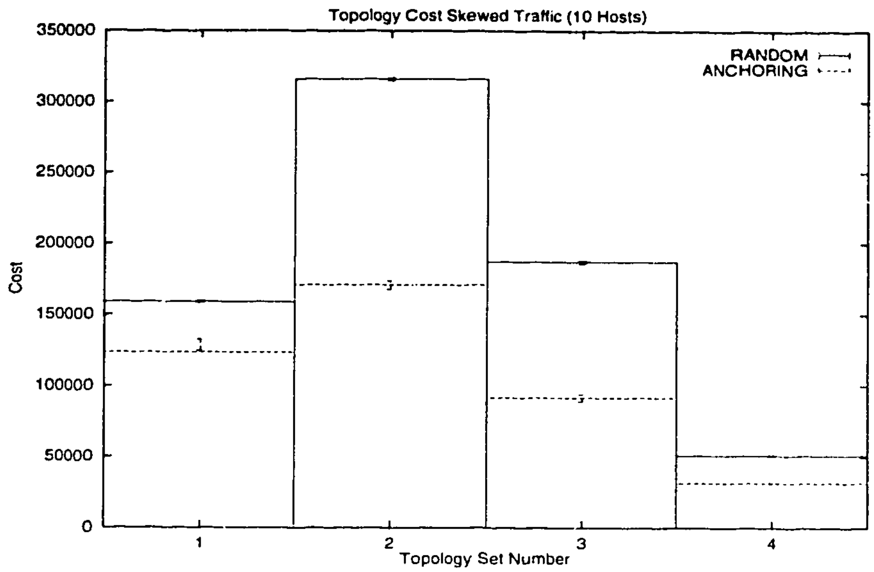

Figure 26: *Comparisons of the Topology Location Algorithms for a 10 Host Network with Skewed Traffic*

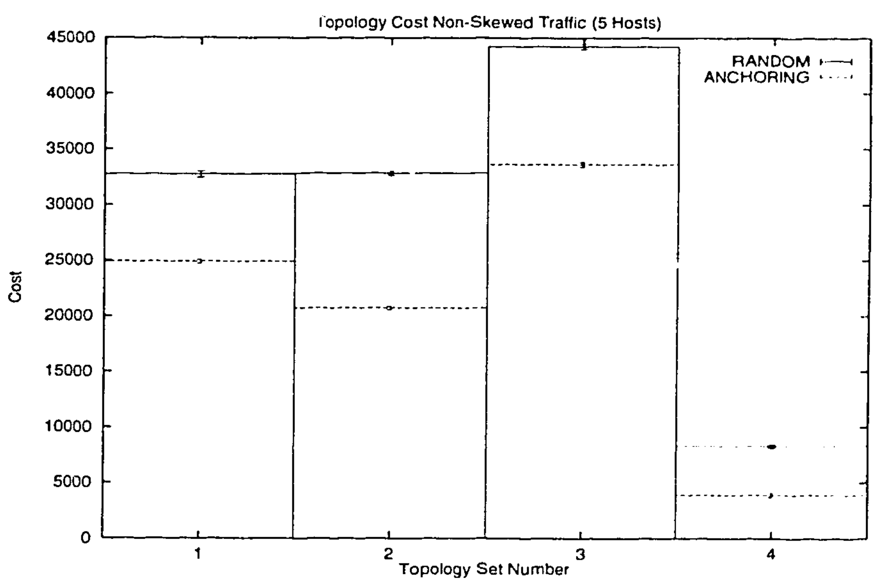

Figure 27: *Comparisons of the Topology Location Algorithms for a 5 Host Network with Non-Skewed Traffic*

two times (Figure 27). This is not surprising at all, especially when some of the switch candidate locations are very far from the host. In the Anchoring algorithm, these locations will be avoided. However, for the Random algorithm, these switch candidate locations can still be chosen as the place to put switches.

