

OPTIMISING EMBEDDED ATM NETWORKS

J. Murphy
T. Curran

Abstract: *In this paper we discuss configuring digital cross-connect systems (DCS) to form an embedded logical asynchronous transfer mode (ATM) network from a physical facility network. A DCS forms a channel, called a pipe, that connects two nodes. Configuration of these switches results in allocation of bandwidth to these pipes.*

The ATM network is modelled as a network of M/M/1 queues and the design of the logical network is formulated as an optimisation problem. The properties of the logical network are investigated and a property of unfragmented flows is found. This is conjectured to apply to more complicated networks.

1. Introduction.

The design of telecommunications networks is an extensively studied subject [2, 6, 11, 16, 17]. The methods for designing these networks are dependent on the network technology being used. For instance, many circuit-switched network design tools, such as Erlangs formulas, are not applicable in packet-switched network design. The models for the technological components of the network vary considerably, depending on their application. For example, in the design of a circuit-switched network switch the model used is generally a blocking one; but when designing a network they are modelled as non-blocking switches. These simplifications and approximations are necessary to enable the network design problem to be solved.

DCS are utilised in both circuit-switched and packet-switched networks [9]. DCS provide a facility cross connection function, in that they do not operate in real time (with respect to the traffic) but rather are part of network management. They allow reconfiguration of the network topology. This can be used to build resilience to node and link failures in a network. It also allows an added layer of flexibility in the topological design of the network.

An ATM network is likely to be built on top of a facility network [7]. A physical network of fibre optic trunks interconnected with DCS will be the facility network. Robustness to failure is only one advantage of the DCS in ATM networks. It is also possible to alter the performance of the ATM network by changing the network topology using the DCS. This is where ATM networks differ from circuit-switched networks. In traditional packet-switched network design the network cost is minimised [6, 17] whereas in this paper the physical network is given and a logical network (using the DCS) is designed to maximise performance.

The goal in designing an ATM network is to produce a cost versus performance comparison. In this way for a particular cost the best performance is found. This network planning in practice would be multiperiod [1] in that future trends would be taken into account. With the flexibility provided by DCS, it is now possible to design a physical network sub-optimally at certain times when optimising the overall performance. Hence the physical network can be different from the desired network. We rely on the DCS to adapt the physical network to the logical network. The aim is to use logical network design with physical network design in the long term planning of the network.

In this paper, we assume that the physical network is given, so that we know the locations of the ATM and DCS nodes as well as the capacities of all the fibre links connecting them. We assume that this physical network is exclusively used for the ATM network, i.e. no sharing of resources will take place between it and other services provided by the facility network. The problem we consider is to map a logical, or embedded, network into this physical network to optimise performance. This might be done as part of long term planning, to evaluate how different physical networks would perform in a multiperiod plan; or it could be part of short term planning, to take recent traffic measurements or node and link failures into account in reconfiguring the network. We allow for the

possibility of disrupting service while the DCS are being readjusted, so that our approach would not be applicable to topology tuning [15].

The rest of the paper is as follows. Sections 2 to 4 introduce the model, the problem formulation and the problem statement. The solution algorithm and the results are presented in sections 5 and 6. In section 7 we conclude with some discussion and suggestions for future work.

2. Model.

ATM switches have cell loss associated with them due to finite buffering. The buffer(s) can be at the input, output, internal or a combination of these. Many ATM switch models have been proposed. However most of the associated queuing models are very complex, so that approximations are used in network design. In this paper we will assume only output buffering. The switch fabric is then assumed to be lossless, as shown in Figure 1. Our approach is to model multilink protocols as queues, with inputs from the ATM switch outputs, and outputs onto 150 Mb/s optic fibre trunks, as in Figure 2.

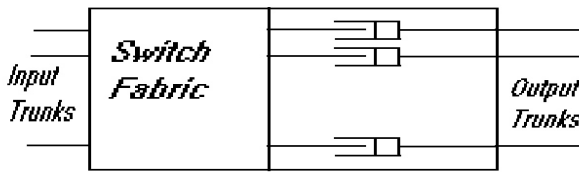


Figure 1. ATM switch model.

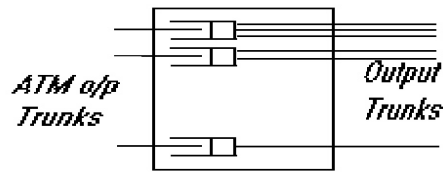


Figure 2. Multilink protocol model.

