High Level B-ISDN/ATM Traffic Management in Real Time

Åke Arvidsson^{a*}

^aDepartment 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.

ABSTRACT

We study real time traffic management in the perspective of a four-level model of traffic variations: cells, bursts, sessions, and links. Our interest is focused on the two latter levels: Using virtual channels (VCs) and virtual paths (VPs) as the managed entities on the two levels respectively, we put forward a simple and robust strategy for traffic management in real time: All nodes simultaneously monitor offered traffics and forward the result to a network management centre (NMC); The NMC computes a new set of optimal VPs and compares the result to the existing network; If a change appears profitable, the necessary information is sent back to the nodes and the new design is implemented.

Applying our strategy to control general networks subject to traffics which in advance are known only as expectations, we develop simple methods for traffic observation, the optimal observation interval, a fast and efficient algorithm to compute VP networks, a test to determine whether to implement a new design in which costs related to lost traffic are compared to processing costs, and the optimal updating interval.

Running our strategy in a simulator, we discover a considerable adaptivity, large savings in transmission capacity, and a performance far better than achieved by fixed redesign according to forecasts. Finally we discuss limits to the range of variations within which our strategy can be applied.

The present work applies both to networks using SDH/SONET and/or networks based on ATM, *i.e.* both to present public switched telephony networks and broadband integrated services digital networks, and is a summary and extension of earlier works within this area [3–9].

Keyword Codes: C.2.1, C.2.3

Keywords: Network Architecture and Design; Network Operations

1. INTRODUCTION**1.1. Traffic Management**

Any telecommunications network is subject to traffic variations, some of which are related to human factors, such as office hours, while others are technology dependent, for example compression of video information. It is the task of network operators to cope

*The work reported herein was partly carried out at the Teletraffic Research Centre, The University of Adelaide, Adelaide, South Australia 5005, Australia.

Table 1

Hierarchical classification of traffic variations, cause, time scale, and suitable management means.

Level	Source of variation	Time scale	Management means
1 Link	User behaviour	h	Capacity reallocation
2 Session	Equipment usage	min	Overflow, routing
3 Burst	Information generation	s	Larger buffers, flow control
4 Cell	Cell formation	ms	Smaller buffers, policing

with these variations and allocate sufficient resources according to demands. We refer to this process as traffic management.

Traffic management issues have been studied for almost as long as telecommunications have existed. It is well known, for example, that fewer calls are placed in the night than in the day, and that not all users use their equipment simultaneously. These variations have usually been handled by applying the concept of a busy hour for the time of the day, and a Poisson model for non-simultaneous usage. Similarly, the bursty nature of computer communications led to the development of packet switching as a means of better coping with varying activity within an ongoing call.

Current interest in traffic management is motivated by a number of factors: Services with widely different traffic characteristics and quality of service demands must be jointly managed in the integrated services digital network (ISDN) and in the broadband-ISDN (B-ISDN); Recent technological achievements affect network design as advances in optical fibre technology means that transmission becoming faster, cheaper and more reliable and switching technology has taken a new course with the introduction of the synchronous digital hierarchy (SDH) or its equivalent the slotted envelope network (SONET); User needs are changing with the continuous deployment of new equipment and new applications at the user's end.

Our view of traffic management follows a layered approach as suggested in [19, 20, 34] and others. Table 1 shows a simple, hierarchical view of traffic variations and traffic management in a B-ISDN using the asynchronous transfer mode (ATM). Each level defines a certain cause and associated time frame, and our idea is to apply similarly structured controls, where separate entities are employed to manage the portion of traffic variations that fall within each of the time scales.

The **first level** refers to variations in the number of potential users, *i.e.* the number users in a position where they may use their equipment. Typical factors on this level include weekdays, weekends, office hours, luncheon breaks *etc.*, for which a suitable time scale is on the order of hours. Traditionally, level one variations have been met by variable routing schemes, *e.g.* DNHR in the U.S.A., but modern technology such as SDH, SONET and ATM VP-addressing permits what in fact is link redimensioning.

The **second level** refers to variations in the number of active users, *i.e.* the number of potential users actually using their equipment. These variations are typically considered to be random and therefore described by stochastic means, typically call arrivals are assumed to form a Poisson process. A suitable time scale may be on the order of minutes. Variations of this type is preferably met by common statistical multiplexing of sessions

combined with overflow arrangements and good routing schemes.

The **third level** refers to variations in user behaviour, *i.e.* the number of active users currently generating information. Such variations typically refer to *e.g.* alterations between listening and talking (voice), screen modifications versus manual typing (data), and completion of video frames (image). It is seen that variations on this level, typically on the order of seconds, may display both a deterministic pattern and a random behaviour. Typical management means include statistical multiplexing via buffers on which flow control is applied. The size of the buffer may depend on application demands such as loss, delay, and jitter, and the flow control, for example window schemes and explicit congestion notifications, is traditionally carried out in software.

The **fourth level** refers to variations in user delivery, *i.e.* the number of users currently presenting information to the network. Variations on this level reflect *e.g.* cell adaptation routines carried out by the ATM adaptation layer (AAL) which takes place, say, on the order of milliseconds. This level is a particular feature of ATM and hence there is no previous experience to build on. Current management proposals include buffers combined with numerous versions of policing, spacing, and fast reservation mechanisms. High transmission rates and traffic integrity suggest that control must be carried out in hardware and that only limited buffering is meaningful.

While most papers on traffic management in B-ISDN deal with buffer problems of ATM, levels three and four in our terminology, our main concern is the two upper levels. Our work is based on virtual paths (VPs) as the managed entity on level one and virtual channels (VCs) on level two. In particular, we propose a simple and robust strategy for managing VPs in SDH/SONET and/or ATM networks in real time. Our strategy consists of on-line measurements of offered traffics followed by the design and possible implementation of a new virtual path network, cyclically repeated in a relevant time scale. We also discuss the role of VCs and advanced call routing on the second level.

1.2. Virtual Paths and Virtual Networks

A 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. A network of VPs, a virtual network (VN), forms a higher layer which is logically independent of the underlying physical network. It is pointed out that the path concept of SDH/SONET, and the path addressing of ATM, both lend themselves well to support VNs.

VNs are designed to handle current traffic demands with an acceptable grade of service, or, if all demands cannot be accommodated, 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 changes. We refer to this process as VP management (VPM), other names include bandwidth switching [1], capacity management [13], bandwidth management [29], and bandwidth control [42].

Besides the general appeal of a structured approach to traffic management, some of the motives behind VPs and VNs [2, 11–16, 21, 33, 34, 41, 42] are: Reduced costs resulting from simplified transit exchanges; Faster call handling by excluding intermediate node processing at set-up time; Improved traffic management capabilities such as possibilities

to redirect traffic in a congested or faulty network; A means for providing customer-dedicated, closed VNs. For B-ISDN type networks, we may additionally gain simplified statistical multiplexing and grade of service control by grouping services into service classes (SCs) according to their characteristics, *e.g.* average rate, peak rate and burst length, and demands, *e.g.* loss, delay and jitter, and carrying each class on a separate VN.

Referring to the model in table 1, VPM can be regarded as the portion of traffic management that is carried out on level one. The three lower layers are typically assumed to be automatic and run with a minimum of manual interaction. Our aim is to extend this to the upper level of the model. To achieve this, we wish to devise an algorithm that can design VNs in real time, and a real time VPM strategy with the following properties: (1) Maximisation of carried traffic through efficient capacity redistribution, (2) Minimisation of resources spent on VPM itself, (3) No excessive or unrealistic demands on input data, and (4) Robustness against uncertain predictions of offered traffics and their variations.

2. PRELIMINARIES

We consider networks with N on which VPs, and consequently VNs, can be formed. Each VP has a number M of VCs associated to it. The number of VCs depends on the capacity of the VP, the buffer space for statistical multiplexing of cells and bursts, and service requirements such as loss, delay and jitter for the class in question. All requests for which a VC 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) $o, d = 1, \dots, N$ is offered a pure chance traffic, expressed in Erlangs, the rate of which changes every T th time unit. We denote the traffic offered between o and d during the k th interval by $A_k(o, d)$. The sequence of traffics is limited to $k = 1, \dots, K$ and cyclically repeated, so that $A_k(o, d)$ is followed by $A_{k+1}(o, d)$ and so on, and $A_K(o, d)$ is followed by $A_1(o, d)$.

