REAL TIME MANAGEMENT OF VIRTUAL PATHS

Åke Arvidsson, Department of Communication Systems, Lund Institute of Technology, Lund, Sweden*†

Abstract We propose a simple and robust automatic strategy for managing SDH/SONET and/or ATM virtual paths in real time to handle slow traffic variations. It consists of on-line measurements of offered traffics followed by the design and possible implementation of a new virtual path network in a repeated cycle. Methods and parameters for traffic measurements and network updating are studied and optimised to achieve maximal traffic carrying capability at minimal cost. Applying our strategy to networks subject to partially unknown, variable traffics we discover a considerable adaptivity, large savings in transmission capacity, and a performance far better than achieved by deterministic management.

Virtual Paths

A virtual path (VP) is a reserved amount of transmission capacity on a series of links which is cross connected through possible, intermediate transit nodes. A network of VPs, virtual path network (VPN), forms a higher layer which is logically independent of the underlying physical network. The VP concepts of SDH/SONET and ATM both lend themselves well to implementing VPNs.

VPNs are designed to handle current traffic demands with an acceptable grade of service, or, if all demands cannot be accomodated, to maximise some performance metric, *e.g.* carried traffic profits minus running costs. Since allocations depend on available link capacities and currently offered traffics, allocations must be re-evaluated in response to *e.g.* traffic variations. A simple, layered view of such variations is shown below.

Level	Variation source (time scale)
1 Path	User behaviour (h)
2 Channel	Equipment usage (min)
3 Burst	Information generation (s)
4 Cell	Cell delivery (ms)

Figure 1: *Layered model of traffic variations.*

The **first level** refers to variations in the number of potential users, *e.g.* the number of users being in their office, the **second level** to variations in the number of active users, *e.g.* the number of potential users currently using their computer, the **third level** to variations in user information generation, *e.g.* the number of active users currently producing data packets, and the **fourth level** to variations in user delivery, *e.g.* the number of generating users currently presenting new cells to the network.

The figure refers to a B-ISDN employing ATM. Traditional telephone networks only experience layers 1 and 2, while packet switched networks also see layer 3. Traditionally, advanced routing of calls and packets respectively has been the single means by which variations are handled. The concept of VPNs can be seen as a new, better tool to handle level 1 traffic variations, while good routing schemes still are required on lower levels. Other motives behind VPNs include reduced costs resulting from simplified transit exchanges, faster call handling by excluding transit node processing, improved network management capabilities such as traffic redirection in faulty networks, and a means for providing customer-dedicated, closed VPNs. For B-ISDN type networks, we may additionally gain simplified statistical multiplexing and grade of service control by grouping services according to their characteristics, *e.g.* peak rate and burstiness, and demands, *e.g.* loss and delay, and use separate VPNs for each class.

Preliminaries

We consider VPNs with N nodes. Each VP carries a designated service class and has M virtual channels associated to it, which depends on the capacity of the VP, the buffer space for level 3 and 4 statistical multiplexing, and service requirements such as loss, delay and jitter for the service class in question, see *e.g.* [12]. All calls for which a channel is available are accepted. For the sake of simplicity, we restrict ourselves to one service class and set its mean call holding time to unity. However, the results are readily extended to multi-service networks.

In terms of session requests, each origin-destination pair (OD-pair) $o, d = 1, \ldots, N$ is offered a pure chance traffic, expressed in Erlangs, the rate of which which changes every Tth time unit. The traffic offered to VP o, d during the kth interval is denoted by $A_k(o, d)$. The sequence is limited to $k = 1, \ldots, K$ and forms a traffic profile which is cyclically repeated, so that $A_K(o, d)$ is followed by $A_1(o, d)$. Typically, cycles cover a day or a week and $A_k(o, d)$s are known as expectations, *e.g.* from long term measurements, but will in reality vary around these values, figure 2.