We amalgamate these queues into a single queue, giving a multilink ATM switch model, shown in Figure 3.



Figure 3. Multilink ATM switch model.

The multilink ATM switch queues are finite and so will have cell loss. As the buffer cell loss model is complex, we take queue length as a measure of cell loss. In turn we take the buffer delay to be a measure of the queue length. Hence we consider the buffer delay to be equivalent to the buffer cell loss:

$$\Pr\{\text{buffer cell loss}\} = (\text{factor}) \cdot (\text{buffer delay})$$

We can't in general evaluate the buffer delay unless we make the classical packet switched network assumptions [6]. We assume that 1) the arrival rate to the buffer is Poisson, 2) the service rate is exponential, even though the packet length is fixed, 3) independence between interarrival times and service times, 4) infinite nodal storage, 5) error free links. Each queue is now modelled as an independent M/M/1 queue. Thus the i^{th} buffer delay is given by $T_i = 1 / (C_i - f_i)$ where C_i is the capacity of the server in cells/s, and f_i is the arrival rate in cells/s, in buffer i .

A DCS introduces a delay, but does not introduce any cell loss while it is fixed in a configuration. Hence we can model DCS as zero cell loss nodes.

3. Problem Formulation.

Users of ATM networks would probably be interested in at least two performance criteria, cell loss probability and cell delay. It appears [7] that the low cell loss probability

(10^{-6} to 10^{-9}) required for some applications is more important than cell delay. Hence we base our performance measure on the probability of successful cell transmission in the network:

$$\begin{aligned}
\Pr\{\text{succ. cell t/x}\} &= 1 - \Pr\{\text{cell loss}\} \\
&= 1 - \sum_{\text{buffers}} \frac{\text{cell arrival rate in buffer} \cdot \Pr\{\text{cell loss in buffer}\}}{\text{cell arrival rate in network}} \\
&= 1 - \sum_{\text{buffers}} \frac{\text{cell arrival rate in buffer} \cdot \text{factor} \cdot \text{cell delay in buffer}}{\text{cell arrival rate in network}}
\end{aligned}$$

Hence by maximising $\Pr\{\text{succ. cell t/x}\}$ we are minimising average cell delay. This contrasts with other proposed performance measures such as the maximum cell delay or a weighted average of cell delays.

The physical network is modelled by a graph $G = (V, L)$, where $V = \{v_1, \dots, v_{N+D}\}$ is the set of nodes and $L = \{l_1, \dots, l_M\}$ is the set of links [4]. There are two types of nodes, ATM and DCS, with N ATM nodes and D DCS nodes and there are M links. Each link $l_n \in L$ has capacity given by the number of cells served per second, C_n . A typical 150 Mb/s trunk with a cell size of 53 bytes would have a capacity of 354,000 cells/s. The total capacity of the network is expressed by the capacity vector $\mathbf{C} = (C_1, \dots, C_M)^T$. A path p_q going from v_i to v_j consists of a list of nodes it passes through $\{v_i, v_{a1}, v_{a2}, \dots, v_{aij}, v_j\}$ where $(v_{a1}, v_{a2}, \dots, v_{aij}) \in \{v_{N+1}, \dots, v_{N+D}\}$ are DCS nodes and $(v_i, v_j) \in \{v_1, \dots, v_N\}$ are ATM nodes. $p_q \in P_{ij}$, where P_{ij} is the set of all possible paths between v_i and v_j . This path can be described by its link-path vector

$$\mathbf{p}_q = (p_{q1}, \dots, p_{qM})^T \text{ where } p_{qn} = \begin{cases} 1 & l_n \in p_q \\ 0 & \text{otherwise} \end{cases}$$

The logical network is modelled by a graph $\bar{G} = (\bar{V}, \bar{L})$, where $\bar{V} = \{v_1, \dots, v_N\}$ is the set of ATM nodes and $\bar{L} = \{\bar{l}_1, \dots, \bar{l}_M\}$ is the set of logical links. A logical link corresponds to one physical path. There can be many physical paths between v_i and v_j and these describe many logical links in \bar{G} . The capacity of a logical link \bar{l}_q is \bar{C}_q which is given in cell/s. A logical path can be described by $\bar{p}_q \in \bar{P}_{ij}$ where \bar{P}_{ij} is the set of all possible logical paths between v_i and v_j . This logical path \bar{p}_q can be represented by its logical path-logical link

$$\text{vector } \bar{\mathbf{p}}_q = (\bar{p}_{q1}, \dots, \bar{p}_{qM})^T \text{ where } \bar{p}_{qn} = \begin{cases} 1 & \bar{l}_n \in \bar{p}_q \\ 0 & \text{otherwise} \end{cases}$$