4.7.6 Algorithm Running Time

Although the Exhaustive algorithm emerge as the winner in Part II, it is not without its downside. The major problem with this algorithm is its exponential increase in execution time with the rise in number of hosts in the network. Figure 28 shows the execution time for the Part II algorithms, and 29 shows the execution time for the Part III algorithms. These are real time not CPU time calculations and may not be completely accurate indicators of the execution speed since other processes may be present on the system. However, the results are consistent over the many runs and the differences in execution time for the different algorithms are large.

As we can see from Figure 28, for a 20 host case the execution time for the Exhaustive algorithm goes up to 8606 seconds, which is about 2.3 hours. These are real time measures and therefore, the differences are consistently significant. Thus, for large host size, for example 100 hosts, the execution time for the Exhaustive algorithm may become too long to be of any use. Nevertheless, since this method is specifically designed for local area networks, chances are that the number of hosts in the network will not reach that high.

For the Anchoring algorithm, even though it is also showing an exponential increase in execution time with the increase of host number, the actual time that it takes to run is very small (in fractions of a second even for 20 hosts). Hence, we believe that long running time will not be the problem for larger hosts numbers.

4.7.7 Compatibility of Part II and Part III

Table 4.4 shows the results of some of the runs, before and after applying the topology location Anchoring algorithm to them.

We can see that in some runs, even though the Random topology improvement

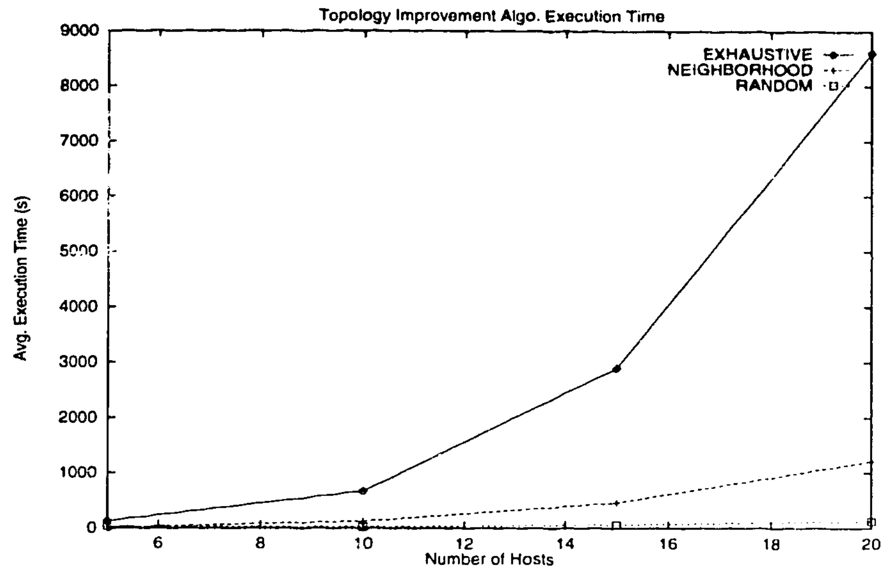

Figure 28: *Average Execution Time for the Topology Improvement Algorithms (in Seconds)*