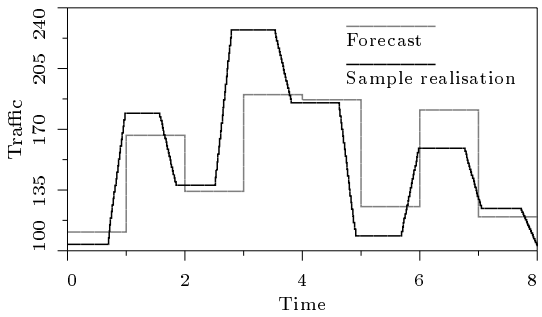

Figure 2: *Traffic profiles.*

*Department of Communication Systems, Lund Institute of Technology, Box 118, S-221 00 Lund, Sweden. Telephone int+46 46 109669, facsimile int+46 46 145823, email akear@tts.lth.se.

†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.

The management strategy is illustrated in figure 3. All nodes of the transport network simultaneously monitor offered traffics during intervals of t_M time units, with new intervals commencing every t_Uth time unit. Traffic estimates are sent to a network management centre (NMC), which computes an updated VPN, evaluates the profitability of the design and possibly returns the results for implementation, a process which, including possible implementation, takes t_E time units.

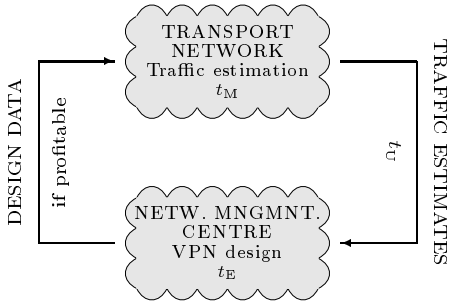

Figure 3: *Management strategy.*

The VPN design algorithm used at the NMC is given in [3]. It requires neither linear equivalent bandwidth nor predefined paths, produces integer valued solutions, converges safely and relatively fast, and the optimisation function can be chosen rather arbitrarily, see also *e.g.* [2, 4, 7, 8, 9, 11, 13]. The other components of the strategy are dealt with in the sequel.

New designs may result in that sessions in progress must be moved, or that VPs no longer can support all sessions in progress. The former is not considered further here, but for the latter we provide two-VP rerouting over the least busy alternative if possible, otherwise premature clearing is enforced. Rerouting is also combined with limited repacking: At updating instants are rerouted sessions moved to direct routes if possible, but without optimisation.

Although redesigning networks aims at increasing the carried traffic, and hence revenues, there are also costs associated, *e.g.* data collection and transmission, and processing of sessions and paths. To account for this, we have included the following costs: C_L for every call attempt lost, C_T for each updating attempt (transmission of data to the NMC, computing and analysing a design), and, if implementing it, C_I for the implementation as such plus C_P for processing a call not routed on its direct path, C_R for processing a call that cannot be carried on its direct path, and C_C for each call that must be dropped.

We have used a set of test networks which comprises 8 different physical networks, each consisting of $N = 20$ nodes and subject to $K = 8$ distinct sets of offered traffics, for more details see *e.g.* [2]. Moreover, we have set the traffic interval length $T = 30$ and chosen the following costs: $C_L = 1$, $C_T = 10$, $C_I = 100$, $C_P = 0.5$, $C_R = 0.5$, $C_C = 10$. Numerical values obtained below are to some extent dependent on our particular choices.

The remainder of this paper is organised as follows: After a preliminary look at which levels of traffic variations that our strategy can handle, we go into details with methods and intervals for traffic observation, the criterion for evaluating the profitability of a new design, and the optimal updating interval. In the concluding section, we examine the performance of our strategy and compare it to alternative solutions.

Traffic Dynamics

Clearly, only variations that last long enough to be detected can be handled by our strategy. Figure 4 shows expected relative errors of the estimates as a function of T. The solid line refers to estimating by optimal measurements as described below, and the dotted line to estimating by averaging expected traffics over all k. Obviously, the strategy is useful only if it does better than average dimensioning, and we observe that this requires T to be greater than a couple of call holding times.