The traffic offered to the network is between ATM nodes. We form a matrix \mathbf{A} which is N x N, and let a_{ij} represent the traffic from v_i to v_j . \mathbf{A} is not necessarily symmetric but $a_{ii} = 0$ and $a_{ij} \geq 0$. The demand a_{ij} flows on the set of paths \bar{P}_{ij} and $r(i, j; q)$ of it flows on path $\bar{p}_q \in \bar{P}_{ij}$. a_{ij} are in units of cell/s and the total arrival rate to the network is given by $\lambda = \sum_i \sum_j a_{ij}$.

Cell loss in the network occurs in the buffers which are located on the logical links. Denote the logical flow on logical link \bar{l}_q by \bar{f}_q . The cell delay in the buffer is then given by the M/M/1 queue formula: $T_q = 1/(\bar{C}_q - \bar{f}_q)$.

4. Problem Statement.

Many formulations of network design are found in the literature [14]. They correspond to different choices of what is given, the objective function, the constraints and the variables. The overall problem as formulated in [7] is:

- Given: - Topology and capacities of physical network
 - Topology of logical network
 - Offered traffic
- Objective: - Minimise average delay
- Variables: - Logical flows and logical capacities in the logical network
- Constraints: - Satisfy the offered traffic
 - Logical link flow is given by the sum of all logical flows on the logical paths that use that logical link
 - Logical flows should not exceed logical capacities
 - Logical link capacities should not exceed physical link capacities

In this paper we wish to examine the effect on the solution of varying the offered traffic matrix. By varying **A** we should see the set of all possible solutions, as opposed to the usual case where it is fixed and one solution is found.

Actual networks could consist of tens of nodes and hundreds of physical links and are very complex [3]. Rather than making more assumptions to simplify the problem further, we resort to picking an example network with a simple structure that we can solve, as shown in Figure 4. The network chosen demonstrates the possibility of forming logical paths to utilise DCS.

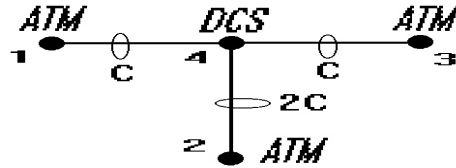


Figure 4. Topology and capacity of physical network

The problem now becomes one of varying the offered traffic matrix and finding the resulting logical network:

- Given: - Topology and capacities of physical network.
- Objective: - Minimise $\left(\frac{1}{\lambda}\right) \sum_i \frac{\bar{f}_i}{\bar{C}_i - \bar{f}_i}$
- Variables: - $\bar{C}_i, \bar{f}_i \quad \forall i \in \{1, \dots, \bar{M}\}$
- Constraints: - 1. $\sum_q r(i, j; q) = a_{ij} \quad \forall v_i, v_j \in \bar{V}$
 - 2. $\bar{\mathbf{f}} = \sum_{i,j} \sum_q r(i, j; q) \bar{\mathbf{p}}_q$
 - 3. $\bar{\mathbf{f}} \leq \bar{\mathbf{C}}$
 - 4. $\sum_q \bar{C}_q \bar{\mathbf{p}}_q \leq \mathbf{C}$
 - 5. $\bar{\mathbf{f}} \geq \mathbf{0}, \bar{\mathbf{C}} \geq \mathbf{0}, \mathbf{C} \geq \mathbf{0}, \mathbf{r} \geq \mathbf{0}$

Constraint 1 states that the offered traffic must be carried on the logical paths. The definition of logical flow is given by constraint 2. Constraint 3 states that the logical flows must not exceed the logical capacities. The logical capacities must not exceed the capacity of the physical network is given by constraint 4. The flows and capacities must be all positive is defined by constraint 5.

This will produce a logical network for every possible \mathbf{A} . The solution will fall into one of three categories; type I, a direct logical path for each demand; type II, combining links to form a single logical link; type III, a combination of both.

