

## Call admission and routing in ATM networks based on virtual path separation

S.A. Berezner

Department of Statistics, University of Natal, 4001 Durban, South Africa.  
fax: +27 31 260 1009, e-mail: berezner@ph.und.ac.za

A.E. Krzesinski

Department of Computer Science, University of Stellenbosch, 7600 Stellenbosch, South Africa. fax: +27 21 808 4416, e-mail: aek1@cs.sun.ac.za

### Abstract

Dynamic reconfiguration is a network management control which reserves transmission capacity on the communication links in order to form dedicated logical paths for each origin-destination flow. We present an efficient deterministic algorithm (XFG) to rapidly compute an optimal network configuration. We apply reconfiguration to a model of an ATM network. The XFG calculation of an optimal configuration requires a few seconds of CPU time (Pentium 100MHz) whereas a standard NLP package requires several hours to compute an optimal network configuration.

### Keywords

ATM networks, call admission control, network resource management, non-linear optimization, reconfiguration, virtual paths.

## 1 INTRODUCTION

Resource management in ATM networks is implemented by a hierarchy of controls at the network, call and cell levels which interact to form a layered

---

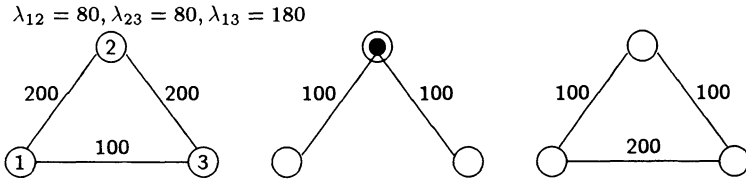
\*This work is supported by grants from the Australian Research Council and the South African Foundation for Research Development.

traffic control. The Network Resource Manager (NRM) collects O-D traffic intensities and call blocking probabilities for the different traffic classes from the Call Admission Control (CAC). If these data exceed threshold values, the NRM may adjust the buffer sizes at the switches and compute new values for the bandwidths of the Virtual Paths (VPs) and update the VP network configuration. This is done by cross connecting groups of circuits in intermediate nodes to form (logical) direct connections between all O-D pairs to create a fully meshed network. Depending upon the traffic/network equipment variations, the update interval may vary from minutes to hours.

The CAC computes the effective bandwidth of an offered call and determines how best to route the call through the network. If a route is available, the call is connected, else the CAC may re-negotiate the call's Quality of Service (QOS) parameters or reject the call. The CAC collects cell loss and cell delay data and if these exceed threshold values, the CAC may adjust the effective bandwidth of the call or other actions may be taken. The cell level controls are responsible for detecting cell congestion and responding by traffic shaping, source rate adaptation, rate- or credit-based flow control, cell priorities and discards, and user/network parameter controls.

This paper presents a model of hierarchical resource management in ATM networks. The NRM component of the model adjusts the logical link capacities in order to optimally size the virtual path (VP) network whenever changed conditions demand such a reconfiguration. The reconfiguration maintains sufficient capacity to carry the established calls in progress. The CAC connects calls on the (logical) direct links. A call is rejected if insufficient capacity is available on the logical link. The model assumes that the effective bandwidths of the various traffic classes are known and are fixed. Cell level behaviour is assumed to be encapsulated into the effective bandwidth parameters. The model therefore contains no explicit description of cell level behaviour or cell-based resource management.

The model shows that the simplicity of reconfiguration is especially useful in dealing with the complex characteristics of multi-rate calls. We compare reconfiguration with another state independent admission/routing control (Mitra 1996) which effectively converts an ATM network into a network with fixed routing. We develop a computationally efficient algorithm for optimal network reconfiguration. We apply the algorithm to model call admission and routing in the NSF backbone network (Mitra 1996) and we demonstrate that reconfiguration achieves a comparable performance to traffic redistribution. We show that reconfiguration has both advantages and disadvantages: the advantages include conceptual simplicity, protection of the GOS for O-D pairs and much lower computational complexity. The main disadvantage of reconfiguration is that it realizes a higher overall blocking, especially for smaller networks.