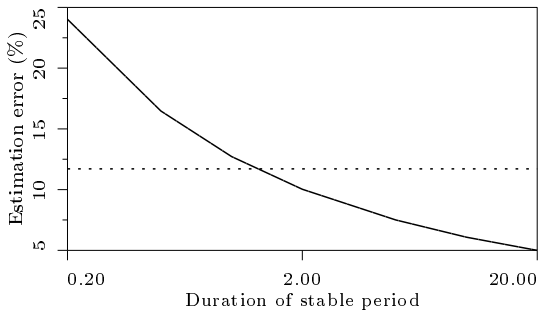

Figure 4: *Bounds imposed by traffic estimation time.*

It is also obvious that our strategy is limited to traffics which prevail long enough to allow for a network redesign. Figure 5 shows expected operation costs per time unit as a function of t_E/T. The curves refer to redimensioning based on optimal measurements (solid line), on perfect knowledge (dashed line), and on an average over all k (dotted line). Again dynamic management is useful only if it performs better average dimensioning, and we observe that this requires T to be greater than a couple of design turn-around times.

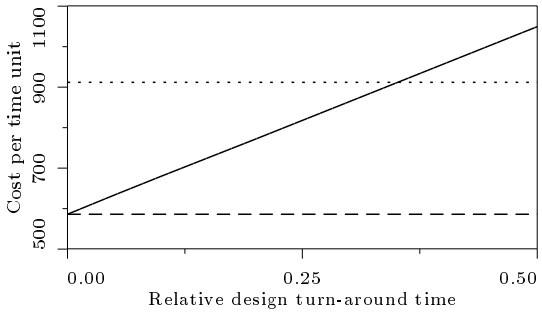

Figure 5: *Bounds imposed by network design time.*

While it is clear that the time scale of level 1 complies with these requirements, figure 6 shows that this is not the case for level 2. The diagram gives some examples of expected times to return to the average operating point for a link with 100 circuits subject to a constant traffic [1]. It is seen that no deviations are expected to last long enough, and this is particularly true for the smaller, and more common, deviations. Summing up, our strategy is limited to level 1, and

level 2 must be handled by other means, *e.g.* traditional overflow routing schemes.

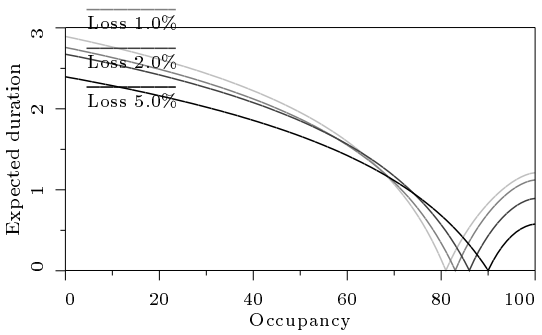

Figure 6: *Time scale of level 2 variations.*

Traffic Estimation

In a real environment, offered traffics are not known but must be estimated from forecasts and/or measurements. For our purpose, only short term, on-line measurements, possibly combined with short term forecasts, are relevant.

We first consider arrival counting (AC) which 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 $\hat{A} = N(t)/t$. The analysis is straightforward and we find $E\{\hat{A}\} = A$ and $V\{\hat{A}\} = A/t$.

Next, we study carried traffic measurements (CT), in which the number of busy circuits at time τ, $a'(\tau)$, is recorded during t time units, $\int_t a'(\tau)\,d\tau$, from which an estimate $\hat{A}'$ of the carried traffic A' is obtained as $\hat{A}' = t^{-1}\int_t a'(\tau)d\tau$. This, in turn, gives an estimate of the offered traffic by "backward Erlang computation", *i.e.* by solving for $\hat{A}$ in $\hat{A}' = \hat{A}(1 - E_M(\hat{A}))$. (For estimation of A' only, see also *e.g.* [5, 6].) Assuming a loss of 0, hence $\hat{A} = \hat{A}'$, and negative exponential call holding times, 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 provides better accuracy if $t \lesssim 1.5936$. However, any loss > 0 requires backward Erlang which, because of the non-linearity of the Erlang loss function, will introduce a positive bias to $\hat{A}$. Figure 7 shows examples of the bias recorded by applying CT on a simulated trunk of 100 circuits. In conclusion, CT cannot be used for real-time estimation of offered traffics and AC is chosen as our estimation method.