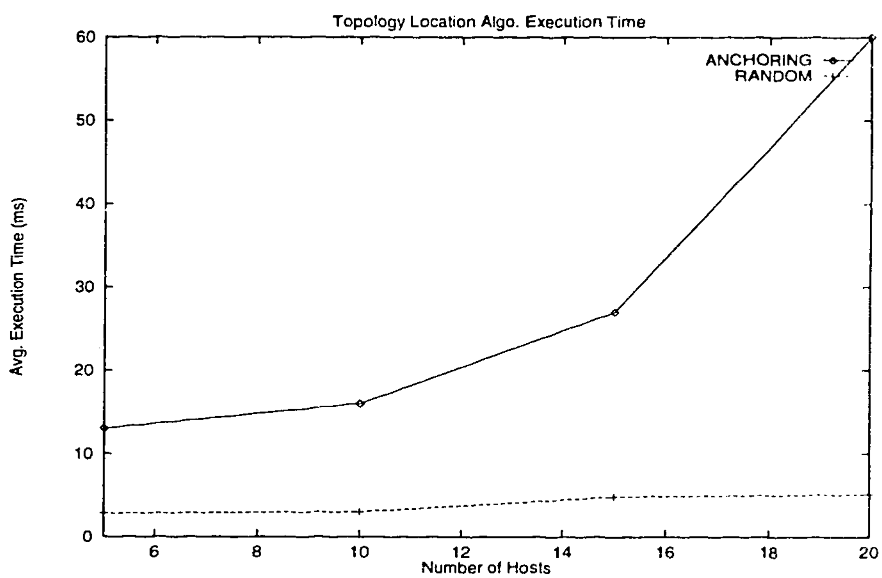

Figure 29: *Average Execution Time for the Topology Location Algorithms (in Milliseconds)*

	Random		*Exhaustive*		*Neighborhood*	
	Before	After	Before	After	Before	After
1.	162235	123292	156210	127473	160164	124131
2.	317830	171938	315964	176189	315088	176699
3.	174901	87669	167203	97435	170006	89114
4.	49336	30867	47636	30808	48343	30825

Table 4.4: ***Results of the Three Topology Improvement Algorithms Before and After Anchoring***

algorithm loses in the beginning, after applying the anchoring algorithm, it reverses and becomes the winner (Sets 1 to 3). This is an interesting observation, as we thought that 'minimizing' the cost in Parts II and III separately should yield good results. However, as we have seen, this is not always true. This implies that it would be better to combine both Part II and Part III together so that we can take into account both switch locations and switch combinations at the same time. Another view is that there is no sense using "intelligence" at the topology improvement stage; just choose something quick.

4.8 Summary

In this chapter, Phase II of our Pipe-Based Topology Design Methodology is presented in detail. The results of the experiments that were run to analyze the algorithms in Phase II are also presented. From the results we can conclude that:

- The Initial Topology algorithm successfully generates a feasible topology which the Part II topology Improvement algorithms will improve.
- The algorithms proposed in Part II successfully reduces the size of the initial topology by combining switches.
- The Exhaustive algorithm in overall exhibits the most savings at the Topology Improvement Stage, followed by the Neighborhood algorithm, then by the Random algorithm.

- However, the Exhaustive algorithm has the longest execution time, followed by the Neighborhood algorithm, then by the Random algorithm.
- The Anchoring Location algorithm assigns switches to better locations than the Random Location algorithm in all cases.
- The separation of Part II and Part III will not always give good results. For future improvements, Part II and Part III should be combined.

Chapter 5

Conclusion

5.1 Summary

In this thesis, we have proposed a novel method for building topologies for local area ATM networks (LATM). This method utilizes our concept of ATM pipes as the fundamental building blocks of the network. In determining the size of these pipes, we have developed a mathematical method (the Multiclass Heterogeneous On/Off Sources Fluid-Flow Method, MCFF) by extending on the work of Tucker [39]. Then, we proposed a 3 part methodology to construct the LATM network. We also introduced and compared 6 variations of the implementation of this methodology.

We started with Chapter 1, where we presented a general discussion on topology design. Then in Chapter 2, an overview of some techniques on topology from previous literature was given.

In Chapter 3, we presented an overview to our methodology for building a LATM network. Basically, there are two phase to our method. The first phase involves determining the pipe sizes required, whereas the second phase concerns with building the topology based on the calculated pipe sizes. The first phase of our method, i.e. the MCFF model was presented in detail. To verify the accuracy of the MCFF, results from some simulation runs that we have performed was presented. If was found that the MCFF is very accurate, since the values calculated with MCFF lie within the 95% confidence interval of their corresponding simulation runs.

