MANAGEMENT OF RECONFIGURABLE VIRTUAL PATH NETWORKS

Åke Arvidsson[a]*

[a]Department of Communication Systems, Lund Institute of Technology, Box 118, S-221 00 Lund, Sweden.

We study dynamically rearrangable end-to-end transport networks by means of SDH/SONET and/or ATM virtual paths. The concept and its advantages are briefly reviewed and we propose a new algorithm for designing such networks. Our algorithm provides integer solutions, handles non-linear equivalent bandwidths, finds its own paths, and allows for an arbitrary design optimisation criterion. It is found to perform equal to or better than an earlier, comparable though more restricted, algorithm. We also put forward a strategy for managing virtual paths in real time. It consists of a repeated cycle of on-line measurement of offered traffics followed by the design and implementation of a new virtual path network. Methods and parameters for traffic measurements and network updating frequency are studied and optimised. Applying the strategy to rearrangeable networks subject to variable traffics in a simulator, we discover considerable gains and a robustness in our parameter settings.

1. Virtual Path Networks

A virtual path (VP) is formed by reserving a certain amount of transmission capacity on a series of links and cross connecting the reserved channels through possible, intermediate transit nodes. Interconnecting all origin-destination pairs (OD-pairs) by means of VPs, a virtual path network (VPN) is obtained, figure 1. Such a network forms a higher layer which is logically independent of the underlying physical network. For multi-service networks, such as the B-ISDN, we may have different VPNs for different service classes (SCs), virtual services networks (VSNs), according to traffic properties (*e.g.* peak rate and burstiness) and service demands (*e.g.* loss and delay).

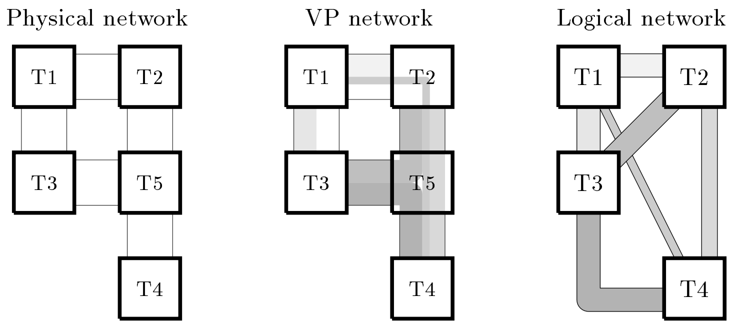


Figure 1. *Formation of a logical network of virtual paths.*

The process of creating and/or rearranging a (logical) network of VPs between a number of end nodes is called capacity management [14], bandwidth management [29], bandwidth switching [1], or bandwidth control [42], and is performed at a network management centre (NMC).

VPs are engineered for each OD-pair respectively so that current traffic demands can be carried with an acceptable grade of service. However, it is not always possible to accomodate all demands to their full extent. Hence capacity allocation must be made in such a way that some performance metric, *e.g.* network profit from carried traffic minus carrying costs, is maximised.

*The work reported herein was partly carried out at the Teletraffic Research Centre, The University of Adelaide, G.P.O. Box 498, Adelaide, South Australia 5001, Australia.

Since the optimality of a certain VP configuration depends on link capacities and currently offered traffics, VP assignments must be re-evaluated in response to changes. We refer to this process as dynamic capacity management (DCM). DCM can be performed either in advance or on demand. The former means that assignments are predefined and changed in an independent manner while individual call attempts control rearranging in the latter. We focus on the former, as this is the one most favoured by low transmission-to-processing costs [14,15].

Some of the motives behind VP networks and DCM [2,12–15,17,22,33,34,41,42] are: Reduced network costs resulting from simplified transit exchanges, simplified multiplexing and grade of service control by service separation through VSNs, faster call handling by excluding intermediate node processing at set-up time, improved network management capabilities such as possibilities to redirect traffic in a congested or faulty network, and a means for providing new services by setting up customer-dedicated (sub)networks as closed VPNs.

2. Algorithms for Dynamic Capacity Management

2.1. Existing Algorithms

DCM must be supported by efficient algorithms to compute VP capacity allocations (VPN designs). We have found DCM algorithms or algorithms closely related to DCM published by Gopal *et al.* [26,27], Herzberg [29–31], Evans [18], Gersht *et al.* [23,24,36] and Mase *et al.* [38]. Algorithms are also outlined by Hui *et al.* [35]. Gerla *et al.* [22] study DCM for packet-switched networks and, finally, Mase *et al.* [39] discuss such algorithms in terms similar to those in [18,23,24,35,36], but without going into any detail. Summing up on these, it is found that: Most algorithms handle bursty services by explicitly or implicitly presuming a linear relationship between the capacity of a VP and its call carrying capability (linear equivalent bandwidth), most algorithms explicitly or implicitly presume the existence of predefined paths for all VPs, and some algorithms produce real valued solutions which are not immediately useful in SDH/SONET-networks.

We have developed a new algorithm that does not require linear equivalent bandwidth nor predefined paths and that produces integer valued solutions. Moreover, the optimisation function can be chosen arbitrarily. The algorithm is a heuristic and as such it does not guarantee that the final solution is a global optimum. On the other hand, the "optimality" guaranteed by some of the algorithms above need not be a global optimum since it is restricted by the fixed set of paths. The importance of this is hard to estimate, but it is known [25] that unrestricted, dynamic path selection is vital to network performance for dynamic routing.

