

A Tool for Multi-Hour ATM Network Design considering Mixed Peer-to-Peer and Client-Server based Services

Contact Author Name: Luis Cardoso

Company / Organization: Portugal Telecom Inovação

Complete Mailing Address: Rua Eng. José F. Pinto Basto, 3810 Aveiro, Portugal

Telephone Number: +351.234.403299

Facsimile Number: +351.234.403550

Email Address: luis-o-cardoso@ptinovacao.pt

Co-Authors: Carlos Lopes; Amaro de Sousa; Rui Valadas

Theme: Revolution in Network Planning: Tools and Methods

Sub theme: Designing for Varying Performance Levels

Abstract

This paper presents a network design tool for ATM networks, which allows for multi-hour dimensioning of networks supporting simultaneously peer-to-peer and client-server based services. The dimensioning procedure assumes aggregation of Virtual Circuit Connections into fixed bandwidth end-to-end Virtual Path Connections. In peer-to-peer services, connections are established between all node pairs of the service. In client-server based services (e.g., video-on-demand, fast Internet access), connections are between service nodes and servers. In this case, the dimensioning procedure allows for the selection of the best server for each node. The design model is formulated as a linear integer programming problem, which is solved through a heuristic based on Lagrangean relaxation with sub-gradient optimization. This approach allows the computation of a theoretical lower bound, which is very useful in assessing the quality of the obtained solution. The network design tool includes a graphical interface for an easy introduction and edition of the network parameters. Results show that the tool can design large networks with reasonable computational speed and good accuracy.

1 Introduction

ATM networks need to support a variety of services with different traffic profiles and Quality-of-Service (QoS) requirements, at both call and cell level. In ATM, flows of Virtual Circuit Connections (VCCs) can be aggregated in Virtual Path Connections (VPCs). Using VPCs, it is possible to configure different logical networks on top of the physical network [1]. We assume that VPCs are end-to-end and are assigned a fixed bandwidth. The dimensioning of an ATM network aims at finding the network configuration (the bandwidth and route of each VPC) that achieves the lowest cost given the operational cost of the network elements and the call and cell level QoS requirements [2] [3]. Several features can be considered in the process of network design. For example, it is common sense that existing services do not have the same traffic behaviour during time. In general, business services have higher offered traffic in periods that are complementary to residential services. If the network has dynamic reconfiguration capabilities, i.e., time is partitioned in periods and the network configuration is allowed to change between periods, a multi-hour design procedure can lead to significant savings in the overall network cost [4].

Typically, existing services are either peer-to-peer or client-server based. In peer-to-peer services, calls or connections (depending on whether the services are switched or permanent) can be requested between any pair of nodes with attached users. Thus an end-to-end VPC must be configured for each user node pair. In client-server services (e.g., video-on-demand, fast Internet access), a user can request a call (or a connection) to one of the server nodes of the service. Thus an end-to-end VPC must be configured between each user node and a server node. As far as we know, existing design tools deal only with peer-to-peer services.

This paper presents a software tool for the design of ATM networks, called PTPlan ATM. PTPlan ATM performs multi-hour design and considers mixed peer-to-peer and client server services. Furthermore, in the case of client-server services the tool allows the selection of the best server node for each user, from a set of pre-defined candidate server nodes. In section 2 we present the model and problem formulation used in the design tool. In section 3 we show the main features of the design tool. In section 4 we discuss a case study and present some conclusions.

2 Model and Problem Formulation

The design procedure can be separated in the following steps: (i) VPC bandwidth assignment; (ii) link capacity calculation, routing optimisation and server selection (for client-server services). The last step can be jointly addressed by the linear integer programming formulation presented below. Since VPCs are end-to-end, the capacity assignment problem can be addressed individually for each VPC by (i) calculating the number of VCCs that observe the specified blocking probability and (ii) calculating the VCC bandwidth that observes the specified cell level performance parameters. The first step can resort to the Erlang B formula, under the usual assumptions of Poisson arrivals and independence between the arrival process and the service distribution. The second step can be solved assuming peak bit rate allocation or, more aggressively, exploiting potential statistical multiplexing gains through the use of appropriate statistical multiplexing models.