Offered traffics are not known *a priori*, the only information available is forecast averages of $A_k(o, d)$, which are obtained *e.g.* through long term measurements. For any given OD-pair and time, the traffic actually offered is assumed to deviate randomly from its forecast value according to a normal distribution. We denote the mean of this distribution by $A_k(o, d)$ and its standard deviation by $C_v A_k(o, d)$ where C_v is the coefficient of variation.

In terms of the model in table 1, we thus have a Poisson model on level two, with a time scale on the order of 1, and partially known, step wise variations with a time scale of T on level one.

Applying our strategy, all nodes simultaneously monitor offered traffics during intervals of t_M time units, with new intervals commencing every t_U th time unit. Traffic estimates are forwarded to the network management centre (NMC) together with information on link states (up/down). Using this information, the NMC then computes an updated VN and analyses the result. If implementing the new design appears profitable, the necessary information is sent back to the nodes and the design is implemented. Transmitting traffic

information to the NMC, computing and analysing the design, returning results to the nodes, and implementing the design is assumed to take a total of t_E time units.

Implementing new designs may result in that sessions in progress must be moved from one physical path to another, or that VPs no longer can support all sessions in progress. We will not consider the former further here, but the latter calls for some attention. Our policy is to provide two-VP rerouting over the least busy alternative if possible: 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. If all paths are blocked, premature clearing is used as a last resort. Rerouting is further combined with limited repacking: At updating intervals are rerouted sessions moved back to direct routes if possible, but without optimisation.

While it is clear that frequent rearrangements of VPs will give faster responses to changes in offered traffics, and hence increase the traffic carrying ability of a network, it is also clear that the more frequent the updating, the more resources are spent on transmitting and processing management information, and on computing and implementing redesigns. Hence, optimal VPM is a balance between maximising the carried traffic and minimising the resources spent on management.

To compare the two, we have chosen the following costs: In between two redesigns, a penalty of C_L is earned for every call attempt lost. At the point of a possible redesign is a fixed penalty of C_T assigned for transmitting data to the NMC, computing and analysing the design. Further, if the design is implemented, a fixed cost of C_I is paid for the implementation itself, and a variable cost for processing sessions in progress: C_P for every call not routed on its direct path, C_R for every call that cannot be carried on its direct path, and C_C for every call that cannot be carried at all.

Throughout this paper, we have used a set of test networks and a simulator which implements the above strategy. The set of test networks comprises eight different physical networks, each of $N = 20$ nodes and each for which a series of $K = 8$ distinct sets of offered traffics was provided. For details on network topologies, transmission capacities and offered traffics, refer to appendix A. 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 divided in two major parts: Algorithms for VN design and strategies for real time management respectively. Focusing on algorithms, we begin by giving a brief survey of algorithms proposed in the literature. Finding shortcomings in all of these, we then give an algorithm of our own that combines speed and simplicity with accuracy and flexibility. Turning our attention to strategies, we first evaluate some simple techniques to estimate offered traffics, after which the the optimal observation interval is derived. Next we study criteria for the profitability of implementing a particular VN design and derive the optimal updating interval. Applying the complete strategy, we then examine the performance of our strategy when actual traffics vary randomly around predicted values and compare the results to alternative methods by which level one variations can be met. Lastly we give some experimental results to highlight the boundary between level one and two in the model of table 1.

3. NETWORK DESIGN ALGORITHMS

3.1. Existing Algorithms

VPM must be supported by efficient algorithms to compute VN designs. We have found such algorithms in Evans [17], Gersht *et al.* [22, 23, 37], Gopal *et al.* [26, 27], Herzberg [29–31], Mase *et al.* [39], and Pióro *et al.* [43]. Algorithms are also outlined by Hui *et al.* [35]. Ash *et al.* [10] consider a similar problem for long term network construction, Gerla *et al.* [21] study DCM for packet-switched networks, and, finally, Mase *et al.* [40] discuss such algorithms in terms similar to those in [17, 22, 23, 35, 37], 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). In general, this is not true, although, for certain sources and a large number of simultaneous connections, it may be a good approximation.
- Most algorithms explicitly or implicitly presume the existence of predefined paths or sets of paths for all VPs. Deciding on paths in advance may be complicated and imposes unnecessary restrictions to the optimisation problem. Moreover, very large data bases are required to be prepared for various situations of outages and unbalanced loads.
- Some algorithms produce real valued solutions which are not immediately useful in SDH/SONET-networks.

In earlier works, we proposed an algorithm that does not require linear equivalent bandwidth nor predefined paths, that produces integer valued solutions and for which the optimisation function can be chosen arbitrarily. We will now outline a new version which is considerably faster (our networks below are typically handled on the order of seconds on a Sun work station SPARC ELC, *i.e.* approximately ten times faster) and in most cases produces more efficient designs (comparing the two for our networks below, the new algorithm gives smaller loss and higher utilisation in more than 9 cases out of 10, and comes within a fraction of a percent in the others).

Similar to the older version, the algorithm is a heuristic hence there is no guarantee that the final solution is a global optimum. On the other hand, the optimum guaranteed by some of the algorithms above need not be a truly global one since it is restricted by a fixed set of paths. The importance of such restrictions is hard to estimate, but it is known that unrestricted, dynamic path selection is vital to network performance for dynamic call routing [25], *i.e.* on level two of our model.

3.2. Our Algorithm

Letting $s = 1, \dots, S$ denote an arbitrary SC and $o = 1, \dots, N$ ($d = 1, \dots, N$) 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, expressed as units of capacity, and number of VCs. (A “unit of capacity” is, for PDH a number of 64

kb/s channels, for SDH/SONET the capacity of the smallest virtual container used, and for pure ATM a rate in cells/second large enough to carry a call of any SC.)

2. Read link capacities $C_{o,d}$ and offered traffics $A_{s,o,d}$ (expressed as Erlangs).
3. Assign high, initially acceptable call loss levels $\epsilon(s, o, d)$ for all SCs s and all OD-pairs o, d .
4. Find the shortest paths, if any, along which any traffic s, o, d can be granted more capacity.
5. Identify all traffics s, o, d that suffer from loss levels above $\epsilon(s, o, d)$ and for which a path is available.
 - (a) If at least one such traffic is found, proceed to step 6.
 - (b) If no such traffic is found because all $\epsilon(s, o, d)$ are ultimately acceptable or because no more capacity can be supplied to traffics s, o, d still suffering from unacceptable loss, then stop.
 - (c) Otherwise lower $\epsilon(s, o, d)$ and repeat step 5.
6. Allocate one unit of capacity to each traffic s, o, d identified above along their shortest paths respectively.
7. If the capacity allocated on each link is within the physical bounds, then repeat from step 4, else
 - (a) For each traffic s, o, d , compute the loss L paid if the currently allocated unit of capacity was removed.
 - (b) For each traffic s, o, d , compute the relative gain G if the currently allocated unit of capacity was removed.
 - (c) Find the traffic $s_{\max}, o_{\max}, d_{\max}$ with the highest gain/loss ratio.
 - (d) Remove the unit of capacity currently allocated to traffic $s_{\max}, o_{\max}, d_{\max}$ and repeat step 7.

Available capacity is successively distributed among VPs in the order of need and so that a minimum amount of capacity is used in each step. In case of overallocation is capacity removed so that conflicts can be resolved at a minimum overall loss. 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 , $i = 1, 2, \dots$, units of capacity, *i.e.* the equivalent number of channels. The tables are computed from traffic characteristics, grade of service demands, buffer space and acceptable loss, see earlier works and *e.g.* [28, 32, 40].

Our initial loss level ϵ in step 3 is set to 99%. It is then successively reduced in step 5c by dividing ϵ by 1.1 until at 5b the ultimately acceptable level of 0.05% is reached.