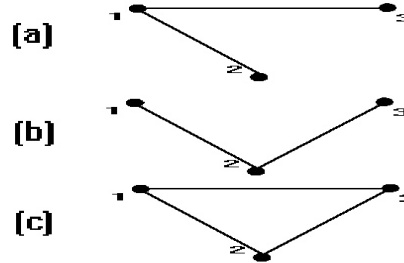


Figure 5. Different logical networks, (a) Type I (b) Type II (c) Type III

5. Solution Algorithm. The objective function is not convex in $(\bar{\mathbf{C}}, \bar{\mathbf{f}})$ and hence there may be multiple local minima [5]. Algorithms have been proposed to achieve the global minimum [12] but they may not terminate within polynomial time [8]. To overcome the problem of the inability to guarantee that a solution is the global minimum, we use different starting solutions and the lowest local minimum is taken to be the global minimum. The objective function is a monotonically increasing function of any a_{ij} . Therefore it is possible to compare the objective function at the solution to that obtained with neighbouring values for \mathbf{A} , to check that when we increase the offered traffic the objective function also increases.

The algorithm proposed is based on a standard IMSL routine for linearly constrained optimisation problems [5, 10].

Algorithm:

- Input* : Physical network description
- Step 0* : Choose $\mathbf{A}(0)$ to be the zero matrix \Rightarrow objective function $\mathbf{O}(0) = \mathbf{0}$
 $k \leftarrow 1$
- Step 1* : Increment $\mathbf{A}(k-1)$ to obtain $\mathbf{A}(k)$
- Step 2* : For each initial solution $s_j(k) \in \mathbf{S}(k)$:
 - Call IMSL routine "LCONG";
 - Find local minimum value of the objective function, $\mathbf{O}_j(k)$ $\mathbf{O}(k) \leftarrow \min_j \mathbf{O}_j(k)$
- Step 3* : Check $\mathbf{O}(k) > \mathbf{O}(k-1)$:
 - No \Rightarrow $\mathbf{O}(k-1)$ not global minimum for $\mathbf{A}(k-1)$;
Add more initial solutions to $\mathbf{S}(k-1)$;
Redo Step 2 for $\mathbf{S}(k-1)$, and check Step 3.
 - Yes \Rightarrow $k \leftarrow k+1$;
Goto Step 1.

6. Results.

For all variations in \mathbf{A} , the results show that the optimum flow on each logical path is either zero or the offered traffic: $(r(i, j; q) \in \{a_{ij}, 0\})$. This means that there are only two possible network solutions: types I and II (Figure 5). By observing the performance function for various values of \mathbf{A} (Figure 6) we note that there are two maxima, one for type I configuration and the other for type II.

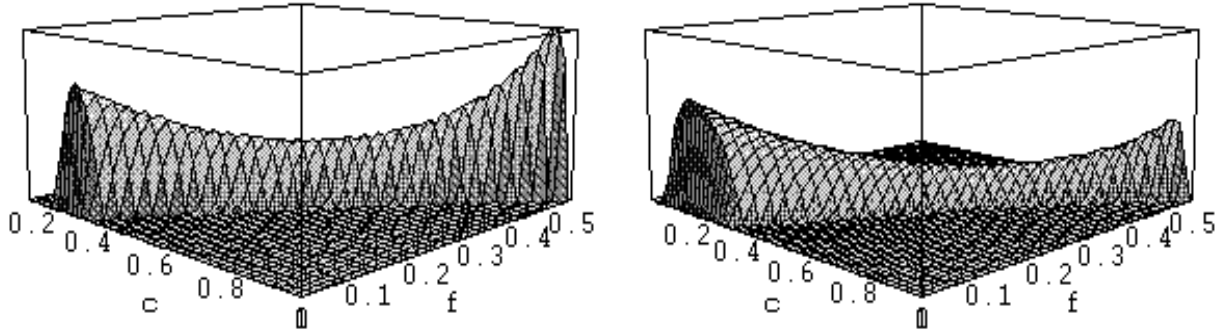


Figure 6. Plot of objective function.

The point at which one type of configuration is better than the other can be evaluated for all values of a_{12} . Thus a plot of a_{13} against a_{12} can be drawn. This gives the two regions of operation depending on a_{12} and a_{13} , as shown in Figure 7.