2.2. Our Algorithm

Letting $s = 1, \ldots, S$ denote an arbitrary SC and $o = 1, \ldots, N$ ($d = 1, \ldots, N$) an arbitrary node of origin (destination), the main steps of the algorithm can be described as follows:

1. Read the tables that provide the relationships between capacities and circuits.
2. Read link capacities $C_{o,d}$ (expressed as units of capacity[1]) and offered traffics $A_{s,o,d}$ (expressed as Erlangs).
3. Assign high, initially acceptable call loss levels $\alpha(s, o, d)$ for all SCs s and all OD-pairs o, d.
4. Find the shortest paths available to each traffic s, o, d.
5. Compute the gain achieved for each traffic s, o, d if one unit of capacity was to be added to its shortest path.
6. Compute the loss paid for each traffic s, o, d if one unit of capacity was to be added to its shortest path.
7. Find the traffic $s_{\max}, o_{\max}, d_{\max}$ that would yield the highest gain/loss ratio.
8. If the highest gain/loss ratio is 0 then go to step 11, else proceed to step 9.

[1] A "unit of capacity" is, for PDH a number of 64 kb/s channels, for SDH/SONET the smallest virtual container used, and for pure ATM a rate in cells/second large enough to carry a call of any SC.

9. Assign one unit of capacity to the traffic $s_{\max}, o_{\max}, d_{\max}$.
10. Go to step 4.
11. If low, ultimately acceptable loss levels $\alpha(s,o,d)$ have been reached for all traffics or all capacity has been assigned then stop, else proceed to step 12.
12. Reduce currently acceptable loss levels $\alpha(s,o,d)$ for all s,o,d.
13. Go to step 4.

Available capacity is successively distributed to VPs so that a minimum amount of capacity is used in each step and that maximum value is obtained for each unit of capacity. The successive reductions of acceptable losses serve to ensure fairness in grade of service and that assigned capacity will be sufficiently utilised. The algorithm terminates when for every VP either (i) a final, predetermined, desirable loss level has been reached or (ii) no more capacity is available to VPs which still suffer from high losses.

The tables in step 1 give, for each SC respectively, the number of simultaneous connections that can be supported by i, $i = 1, 2, \ldots$, units of capacity, *i.e.* the equivalent number of circuits. The tables are computed from traffic characteristics, grade of service demands, buffer space and acceptable loss, see *e.g.* [6,7,28,32,39].

In step 3, our initial loss level is 50%. In step 11, it is reduced to the ultimately acceptable level of 0.05% through two intermediate levels of 5% and 0.5% respectively.

Shortest paths in step 4 are determined using Floyd's algorithm [37] with the length l associated to link o,d designed to find the shortest path in number of links traversed, with preferential treatment to paths having more spare capacity left than other paths of equal length

$$l(o,d) = \begin{cases} 1 + \dfrac{1}{{C'_{o,d}}^{4}} & C'_{o,d} > 0 \\ \infty & C'_{o,d} = 0 \end{cases}$$

where $C'_{o,d}$ denotes the remaining, not yet assigned capacity on link o,d.

$G(s,o,d)$ in step 5 is the additional s-traffic that would be carried from o to d if one unit of capacity was added to its currently shortest path

$$G(s,o,d) = \begin{cases} A_{s,o,d}[E_{m'_{s,o,d}}(A_{s,o,d}) - E_{m_{s,o,d}}(A_{s,o,d})] \\ \quad \text{if } (E_{m_{s,o,d}}(A_{s,o,d}) > \alpha(s,o,d)) \text{ and } (l(o,d) < \infty) \\ 0 \\ \quad \text{if } (E_{m_{s,o,d}}(A_{s,o,d}) \leq \alpha(s,o,d)) \text{ or } (l(o,d) = \infty) \end{cases}$$

where $E_m(A)$ is the Erlang-B formula. $m_{s,o,d}$ is the present number of circuits available to s,o,d, while $m'_{s,o,d}$ refers to the case where one more unit of capacity has been added to the shortest path. Both m and m' are determined for each route of the VP individually, by means of the tables referred to in step 1, and then summed.

$L(s,o,d)$ in step 6 is the sum of all gains that can be achieved at the same point and that require some of the capacity also requested by s,o,d:

$$L(s,o,d) = \sum_{s'=1}^{S} \sum_{o'=1}^{N-1} \sum_{d'=o'+1}^{N} I(\mathcal{L}_{s,o,d} \cap \mathcal{L}_{s',o',d'} \neq \emptyset)\, G(s',o',d')$$

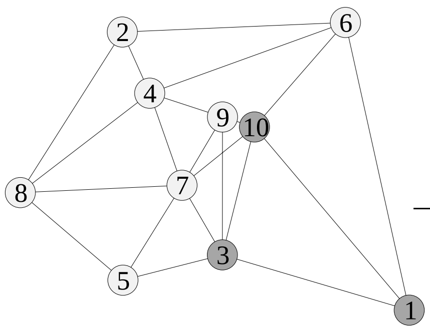
where $\mathcal{L}_{s,o,d}$ is the set of links traversed by the shortest path for s,o,d and $I(\cdot)$ is an indicator function taking the value of 1 if its argument is true, otherwise 0.

2.3. Discussion

Unlike methods based on mathematical programming, our algorithm will remain stable and converge at the same speed for all types of non-linearities and discontinuities in gain and loss functions and irrespective of the ways in which routes for VPs are chosen. This leaves full freedom to modify and extend the algorithm to meet particular demands such as: Biased selection of routes, a limitation to the number of distinct physical routes, predetermined routes, arbitrary profit maximisation function G, and further encouragement of traffic concentration.

2.4. Numerical Results

To investigate the power of the proposed algorithm (A), it was applied to a series of eight distinct networks, each consisting of ten nodes and subject to eight different traffic patterns, each summing up approximately 7,000 Erlangs. A typical network (with one unit of capacity set to $10 \times 64 = 640$ kb/s) and traffic pattern (with traffic given in Erlangs) is shown in figure 2. A complete example and more details on the networks are found in [6,10] or appendix A of [8].