The second phase of our method was presented in Chapter 4. Each of the proposed algorithms was outlined, and performance of the algorithms are given. It was found that for topology improvement, the best algorithm is the Exhaustive algorithm, followed by the Neighborhood algorithm, then the Random algorithm. For topology location, Anchoring performs better than Random topology location in all cases. However, overall when Part 3 is applied it may reverse the result of Part 2, and hence may mean that we should not separate the 2 functions. Other results are, skewness of pipe sizes has little effect on the results, and that the execution times of the algorithms are significantly different.

Conclusions and future work are then summed up in this Chapter.

5.2 Conclusions and Future Work

In this section we conclude our method successfully generates LATM topologies while satisfying the QOS (specifically, the traffic loss probability) requirements. One of the advantages of our work is that the method is highly modular. That is if a better solution exists for the parts, we can simply substitute the new solution into that part. Our method is also practical, since pipes can be implemented as virtual paths. Another conclusion is that the MCFF model is very accurate. However, improvements can still be done to model other types of traffic as well. A disadvantage of our method is that since the topology construction phase is further divided into three separate parts, problems of incompatibility between the parts may occur. We have seen this with the overall cost of the finished network. One algorithm at the topology improvement stage may be the best but is poorer at the final stage.

Possible future work for this thesis are

- *Pipe Calculation* Currently, we assume that the traffic in the pipes are fixed rate. Research can be done, so that other traffic types, such as variable bit-rate traffic (VBR) such as VBR voice, compress video can be taken into account. At this point, this traffic would be handled by this model if peak rate is applied. Another possible research work is modelling the effects of cell scale as well as burst scale. MCFF is currently a burst scale model.

- *Pipe Usage* This involves investigating the effects of grouping certain types of traffic together, or the usage of pipes. In theory, a source can send traffic to the destination in more than one pipe. By having multiple pipes for a single source to the destination, essentially that means that a source can have a choice of different loss rates to send its traffic. Another good item for investigation is the idea of a broadcast pipe. This is in anticipation of the multicast or broadcast traffic on the network.

- *Pipe Configuration* After a network has been installed, the traffic that runs on the network changes through time. The pipe configuration that was initially done at the time of installation may not apply anymore. Research can be done to see how a new pipe configuration can be derived from the underlying switched network without changing the physical topology.

- *Physical Topology Redesigning* Research can be carried out in this area to determine how a new physical topology using ATM pipes can be added on a existing ATM pipe network with little or no change to the existing topology.

Appendix A

Tucker's Homogeneous On/Off Sources Fluid-Flow Model

This appendix presents Tucker's Fluid-Flow Model for multiplexed homogeneous On/Off voice sources, on which the model presented in this thesis for the calculation of a multiclass, heterogeneous On/Off sources is extended in Chapter 3.

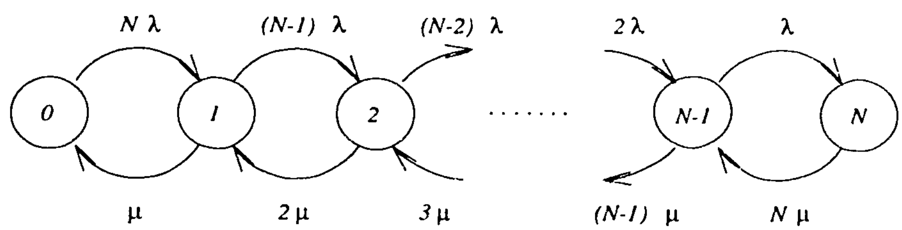

Figure 1: *The Talker Activity Model Markov Chain*

The Fluid-Flow Model is based on a Markov Chain, as shown in figure 1, This Markov Chain is also known as the Talker Activity Model.