Shortest paths in step 4 are determined using Floyd's algorithm [38], with the length l associated to link o, d designed to avoid nodes subject to heavy load and links with little remaining capacity:

$$l(o, d) = \begin{cases} N^{1/2} \sqrt{\frac{\max(\delta, \max(T'_o, 0) + \max(T'_d, 0))}{C'_{o,d}}} & C'_{o,d} > 0 \\ \infty & C'_{o,d} = 0 \end{cases} \quad (1)$$

where T'_n is the total, remaining number of capacity units required by traffics originating from n to fulfill our ultimately acceptable blocking level, δ is a very small number (we have used $\delta = 10^{-9}$), and $C'_{o,d}$ denotes the remaining, not yet assigned capacity on link o, d . Finally, realising that equation (1) may favour series of very high capacity links to direct ones of lower capacity, paths are "flattened" to remove such anomalies.

Traffic selection in step 5 aims at keeping variations in grade of service between OD-pairs to a minimum. Noting that this can reduce the total traffic carried by the network, we give slightly preferential treatment to traffics carried on shorter paths by relating the current loss to the requested capacity and obtain a modified loss $\eta_{m(s,o,d)}(A(s, o, d))$ on which the need for further capacity is assessed

$$\eta_{m(s,o,d)}(A(s, o, d)) = \frac{E_{m(s,o,d)}(A(s, o, d))}{\ln(h(s, o, d) + 1)} \quad (2)$$

where $E_m(A)$ is the Erlang-B formula, A the offered traffic, m the number of channels currently available, and h is the number of links included in the proposed, shortest path from o to d . In other words, we compute losses $E_m(A)$, which are modified to $\eta_m(A)$, and then compared to $\epsilon_m(A)$. Allowing for non-linear equivalent bandwidths, we sum circuits rather than bandwidths and compute the number of circuits for each route of the VP individually, by means of the tables referred to in step 1, and then sum to get m .

$L(s, o, d)$ in step 7a is the s, o, d -traffic that the most recently added unit of capacity to VP s, o, d is expected to carry,

$$L(s, o, d) = A(s, o, d) [E_{m(s,o,d)}(A(s, o, d)) - E_{m'(s,o,d)}(A(s, o, d))] \quad (3)$$

where m and m' respectively refer to the number of channels currently available and what would be available if the most recently added unit of capacity was removed. Again, m and m' are determined by summing the values for each physical route.

$G(s, o, d)$ in step 7b is the sum of all losses that can be avoided at the same point if traffic s, o, d was selected to loose its most recently added unit of capacity,

$$G(s, o, d) = \sum_{s'=1}^S \sum_{o'=1}^{N-1} \sum_{d'=o'+1}^N I(\mathcal{L}_{s,o,d} \cap \mathcal{L}_{s',o',d'} \neq \emptyset) L(s', o', d') \quad (4)$$

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.

Table 2

Comparison of network performance for designs obtained from our algorithm (A) and an earlier proposal (B).

Algo- rithm	Network loss			OD-pair loss			
	E{Loss}	Best	Exp./Max.	E{Loss}	Best	S{Loss}	Best
A	1.6%	57	0.4%/1.4%	1.3%	64	1.6%	64
B	2.0%	7	0.1%/0.4%	3.1%	0	7.6%	0
	Utilisation		Usage factor				
	E{Util.}	Best	E{Us. f.}	Best			
A	4.0	64	1.3	8			
B	3.9	0	1.4	0			

3.3. Discussion

Our method aims at distributing the available capacity among traffics and OD-pairs in a fair way, yet so that the total carried traffic is maximised. Unlike methods based on mathematical programming, our algorithm will remain stable and converge at the same speed for a wide range of non-linearities and discontinuities and irrespective of the ways in which routes for VPs are chosen.

These properties allow us to modify distances and gains in order to include *e.g.* differences in revenues related to OD-pairs, SCs, and allocated resources. For example, callers may be charged according to the distance from origin to destination and the time of the day. Similarly, the cost for the capacity carrying the call may depend on a complicated tariff structure set by an external provider of transmission facilities, in particular for long-distance and international connections. We may also impose biased selection of routes, a limitation to the number of distinct physical routes, and predetermined routes.

3.4. Numerical Results

To investigate the power of the proposed algorithm (A), it was applied to our set of test networks. To enable simple comparisons to the comparable algorithm (B) of [26, 27], the number of SCs was set to one. (Results on two SCs, voice and frame relaying, are found in earlier works.) The predefined paths required by B were taken as the four most used ones found by A.

Results are summarised in table 2. Network loss gives is the loss averaged over the entire networks; for each algorithm we give average loss, the number of times it scored the best, and, for these occasions the absolute values of the maximum and expectation of the differences noted. OD-pair loss is the loss averaged over the OD-pairs; for each algorithm we give the expected loss, the times it scored the best, and the standard deviation of the loss. Utilisation records the carried traffic per allocated unit of capacity; we give expected utilisations and the number of times each algorithm scored the best. Finally, the usage factor is the number of OD-pairs served per physical unit of capacity under different traffic conditions; we give the expectation for each algorithm and, noting that a low number means less switching, the number of times it scored the best.

We note that, for the networks and traffic patterns considered, our algorithm will in most cases give a lower overall loss and that, if this is not the case, the difference to the other algorithm will be small. Turning our attention to performance per OD-pair, our

algorithm significantly outperforms the other one in that it for all cases not only produces a lower loss but also distributes losses more evenly. Finally, the table suggests that this is achieved without sacrificing network utilisation, for which a small but distinct increase is achieved, and without increasing switching, for which a small but distinct reduction is recorded.

4. APPLICATION STRATEGY

4.1. Traffic Observation

The first step in our strategy consists of nodes observing offered traffics and reporting results to the NMC. We therefore need a method according to which traffics are observed and a way to compute t_M , the time during which observations takes place.

4.1.1. Method

For our purpose, only short term, on-line measurements, possibly combined with short term forecasts, are relevant. We have chosen two basic approaches, arrival counting (AC) and carried traffic measurements (CT).

We first consider 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

$$E\{\hat{A}\} = \sum_{k=0}^{\infty} \frac{k}{t} \text{Prob}(N(t) = k) = \dots = A \quad (5)$$

and

$$V\{\hat{A}\} = \sum_{k=0}^{\infty} \left(\frac{k}{t}\right)^2 \text{Prob}(N(t) = k) - A^2 = \dots = \frac{A}{t} \quad (6)$$

Proceeding to CT, the number of busy channels 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 $\hat{A}' = \frac{1}{t} \int_t a'(\tau) d\tau$. This, in turn, gives an estimate of the offered traffic by “backward Erlang computation”, *i.e.* by solving for $\hat{A}$ in $\hat{A}' = \hat{A}(1 - E_M(\hat{A}))$. (For estimation of A' only, see also *e.g.* [18, 20].) First, assuming a loss of 0, hence $\hat{A} = \hat{A}'$, and negative exponential call holding times, we find

$$E\{\hat{A}\} = E\left\{\frac{1}{t} \int_{\tau=0}^t a'(\tau) d\tau\right\} = \dots = A \quad (7)$$

and

$$V\{\hat{A}\} = V\left\{\frac{1}{t} \int_{\tau=0}^t a'(\tau) d\tau\right\} = \dots = \frac{2A}{t} \left(1 - \frac{1 - e^{-t}}{t}\right) \quad (8)$$

Comparing AC to CT, it is noted that the latter provide better accuracy if (8) < (6), *i.e.* if $t \lesssim 1.5936$.

For a loss > 0 , however, CT fails as backward Erlang will introduce a positive bias to $\hat{A}$. In short, this is because of the non-linearity of the Erlang loss function. Figure 1 shows

Figure 1. Traffic estimation bias for CT measurement technique versus observation interval for various points of operation.

the bias recorded by applying CT measurements on a simulated trunk with $M = 100$ (left) and $M = 1000$ (right) channels and with $E_M(A) = 0.5\%$, 1.0% , 2.0% and 5.0% respectively.

In conclusion, CT cannot be used for real-time estimation of offered traffics, and AC is chosen as our estimation method. AC also has the advantage of ease of implementation: Counting the number of arriving service requests is straightforward, and is actually already implemented in many switching systems, *e.g.* as a part of the overload control of the system itself.