**Figure 1** (a) Physical network before reconfiguration (b) cross connections (c) logical network after reconfiguration

## 2 DYNAMIC RECONFIGURATION

Consider a network which consists of  $J$  nodes and carries  $S$  services (call classes). Let  $d_s$  denote the effective bandwidth requirement of a call of class  $s$  where  $s = 1, \dots, S$ . Let  $C_{ij}$  denote the capacity of a uni-directional link  $(i, j)$  such that  $C_{ij} > 0$  denotes the existence of a physical link between nodes  $i$  and  $j$ . Let  $L$  denote the number of physical links. Let  $\lambda_{ij}^s$  denote the Poisson arrival rate of calls of class  $s$  at link  $(i, j)$  and let  $1/\mu_{ij}^s$  denote the mean holding time of class  $s$  calls on link  $(i, j)$ . The arrival intensity for calls of class  $s$  on link  $(i, j)$  is  $\rho_{ij}^s = \lambda_{ij}^s/\mu_{ij}^s$ . A call of class  $s$  connected between nodes  $i$  and  $j$  generates revenue at a rate  $\theta_{ij}^s$ . The revenue generated by a call does not depend on how the call is routed, although this assumption can be relaxed if cross connection costs or other costs are taken into account. The effective bandwidth of a class  $s$  call is also independent of how the call is routed.

Consider the following network reconfiguration scheme which copes with mismatches between offered traffic and available capacity by making logical direct connections between nodes. Each such logical link is composed from several routes, where a route  $r = (i, i_1, \dots, i_k, j)$  is a sequence of nodes. All traffic classes are multiplexed onto each link and onto each route, provided the route has sufficient capacity.

Reconfiguration is performed by cross connecting circuits using configurable switches at the transit nodes. For example, consider the 3-node network presented in Figure 1. Before reconfiguration – Figure 1(a) – the network blocking probability is 24%. Figure 1(b) illustrates the cross connection of 100 circuits at node 2. After reconfiguration – Figure 1(c) – the network blocking probability is 1%.

The number  $x_r$  of circuits that should be cross connected along route  $r$ , as well as the number  $x_{(i,j)}$  of circuits that should be used to carry direct traffic on the link  $(i, j)$  are obtained as a solution to the optimization problem

presented below. Let  $\mathcal{R}_{ij}$  denote the set of *admissible* routes between nodes  $i$  and  $j$ , including the direct route  $(i, j)$  if it exists. A route  $r$  is said to be *admissible* if circuits can be reserved on each link along this route for the sole use of calls between nodes  $i$  and  $j$ . A route is not admissible if restrictions are imposed on the route: for example there could be a limit on the number of links in the route. Routes with loops are not admissible. Let  $\mathcal{R} = \cup_{i,j} \mathcal{R}_{ij}$  denote the set of all admissible routes. Let  $\mathbf{x} = \{x_r\}_{r \in \mathcal{R}}$ ,  $x_r \in \mathbb{I}^+$  denote the route configuration. The capacity reserved for the sole use of the traffic between nodes  $i$  and  $j$ , which is the capacity of the VP connecting nodes  $i$  and  $j$ , is given by  $C_{ij}(\mathbf{x}) = \sum_{r \in \mathcal{R}_{ij}} x_r$ .

Let  $B_{ij}^s(C_{ij}(\mathbf{x}))$  denote the blocking probability (Ross 1995) experienced by class  $s$  calls on a link  $(i, j)$  of capacity  $C_{ij}(\mathbf{x})$  carrying services  $s = 1, \dots, S$  with bandwidth  $d_s$  and traffic intensity  $\rho_{ij}^s$ . The total rate of earning revenue is

$$F(\mathbf{x}) = \sum_{ij} F_{ij}(C_{ij}(\mathbf{x})) = \sum_{ij} \sum_s \theta_{ij}^s \rho_{ij}^s (1 - B_{ij}^s(C_{ij}(\mathbf{x}))). \quad (1)$$