Let the network be represented by an undirected graph (N, A) whose nodes and arcs represent ATM switch locations and available transmission facilities between switch locations. Each element of A is defined by an undirected arc (i, j) , with $i, j \in N$. The set of possible interface

types to install in any arc is denoted by T and α_t is the bandwidth of interface type $t \in T$. Let Y_{ij}^t denote the maximum number of interfaces of type $t \in T$ that can be installed on (i, j) . The operational and maintenance cost associated with the use of one interface of type $t \in T$ in the arc (i, j) is denoted by C_{ij}^t . Let H be the set of time periods h . Each VPC k_h is defined by the origin node $o(k_h)$, the set of possible destination nodes $d(k_h) \in D(k_h)$, the bandwidth in the direction from origin to destination $b(k_h)$, and the bandwidth in the direction from destination to origin $\underline{b}(k_h)$. Set $D(k_h)$ is a single destination for peer-to-peer services and the set of server nodes for client-server services. The optimisation model uses the following set of variables. Integer variables y_{ij}^t that define the number of interfaces of type t that are installed on arc $(i, j) \in A$. Route binary variables x_{ij}^{kh} ; when equal to one, defines that VPC k_h passes through arc $(i, j) \in A$ in the direction from node i to node j . Route binary variables \underline{x}_{ij}^{kh} ; when equal to one, defines that VPC k_h passes through arc $(i, j) \in A$ in the direction from node j to node i . The following integer programming model determines the lowest cost physical network given by the optimal routes for each VPC and the optimal server locations:

$$\text{Minimize} \quad \sum_{(i,j) \in A} \sum_{t \in T} C_{ij}^t y_{ij}^t \quad (1a)$$

Subject to:

$$\{x_{ij}^{kh}, \underline{x}_{ij}^{kh} : x_{ij}^{kh} = 1 \wedge \underline{x}_{ij}^{kh} = 1\} \text{ is a path from } o(k_h) \text{ to } d(k_h) \in D(k_h), k_h \in K_h, h \in H \quad (1b)$$

$$\sum_{k_h \in K_h} (b(k_h) \cdot x_{ij}^{kh} + \underline{b}(k_h) \cdot \underline{x}_{ij}^{kh}) < \sum_{t \in T} \alpha_t y_{ij}^t, \quad (i, j) \in A, h \in H \quad (1c)$$

$$\sum_{k_h \in K_h} (b(k_h) \cdot \underline{x}_{ij}^{kh} + \underline{b}(k_h) \cdot x_{ij}^{kh}) < \sum_{t \in T} \alpha_t y_{ij}^t, \quad (i, j) \in A, h \in H \quad (1d)$$

$$y_{ij}^t < Y_{ij}^t, \quad (i, j) \in A, t \in T \quad (1e)$$

$$x_{ij}^{kh} \in \{0, 1\}; \quad \underline{x}_{ij}^{kh} \in \{0, 1\}; \quad y_{ij}^t > 0 \text{ and integer}$$

The objective function (1a) represents the total cost of the network solution as a function of the number of interfaces of each type installed in each network arc. Constraint (1b) forces the solution to choose a path from origin to destination for all VPCs to be supported by the network. Constraints (1c) and (1d) impose that the total bandwidth installed in each arc is enough to support the bandwidth occupied by the VPCs that cross the arc defined in (1a) and (1b) in all time periods. Finally, constraint (1e) imposes that the number of interfaces of each type in each arc is not greater than a defined maximum value. This model does not separate the determination of the network topology from the link capacity dimensioning. Some existing solutions determine the network topology a priori in order to reduce the arc set A and reach a network dimensioning problem which is smaller and easier to solve. Separation of the two problems results in a procedure that, although faster, leads to higher cost solutions.