4.1.2. Observation Interval

We now turn to the problem of selecting a proper observation interval $t = t_M$ for AC. On one hand, equation (6) shows that the variance of the estimate becomes smaller the longer the observation interval. On the other hand, we consider traffic processes the rate of which varies with time, and the longer the interval, the more old, possibly invalid, information will be contained in the estimate. In other words, the optimal observation interval is a balance between low variance (large t), and exclusion of invalid information (small t).

Consider an arbitrary OD-pair o, d , an arbitrary traffic interval k of length T , and let $\hat{A}$ be an estimate of $A_k(o, d)$ used to redesign the network during this interval. We define the optimal observation interval with respect to k, o, d , $t_M^{\text{opt}}(k, o, d)$, as the one for which the expected, squared error of $\hat{A}$ takes its minimum,

$$t_M^{\text{opt}}(k, o, d) = t_M : \min_{t_M} E\{(\hat{A}(t_M) - A_k(o, d))^2\} \quad (9)$$

For convenience we temporarily drop the reference to o, d . Moreover, we assume that A_k confirms exactly with the forecast average, and ignore possible information on expected deviations in our analysis.

Due to transmission and processing time, designs implemented during the first t_E time units of the interval (1), will be based on measurements made during $k - 1$. Similarly, designs completed during the following t_M time units (2) will be based on measurements covering both $k - 1$ and k . For the rest of the interval (3), however, all designs completed will be based on information relating to k . Assuming that designs are completed with the same probability throughout the interval, we find

$$\int_{\tau=0}^{t_E} E\{(\hat{A} - A_k)^2\} \frac{d\tau}{T} = \dots = \left((A_{k-1} - A_k)^2 + \frac{A_{k-1}}{t_M} \right) \frac{t_E}{T} \quad (10)$$

$$\int_{\tau=0}^{t_E+t_M} E\{(\hat{A} - A_k)^2\} \frac{d\tau}{T} = \dots = \left(\frac{(A_{k-1} - A_k)^2}{3} + \frac{A_{k-1} + A_k}{2t_M} \right) \frac{t_M}{T} \quad (11)$$

$$\int_{\tau=t_E+t_M}^T E\{(\hat{A} - A_k)^2\} \frac{d\tau}{T} = \dots = \left(\frac{A_k}{t_M} \right) \frac{T - t_E - t_M}{T} \quad (12)$$

for the three subintervals respectively. Minimising the sum of (10)–(12) by differentiating with respect to t_M and solving for a positive, real valued root gives

$$t_M^{\text{opt}} = \frac{\sqrt{3A_k T + 3(A_{k-1} - A_k)t_E}}{|A_{k-1} - A_k|} \quad (13)$$

If the difference between A_{k-1} and A_k is small, (13) may give $t_M^{\text{opt}} > T - t_E$, 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_M^{opt} as the minimum of $T - t_E$ and (13).

Extending the result to networks, we compute an overall t_M^{opt} by weighting $t_M^{\text{opt}}(k, o, d)$ for each k, o, d by the expected errors respectively

$$t_M^{\text{opt}} = \sum_{k,o,d} \frac{|\hat{A} - A_k(o, d)|}{\sum_{k,o,d} |\hat{A} - A_k(o, d)|} t_M^{\text{opt}}(k, o, d) \quad (14)$$

Applying (14) to our set of test networks, we typically obtain $t_M^{\text{opt}} \approx 6, 5$, and 4 for $t_E = 0, 3$, and 6 respectively. Figure 2 shows network loss versus t_M/t_M^{opt} as observed in our simulator, with different values of t_E indicated by solid lines. Dotted and dashed lines indicate upper and lower bounds respectively: The former refers to a fixed network which is permanently dimensioned for average traffics throughout all k , and the latter to deterministic, “ideal” management were exact information about all traffics is available in advance, and the network is updated exactly when traffics change.

The figure confirms the optimum and we observe an overall insensitivity to t_M^{opt} : Setting t_M to $t_M^{\text{opt}}/2$ or $2t_M^{\text{opt}}$ has very little impact on the loss. This means that (14) provides a robust estimate in the sense that even poor forecasts of $A_k(o, d)$ will yield a t_M^{opt} that performs close to optimal.

The figure also shows that for $t_E = 0$, performance is slightly improved if t_M^{opt} is reduced. This is explained by additional gain from exploitation of level 2 variations, the stochastic nature of which demands quick responses, *i.e.* short observation intervals and short

Figure 2. Overall network loss versus observation interval for various strategies and implementation times.

execution times. We also note the significant increase in loss caused by longer execution times. This leads to the conclusion that, because of traffic dynamics, a fast algorithm may be preferred to a slower one even if the latter is known to produce slightly better network designs.

4.2. Network Updating

Upon receiving the completed traffic estimates, the NMC computes a VN design, in our case by means of the algorithm in section 3.2, after which it must decide whether to implement the result or not. Basing this decision on a comparison of costs, we need to evaluate the expected costs for implementing the new design and for keeping the old one. We also need to determine the interval t_U at which the cycle of measurements and possible updating should be repeated.

4.2.1. Evaluation

Updating a network means, on one hand, that we can expect higher revenues as resources are reallocated according to current demands. On the other hand, however, updating means costs, for example related to connecting, disconnecting and redirecting VPs and VCs. Clearly, updating should be made only when increasing revenues will exceed the costs. As costs are paid when an update is made but increasing revenues follow later, a decision on whether to implement a design or not must be based on predictions.

Consider a network with channel allocations $m(o, d)$, which are based on traffic estimates $A(o, d)$, and suppose that a new design $m^*(o, d)$, which is based on the most recent traffic estimates $A^*(o, d)$, is considered for implementation. Denoting the costs for implementing the new design by C and the expected increase in revenues by G , our decision

criterion becomes: "Implement the design if $G > C$ ".

The costs C of implementing the design, consists of a fixed cost C_I for the implementation itself, and a variable cost depending on the state of the network. The latter is composed from 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 exactly by performing an imaginary update. The total cost can thus be expressed as

$$C = C_I + \sum_{\text{Packed calls}} C_P + \sum_{\text{Rerouted calls}} C_R + \sum_{\text{Cleared calls}} C_C \quad (15)$$

The gain G from implementing the design is the additional traffic carried in the future if the design is implemented, but that otherwise would have been rejected. We assume that such a gain can only be achieved if the new design is based changes that refer to changing arrival rates, which we expect to be durable, but not if the difference $\Delta A(o, d)$ between $A(o, d)$ and $A^*(o, d)$ merely reflect random fluctuations. The expected gain is therefore

$$G = \sum_{o,d} P(o, d) A^*(o, d) \left(E_m(o, d)(A^*(o, d)) - E_{m^*(o,d)}(A^*(o, d)) \right) \tau \quad (16)$$

where $P(o, d)$ is the probability that $\Delta A(o, d)$ is a significant change on level one, and not a result of stochastic variations on level two, and τ is the time during which present conditions are expected to last. It is observed that negative terms will appear in (16) when a reduced number of circuits is proposed in the interest of a total, global gain.

To compute $P(o, d)$, we assume that traffic estimates are normally distributed, with the mean and variance given by equations (5) and (6) respectively,

$$E\{A(o, d)\} = A(o, d) \quad (17)$$

$$V\{A(o, d)\} = \frac{A(o, d)}{t_M} \quad (18)$$

It follows that, if $A^*(o, d)$ is an independent sample of the same traffic as $A(o, d)$, both have the same mean and variance, hence

$$E\{\Delta A(o, d)\} = E\{A(o, d) - A^*(o, d)\} = E\{A(o, d)\} - E\{A^*(o, d)\} = 0 \quad (19)$$

$$V\{\Delta A(o, d)\} = V\{A(o, d) - A^*(o, d)\} = V\{A(o, d)\} + V\{A^*(o, d)\} = 2 \frac{A(o, d)}{t_M} \quad (20)$$

and we may define a normalised difference $Z(o, d)$

$$Z(o, d) = \frac{\Delta A(o, d) - E\{\Delta A(o, d)\}}{\sqrt{V\{\Delta A(o, d)\}}} = \frac{\Delta A(o, d)}{\sqrt{2 \frac{A(o, d)}{t_M}}} \in N(0, 1) \quad (21)$$

which allows us to approximate $P(o, d)$ as

$$P(o, d) \approx 2\Phi(|Z(o, d)|) - 1 \quad (22)$$

Figure 3. Overall network cost (left) and loss (right) versus assumed duration of an update for various strategies and implementation times.

where Φ denotes the distribution function for the standard normal distribution, and we have used a two-sided test.