The Dynamic Reconfiguration and Optimization (DROP) problem is defined as: Maximize  $F(\mathbf{x})$  subject to

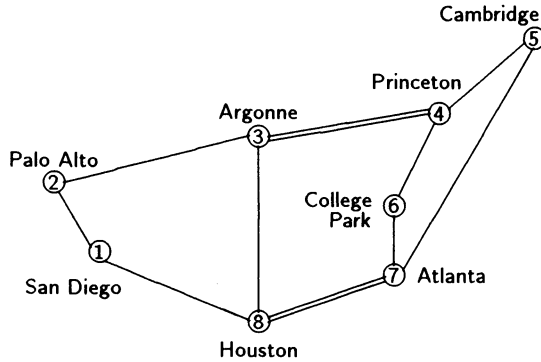
$$\sum_{r \in \mathcal{A}_{ij}} x_r \leq C_{ij}, x_r \geq 0. \quad (2)$$

where  $\mathcal{A}_{ij}$  denotes the set of admissible routes that use link  $(i, j)$  including the direct route  $(i, j)$ . A vector  $\mathbf{x}$  which satisfies constraint (2) is called a solution to the DROP problem. A solution  $\mathbf{x}$  which (locally) maximizes the revenue function (1) is called a (local) optimal solution to the DROP problem.

### 3 TRAFFIC REDISTRIBUTION

Another method (Mitra 1996) of dealing with traffic/capacity mismatch is to redistribute the offered traffic by random sampling of the arriving flows. Let  $p_{ijr}$  denote the probability that a call arriving to node  $i$  and destined for node  $j$  is offered to route  $r$ . Thus  $1 - \sum_{r \in \mathcal{R}_{ij}} p_{ijr}$  is the probability that an arriving call is not offered to the network, which is a form of call admission control. If the call is offered to the network, and if the call is blocked on route  $r$ , it is lost. This model is equivalent to a loss network with fixed flows on fixed routes and has an analytical solution (Kelly 1991).

A sequence of values is computed for the probabilities  $p_{ijr}$  which successively improve the network revenue. At each step, for the current set of



**Figure 2** Core NSF network

probabilities  $p_{ijr}$ , the link blocking probabilities are calculated using the Erlang fixed point approximation. This yields a system of  $SL$  linear equations for the implied costs. The linear equations are solved to obtain the derivatives of the revenue function. The complexity of this step is  $O(S^3L^3)$  which can be reduced to  $O(L^3)$  if a special approximation technique is used. The derivatives are used to compute the direction of steepest ascent for the revenue function in order to obtain new values for the probabilities  $p_{ijr}$  which yield an improved revenue. This optimization procedure converges to a value for the network revenue which is not necessarily a local maximum.

Note that our reconfiguration scheme differs from the logical reconfiguration method described in Faragó (1995) which assumes that the network revenue is a convex function of the link capacities. Optimal routes are computed to carry each service class between all O-D pairs and an exact lower bound to the network revenue is calculated. This global optimum provides an initial point to compute a better estimate of the optimal network revenue. Our model assumes that the link revenues are convex functions of the link capacities so that the network revenue is not a convex function. Optimal routes are computed between all O-D pairs and many service classes are multiplexed onto each route.

#### 4 RECONFIGURATION VERSUS REDISTRIBUTION

Consider the model (Mitra 1996) presented in Figure 2 which is a fictitious representation of the core NSF ATM backbone consisting of eight nodes

**Table 1** Traffic intensity matrix

<i>nodes</i>	1	2	3	4	5	6	7	8
1	–	6	7	1	9	5	2	3
2	7	–	24	3	31	15	6	9
3	8	25	–	4	37	18	7	11
4	1	3	3	–	4	7	1	1
5	11	33	39	5	–	24	9	15
6	5	14	16	2	21	–	4	6
7	2	5	6	1	8	4	–	2
8	3	8	10	1	12	6	2	–

each connected by two uni-directional links. Each link carries traffic in one direction. The double lines between nodes 3 and 4 and 7 and 8 indicate that there are two links in each direction connecting these nodes. The network carries 6 services: the bandwidth requirement of the 1st service is 1 unit (16 Kbps) and the bandwidth requirements of services 2 through 6 are 3, 4, 6, 24 and 40 units respectively. The capacity of each uni-directional link is 2812 units (45 Mbps). The traffic intensity for service 3 is given in Table 1. For all other services, the values in Table 1 are multiplied by 0.5. The revenue earned by a carried call per unit of time is assumed to be independent of how the call is routed and is equal to the bandwidth of the call.