OD	A	OD	A	OD	A	OD	A	OD	A
1– 2	274.5	1– 3	357.8	1– 4	120.8	1– 5	68.9	1– 6	245.8
1– 7	210.7	1– 8	149.9	1– 9	12.8	1–10	329.7	2– 3	52.6
2– 4	213.2	2– 5	155.1	2– 6	23.6	2– 7	69.8	2– 8	79.9
2– 9	143.3	2–10	278.8	3– 4	221.6	3– 5	54.0	3– 6	303.3
3– 7	39.0	3– 8	131.3	3– 9	292.9	3–10	295.4	4– 5	18.7
4– 6	197.0	4– 7	157.7	4– 8	200.3	4– 9	106.8	4–10	172.2
5– 6	249.3	5– 7	55.7	5– 8	120.5	5– 9	46.3	5–10	145.9
6– 7	24.3	6– 8	215.7	6– 9	255.1	6–10	37.2	7– 8	138.1
7– 9	9.6	7–10	285.1	8– 9	40.9	8–10	28.5	9–10	61.9

OD	C	OD	C	OD	C	OD	C	OD	C
1– 3	148	1– 6	32	1–10	28	2– 4	58	2– 6	95
2– 8	62	3– 5	97	3– 7	51	3– 9	40	3–10	40
4– 6	97	4– 7	63	4– 8	31	4– 9	72	5– 7	12
5– 8	68	6–10	52	7– 8	25	7– 9	2	7–10	36
9–10	7								

Figure 2. *Typical network layout, capacities (C), and traffics (A).*

To enable comparisons to the comparable algorithm (B) of [26,27], the number of SCs was set to one.[2] The predefined paths required by B were taken as the four most used ones found by A. Results are summarised in table 1. E_{Call} is the loss averaged over all calls in the network, $E_{\mathrm{OD-pair}}$ is the loss averaged over all OD-pairs, U_{tot} is the mean carried traffic per seized unit of capacity and $P_{\mathrm{OD-pair}}$ is the mean number of distinct routes used per OD-pair. For each algorithm is given the number of times it produced the *best* result with the *mean* result over all 64 configurations within parenthesis. Loss is expressed in % and utilisation in carried Erlangs per unit of capacity. Finally, R_{tot} is the ratio between the virtual capacities of the rearranged networks and the actual capacities of the physical networks, averaged over all 8 networks.

Algorithm	E_{Call}	$E_{\mathrm{OD-pair}}$	U_{tot}	$P_{\mathrm{OD-pair}}$	R_{tot}
A	39 (1.9)	59 (1.8)	64 (5.1)	(1.4)	5 (1.3)
B	25 (2.0)	5 (2.9)	0 (5.0)	(1.2)	3 (1.3)

Table 1. *Comparison of network performance using different DCM-algorithms.*

The table suggests that, for the networks and traffic patterns considered, our algorithm results in a slightly better performance in terms of E_{Call} and even better in terms of $E_{\mathrm{OD-pair}}$. Also, a distinct, slightly higher degree of network utilisation U_{tot} is recorded and we observe equal savings in transmission capacity R_{tot} of about 30% by using different VP arrangements for different traffic patterns.

3. Applying Dynamic Capacity Management

3.1. Operational Considerations

In order to employ the above algorithm to dynamically control a VP network subject to variable traffics, a simulator comprising an arbitrary network like the one in figure 2, traffic generators and traffic monitors for each OD-pair, and an NMC was constructed. The traffic generators generate call attempts according to independent Poisson processes the rates of which can be changed with time. The monitors estimate offered traffics and report their results to the NMC which, in turn, computes a network design by means of the algorithm which is then implemented.

[2]Results on two SCs, voice and frame relaying, are found in [6,7].

We normalise time by the mean call holding time and define the following parameters: T is the length of a period during which offered traffics are constant, $A_k(s,o,d)$ is the offered s-traffic from o to d during the kth traffic period, t_{U} is the interval between two successive network updates, t_{M} is the time needed to produce an estimate of the currently offered traffics, and t_{E} is the time required to report measurement results from the nodes to the NMC, compute the new design at the NMC by executing the algorithm, reporting the new design back to the nodes, and implementing it.

A new design implies that physical routes and/or capacity assignments of VPs may change. The former means that calls must be moved from one physical path to another, and the latter may result in VPs no longer being able to support all calls in progress. Such calls must either be rerouted over tandem nodes or prematurely cleared. Neither of these alternatives are very attractive: The former means increased demands on node processing and transmission capacity, while the latter is unacceptable from subscribers' point of view.

Our policy is to provide one-hop rerouting if this is possible. The alternative route is selected according to the Least Busy Alternative (LBA) strategy [40]: For each pair of VPs between the nodes in question is the highest utilisation computed after which the pair with the lowest maximum utilisation is chosen. Tandem routing over more than one node is prohibited in the interest of utilisation efficiency. Hence, if all two-hop paths are blocked, premature clearing is used as a last resort. Further, rerouting is combined with limited repacking so that, at every network updating point, rerouted calls are moved back to direct routes as far as possible, but without optimisation.

3.2. Traffic Dynamics

On the call arrival time scale, traffics basically exhibit two kinds of variations [19]: Those which can be seen as a result from the stochastic nature of a Poisson process of constant rate (micro dynamics), and those which are better modelled by rate variations (macro dynamics). Micro dynamics is thus characterised by stochastic variations on the order of minutes, while macro dynamics is more regular and on the orders of hours. It follows that micro dynamics requires faster and more frequent network updating than macro dynamics.