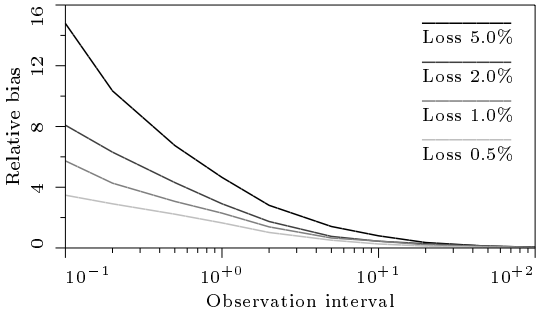

Figure 7: *Bias for CT measurements.*

Observation Interval

We now turn to the problem of selecting a proper observation interval $t = t_{\mathrm{M}}$ for AC, a balance between low variance (large t), and exclusion of old, invalid information (small t). We define the optimal observation interval as the one for which the expected, squared error of an estimate takes its minimum, $t_{\mathrm{M}}^{\mathrm{opt}}(k) = \min_{t_{\mathrm{M}}} E\{(\hat{A} - A_k)^2\}$, where A_k is the offered traffic during interval k, where we have dropped the references to o, d.

Designs completed during the first t_{E} time units will be based on measurements made during $k-1$, those completed during the following t_{M} time units will be based on measurements covering both $k-1$ and k, and for the rest of the interval all designs completed will be based on information relating to k. Computing the expected errors for each case, minimising the sum by differentiating with respect to t_{M}, and solving for a positive, real valued root gives $t_{\mathrm{M}}^{\mathrm{opt}}(k) = \sqrt{3A_kT + 3(A_{k-1} - A_k)t_{\mathrm{E}}}/|A_{k-1} - A_k|$. If the difference between A_{k-1} and A_k is small, we may get $t_{\mathrm{M}}^{\mathrm{opt}} > T - t_{\mathrm{E}}$, a result for which our model is not valid, but a sequence of more than two traffics must be considered. We refrain from this and simply set $t_{\mathrm{M}}^{\mathrm{opt}}$ as the minimum of $T - t_{\mathrm{E}}$ and the above.

Extending the result to networks, we compute an overall $t_{\mathrm{M}}^{\mathrm{opt}}$ by a weighted sum of $t_{\mathrm{M}}^{\mathrm{opt}}(k)$, $t_{\mathrm{M}}^{\mathrm{opt}} = \sum_{o,d,k} t_{\mathrm{M}}^{\mathrm{opt}}(k)\,(|\hat{A}_k - A_k| / \sum_{o,d,k} |\hat{A}_k - A_k|)$.

Figure 8 shows the loss for different values of $t_{\mathrm{M}}/t_{\mathrm{M}}^{\mathrm{opt}}$ and t_{E}. Again solid lines indicate our strategy, while dashed and dotted lines refer to bounds set by perfect knowledge and average over all k respectively. The figure confirms the optimum, although we note some level 2 gain for the smallest design time, and we observe an overall insensitivity to $t_{\mathrm{M}}^{\mathrm{opt}}$: Setting t_{M} to $t_{\mathrm{M}}^{\mathrm{opt}}/2$ or $2t_{\mathrm{M}}^{\mathrm{opt}}$ has very little impact on loss.

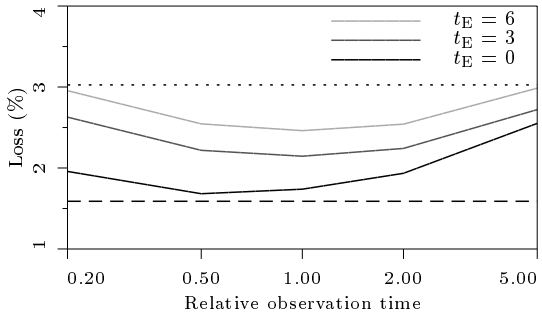

Figure 8: *Network loss vs. observation interval.*

Design Evaluation

The question of whether to update a network or not is a balance between the costs of updating and the gains resulting from improved grade of service. Since the latter are not exactly known when a decision is made, a decision must be based on predictions.

Consider a network with channel allocations $m(o,d)$, which are based on traffic estimates $\hat{A}(o,d)$, and suppose that a new design $m^*(o,d)$, which is based on the most recent traffic estimates $\hat{A}^*(o,d)$, is considered for

implementation. Denote the cost of implementing the design by C and the expected gain by G and again for convenience drop the reference to o, d.