The state of the Markov Chain, is represented by the number in the circle denotes the the number of On/Off sources in the On state out of a total of N sources. Each source is a voice source that alternates between an identical exponentially distributed On period (talkspurt), where voice data is generated and an exponentially distributed Off period, where no voice data is produced (silence). These On/Off periods has a mean of

$$
\begin{aligned}
1/\mu &= \text{mean talkspurt length} \\
1/\lambda &= \text{mean silence length}
\end{aligned}
$$

With the exponentially distributed On/Off periods, the birth and death rates for the Talker Activity Model are then

$$
\begin{aligned}
p(i, i+1) &= (N-i)\lambda, i \neq N \\
p(i, i-1) &= i\mu, i \neq 0
\end{aligned}
$$

where $p(i,j)$ denotes the transition rate from state i, where i sources are in their On periods, and to state j, where j sources are on. The total flow-rate out of state i is then

$$p^*(i) = p(i, i+1) + p(i, i-1) = (N-i)\lambda + i\mu$$

The multiplexed data produced by these N voice sources, after being fed into a FIFO buffer, are then transmitted on to a link with capacity c units. Since Tucker assumes that each voice source generates voice data at a capacity of 1 unit, the contents in the buffer can be observed as follows:

1. If the number of On sources i is less than c ($i < c$), the queue decreases steadily at a rate of $(c-i)$. If the queue becomes empty, it will remain empty.

2. If ($i > c$), the queue increases steadily at a rate of $(i-c)$. If the queue reaches its limit, it will stay on its limit.

3. If ($i = c$), the queue length does not change.

Let $P_i(t,x), 0 \leq i \leq N, 0 \leq t, 0 \leq x \leq m$ be the probability that i sources are on at time t, and the queue length in the buffer is $\leq x$, and that the queue limit is m. Then

$$
\begin{aligned}
P_i(t+\Delta t, x) &= P_{i-1}\{t, x-(i-c)\Delta t\}p(i-1,i)\Delta t \\
&\quad + P_{i+1}\{t, x-(i-c)\Delta t\}p(i+1,i)\Delta t \\
&\quad + P_i\{t, x-(i-c)\Delta t\}(1-p^*(i)\Delta t) + o(t) \qquad \text{(A.1)}
\end{aligned}
$$

Dividing both sides by $\triangle t$ and taking $\triangle t \to 0$, with a little manipulation equation (A.1) becomes

$$\delta P_i(t,x)/\delta t + (i-c)\delta P_i(t,x)/\delta x = p(i-1,1)P_{i-1}(t,x)$$
$$+p(i+1,i)P_{i+1}(t,x) - p^*(i)P_i(t,x), \quad 0 < x < m. \qquad (A.2)$$

If we let $F_i(x) \triangleq$ the equilibrium probability ($\lim_{t\to\infty} P_i(t,x), \delta P_i/\delta t = 0$) there are i on sources and the queue length $\leq x$, equation (A.2) can be rewritten as

$$(i-c)\, dF_i/dx = p(i-1,i)F_{i-1}(x) + p(i+1,i)F_{i+1}(x)$$
$$-p^*(i)F_i(x), \quad 0 < x < m. \qquad (A.3)$$

Or in matrix form

$$\mathbf{D}\, d\mathbf{F}(x)/dx = \mathbf{MF}(x), 0 < x < m \qquad (A.4)$$

where

$$\mathbf{D} = \begin{bmatrix} -c & & & & \\ & 1-c & & & \\ & & 2-c & & \\ & & & \ddots & \\ & & & & N-c \end{bmatrix}$$

and

$$\mathbf{M} = \begin{bmatrix} -p^*(0) & p(1,0) & & & & & \\ p(0,1) & -p^*(1) & p(2,1) & & & & \\ & p(1,2) & -P^*(2) & p(3,2) & & & \\ & & & \ddots & & & \\ & & & & p(N-2,N-1) & -p^*(N-1) & p(N,N-1) \\ & & & & & p(N-1,N) & -p*(N) \end{bmatrix}$$

Hence, there are $(N+1)$ state equations to be solved, assuming c is not an integer. If z_r be an eigenvalue of $\mathbf{D}^{-1}\mathbf{M}$ and ϕ_r be its corresponding right eigenvector. The solution to (A.4) becomes

$$\mathbf{F}(x) = \sum_{r=0}^{N} \exp(z_r x) a_r \phi_r, \quad 0 < x < m \qquad (A.5)$$