The DROP problem can be solved by NLP methods. The NSF backbone model has 332 routes and 56 VPs. Two NLP problems were defined: the first problem restricted the route lengths to 5 links (260 routes) so that the NLP has 260 variables and 56 linear constraints. The second problem does not restrict the route lengths which can be up to 7 links long. In this case the network has 332 routes and the corresponding NLP has 332 variables and 56 linear constraints. Table 2 presents the computational resources required to calculate an optimal VP configuration using the NLP package CFSQP (Lawrence 1996). The NLP terminates when the values of the objective function at the previous and current iterates differ by less than  $10^{-\delta}$ . An accurate CFSQP calculation of the optimum reconfiguration for this, not very large, network can take several hours (Pentium 100MHz). This suggests that general NLP solvers may not be able to compute an optimal reconfiguration within the time frame required to reconfigure the network.

Let the set of admissible routes  $\mathcal{R}$  be partitioned into  $\mathcal{R} = \cup_{k=0,1,\dots} \mathcal{R}^k$

Table 2 NLP

$\delta$	5 links			7 links		
	steps	time secs	revenue	steps	time secs	revenue
0	76	760	19 970	95	1698	19 973
1	132	1048	19 995	123	2212	19 945
2	198	2618	20 003	217	3886	20 004
3	254	6256	20 004	223	5635	20 004
6	291	7375	20 004	363	18 679	20 004

where  $\mathcal{R}^k$  is the set of routes for which the number of links in route  $r \in \mathcal{R}_{ij}$  is no greater than  $k_{ij} + k$  where  $k_{ij}$  is the length of the shortest route connecting nodes  $i$  and  $j$ . First consider the route set  $\mathcal{R}^0$ : traffic redistribution yields a revenue of 20 401 and the network blocking is 5.6%. Reconfiguration yields a revenue of 20 004 and a blocking of 7.4%. When applied to the route set  $\mathcal{R}^1$ , redistribution yields a revenue of 21 012 and reconfiguration again yields 20 004.

*Overall blocking.* The network revenue obtained from traffic redistribution is some 2 to 5% higher than the revenue obtained from reconfiguration. The reason for this is that redistribution employs a fully shared network where calls for many O-D pairs are multiplexed onto one route, which can result in the more efficient use of resources. Reconfiguration reserves capacity on a distinct VP for each O-D pair which results in a higher blocking. This difference will be noticeable in networks with relatively small link capacities, low connectivity where many VPs use the same links and light traffic where higher capacities are reserved for O-D pairs with light traffic.

*GOS preservation.* Capacity reservation, which can be considered a drawback of reconfiguration since it results in higher overall blocking, is at the same time an important instrument for GOS protection. For example, suppose the links on all routes connecting nodes  $i$  and  $j$  have 5% blocking and the length of these routes is equal to 5. Traffic redistribution will result in all calls from  $i$  to  $j$  experiencing 25% blocking, which is unacceptable. Reconfiguration does not have such a problem when calls must be carried on long routes since dedicated VPs (logical direct routes) are established for all O-D pairs.

*Conceptual Simplicity.* Traffic redistribution and configuration are relatively simple controls when compared to alternative routing. However, reconfiguration is simpler to implement, operate and optimize. Once the

VPs have been configured, the network transports direct traffic on (logical) direct links. The calculation of the blocking probabilities is straightforward, unlike traffic redistribution which employs the multi-rate Erlang fixed point approximation to compute the blocking. If the traffic conditions change, any resulting performance problems can be isolated and corrected. Thus if blocking for an O-D pair increases, this must be because the capacity assigned to the VP for this O-D pair is no longer adequate and must be increased.