The cost C consists of a fixed cost C_{I} for the implementation itself and costs for packing, rerouting and clearing, which amount to C_{P}, C_{R}, and C_{C} respectively per transaction. The number of transactions can be worked out by performing an imaginary update.

The gain G is the sum of all traffics carried in the future if and only if the design is implemented. Excluding random fluctuations between traffic estimates $\Delta\hat{A} = \hat{A} - \hat{A}^*$ we therefore write $G = \sum_{o,d} P\hat{A'}^*\tau$, where P is the probability that $\Delta\hat{A}$ is not a result of stochastic variations, $\hat{A'}^* = \hat{A}^* \left[E_m(\hat{A}^*) - E_{m^*}(\hat{A}^*)\right]$ is the additional traffic carried, and τ is the time during which present conditions are expected to last (we have set τ to its minimum value t_{U}).

To estimate P, we assume that traffic estimates are normally distributed with the mean and variance given above. Then, if $\hat{A}^*$ is an independent sample of the same traffic as $\hat{A}$, both have the same mean and variance and we may define a normalised difference $Z = \Delta\hat{A}/\sqrt{2\,\hat{A}/t_{\mathrm{M}}}$ which is N(0,1)-distributed and we get $P \approx 2\Phi(|Z|) - 1$, where Φ denotes the distribution function for the standard normal distribution, and we have used a two-sided test.

Figure 9 shows network cost versus τ/t_{U} as observed in our simulator, with different values of t_{E} and bounds as in figure 8. The curves seems to confirm our conservative approach in estimating the life time $\tau = t_{\mathrm{U}}$ of a network update.

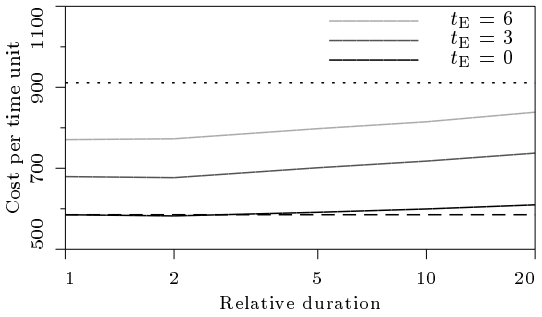

Figure 9: *Network cost vs. assumed design life time.*

Updating Interval

Selecting a proper updating interval t_{U} is again a question of balancing costs and gains. We define the optimum updating interval as the one for which the expected, total cost takes its minimum, $t_{\mathrm{U}}^{\mathrm{opt}} = t_{\mathrm{U}} : \min_{t_{\mathrm{U}}} \sum_{k=1}^{K} C_k(t_{\mathrm{U}})$, where $C_k(t_{\mathrm{U}})$, the costs paid during period k with an updating interval of t_{U}, is a sum of management costs and costs for lost traffic.

Management costs relate to transmission of traffic measurement data to the NMC, and to computing and analysing a new design at the NMC. This process is repeated at intervals of t_{U}, at a cost of C_{T} each time.

Costs for lost traffic are caused by delaying network updates. The total delay can be seen as a sum of an initial delay and a possible additional delay, where the former refers to the time until the first, new design can be implemented, and the latter, which occurs if the first redesign is rejected, is the remaining time until an update is made. The initial delay consists of the time until a change is first reported to the NMC, on the average $t_{\mathrm{U}}/2$, plus the time t_{E} required to compute, evaluate and implement a new design, and the average additional delay is $H_k(t_{\mathrm{U}})\,t_{\mathrm{U}}$, where $H_k(t_{\mathrm{U}})$ is the expected number of updating cycles until a new design is accepted and implemented. Assuming that call loss only occurs when $A_k > A_{k-1}$ and no redimensioning takes place between $k-1$ and k, we get $C_k(t_{\mathrm{U}}) = T/t_{\mathrm{U}}\, C_{\mathrm{T}} + [t_{\mathrm{U}}/2 + t_{\mathrm{E}} + H_k(t_{\mathrm{U}})\,t_{\mathrm{U}}] \sum_{o,d} \Delta A_k^+$, where $\Delta A_k^+ = \max(A_k - A_{k-1}, 0)$ and we again have omitted references to o, d.