By setting suitable boundary conditions, the coefficients a_r for (A.5) can be obtained. Defining $F_i(m-)$ as $\lim_{x \to m} F_i(x)$, and b_i = the equilibrium probability that i lines are on, that is

$$b_i = \frac{(\lambda/\mu)^i \binom{N}{i}}{(1+\lambda/\mu)^N}$$

the boundary conditions can be set by observing the fact that

1. for $(i > c)$, the buffer contents is always increasing, therefore the probability that the buffer contents will be less than or equals to zero is nil. In other words, $F_i(0) = 0$

2. for $(i < c)$, the buffer contents is always decreasing. Hence, $F_i(m-) = b_i$. Equation (A.4) can then be solved by standard numerical algorithms.

The loss can be calculated by

$$\text{Loss} = (1/(\alpha N)) \sum_{i=C}^{N} (i-c)(b_i - F_i(m-)) \qquad \text{(A.6)}$$

where α is the average fraction of lines on $\alpha = \lambda/(\mu + \lambda)$, and $C = \lceil c \rceil$.

Bibliography

[1] D Anick, D Mitra, and M Sondhi. Stochastic theory of a data-handling system with multiple sources. *Bell System Technical Journal*, 61:1253–1257, 1982.

[2] Sara Baase. *Computer Algorithms: Introduction to Design and Analysis*. Addison-Wesley Publishing Company, 1988.

[3] Andrew Campbell, Geoff Coulson, and David Hutchison. A quality of service architecture. *Computer Communication Review*, 24(2):6–27, Apr 1994.

[4] John Daigle and Joseph Langford. Models for analysis of packet voice communications systems. *IEEE Journal of Selected Areas in Communications*, 4(6):847–855, Sep 1986.

[5] M. de Prycker, R. Peschi, and T. Van Landegem. B-ISDN and the OSI protocol reference model. *IEEE Network*, 7(2):10–18, March 1993.

[6] Martin de Prycker. *Asynchronous Transfer Mode: Solution for Broadband ISDN*. Ellis Horwood, 1991.

[7] Anwar Elwalid and Mitra Debasis. Fluid models for the analysis and design of statistical multiplexing with loss priorities on multiple classes of bursty traffic. *IEEE Infocom*, 1:415–425, 1992.

[8] Noboru Endo, Toshiya Ohuchi, Takahiko Kozaki, Hiroshi Kuwahara, and Makoto Mori. Traffic characteristics evaluation of a shared buffer switch. *IEEE Globecom*, 3:1913–1918, 1990.

[9] Cem Ersoy and Shivendra Panwar. Topological design of interconneccted LAN-MAN networks. *IEEE Infocom*, 3:2260–2269, 1992.

[10] Michael Garey and David Johnson. *Computers and Intractability. A Guide to the Theory of NP-Completeness.* W. H. Freeman and Co., 1979.

[11] Mario Gerla, Jose Augusto Monteiro, and Rodolfo Pazos. Topology design and bandwidth allocation in ATM nets. *IEEE Journal on Selected Areas in Communications*, 7(8):1253–1262, October 1989.

[12] Donald Gross and Carl M. Harris. *Fundamentals of Queueing Theory.* John Wiley and Sons, 1985.

[13] Harry Heffes and David M Lucantoni. A markov modulated characterization of packetized voice and data traffic and related statistical multiplexer performance. *IEEE Journal of Selected Areas in Communications*, 4(6):856–867, September 1986.

[14] Ellis Horowitz and Sartaj Sahni. *Fundamentals of Computer Algorithms.* Computer Science Press, 1978.

[15] W Hsieh, M Gerla, P McGregor, and J. Eckl. Locating backbone switches in a large packet network. In *Proceedings of the 3rd International Conference on Computer Communication*, pages 330–339, Toronto, 1976.

[16] Aaron Kershenbaum. *Telecommunications Network Design Algorithm.* McGraw-Hill, 1993.

[17] Aaron Kershenbaum, Parviz Kermani, and George A. Grover. Mentor: An algorithm for mesh network topological optimization and routing. *IEEE Transactions on Communications*, 39(4):503–513, April 1991.