*Computational Complexity.* Reconfiguration and traffic redistribution are intended to operate in near real-time. The computational resources required to calculate an optimal reconfiguration/redistribution are therefore important.

The solution of the traffic redistribution problem (Mitra 1996) has a high computational complexity since each step of the optimization requires solving a large system of linear equations, calculating the link blocking and solving the fixed point equations. In contrast, network reconfiguration is simple and requires much less computation. The multi-rate link blocking probabilities  $B_{ij}^s(c)$  are calculated for  $c = 0, 1, \dots, C$  where  $C = \max_{ij}(C_{ij})$  using a recursive algorithm (Ross 1995) whose computational complexity is  $O(SC)$ . This algorithm is adapted to work with normalized probabilities so that the calculation of the multi-rate blocking probabilities is numerically stable for large values of  $C$ . The probabilities  $B_{ij}^s(c)$  are calculated once only.

The link blocking probabilities are used to compute the rate of earning revenue  $F(\mathbf{x})$  which is given in equation (1). The dimensionality of  $\mathbf{x}$  is equal to the number of routes in the network. For example in a network of 100 nodes, the number of shortest routes between two nodes could be  $O(10^4)$ . Thus even if the NLP was restricted to the shortest routes it could encounter computational difficulties. Discarding longer routes will yield a sub-optimal solution.

*Bandwidth allocation within a VP.* For each O-D pair the reconfiguration procedure constructs a single VP from several routes. In a multi-service environment the blocking on this VP can be higher than the blocking on a shared link of the same capacity. However, if there is a large capacity route in a VP, then high bandwidth calls can be sent over this route and low bandwidth calls can be sent to routes of smaller capacity within the VP. In this case, blocking on the VP will be almost identical to blocking on a single link of the same capacity. Thus an appropriate intra-VP routing strategy can convert this disadvantage of reconfiguration into an advantage: admission policies within each VP can be chosen to optimize the blocking/revenue for all services.



## 5 XFG: AN OPTIMIZATION ALGORITHM

The DROP problem can be solved using NLP methods. However, NLP methods have a high computational complexity. This section presents an efficient deterministic algorithm named XFG to compute an optimal value for the revenue function given in equation (1) subject to constraints of equation (2). The algorithm contains a mechanism to extract itself from inferior local optima.

The revenue function  $F_{ij}(c)$  for each link  $(i, j)$  was defined in equation (1) for integral values of  $c$ . However, most optimization techniques require that  $F_{ij}(c)$  be defined for all real nonnegative  $c$ . Therefore, instead of optimizing  $F(\mathbf{x})$  we optimize  $\tilde{F}(\mathbf{x}) = \sum_{ij} \tilde{F}_{ij}(C_{ij}(\mathbf{x}))$  where  $\tilde{F}_{ij}(c)$  is obtained by fitting a differentiable increasing convex function to  $F_{ij}(c)$  where  $c = C_{ij}(\mathbf{x})$ . We also require that  $\tilde{F}'_{ij}(c) = \partial \tilde{F}_{ij}(c) / \partial c = \theta$  at  $c = 0$  for all  $i$  and  $j$  such that  $C_{ij} > 0$  where  $\theta = \max_{ij}(\theta_{ij}^*/d_s)$  is the maximal rate of earning revenue per unit of capacity. The meaning of this requirement is explained below.

An  $\mathbf{x}$  that realizes an optimal revenue for  $\tilde{F}_{ij}(\mathbf{x})$  will realize an approximately optimal revenue for  $F_{ij}(\mathbf{x})$ . The following arguments support this statement. Although the multi-service link revenue function may be non-monotone, it is asymptotically convex and monotone and in fact it becomes convex and monotone for moderate values of  $c$  in most practical cases. We expect the optimal network configuration to have relatively large capacities for all source-destination pairs, so that the difference  $\tilde{F}_{ij}(\mathbf{x}) - F_{ij}(\mathbf{x})$ , which is the error in the approximation, will be small.