A pessimistic estimation of the life time τ , suggests that designs will change at every opportunity, *i.e.* $\tau = t_U$, while a more optimistic one assumes that designs will remain intact throughout a traffic period, *i.e.* $\tau = T$. To avoid excessive updating and ensure the profitability of a redesign, we have taken a conservative approach and set $\tau = t_U$.

The left diagram in figure 3 shows network cost versus τ/t_U as observed in our simulator, with different values of t_E , and upper and lower bounds as in section 4.1.2. The results confirm our conservative estimate of the life time of a network update, as we observe that network costs appear to be the smallest for $\tau/t_U = 1$ –2, and successively increasing, the longer a redesign is assumed to last.

The diagram to the right shows in the same way the longer a design is assumed to last, the lower the loss. This is a consequence of the fact that the longer the assumed duration τ , the more updates will be considered profitable and implemented, the more trimmed will the network be and, hence, the less calls will be lost. However, the diagram to the left says that reducing loss to the extent below what is achieved for $\tau = t_U$ is not profitable.

4.2.2. Interval

Selecting the proper interval t_U between two attempts to update a network, is again a question minimising costs: The more immediate the response to changing demands, the more resources must be spent on management, but the less traffic will be lost. We define the optimum updating interval as the one for which the expected, total costs takes its

minimum

$$t_U^{\text{opt}} = t_U : \min_{t_U} \sum_{k=1}^K C_k(t_U) \quad (23)$$

where $C_k(t_U)$, the costs paid during period k with an updating interval of t_U , is a sum of management costs and costs for lost traffic.

As for management, results on traffic measurements are transmitted to the NMC which computes a new design and analyses it. This process is repeated at intervals of t_U at a cost of C_T each time.

Proceeding to losses, costs are associated to delaying a network update. For each traffic period k , the first traffic estimate $A^*(o, d)$ that contains information about the new traffic will, on the average, be completed after $t_U/2$ time units, after which computation, evaluation and possible implementation will take another t_E time units. We call this time the initial updating delay. If a redesign is not made at this point, an additional updating delay, a multiple of updating intervals t_U , will be added until a proper redesign is implemented. We denote the average number of such attempts by $H = H_k(t_U)$. To simplify the analysis, we assume ideal multiplexing such that changing traffics from A_{k-1} to A_k will result in zero loss on VPs for which $A_k \leq A_{k-1}$, while VPs for which $A_k > A_{k-1}$ will loose all additional traffic.

Summing up, we can expect the following costs over a traffic period k of length T

$$\begin{aligned} C_k(t_U) = & \frac{T}{t_U} C_T + \left(\frac{t_U}{2} + t_E \right) \sum_{o,d} \max(A_k(o, d) - A_{k-1}(o, d), 0) + \\ & + H_k(t_U) t_U \sum_{o,d} \max(A_k(o, d) - A_{k-1}(o, d), 0) \end{aligned} \quad (24)$$

where the first term refer to processing, the second one to traffic lost during the initial delay, and the third one to traffic lost during the additional delay.

To find $H_k(t_U)$, consider the probability $h_k(t_U)$ that the first traffic estimate containing information about a new traffic k will lead to a network redesign. According to section 4.2.1, a redesign will take place only if the cost C is less than the expected gain G , and we may write

$$h_k(t_U) = \text{Prob}(c_k < g_k) \quad (25)$$

where c_k and g_k is the cost and gain respectively, as predicted by the first sample containing k -traffic after a change.

Starting with the cost c_k , we simplify (15) by not going into details with the processing of individual calls, which would require approximations of computed designs and network states at updating instants, and only consider the cost for implementing the design, hence c_k is fixed and we get

$$c_k \approx C_I \quad (26)$$

Proceeding to the gain g_k , let $A(o, d)$ denote the estimates of $A_{k-1}(o, d)$ according to which the network is dimensioned prior to traffic period k , let $A^*(o, d)$ be the estimates

which contain the first information on $A_k(o, d)$, and define $\Delta A^+(o, d) = \max(A^*(o, d) - A(o, d), 0)$. Using ideal multiplexing as before we get

$$g_k \approx \sum_{o,d} p_k(o, d) \Delta A^+(o, d) \tau \quad (27)$$

where $p_k(o, d)$ is the significance of a difference $\Delta A^+(o, d)$

$$p_k(o, d) \approx 2\Phi(z(o, d)) - 1 \quad (28)$$

with

$$z(o, d) = \frac{\Delta A^+(o, d)}{\sqrt{2 \frac{A(o, d)}{t_M}}} \quad (29)$$

Equation (27) is a sum of functions of independent stochastic variables, hence g_k is approximately normally distributed. Using standard formulae for moments of functions of stochastic variables, see *e.g.* [36], the mean $E\{g_k\}$ and variance $V\{g_k\}$ may be approximated as

$$E\{g_k\} \approx \sum_{o,d} 2\Phi\left(\frac{E\{X\}}{E\{Y\}}\right) \frac{E\{X\}}{E\{Y\}} \tau \quad (30)$$

and

$$\begin{aligned} V\{g_k\} \approx & \sum_{o,d} V\{X\} \tau^2 \left(2\varphi\left(\frac{E\{X\}}{E\{Y\}}\right) \frac{E\{X\}}{E\{Y\}} + 2\Phi\left(\frac{E\{X\}}{E\{Y\}}\right) + 1 \right)^2 + \\ & + V\{Y\} \tau^2 \left(2\varphi\left(\frac{E\{X\}}{E\{Y\}}\right) \frac{E\{X\}^2}{E\{Y\}^2} \right)^2 \end{aligned} \quad (31)$$

where φ is the probability density function of the standard normal distribution,

$$X = \Delta A^+(o, d) \Rightarrow \quad (32)$$

$$\begin{aligned} E\{X\} &= \dots = \frac{\max(A_k(o, d) - A_{k-1}(o, d), 0)}{2} \frac{t_U}{t_M} \\ V\{X\} &= \dots = \frac{2A_{k-1}(o, d)}{t_M} + \frac{E\{X\}}{t_M} + \frac{E\{X\}^2}{3t_M} \end{aligned}$$

and

$$Y = \sqrt{\frac{2}{t_M}} \sqrt{A(o, d)} \Rightarrow \quad (33)$$

$$\begin{aligned} E\{Y\} &= \dots \approx \sqrt{\frac{2}{t_M}} \sqrt{A_{k-1}(o, d)} \\ V\{Y\} &= \dots \approx \frac{1}{2t_M^2} \end{aligned}$$

Figure 4. Overall network cost (left) and loss (right) versus updating interval for various strategies and implementation times.

We may now obtain $h_k(t_U)$ by subtracting the cost (26) from the gain (27) and convert the result into a standard normal distribution:

$$h_k(t_U) \approx \Phi\left(\frac{E\{g_k\} - c_k}{\sqrt{V\{g_k\}}}\right) \quad (34)$$

As for samples following the first one, these will contain successively more and more information about the new k -traffic, hence if an update is rejected the first time, it will have a greater chance of being implemented the second time and so on. However, we assume that this case is so infrequent that the probability of implementing a new design always is $h_k(t_U)$, *i.e.* independent of the number of previously failed attempts, and finally obtain

$$H(t_U) = \sum_{k=0}^{\infty} k (1 - h(t_U))^k h(t_U) = \frac{1 - h(t_U)}{h(t_U)} \quad (35)$$

Inserting (24)–(35) in (23), we obtain an expression which is preferably solved by interval minimisation, suitable limits are *e.g.* 0 and t_M .

For our series of networks, we typically obtain $t_U^{\text{opt}} \approx 1.5, 1.3$, and 1.1 for $t_E = 0, 3$, and 6 respectively. Of the proposed updates, typically about 25% are found profitable, a number which corresponds to the number of updating instants it takes until a change in offered traffics has fully penetrated the traffic estimates.