To estimate $H_k(t_{\mathrm{U}})$ we consider the probability $h_k(t_{\mathrm{U}})$ that the first traffic estimate containing information about a new traffic k will lead to a network redesign. Denoting estimated costs and gains for a redesign by c_k and g_k respectively, we get $h_k(t_{\mathrm{U}}) = \mathrm{Prob}(c_k < g_k)$.

Neglecting the processing of individual calls, the cost c_k is fixed at $c_k = C_{\mathrm{I}}$, and the simplifiying assumption on loss gives the gain g_k as $g_k \approx \sum_{o,d} p_k \Delta A_k^+ \tau$, where p_k is an approximation of the significance of ΔA_k^+. Analogous to above we set $p_k \approx 2\Phi(z_k) - 1$ with $z_k = \Delta A_k^+ / \sqrt{2\,A_k/t_{\mathrm{M}}}$. Noting that design evaluation is based on estimates, we observe that g_k is a sum of functions of independent random variables hence the mean $E\{g_k\}$ and variance $V\{g_k\}$ may be be obtained by summing means and variances of each term. (These are obtained from from the moments of ΔA^+ and A, see *e.g.* [10, pp. 231–232]). Setting $h_k(t_{\mathrm{U}}) \approx \Phi([E\{g_k\} - c_k]/\sqrt{V\{g_k\}})$ and assuming that successive updating attempts also are equal to $h_k(t_{\mathrm{U}})$ and independent, we finally obtain $H(t_{\mathrm{U}}) = h(t_{\mathrm{U}})/[1 - h(t_{\mathrm{U}})]$.

Figure 10 shows network cost versus $t_{\mathrm{U}}/t_{\mathrm{U}}^{\mathrm{opt}}$, with different values of t_{E} and bounds as before. As before, the figure confirms the optimum and again displays an insensitivity to deviations in t_{U} from the optimal value.

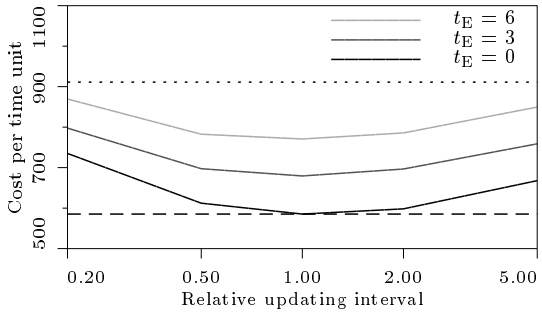

Figure 10: *Network cost vs. updating interval.*

Performance

In a real network, T are A_k are neither known, nor do they actually exist. However, fixing T to a period of reasonable stability, say one hour, one would have a fair idea of of average traffics per hour and OD-pair for a complete 24 hour period. To represent the errors of such a forecast, and natural fluctuations such as from

day to day, we assume that actual values of T and $A_k(o,d)$ are drawn from a family of independent, normal distributions all of which have the forecast mean and same coefficient of variation which is taken as the measure of uncertainty.

Figure 11 shows cost as a function of forecast uncertainty, with different values of t_E and bounds indicated as above. It is seen that our strategy lies close to the lower bound already for perfect forecasts and, starting at an uncertainty of 2% for the shortest design turn-around time, successively outperforms the "lower bound" as the uncertainty increases.

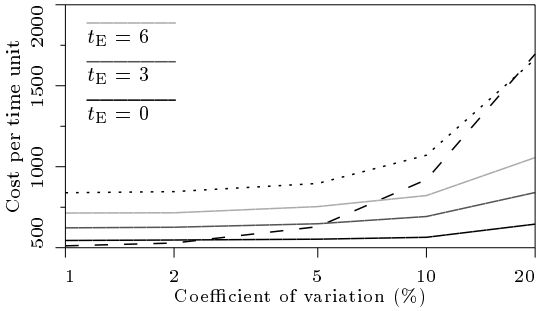

Figure 11: *Network cost vs. forecast uncertainty.*

Having demonstrated the efficiency of our strategy, we now turn to a number of alternatives. An obvious first alternative is to skip the dynamics and redesign VPNs directly according to forecasts. This is, in fact, the lower bound in figure 11. Clearly, this alternative works only with accurate predictions and small daily variations.