Consider a link on which the average occupancy is M and let $U_{m,M}$ denote the expected time elapsed from the moment at which an occupancy of m is detected until the occupancy is again M for the first time [5]. Figure 3 displays $U_{m,M}$ for links with 100 (left) and 1000 (right) circuits operating at engineered losses of 0.5%, 1.0%, 2.0% and 5.0%.

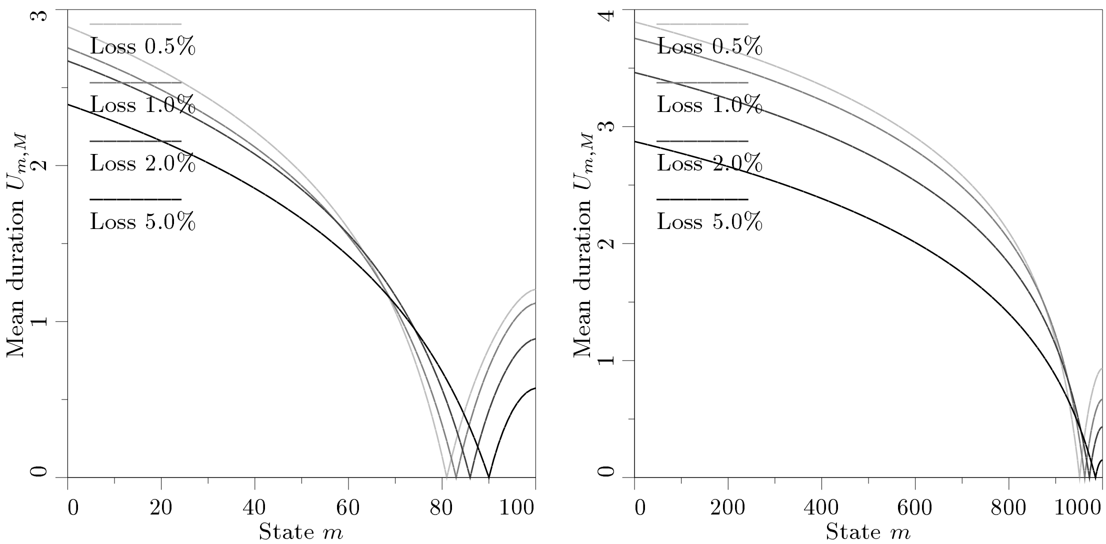


Figure 3. *Expected duration of load states.*

It is observed that the expected time required to return to the average point of operation increases up to a maximum of 3 and 4 time units respectively for the largest initial deviation, while the more common deviations of, say, $\pm 10\%$ around a only are expected to last a fraction of a call holding time.

Realising that network updating must not be slower than the variations it is supposed to compensate for, it is concluded that DCM cannot be accomplished in the time scale of micro dynamics. Instead, it appears that solutions traditionally employed to cope with this and other types of variations — various arrangements of overflow systems with alternative routing — would be adequate for rearrangable VP networks too. In fact, applying the two methods in parallel they will compliment each other: DCM rearranges the network on the macro scale and overflow arrangements rearrange calls on the micro scale.

3.3. Traffic Estimation

In a real environment, offered traffics are not known but must be estimated from forecasts and/or on-line measurements. We restrict ourselves to short term, on-line measurements, as we wish to device a dynamic, fully automatic management system, and distinguish between two methods: A direct one through arrival counting (AC), and an indirect one through carried traffic measurements (CT).[3]

AC means counting the number of call attempts received during an observation interval of length t, $N(t)$, from which an estimate $\hat{A}$ of the offered traffic A is obtained as $\hat{A} = N(t)/t$. In CT is the number of busy circuits at time τ, $a'(\tau)$, recorded during an interval of length t, $\int_t a'(\tau)\,d\tau$, from which an estimate $\hat{A}'$ of the carried traffic A' is obtained as $\hat{A}' = \frac{1}{t}\int_t a'(\tau)\,d\tau$. This, in turn, gives an estimate of the offered traffic by "backward Erlang computation", *i.e.* by solving for $\hat{A}$ in $\hat{A}' = \hat{A}(1 - E_m(\hat{A}))$.

The analysis of AC is straightforward and we find the expectation E and the variance V of $\hat{A}$ as $E\{\hat{A}\} = A$ and $V\{\hat{A}\} = A/t$ respectively.

For CT, first consider an infinite group for which the probability of loss is 0, hence $\hat{A} = \hat{A}'$. We find $E\{\hat{A}\} = A$ and $V\{\hat{A}\} = 2A/t[1 - (1 - e^{-t})/t]$. Comparing AC to CT, it is noted that the latter provide better accuracy if $t < 1.5936$. For a finite group, however, loss is > 0 and $\hat{A} > \hat{A}'$. We will not attempt to analyse the procedure in detail. In short, $\hat{A}'$ is an unbiased estimate of A', but estimating A by $\hat{A}$ through $\hat{A}'$ and backward Erlang gives a positive bias for finite observation intervals because of the non-linearity of the Erlang loss function.

In conclusion, AC is chosen as our estimation method.