The diagram to the left in figure 4 shows network cost versus t_U/t_U^{opt} as observed in our simulator, with different values of t_E , and upper and lower bounds as before. Similar to

what was observed for measurements, the figure confirms the optimum and shows that it is flat: Setting t_U to $t_U^{\text{opt}}/2$ or $2t_U^{\text{opt}}$ only marginally increases the cost.

A similar looking diagram to the right in figure 4 shows network loss versus the relative updating interval. We observe, as expected, that neither increasing nor decreasing the updating interval from the optimal value will reduce loss, as both means postponing network updating longer than necessary: Too long intervals will increase the initial delay while too short intervals will increase the additional delay.

4.3. Performance

In deriving our strategy we have assumed that, for each period k of length T , offered traffics $A_k(o, d)$ remain constant, are known in advance, and that changes take place instantaneously. Clearly, if this was true, there would be no better strategy than deterministic rearrangements, our lower bound in the figures above. However, for any realistic network, traffics change gradually and in a somewhat irregular way, and there will always be a difference between forecast and actual traffics. It is these differences that motivate our strategy of dynamic rearrangements.

To quantify the differences between predicted and actual values of T and $A_k(o, d)$, we assume that the latter, denoted by $\tilde{T}$ and $\tilde{A}_k(o, d)$ respectively, deviate randomly from the predicted values according to independent, normal distributions and vary the expectation of this deviation:

$$\begin{aligned}\tilde{T} &\in N(T, C_V T) \\ \tilde{A}_k(o, d) &\in N(A_k(o, d), C_V A_k(o, d))\end{aligned}$$

where C_V , the coefficient of variation, specifies the degree of uncertainty. Moreover, to further resemble real networks, we let actual traffics change gradually during 10 time units of each interval subject to the same random variations. Figure 5 gives an idea of resulting differences over one cycle $k = 1, \dots, K$ between predicted traffic profiles (dotted lines) and some actual realisations (solid lines) for $C_V = 0.05$ (left) and $C_V = 0.20$ (right).

The same networks and traffic profiles as above were simulated for different values of C_V , while VP management parameters t_M and t_U consistently were set according to sections 4.1.2 and 4.2.2, using the predicted traffic profiles given by the expected values of T and $A_k(o, d)$. The results are shown in figure 6 in terms of network cost (left) and lost calls (right) respectively. Again, solid lines refer to different values of t_E , dotted lines indicate no updating, and dashed lines indicate deterministic updating.

It is noted that very small deviations from forecasts have none or only a small impact on performance in terms of costs and loss. For larger variations, however, performance slowly begins to deteriorate, and we note that our strategy does better in counteracting this than deterministic updating based on averages. The degree of uncertainty at which this happens depends basically on t_E and on our particular choice of costs: Lesser deviations would be required if management costs were lower would and *vice versa*.

In general, we also note a considerable adaptivity of our strategy: In fact, for reasonably short execution times, our strategy appears to fully compensate for deviations from forecasts of up to 10%. We also observe that for variations of about 20% or more, deterministic updating performs worse than no updating at all.

Another advantage of our strategy as compared to deterministic updating refers to

Figure 5. Examples of theoretical traffic profile and some actual realisations for various degrees of uncertainty.

handling focused overloads, link breakdowns and other unsystematic traffics deviations. For this type of events, our strategy provides a considerable degree of flexibility in that traffics are actually measured and paths are chosen freely.

4.4. Alternatives

We will now compare our VPM strategy to two alternative ways of solving the problem of long term traffic variations of which only limited information is available. The first of these aims at increasing transmission capacities (the idea being that the need for level one management can be eliminated by supplying more capacity), and the second one aims at solving the problems on level two (the idea being that slow traffic variations can be handled at level two by deploying advanced overflow routing schemes for individual calls).

4.4.1. Increasing Network Capacity

The first alternative is illustrated in the diagram to the left in figure 7. The curve shows the average number of OD-pairs serviced per unit of capacity for various values of C_V . This number gives an idea of the additional capacity required to maintain service under traffic variations without VPM: about 50% for $C_V = 0.01$ and up to 150% for $C_V = 0.20$. It is concluded that considerable amounts of transmission capacity can be saved by VPM, and, moreover, even with overprovisioning, VPM will still be required for situations like faulty links or nodes. The figure is obtained analytically for the current series of networks by averaging designs for 50 realisations of each traffic pattern and each network.

Figure 6. Overall network cost (left) and loss (right) versus traffic profile coefficient of variation for various strategies and implementation times.

4.4.2. Advanced Call Routing

The diagram to the right illustrates the performance of the second alternative. The costs displayed refer to networks in which overflow routing, more specifically dynamic alternative routing (DAR) with optimal trunk reservation [24], is employed. The curves are essentially the same as in figure 6, the only difference being that costs are generally reduced by the overflow facilities. The figure leads to two major conclusions: Firstly, path rearrangements adds efficiency also to a network in which alternative routing is used, as was also shown in [27], and, secondly, the dynamics of our real time strategy adds efficiency also on a network with path rearrangements and alternative routing.

(In DAR, calls are primarily routed on the direct path and secondarily over a tandem node n . Should the secondary choice fail is the call lost and a new n for future overflows is selected. In our implementation, n is selected at random with equal probabilities for all nodes.

Trunk reservation levels are selected for each path of a particular design by optimising the profit earned per path, 1 for all direct calls carried and 1/2 for two-hop calls. In our implementation, direct traffics are assumed equal to the most recent estimates. Overflow traffics are approximated by computing lost traffics per OD-pair under the assumption of no other existing alternatives but the direct one, summing up the overflows at both ends of the path, and assuming the character of the merged stream to be Poisson. Taking into account that we have ignored both the peaked character of the overflow traffic (which would increase loss) and the fact that overflows are spread out over several links (which would decrease loss), we believe that the final result is sufficiently accurate for our purpose.)

Figure 7. Results for replacing rearrangements by increasing capacity (left) and by advanced call routing (right).

4.5. Applicability

We have studied traffic variations which on the two upper levels of table 1 can be modelled as a Poisson process (session level) of variable rate (link level). Naturally, a clear distinction between levels cannot be made in a real network, where traffics appear to vary constantly. Applying our control strategy, one is therefore faced with the problem of identifying a suitable time scale T of rate variations on the link level, the length of a “stable” period.

We will consider two aspects of this with reference to our strategy, *viz.* the extent to which rate variations are noticeable in practice, and hence worth attempting to control, and the extent to which such control is feasible with respect to the time required to produce traffic estimates t_M and the design turn-around time t_E .

Figure 8 attempts to illustrate the degree to which rate variations can be detected in network performance. The figure refers to networks with fixed VP-assignments based on average traffics, the diagrams to the left and right refer to networks with and without overflow facilities respectively. Solid lines show network loss versus the time scale of the variations, dotted lines indicate lower and upper bounds: $T = 0$ (average applies at any time) and $T = \infty$ (average applies only over an infinitely long period). It is seen that the longer the time scale over which the average applies, the worse does a network permanently set to average traffics perform. In particular, we observe that not even rate variations which occur on a time scale faster than a call holding time are fast enough to be perceived as average by the network, a fact which is not affected by employing overflow facilities. It is concluded that tuning networks on a link level in accordance with rate variations is potentially profitable even for very fast variations.

Figure 8. Overall network loss versus rate variations for direct routing only (left) and combined with alternative routing (right).

As for restrictions imposed by the traffic observation time t_M , it follows from equation (13) that the smaller T , the shorter the observation interval t_M and, according to (6), the less accurate will the estimates be. The solid line in figure 9 shows the average relative error of the traffic estimates used in our strategy versus the stable period T , and for comparison we give the same quantity with the estimate replaced by a precomputed average (dotted line). The figure refers to $t_E = 0$, similar curves are obtained for other values of t_E . It is seen that for small T , we can expect precomputed averages over all traffic periods $k = 1, \dots, K$ to be more accurate than on-line estimates. For our case, the point at which estimating traffic becomes meaningful occurs at $T \lesssim 2$. In general, it is concluded that our strategy will perform better for longer T , and that T must be on the order of at least a couple of call holding times before our strategy can be applied at all.