A second method may be to use advanced routing schemes. This will certainly reduce loss, but, as shown in [8], a better way to use such a technique is to use VPM as suggested for level 1 variations and reserve session routing for level 2 variations. In this way is each technique used for what it is best suited, and a double improvement is in fact obtained.

Finally, a third method may be to overprovision capacity rather than reallocating the existing one. Figure 12 shows the capacity increase that would be required to compensate for the rearrangements taking place in our examples, and we note an increase from 50% for the practically ideal case up to considerable 150% for variations of 20%.

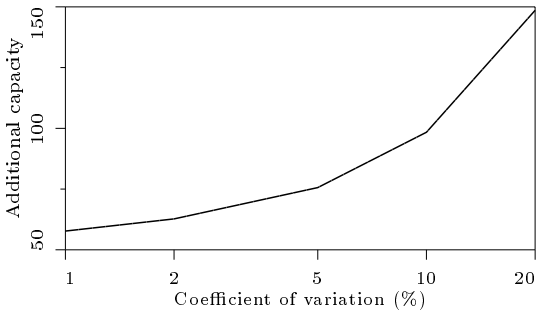

Figure 12: *Capacity increase vs. uncertainty.*

Conclusions

We have presented a simple and robust technique to control slow traffic variations by means of VPs. Given rough estimates of expected traffic profiles and of the relationship between costs for lost calls and traffic management respectively, the strategy reallocates transmission resources according to demands in a cost-effective way.

Further research in this area includes improved measurement techniques, possibly with filtering and forecasting, refined models for optimal updating, and decentralised strategies which do not depend on a single NMC.

Acknowledgement

The generous supports of the Department of Communications Systems, Lund Institute of Technology, Sweden, and of the Teletraffic Research Centre, The University of Adelaide, S.A., Australia, are gratefully acknowledged.

References

[1] Arvidsson, Å.: *On the Performance of a Circuit Switched Link with Priorities*, IEEE J. Sel. Areas in Commun., vol. 9, no. 2, pp. 212–219 (1991).

[2] Arvidsson, Å.: *Management of Reconfigurable Virtual Path Networks*, pp. 931–940 in Proc. 14th Int. Teletraffic Cong., Antibes Juan-Les-Pins, France (1994).

[3] Arvidsson, Å.: *High Level B-ISDN/ATM Traffic Management in Real Time*, paper no 2.5 i Proc. 2nd IFIP TC6/WG6.4 Workshop on Perf. Modelling and Eval. of ATM Netw., Bradford, U.K. (1994).

[4] 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).

[5] Filipiak, J. & Chemouil, P.: *Modeling and Prediction of Traffic Fluctuations in Telephone Networks*, IEEE Trans. on Commun., vol. 35, no. 9, pp. 931–941 (1987).

[6] Filipiak, J.: *Automatic Network Management Controls*, IEEE Trans. on Commun., vol. 39, no. 12, pp. 1776–1786 (1991).

[7] Gersht, A. & Shulman, A.: *Optimal Routing in Circuit Switched Networks*, IEEE Trans. on Commun., vol. 37, no. 11, pp. 1203–1211 (1989).

[8] Gopal, G., Kim, C.-K. & Weinrib, A.: *Algorithms for Reconfigurable Networks*, pp. 341–347 in Proc. 13th Int. Teletraffic Cong., Copenhagen, Denmark (1991).

[9] 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).

[10] Kendall, M. & Stuart, A.: *The Advanced Theory of Statistics*, Charles Griffin & Co. Ltd., London (1962).

[11] Mase, K. & Imase, M.: *An Adaptive Capacity Allocation Scheme in Telephone Networks*, IEEE Trans. on Commun., vol. 30, no. 2, pp. 354–359 (1982).

[12] Mase, K. & Shioda, S.: *Real-Time Network Management for ATM Networks*, pp. 133–140 in Proc. 13th Int. Teletraffic Cong., Copenhagen, Denmark (1991).

[13] Pióro, M. & Gajowniczek, P.: *Stochastic Allocation of Virtual Paths to ATM Links*, paper no 2.1 i Proc. 2nd IFIP TC6/WG6.4 Workshop on Perf. Modelling and Eval. of ATM Netw., Bradford, U.K. (1994).