3.4. Observation Interval

We now turn to the problem of selecting a proper observation interval $t = t_{\mathrm{M}}$ for AC. On one hand, the larger t_{M}, the lower the variance of the estimate, but, on the other hand, since the call arrival rate varies with time, the larger t_{M}, the more old, possibly invalid information is contained in a sample. We define the optimal observation interval as the one for which the expected, squared error of an estimate takes its minimum,

$$t_{\mathrm{M}}^{\mathrm{opt}} = \min_{t_{\mathrm{M}}} E\{(\hat{A} - A_k)^2\}$$

where A_k is the offered traffic during interval k. For this interval, network designs completed during the first t_{E} time units (1) will be based on measurements that were made during the previous interval $k-1$ and designs completed during the following t_{M} time units (2) will be based on measurements covering both the previous and the present intervals $k-1$ and k. For the rest of the interval (3), all designs completed will be based on information relating to the present interval k. Computing the expected error for (1)–(3), weighting and minimising we find

$$t_{\mathrm{M}}^{\mathrm{opt}} = \min\left(\frac{\sqrt{3A_kT + 3(A_{k-1} - A_k)t_{\mathrm{E}}}}{|A_{k-1} - A_k|}, T - t_{\mathrm{E}}\right)$$

Extending the result to networks, and assuming cyclic traffic sequences for each OD-pair, we compute $t_{\mathrm{M}}^{\mathrm{opt}}$ for each OD-pair and each interval k and compute an overall $t_{\mathrm{M}}^{\mathrm{opt}}$ by weighting over the expected, absolute errors. For the examples considered here, typical values obtained for $T = 20$ (50) are $t_{\mathrm{M}}^{\mathrm{opt}} \approx 4$ (6).

Figure 4 shows loss as a function of $t_{\mathrm{M}}/t_{\mathrm{M}}^{\mathrm{opt}}$, as observed in our simulator for the same networks as in table 1 with $T = 20$ (left) and $T = 50$ (right). Solid lines refer to different values of t_{E} and dotted lines to "bounds": The upper ones refer to networks without updating but that are permanently dimensioned

[3] Carried traffic measurements by means of discrete time sampling is considered in [19]. More advanced alternatives include *e.g.* moving average, filtering and adaptive filtering, possibly including forecasting.

for average traffics and the lower ones to "perfect" updating, *i.e.* updating only at the exact instants of traffic changes and with error free estimates of upcoming traffics.

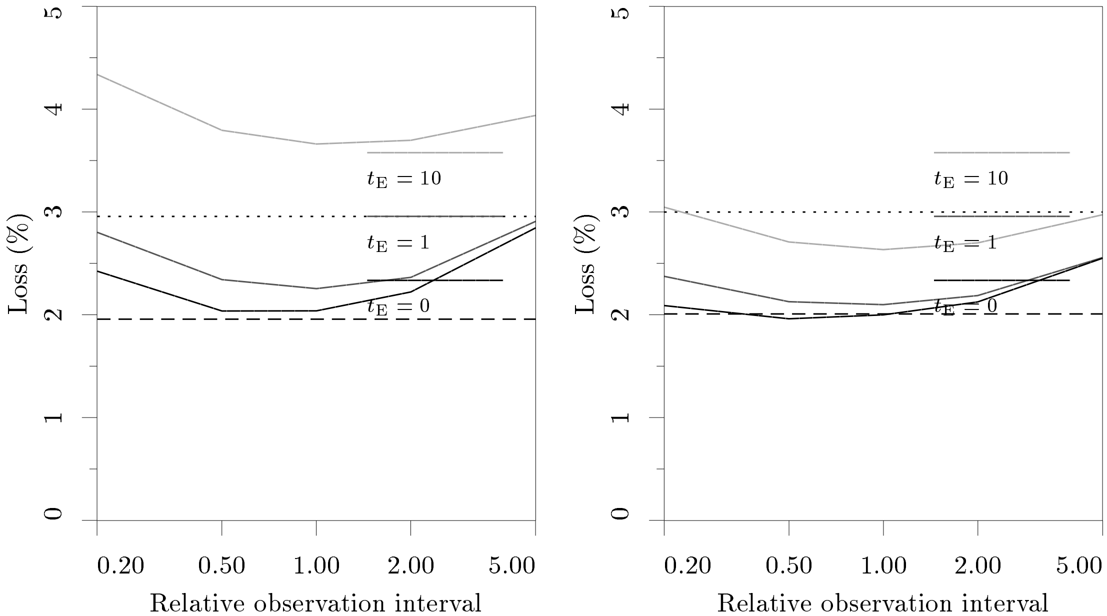


Figure 4. *Network loss versus observation interval for different durations of stable states.*

The figure confirms the optimum and we observe an overall insensitivity to t_M^{opt}: Setting t_M to $t_M^{opt}/2$ or $2t_M^{opt}$ has very little impact on loss. It may be added that similar curves are obtained from simulations in which traffics are changed over several units of time respectively rather than at distinct points.

In two cases are the bounds exceeded: Firstly, if $t_E \approx T$ will average dimensioning perform better than repeated updating, as for $t_E = 10$ and $T = 20$. Secondly, if $t_E \ll T$ will micro dynamics be captured provided the measurement interval is sufficiently short, as for $t_E = 0$ and $T = 50$ with $t_M = t_M^{opt}/2$. For a given t_E, the former case puts a lower limit on rate variations T that fall within the scope of DCM, while the latter is of less practical importance as $t_E \approx 0$ is not feasible in practice. Finally, we point at the big influence that t_E has on network performance.

In a real network, T are A_k are neither known, nor do they actually exist. T can, however, be regarded as a target reflecting DCM ambitions. Further, fixing T to, say, one hour, it is reasonable to assume that one would have a fair idea of average traffics per hour and OD-pair for a complete 24 hour period. Due to the robustness of the optimum, the impact of minor differences between "a fair idea" and the exact value will be little or none.

3.5. Updating Interval

We now consider the updating interval t_U. While it is clear that frequent updating will give faster responses to changes in offered traffics, it is also clear that the more frequent the updating, the more resources are spent on transmitting and processing DCM-information and on implementing VP designs. Updating is motivated by changes in offered traffics hence we wish to update the network neither more nor less often than required to meet these changes.