The optimization is based on the concept of link cost. If the route configuration is  $\mathbf{x}$ , the cost of carrying traffic from node  $i$  to node  $j$  is defined as the derivative of  $\tilde{F}_{ij}(c)$  estimated at  $c = C_{ij}(\mathbf{x})$ . This cost is (approximately) equal to the loss in revenue when the direct traffic from node  $i$  to node  $j$  is carried on a VP whose capacity is increased (decreased) from  $c$  to  $c + 1$  ( $c - 1$ ). In this context, the requirement that  $\tilde{F}'_{ij}(0) = \theta$  for all physical links  $(i, j)$  implies that the cost of a physical link from which all direct traffic has been displaced is equal to an overall network maximum. Since the algorithm is based on balancing link costs, this situation where a link is entirely used by nondirect traffic will not occur in an optimal solution.

The necessary condition for  $\mathbf{x}$  to be a local optimum for  $F(\mathbf{x})$  is that for any pair  $(i, j)$  and for any route  $r \in \mathcal{R}_{ij}$  such that  $x_r > 0$  we have

$$\frac{\partial F_{ij}(C_{ij}(\mathbf{x}))}{\partial c} = \sum_{(i,j) \in r} \frac{\partial F_{ij}(C_{ij}(\mathbf{x}))}{\partial c}. \quad (3)$$

Equation (3) can immediately be derived from the fact that all derivatives of the revenue function at the point of extremum must be equal to zero.

*XFG*: an algorithm for finding an optimal VP configuration.

**Step  $n = 0$ :** Construct an initial route configuration  $\mathbf{x}[0]$  which corresponds to the original network where  $x_r = C_{ij}$  for  $r = (i, j)$  and  $x_r = 0$  for all other  $r \in \mathcal{R}_{ij}$ .

**Step  $n \rightarrow n + 1$ :** Compute a configuration  $\mathbf{x}[n + 1]$  which yields a larger revenue than  $\mathbf{x}[n]$ . Construct a weighted graph of the physical network where each link is assigned its current cost. Select an arbitrary pair of nodes  $i$  and  $j$  and compute the least cost route  $r \in \mathcal{R}_{ij}$  between  $i$  and  $j$ . The computational complexity of this step is  $O(L)$ . The following procedures yield a vector  $\mathbf{x}[n + 1]$  which improves the network revenue:

*Cross connect.* If the current cost of the VP connecting nodes  $i$  and  $j$  is greater than the cost of the shortest route  $r \in \mathcal{R}_{ij}$ , then cross connect  $x \in \mathbf{R}^+$  units of capacity along the route  $r$  – this capacity will be subtracted from each link of the route  $r$ . The amount of capacity  $x$  to be cross connected can be found so that it maximizes the increase in revenue (line search).

*Release.* If the current cost of the VP connecting nodes  $i$  and  $j$  is less than the cost of the shortest route  $r \in \mathcal{R}_{ij}$ , then release  $x \in \mathbf{R}^+$  units of capacity along the route  $r' \in \mathcal{R}_{ij}$  such that  $x_{r'} > 0$  and the cost of this route is maximal among all routes  $r \in \mathcal{R}_{ij}$  – this capacity will be returned for the use of the direct traffic offered to each link of the route  $r'$ . Again, the amount of capacity  $x$  to be released can be found so that it maximizes the increase in revenue (line search).

**Stopping rule:** If for all pairs  $(i, j)$  neither cross connection nor release leads to an increase in revenue, then stop.

The order in which the cross connect and release operations are performed, and the order in which the pairs  $(i, j)$  are selected has to be specified. For example, the procedure can be cyclically applied to all pairs, or the pair that yields highest increase in revenue could be selected. The order of cross connect/release operations may influence the speed of convergence of the algorithm.