[18] S Kirkpatrick, C. D. Gelatt, and M Vecchi. Optimization by simulated annealing. *Science*, pages 671–680, May 1983.

[19] Leonard Klienrock. *Queueing Systems: Vol I.* John-Wiley and Sons, 1976.

[20] L Kosten. Stochastic theory of a multi-entry buffer. *Delft Progress Report*, pages 10–18, 1974.

[21] H. Kroner, M. Eberspacher, T. H. Theimer, P Kuhn, and U. Briem. Approximate analysis of the end-to-end delay in ATM networks. *IEEE Infocom*, pages 978–986, 1992.

[22] Hiroshi Kuwahara, Nonoru Endo, Mineo Ogino, and Takahiko Kozaki. A shared buffer memory switch for an ATM exchange. In *Proceedings: Internationl Conference on Communications*, pages 118–122, 1989.

[23] Averill M. Law and Kelton W. David. *Simulation Modeling And Analysis*. McGraw-Hill, Inc., 1991.

[24] Ming-Jing Lee and James R. Yee. A design algorithm for reconfigurable ATM networks. *IEEE Infocom*, 1:144–151, 1993.

[25] Ian Leslie, McAuley R., and David Tennenhouse. ATM everywhere? *IEEE Network*, 7(2):40–46, March 1993.

[26] Ke-Qiang Liao and Lorne Mason. A heuristic approach for performance analysis of ATM systems. *IEEE Globcom*, 3:1931–1935, 1990.

[27] Chu-Kiat Lim and Ursula Maydell. Xpress transfer protocol in a medical communication system. In *(to appear) Proceedings of SPIE, The International Society of Optical Engineering. Medical Imaging V: PACS Design and Evaluation*, Newport Beach, February 1994.

[28] J Bryan Lyles and Daniel C. Swinehart. The emerging gigabit environment and the role of local ATM. *IEEE Communications Magazine*, pages 52–58, Apr 1992.

[29] B. Martos. *Nonlinear Programming Theory and Methods*. Amsterdam and Oxford: North-Holland, 1975.

[30] Claude Mcmillan Jr. *Mathematical Programming: An Introduction to the Design and Application of Optimal Decision Machines*. John Wiley and Sons, Inc., 1970.

[31] Steven E. Minzer. Broadband ISDN and Asynchronous Transfer Mode (ATM). *IEEE Communications Magazine*, pages 17–24, September 1989.

[32] Jose Augusto Monteiro and Mario Gerla. Topological reconfiguration of ATM networks. In *Proceedings of IEEE Infocom*, pages 207–214, San Francisco, California, June 1990.

[33] Yuji Oie, tatsuya Suda, Masayuki Murata, David Kolson, and Hideo Miyahara. Survey of switching techniques in high-speed networks and their performance. In *IEEE Infocom*, pages 1242–1251, 1990.

[34] Tadanobu Okada, Hirokazu Ohnishi, and Naotaka Morita. Traffic control in Asynchronous Transfer Mode. *IEEE Communications Magazime*, pages 58-62, September 1991.

[35] R. Onvural. *Asynchronous Transfer Mode Networks: Performance Issues*. Archtech House, Inc., 1994.

[36] HuanXu Pan, Hiroyuki Okazaki, and Issei Kino. Analysis of a gradual input model for bursty traffic in ATM. *Teletraffic and Datatraffic (IAC)*, pages 795–800, 1991.

[37] Craig Patridge. *Gigabit Networking*. Addison-Wesley Publishing Company, 1994.

[38] Youichi Sato and Ken-Ichi Sato. Virtual path and link capacity design for ATM networks. *IEEE Journal of Selected Areas in Communications*, 9(1):104–109, January 1991.

[39] Roger C. F. Tucker. Accurate method for analysis of a packet-speech multiplexer with limited delay. *IEEE Transactions on Communications*, 36(4):479–483, April 1988.