We define the optimal updating interval, $t_U = t_U^{opt}$, as the one for which a change in traffics from A_{k-1} to A_k will result in that the first estimate containing some A_k traffic is expected to differ from the previous, jth estimate of A_{k-1} by at least the standard deviation of the latter,

$$t_U^{opt} = \min t_U : E\{|\hat{A}_k^1 - \hat{A}_{k-1}^j|\} \geq E\{|A_{k-1} - \hat{A}_{k-1}^j|\}$$

where $\hat{A}_k^j$ denotes the jth estimate of A_k. We find

$$t_U^{opt} = \min\left(\frac{2\sqrt{A_{k-1}t_M}}{|A_k - A_{k-1}|}, t_M\right)$$

Extending the result to networks, an overall $t_{\mathrm{U}}^{\mathrm{opt}}$ is obtained as above, *i.e.* we assume cyclic traffic sequences for each OD-pair, compute $t_{\mathrm{U}}^{\mathrm{opt}}$ over all OD-pairs and traffics, and weight over the traffics. Typical values obtained for our examples are $t_{\mathrm{U}}^{\mathrm{opt}} \approx 2$ (3) for $T = 20$ (50).

Simulating the same networks as above for various values of $t_{\mathrm{U}}/t_{\mathrm{U}}^{\mathrm{opt}}$, we observe a slight increase in loss the longer the updating interval and a sharp decrease in the fraction of calls subject to rerouting or packing. Typical loss values are indicated in figure 4, and typical processing values observed are: Fraction of lost calls in progress 0–10^{-6}, fraction of rerouted calls 10^{-2}–10^{-4}, and fraction of packed calls 10^{-2}–10^{-4}.

To see the trade off between response time and processing, let us assume the following costs: In between two network updates, a penalty of 1 is earned for every call attempt lost. At the point of a network update is a fixed penalty of 10 assigned for the actual update and a variable cost for processing calls in progress: 0.1 for every call which currently is not routed on its direct path, 0.1 for every call that cannot be carried on its direct path after the update, and 10 for every call that cannot be carried at all.

The result is shown in figure 5 as total cost per time unit as a function of $t_{\mathrm{U}}/t_{\mathrm{U}}^{\mathrm{opt}}$, with various values of t_{E} and bounds indicated as above.

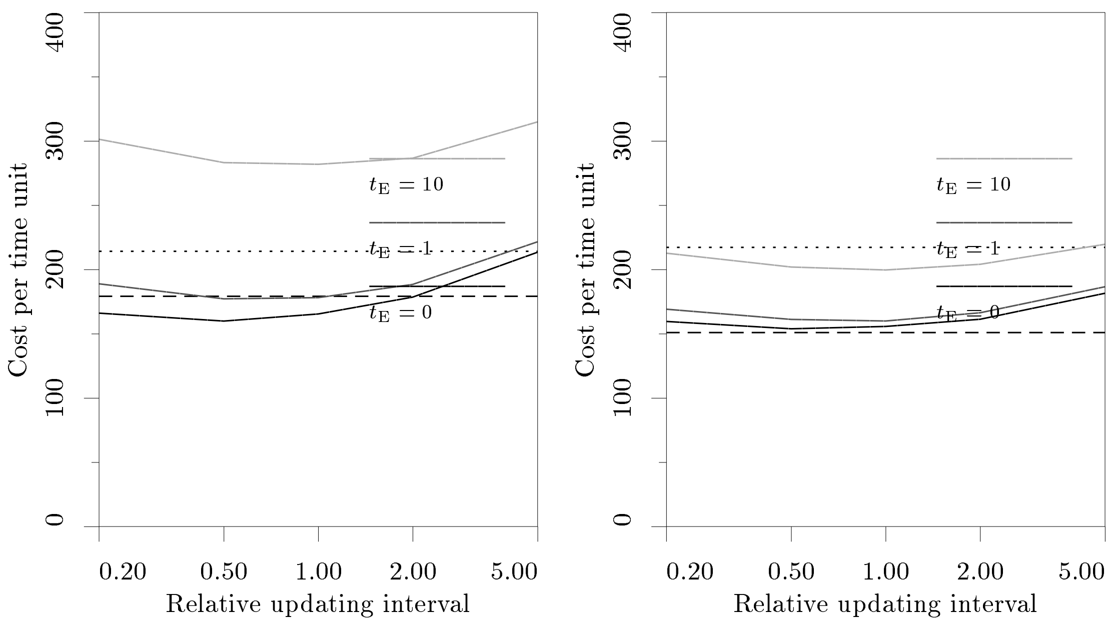


Figure 5. *Network cost per time unit versus updating interval for different execution times and durations of stable states.*

The curves show that a minimum is obtained around $t_{\mathrm{U}} = t_{\mathrm{U}}^{\mathrm{opt}}$ and that, similar to t_{M}, the minimum for t_{U} is flat hence minor differences between estimated and actual values of T and $A_k(o,d)$ are of no great importance to network performance. (It is pointed out that assigning costs differently may, of course, shift the actual value of $t_{\mathrm{U}}^{\mathrm{opt}}$ in either direction.)

We also note that DCM does not work for fast variations relative to the execution time, as for $t_{\mathrm{E}} = 10$ and $T = 20$. However, for moderate to slow variations relative to the execution time, will DCM be almost equal to or even better than perfect updating. The latter is explained by the fact that it takes a while before a change has had any impact on the state of the network. Updating exactly when the changes take place is therefore a bit too early and, consequently, a considerable amount of calls must be processed. This suggests that prescheduled updating requires much more precise information than our approach. Finally, we again observe the large impact of t_{E} on network performance.