## 6 AN APPLICATION OF XFGI

An initial version of the algorithm has been implemented where cross connections and releases are made in integer units of capacity. A piecewise linear convex hull  $\tilde{F}_{ij}(c)$  is fitted to the revenue function  $F_{ij}(c)$  and a cubic spline is used to interpolate among the values of  $\tilde{F}_{ij}(c)$ . The convex hull

**Table 3** XFGi VP configuration

	$R^0$	$R^1$	$R^3$	$R^4$
cross connections	17 968	3538	380	25
routes	47	29	13	7

affords a good approximation since  $F_{ij}(c)$  is nearly monotone, and the cubic spline interpolation ensures that the revenue function and its first and second derivatives are continuous. We call this version of the algorithm XFGi where the “i” is an abbreviation for “integral”.

The XFGi algorithm is applied to compute an optimal VP configuration for the model of the NSF backbone network presented in Figure 2. The algorithm constructs VPs in increments of 1 bandwidth unit. The algorithm converges in 21 920 steps, making 21 920 cross connections and 497 releases using 96 routes. The lengths of the cross connected routes are shown in Table 3. Note that the VPs consist primarily of circuits cross connected along the shortest routes connecting the O-D pairs. Twenty direct routes and 79 other routes are used to construct 56 VPs. The XFGi reconfiguration yields a rate of earning revenue of 20 004.2 and the NLP reconfiguration yields 20 004.3.

The computational requirements of the XFGi algorithm are determined by the search for the shortest (in terms of cost) routes which has complexity  $O(JL)$ . If the link capacities  $C_{ij}$  are large, the unit of bandwidth increment in the XFGi algorithm can be increased from 1 to  $x > 1$ . This will decrease the computational complexity of the algorithm although the computed revenue may be lower. Table 4 presents the revenue computed after reconfiguration for various values of  $x$ .

## 7 CONCLUSION

This paper investigates the use of capacity reconfiguration as a resource management control in ATM networks where transmission capacity is reserved on the communication links in order to form dedicated logical paths for each origin-destination flow. We present a model of dynamic reconfiguration which calculates the optimal network configuration in terms of a constrained nonlinear programming problem (NLP). An efficient algorithm is presented to reconfigure a multi-rate network and compute an optimal VP configuration. The algorithm is applied to reconfigure a model of the

**Table 4** XFGi convergence

$x$	<i>connects</i> <i>/releases</i>	<i>routes</i> <i>connected</i>	<i>time</i> <i>secs</i>	<i>revenue</i>
1	21 920/497	79	19	20 004
3	9615/756	80	9	20 004
4	7963/726	79	7	20 004
6	6667/904	80	7	20 004
24	4118/783	77	4	20 004
40	3592/738	74	4	20 004
NLP: CFSQP			3886	20 004

NSF backbone network. We show that reconfiguration can be advantageous for large networks where its drawbacks namely higher overall blocking due to VP service separation, are outweighed by its advantages namely simplicity, efficiency and tractability.

## REFERENCES

- Faragó A., Blaabjerg S., Ast L., Gordos G. and Henk T.. A New Degree of Freedom in ATM Network Dimensioning: Optimizing the Logical Configuration. *IEEE J. on Selected Areas in Comms.*, **13:7**, 1199–1206.
- Kelly F.P. (1991) Loss networks. *Annals of Applied Prob.*, **1:3**, 473–505.
- Lawrence C., Zou J.L. and Tits A.L. (1996) User's Guide for CFSQP Version 2.4. Report Number TR-94-16r1, Electrical Engineering Department and Institute for Systems Research, University of Maryland, College Park, MD 20742 USA.
- Mitra D. and Seery J.B. (1990) Comparative Evaluations of Randomized and Dynamic Routing Strategies for Circuit-Switched Networks. *IEEE Trans. Comm.*, **31**, 102–116.
- Mitra D., Morrison J.A. and Ramakrishnan K.G. (1996) ATM network design and optimization: a multirate loss network framework. *Proc IEEE INFOCOM '96*, 994–1003.
- Ross K.W. (1995) *Multiservice Loss Models for Broadband Telecommunication Networks*. Springer-Verlag, Berlin.