Proceeding to the design turn-around time t_E , it is clear that updating is meaningful only for variations which are slower than the time required to detect them and make the appropriate adjustments. Figure 10 shows the impact of the relative network updating time t_E/T (solid lines) to network cost (left) and call loss (right) with bounds (dotted and dashed lines) referring to no updating and ideal updating respectively. The curves are obtained for $T = 30$, similar curves are obtained for other values of T . We note that both cost and loss increase linearly with t_E , and that fixed, average dimensioning performs better than dynamic reallocation when $t_E/T \lesssim 0.35$. It is concluded that our strategy performs better, the faster new designs can be implemented, and that it is restricted to variations which are on the order of four times slower than the turn-around time.

Summing up, tuning networks on a link level in accordance with rate variations, as is the aim of our strategy, is potentially profitable even for very fast variations. However,

Figure 9. Expected, relative error of traffic estimates versus traffic observation time.

restrictions are imposed by the on-line estimates used in our strategy, which require that T must be the order of a couple of call holding times, and the delay in implementing designs, which means that T must be a couple of turn-around times. The exact limits of our strategy, *i.e.* the boundary between the link level and the session level, depends on the specific networks, traffics, relationship between costs for lost calls and management costs *etc.*

5. CONCLUSIONS AND FURTHER WORK

We have discussed and classified traffic variations in B-ISDNs in terms of their causes and management means. Proposing a hierarchical model where each layer defines a certain time frame, we have concentrated on the two uppermost layers: The link layer (with a time frame of several call holding times) and the session layer (with a time frame of about a call holding time).

Assuming that traffic variations are partially predictable on a link level (*e.g.* from one hour to the next) but random on a session level (*e.g.* from one minute to the next), we have devised a traffic management strategy based on a repeated cycle of traffic measurements and network design updates. Methods and parameters for on-line traffic measurements were discussed and optimised, algorithms for network design were discussed and a new fast and simple one was put forward, a network update profitability criterion was formulated and the optimal updating frequency was derived.

Applying our strategy to a series of test networks in a simulator, we have shown that our strategy successfully manages network resources even when predicted and actual traffics differ considerably. In particular, we noted that fixed management based solely on

Figure 10. Overall network cost (left) and loss (right) versus design turn-around time.

predictions is outperformed even for rather small deviations from expected averages, typically $\pm 5\%$ is enough. Moreover, it was seen that our strategy saves considerable amounts of transmission capacity and that it does not replace nor can it be replaced by session level management such as overflow rerouting.

Finally, investigating the range of variations to which our strategy can be applied, we saw that even very fast variations are potentially profitable. However, the times required for traffic estimation and design implementation define a lower bound below which session management is superior: Typically, traffic estimation puts a lower limit of stable periods amounting to a few call holding times while the limit imposed by the implementation time amounts to a couple of design turn-around times.

Further work includes, but is not limited to, local strategies for VPM where no central NMC is involved, refining traffic measurement methods and network design algorithms, and preparing up to date, detailed examples of multi service networks, for example one service class per AAL type, preferably down to the burst level.

A. NETWORKS AND SIMULATIONS

To study the efficiency of our algorithm and our strategy, a series of sample networks and traffic patterns were needed. We employed a computer programme to generate 8 distinct networks of 20 nodes, about 50 links, and a total transmission capacity of around 7,500 units of capacity (75,000 channels). For each network, 8 different busy hour patterns were generated, each amounting to approximately 30,000 Erlangs. The programme works as follows:

1. Initialise random number generation.

- (a) Read a number used to preset the pseudo random number used when creating network topology, routing pattern, transmission capacities and basic traffic pattern.
- (b) Read a number used to preset the pseudo random number used when creating the specific traffic pattern.

2. Network topology:

- (a) Nodes: Place $n = 1, \dots, N$ nodes over a surface of size $X \times Y$ at randomly chosen coordinates $x_n \in \text{Rect}(0, X)$ and $y_n \in \text{Rect}(0, Y)$.
- (b) Minimal tree of links: Compute the euclidean distances between all nodes. Add one link at a time by joining the two not yet interconnected nodes $o, d = 1, \dots, N$ that are the closest to each other, until all nodes are interconnected (possibly through one or more other nodes).
- (c) Additional links: Add one link at a time by joining the two not yet directly interconnected nodes $o, d = 1, \dots, N$ that are the closest to each other, until $\mu \frac{N(N-1)}{2}$ link have been added, where μ is a “mesh factor”, we used $\mu = 0.5$.
- (d) Protection links: Add links to provide protection against one node and its associated links being down by removing one node $n = 1, \dots, N$ at a time and checking for full connectivity among the remaining nodes. If “islands” occur, identify the members of these and add one link between the two nodes that are the closest to each other yet members of different islands.

3. Routing pattern:

- (a) Compute the shortest paths between all OD-pairs o, d in terms of number of links traversed. For two paths of equal length, let shortest one be the one having the smallest euclidean length.

4. Transmission capacities:

- (a) Initially define the capacities of all links to be zero.
- (b) For each OD-pair o, d , define a capacity of $1, \dots, 36$ units at random with equal probability for each value. Add this value to the capacities of all links belonging to the shortest path of the OD-pair.

5. Basic traffics:

- (a) For each OD-pair o, d , assign an offered traffic such the expected loss will be exactly 1% given the transmission capacity associated to the OD-pair.

6. Specific traffics:

- (a) Nodes: Place a busy hour centre over the defined area at randomly chosen coordinates $x_h \in \text{Rect}(0, X)$ and $y_h \in \text{Rect}(0, Y)$.

Figure 11. Network design algorithm output: Example of a topology of a network with 20 nodes.

- (b) Draw a circle around x_h, y_h with a radius large enough to enclose at least $N/4$ nodes. These nodes are referred to as busy nodes.
- (c) For each OD-pair o, d , if
 - Both nodes are busy nodes: Scale the assigned traffic randomly by multiplying by a factor $\in \text{Rect}(1.1, 1.3)$.
 - One node is a busy node and one is not: Scale the assigned traffic randomly by multiplying by a factor $\in \text{Rect}(0.9, 1.1)$.
 - None of the nodes are busy nodes: Scale the assigned traffic randomly by multiplying by a factor $\in \text{Rect}(0.7, 0.9)$.

An example of a network and a traffic pattern is given below: Figure 11 shows the topology, table 3 gives transmission capacities and table 4 a traffic pattern. Transmission capacities are given in units of 10 channels and offered traffics in Erlangs. The busy nodes of traffic pattern are indicated by darker circles in figure 11.

In our simulations, we simulate all 8 networks, each subject to its own series of traffic patterns, and average over the networks. Per network, each point in figures 2 and 4 represents approximately 15 million connections and about 30 million connections in figure 6. To speed up simulations we assumed negative exponential call holding times.

Table 3

Network design algorithm output: Example of transmission capacities for the network of the example.

OD	C	OD	C	OD	C	OD	C	OD	C	OD	C
1-9	150	1-11	30	1-14	210	1-16	65	1-19	100	2-3	309
2-4	322	2-7	137	2-8	328	2-12	181	2-16	303	2-18	54
3-8	168	3-9	112	3-11	246	3-16	109	4-5	194	4-7	148
4-12	154	4-14	159	4-15	96	5-7	72	5-15	46	6-8	147
6-9	138	6-13	137	6-20	64	7-14	91	7-18	136	7-19	179
8-13	72	8-17	104	8-18	322	8-20	174	9-17	101	9-20	77
10-12	179	10-13	153	10-18	203	11-17	101	12-15	329	12-18	184
13-18	108	14-16	172	14-18	145	14-19	63	16-17	164	17-20	43

ACKNOWLEDGEMENT

The author would like to express his appreciation of the inspiring discussions with Dr. Meir Herzberg at Telecom Australia Research Laboratories, and with Dr. Gary Anido and Mr. David Atkinson at the University of Wollongong, N.S.W., Australia.

The generous support of the Department of Communications Systems, Lund Institute of Technology, Sweden, and of the Teletraffic Research Centre, The University of Adelaide, S.A., Australia, which has made this work possible is also gratefully acknowledged.