4. Conclusions

VPs and DCM are two important issues in broadband networks based on SDH/SONET and/or ATM. Besides reviewing the motives for VPs and DCM and possible benefits, this paper has addressed two major issues: We have presented a new VP design algorithm and put forward a DCM strategy, *i.e.* a real time strategy for managing VPs.

Contrary to earlier proposals, our VP design algorithm provides integer solutions, handles non-linear equivalent bandwidths, finds its own paths, and allows for an arbitrary design optimisation criteria. Applying it to a large number of networks and traffic patterns, it was found to perform equal to or better than an earlier, comparable though more restricted, algorithm.

Our DCM strategy suggests cyclically repeated on-line measurements of offered traffics followed by the design and implementation of new VP networks. The possible time frame of this strategy was analysed, we studied simple traffic measurement methods and optimal parameter settings for such measurements and we have showed how VP redesign intervals can be optimised so that network cost is minimised.

It was seen that the time scale of DCM is, at least, on the order of a few mean call holding times. Comparing two simple methods of estimating offered traffics, namely directly by counting arrivals and indirectly by observing the carried traffic, it was found that the latter is biased hence only direct methods can be used. The optimal interval during which arrivals should be counted was determined, and it was found that minor deviations from this value are not very critical. Studying the interval between updates and assigning costs to loosing and processing calls, we found a point at which the total cost reaches a minimum, which again was found to be robust against minor variations. Our study has also demonstrated a dependency between network performance and the time needed to compute and implement a new VP design which suggests that optimal design is not only minimising loss but also minimising execution time.

5. Further Work

The results presented above only represent a sample of important issues related to DCM algorithms and application strategies. The present paper, which is a part of a long term project, [6–10], reports on results achieved so far. Further areas include, alternative DCM algorithms and improving speed of the proposed one, thresholds to prevent unnecessary updating such as significant loss increase, combining alternate routing and DCM, and algorithms for traffic estimates, their parameters and performance.

REFERENCES

1. Addie, R. & Warfield, R.: *Bandwidth Switching and New Network Architectures*, paper no. 2.3iiA.1 in Proc. 12th Int. Teletraffic Cong., Torino, Italy (1988).
2. Addie, R., Burgin, J. & Sutherland, S.: *Information Transfer Protocols for the Broadband ISDN*, paper no. 22.6 in Proc. IEEE Globecom '88, Hollywood, Florida, U.S.A. (1988).
3. Ahmed, H., Brgi, P., Haag, J.-P. & Wiest, B.: *Synchronous Transmission Systems and Cross Connects for Broadband Networks*, Electr. Commun., vol. 64, no. 2/3, pp. 194–203 (1990).
4. Andersson, J.-O., Improta, S. & Evangelisti, G.: *Digital Cross-Connect Systems — A Tool for Managing the Transport Network*, paper no. C6.5 in Proc. XIIIth Int. Switching Symp., Stockholm, Sweden (1990).
5. Arvidsson, Å.: *Priorities in Circuit Switched Networks*, Dis. thesis, Department of Communication Systems, Lund Institute of Technology, Lund, Sweden (1990).
6. Arvidsson, Å.: *A Study on Statistical Multiplexing and Dynamic Capacity Management in Voice/Frame Relaying Networks*, Report no. 7/92, Teletraffic Research Centre, The University of Adelaide, Adelaide, South Australia, Australia (1992).
7. Arvidsson, Å.: *On Dynamic Capacity Management in Voice/Frame Relaying Networks*, paper no. 2.3 in Proc. 10th Nordic Teletraffic Sem., Århus, Denmark (1992).
8. Arvidsson, Å.: *Strategies for Dynamic Network Management*, paper no. 7.2 in Proc. 11th Nordic Teletraffic Sem., Stockholm, Sweden (1993).
9. Arvidsson, Å.: *Strategies for Dynamic Capacity Management*, pp. 291–300 in Proc. 8th Australian Teletraffic Research Sem., Melbourne, Victoria, Australia (1993).
10. Arvidsson, Å.: *Real Time Management of Reconfigurable Virtual Path Networks in B-ISDN*, Report in preparation, Teletraffic Research Centre, The University of Adelaide, Adelaide, South Australia, Australia (1994).
11. Astani, K., Harrison, R. & Ballart, R.: *CCITT Standardization of Network Node Interface of Synchronous Digital Hierarchy*, IEEE Commun. Mag., vol. 28, no. 8, pp. 15–20 (1990).
12. Atkinson, D., Anido, G. & Bradlow, H: *Designing ATM Virtual Networks*, pp. 183–192 in Proc. 6th Australian Teletraffic Sem., Wollongong, New South Wales, Australia (1991).
13. Brungard, D., Grotjohann, H. & Kallenberg, P.: *Impact of New Transmission Technologies on the Network Architecture, Network Management and Economics*, paper no. A1.2 in Proc. XIIIth Int. Switching Symp.,