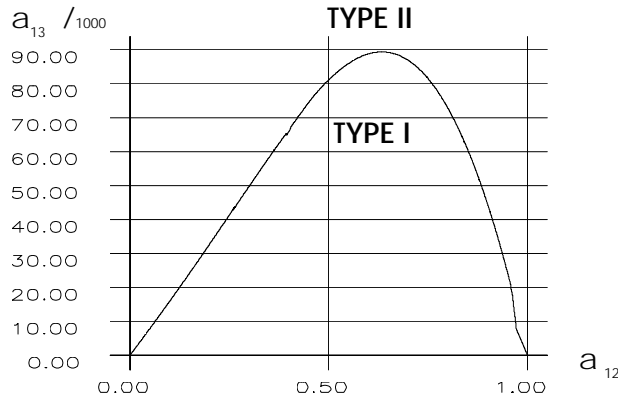


Figure 7. Regions of types of configuration for normalised offered traffic

7. Conclusions and Future Work.

For the network under consideration, the offered traffic flows together in the logical network: it is not fragmented over multiple logical paths. The example in [15] of a more complicated network has no fragmented traffic in the logical network. Larger networks, or networks with more complicated models, might require heuristics for their solution. Our conjecture is that this principle will apply to these networks. However, if the unfragmented transport of the demands would violate the physical path capacity constraint, the traffic must be fragmented. How to fragment in this case is an area for future study.

We have dealt with a simple network example in this paper. It remains to expand this to include either more complicated network models, or more complex networks, or both.

Acknowledgements.

The authors wish to thank Prof. E. C. Posner, California Institute of Technology, for helpful discussions about this work. We also acknowledge the useful suggestions and assistance of Dr. M. Mandell, California Institute of Technology, and Dr. L. Murphy, University of California at Berkeley. We are grateful for the continuing encouragement of this work by Prof. C. McCorkell, Dublin City University.

References.

- [1] A. Dutta and J. Lim, "A multiperiod capacity planning model for backbone computer communication networks" *Opns. Res.* vol. 40, pp. 689-705, 1992.
- [2] V. E. Benes, "Programming and control problems arising from optimal routing in telephone networks", *Bell Syst. Tech. J.*, vol. 45, pp. 1373-1438, 1966.
- [3] V. E. Benes, "Growth, complexity and performance of telephone connecting networks", *Bell Syst. Tech. J.*, vol. 62, pp. 499-539, 1983.
- [4] Dimitri Botvitch, "Optimization of static models of cross connect systems". *Unpublished manuscript, DCU*, 1992.
- [5] R. Fletcher, *Practical Methods of Optimization, Vol. II: Constrained Optimization*, Wiley-Interscience, 1981.
- [6] M. Gerla and L. Kleinrock, "On the topological design of distributed computer networks" *IEEE Trans. Commun.*, vol. COM-25, pp. 48-60, 1977.
- [7] M. Gerla, J. A. S. Monteiro and R. Pazos, "Topology design and bandwidth allocation in ATM nets", *IEEE J. Select. Areas Commun.*, vol. 7, pp. 1253-1262, 1989.
- [8] F. Glover and H. J. Greenberg, "New approaches for heuristic search: A bilateral linkage with artificial intelligence", *EJOR*, vol. 39, pp. 119-130, 1989.
- [9] G. Gopal, C. Kim and A. Weinrib, "Algorithms for reconfigurable networks", in *Proc. 13th Int. Teletraffic Congr.*, pp. 341-347, 1991.
- [10] IMSL, *IMSL Math/Library version 2.0*, vol. 3, pp. 1089-1095, 1991.
- [11] F. P. Kelly, "Blocking probabilities in large circuit-switched networks", *Adv. Appl. Prob.*, vol. 18, pp. 473-505, 1986.
- [12] S. Kirkpatrick, C. D. Gelatt Jr. and M. P. Vecchi, "Optimization by simulated annealing", *Science*, vol. 220, pp. 671-680, 1983.
- [14] L. Kleinrock, *Queueing Systems, Vol. II: Computer Applications*. New York: Wiley-Interscience, 1976.
- [15] J. A. S. Monteiro and M. Gerla, "Topological Reconfiguration of ATM Networks", *IEEE INFOCOM '90*, pp. 207-214, 1990.
- [16] T. M. J. Ng and D. B. Hoang, "Joint optimization of capacity and flow assignment in a packet-switched communications network", *IEEE Trans. Commun.*, vol. 35, pp. 202-209, 1987.
- [17] V. R. Saksena, "Topological analysis of packet networks", *IEEE J. Select. Areas Commun.*, vol. 7, pp. 1243-1252, 1989.