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 Broad-band ISDN*, paper no. 22.6 in Proc. IEEE Globecom '88, Hollywood, Florida, U.S.A. (1988).
3. 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).
4. Arvidsson, Å.: *On Dynamic Capacity Management in Voice/Frame Relaying Networks*, paper no. 2.3 in Proc. 10th Nordic Teletraffic Sem., Århus, Denmark (1992).
5. Arvidsson, Å.: *Strategies for Dynamic Network Management*, paper no. 7.2 in Proc. 11th Nordic Teletraffic Sem., Stockholm, Sweden (1993).
6. Arvidsson, Å.: *Strategies for Dynamic Capacity Management*, pp. 291-300 in Proc. 8th Australian Teletraffic Research Sem., Melbourne, Victoria, Australia (1993).
7. Arvidsson, Å.: *Management of Reconfigurable Virtual Path Networks*, paper no. D31.4 in Proc. 14th Int. Teletraffic Cong., Antibes, France (1994).
8. Arvidsson, Å.: *High Level B-ISDN/ATM Traffic Management in Real Time*, paper no. 2.7 in Proc. 2nd IFIP TC6 Workshop on Perf. Modelling and Eval. of ATM Netw., Bradford, U.K. (1994).
9. Arvidsson, Å.: *Real Time Management of Virtual Paths*, pp. 1399-1403 in Proc. IEEE Globecom '94, San Francisco, California, U.S.A. (1994).

10. Ash, G. & Chang, F.: *Transport Network Design of Integrated Networks with Real-Time Dynamic Routing*, J. of Netw. and Sys. Mngmnt, vol. 1, no. 4, pp. 319–336 (1993).
11. 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).
12. 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).
13. Burgin, J.: *Management of Capacity and Control in Broadband ISDN*, Int. J. of Dig. and Ana. Cabled Sys., vol. 2, pp. 155–165 (1989).
14. Burgin, J.: *Broadband ISDN Resource Management*, Computer Netw. and ISDN Syst., vol. 20, no. 1–5, pp. 323–331 (1990).
15. Burgin, J.: *Traffic Control and Resource Management for the Broadband ISDN*, pp. 93–100 in Proc. Australian Broadband Switching and Services Symposium '92, Melbourne, Victoria, Australia (1992).
16. Dutkiewicz, 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).
17. 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).
18. 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).
19. Filipiak, J.: *M-Architecture: A Structural Model of Traffic Management and Control in Broadband ISDNs*, IEEE Comm. Mag., vol. 25, no. 5, pp. 25–31 (1989).
20. Filipiak, J.: *Automatic Network Management Controls*, IEEE Trans. on Commun., vol. 39, no. 12, pp. 1776–1786 (1991).
21. 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).
22. Gersht, A. & Shulman, A.: *Optimal Routing in Circuit Switched Networks*, IEEE Trans. on Commun., vol. 37, no. 11, pp. 1203–1211 (1989).
23. 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).
24. Gibbens, R., Kelly, F. & Key, P.: *Dynamic Alternative Routing — Modelling and Behaviour*, paper no. 3.4A.3 in Proc. 12th Int. Teletraffic Cong., Torino, Italy (1988).
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. Kendall, M. & Stuart, A.: *The Advanced Theory of Statistics*, Charles Griffin & Co. Ltd., London (1962).
 37. 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).
 38. Kleinrock, L.: *Queueing Systems, Volume II: Computer Applications*, John Wiley & Sons, New York (1976).
 39. Mase, K. & Imase, M.: *An Adaptive Capacity Allocation Scheme in Telephone Networks*, IEEE Trans. on Commun., vol. 30, no. 2, pp. 354–359 (1982).
 40. Mase, K. & Shioda, S.: *Real-Time Network Management for ATM Networks*, pp. 133–140 in Proc. 13th Int. Teletraffic Cong., Copenhagen, Denmark (1991).
 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. Pióro, M. & Gajowniczek, P.: *Stochastic Allocation of Virtual Paths to ATM Links*, paper no. 2.1 in Proc. 2nd IFIP TC6 Workshop on Perf. Modelling and Eval. of ATM Netw., Bradford, U.K. (1994).

Table 4

Network design algorithm output: Example of offered traffics for the network of the example.

OD	A	OD	A	OD	A	OD	A	OD	A
1- 2	59.0	1- 3	182.3	1- 4	137.5	1- 5	156.6	1- 6	82.3
1- 7	94.2	1- 8	248.3	1- 9	223.0	1-10	181.9	1-11	82.0
1-12	104.8	1-13	40.7	1-14	261.2	1-15	107.9	1-16	68.4
1-17	31.8	1-18	319.7	1-19	111.9	1-20	53.7	2- 3	191.0
2- 4	204.7	2- 5	10.8	2- 6	86.8	2- 7	212.2	2- 8	231.7
2- 9	34.8	2-10	130.6	2-11	236.4	2-12	106.7	2-13	94.3
2-14	179.7	2-15	172.0	2-16	227.5	2-17	26.3	2-18	96.5
2-19	9.7	2-20	120.0	3- 4	14.1	3- 5	317.6	3- 6	426.7
3- 7	100.0	3- 8	136.8	3- 9	33.3	3-10	105.3	3-11	269.3
3-12	97.0	3-13	177.6	3-14	261.9	3-15	268.2	3-16	212.1
3-17	95.7	3-18	183.7	3-19	221.3	3-20	163.7	4- 5	24.4
4- 6	88.9	4- 7	164.2	4- 8	192.9	4- 9	9.7	4-10	9.7
4-11	229.8	4-12	185.0	4-13	243.0	4-14	156.8	4-15	183.2
4-16	261.5	4-17	194.2	4-18	28.8	4-19	230.0	4-20	160.6
5- 6	28.5	5- 7	57.5	5- 8	347.6	5- 9	48.2	5-10	155.0
5-11	165.1	5-12	147.6	5-13	343.0	5-14	113.3	5-15	65.0
5-16	161.6	5-17	179.8	5-18	236.3	5-19	49.7	5-20	35.7
6- 7	265.6	6- 8	162.7	6- 9	270.9	6-10	96.8	6-11	168.7
6-12	309.5	6-13	62.3	6-14	20.6	6-15	161.8	6-16	279.8
6-17	255.4	6-18	146.5	6-19	142.9	6-20	323.1	7- 8	93.9
7- 9	19.1	7-10	236.7	7-11	129.5	7-12	297.1	7-13	36.4
7-14	255.8	7-15	191.2	7-16	176.5	7-17	263.4	7-18	12.0
7-19	244.2	7-20	110.9	8- 9	235.7	8-10	148.6	8-11	232.0
8-12	299.2	8-13	84.0	8-14	13.6	8-15	219.7	8-16	178.1
8-17	154.2	8-18	171.8	8-19	195.2	8-20	196.8	9-10	208.1
9-11	99.3	9-12	76.5	9-13	75.5	9-14	223.5	9-15	42.9
9-16	138.0	9-17	268.3	9-18	22.1	9-19	252.8	9-20	274.2
10-11	44.0	10-12	246.3	10-13	204.2	10-14	215.9	10-15	267.9
10-16	161.8	10-17	159.6	10-18	109.8	10-19	57.6	10-20	16.5
11-12	268.1	11-13	22.1	11-14	64.5	11-15	244.9	11-16	170.7
11-17	207.1	11-18	145.4	11-19	18.3	11-20	111.9	12-13	5.5
12-14	272.2	12-15	228.5	12-16	87.4	12-17	43.5	12-18	203.9
12-19	132.9	12-20	145.9	13-14	178.6	13-15	171.9	13-16	201.6
13-17	130.4	13-18	295.0	13-19	201.3	13-20	341.5	14-15	129.7
14-16	112.2	14-17	127.5	14-18	30.3	14-19	198.8	14-20	204.1
15-16	38.8	15-17	207.5	15-18	275.7	15-19	113.7	15-20	168.7
16-17	120.3	16-18	49.5	16-19	39.6	16-20	119.2	17-18	165.9
17-19	43.9	17-20	86.0	18-19	92.0	18-20	164.5	19-20	144.3