Stockholm, Sweden (1990).
14. Burgin, J.: *Management of Capacity and Control in Broadband ISDN*, Int. J. of Dig. and Ana. Cabled Sys., vol. 2, pp. 155–165 (1989).
15. Burgin, J.: *Broadband ISDN Resource Management*, Computer Netw. and ISDN Syst., vol. 20, no. 1–5, pp. 323-331 (1990).
16. Cheung, N.: *The Infrastructure for Gigabit Computer Networks*, IEEE Commun. Mag., vol. 30, no. 4, pp. 60–68 (1992).
17. Dutkievicz, E. & Anido, G: *Traffic Management and Control in Broadband Networks*, pp. 105–112 in Proc. 5th Australian Fast Packet Switching Workshop, Melbourne, Victoria, Australia (1990).
18. Evans, S.: *A Mathematical Model and Related Problems of Optimal Management and Design in a Broadband Integrated Services Network*, J. Australian Math. Soc., ser. B, vol. 31, part 2, pp. 150–175 (1989).
19. Filipiak, J.: *Automatic Network Management Controls*, IEEE Trans. on Commun., vol. 39, no. 12, pp. 1776–1786 (1991).
20. CCITT Recommendations G.702–G.704: *Digital Hierarchies*, CCITT, ITU, Geneva, Switzerland.
21. CCITT Recommendations G.707–G.709: *Synchronous Digital Hierarchy*, CCITT, ITU, Geneva, Switzerland.
22. Gerla, M., Monteiro, J. & Pazos, R: *Topology Design and Bandwidth Allocation in ATM Nets*, IEEE J. Sel. Areas in Commun., vol. 7, no. 8, pp. 1253–1262 (1989).
23. Gersht, A. & Shulman, A.: *Optimal Routing in Circuit Switched Networks*, IEEE Trans. on Commun., vol. 37, no. 11, pp. 1203–1211 (1989).
24. Gersht, A. & Kheradpir, S.: *Integrated Traffic Management in SONET-Based Multi-Service Networks*, pp. 67–72 in Proc. 13th Int. Teletraffic Cong., Copenhagen, Denmark (1991).
25. Girard, A. & Hurtubise, S.: *Dynamic Routing and Call Repacking in Circuit-Switched Networks*, IEEE Trans. on Commun., vol. 31, no. 12, pp. 1290–1294 (1983).
26. Gopal, G., Kim, C.-K. & Weinrib, A.: *Dynamic Network Configuration Management*, paper no. 302.2 in IEEE Int. Conf. on Commun., Atlanta, Georgia, U.S.A. (1991).
27. Gopal, G., Kim, C.-K. & Weinrib, A.: *Algorithms for Reconfigurable Networks*, pp. 341–347 in Proc. 13th Int. Teletraffic Cong., Copenhagen, Denmark (1991).
28. Guérin, R., Ahmadi, H. & Naghshineh, M.: *Equivalent Capacity and Its Application to Bandwidth Allocation in High-Speed Networks*, IEEE J. Sel. Areas in Commun., vol. 9, no. 7, pp. 968–981 (1991).
29. Herzberg, M.: *Network Bandwidth Management — A New Direction in Network Management*, pp. 218–225 in Proc. 6th Australian Teletraffic Sem., Wollongong, New South Wales, Australia (1991).
30. Herzberg, M. & Byes, S.: *Bandwidth Management for Reconfigurable Multi-Service Networks*, Australian Teletraffic Research, vol. 25, no. 2, pp. 1–13 (1993).
31. Herzberg, M. & Byes, S.: *Bandwidth Management in Reconfigurable Networks*, pp. 280–290 in Proc. 8th Australian Teletraffic Sem., Melbourne, Victoria, Australia (1993).
32. Hong, D. & Suda, T.: *Congestion Control and Prevention in ATM Networks*, IEEE Netw. Mag., vol. 5, no. 4, pp. 10–16 (1991).
33. Hughes, D., Bradlow, H. & Anido, G: *Congestion Control in an ATM Network*, pp. 835–840 in Proc. 13th Int. Teletraffic Cong., Copenhagen, Denmark (1991).
34. Hughes, D. & Wajda, K: *Comparison of Virtual Bandwidth Assignment and Routing Methods*, Annales des Telecommunication, to appear.
35. Hui, J., Gursoy, M., Moayeri, N. & Yates, R.: *A Layered Broadband Switching Architecture with Physical or Virtual Path Configurations*, IEEE J. Sel. Areas in Commun., vol. 9, no. 9, pp. 1416–1426 (1991).
36. Kheradpir, S., Gersht, A. & Stinson, W.: *Performance Management in SONET-Based Multi-Service Networks*, paper no. 39.3 in Proc. IEEE Globecom '91, Phoenix, Arizona, U.S.A. (1991).
37. Kleinrock, L.: *Queueing Systems, Volume II: Computer Applications*, John Wiley & Sons, New York (1976).
38. Mase, K. & Imase, M.: *An Adaptive Capacity Allocation Scheme in Telephone Networks*, IEEE Trans. on Commun., vol. 30, no. 2, pp. 354–359 (1982).
39. Mase, K. & Shioda, S.: *Real-Time Network Management for ATM Networks*, pp. 133–140 in Proc. 13th Int. Teletraffic Cong., Copenhagen, Denmark (1991).
40. Mitra, D. & Gibbens, R.: *State-Dependent Routing on Symmetric Loss Networks with Trunk Reservations. II: Asymptotics, Optimal Design*, Ann. Op. Res., vol. 35, no. 1–4, pp. 3–30 (1992).
41. Moondra, S.: *Impact of Emerging Switching-Transmission Cost Tradeoffs on Future Telecommunications Network Architectures*, IEEE J. Sel. Areas in Commun., vol. 7, no. 8, pp. 1207–1218 (1989).
42. Ohta, S. & Sato, K.-I.: *Dynamic Bandwidth Control of the Virtual Path in an Asynchronous Transfer Mode Network*, IEEE Trans. on Commun., vol. 40, no. 7, pp. 1239–1247 (1992).
43. Sato, K.-I., Ohta, S. & Tokizawa, I.: *Broad-Band ATM Network Architecture Based on Virtual Paths*, IEEE Trans. on Commun., vol. 38, no. 8, pp. 1212–1222 (1990).