The solution to the optimisation problem is obtained using Lagrangean relaxation with sub-gradient optimisation. Departing from the original optimisation problem, a new optimisation problem is obtained by applying Lagrangean relaxation to constraints (1c) and (1d), which we

will refer to as the LLBP (Lagrangean Lower Bound Problem). To derive the LLBP, a set of Lagrangean multipliers is introduced, one for each of the relaxed constraints. For any arbitrary set of non-negative Lagrangean multipliers, the solution of LLBP is a lower bound of the original problem [5]. To compute different sets of Lagrangean multipliers, we use sub-gradient optimisation [6]. This technique is an iterative process that, for a given set of Lagrangean multipliers, calculates another set of multipliers that try to maximize the objective function value of the LLBP. At the end of the procedure, a final solution and a theoretical lower bound for the original problem can be calculated. The solution of the LLBP for the x variables is a shortest path calculation. The standard procedure uses the Dijkstra algorithm in each iteration and, in this case, any VPC route can become part of the final solution. Alternatively, the user can specify a priori a set of candidate routes for each VPC and, in this case, the LLBP will select only from this set of candidate routes. This feature can be used to trade-off accuracy and execution time, in networks with a large number of nodes. The detailed procedure can be found in [7].

3 The Network Design Tool

In PTPlan ATM, the user inputs the network topology, the service characteristics, a case study definition and (optionally) it can impose one or more routes. Using these data the tool performs network design and traffic growth analysis. PTPlan ATM runs on a Windows platform and includes a graphical interface (Figure 1) where the user can enter and edit the network topology (nodes and links between nodes). The user can create a library of interfaces that the network nodes can select from. Examples of interface types are PDH 34 Mb/s and SDH 155 Mb/s. A library of nodes can also be created. Each node is defined by the set of interfaces it can support. After node definition, each link is automatically assigned the set of interfaces that are common to its adjacent nodes. In addition the user can associate a distance and a maximum utilization to each link and define the maximum number of interfaces of each type that the link can support. Both switching and transmission costs can be assigned to each interface in each link. Transmission costs can be based on link length. The cost of each interface is calculated as $2 \times \text{switching cost} + \text{transmission cost} \times \text{link length}$. There is also the possibility of creating subnets. In the “service characterization” window (Figure 1.a), services can be selected as being permanent or switched, VBR or CBR, and peer-to-peer or client-server. Permanent services are characterized by cell level QoS requirements and switched



Figure 1 – Graphical user interface of PTPlan ATM.

services by both call level and cell level QoS requirements. At this window, the user can also define the blocking probability, the mean call duration, the source traffic model parameters (peak bit rate, mean burst size and utilization), and also the buffer size and cell loss ratio required for the calculation of the effective bandwidth. There is a choice of effective bandwidth models. In the “case study” window (Figure 1.b), the user defines the number of time periods (for multi-hour designs), which nodes have attached users and, in the case of client-server nodes, which nodes can have servers. The tool automatically generates the origin-destination pairs, in the case of peer-to-peer services, and the origin nodes in the case of client-server services. After that the user is asked to define the offered traffic for each origin-destination pair. After defining all network parameters, the user may perform uni-hour or multi-hour designs. The results are the VPC routes, the number of interfaces of each type to install in each link and the overall network cost.

4 Case study

In order to illustrate the capabilities of the design tool we have defined a network scenario with the following characteristics. The network topology is the one represented in Figure 1, with 21 nodes and 46 links. There are 3 interface types: PDH 2 Mb/s, PDH 34 Mb/s and SDH 155 Mb/s with associated switching costs 100, 1000 and 3000, respectively, and associated transmission costs 1, 10, and 40, respectively. We have considered two services and two time periods. The services are (i) a switched VBR peer-to-peer service with peak bit rate 128 Kb/s, mean burst period 10 s, utilisation 0.2, cell loss ratio 0.01, grade of service 0.01, mean call duration 20 s; (ii) a permanent CBR client-server service with a rate of 40 Kb/s in the client-server direction and 4000 Kb/s in the server-client direction. There are end users for both services at all the nodes and two servers for the client-server service at Porto and Lisboa. The offered traffic was assigned randomly using a uniform distribution between 0 and 100 channels for the CBR service and 0 and 100 calls/s for the VBR service. We designed this network in several cases: shortest path VPC routes (with distance based link metric), uni-hour and multi-hour design. In addition, for the client-server service, we considered two cases: (i) fixed server, where each user is attached to the closest (in number of hops) server, and (ii) selected server, where the tool is allowed to select the best server for each node. The results are as follows:

Cost (time)	Shortest path	Uni-hour	Multi-hour
Fixed Server	541274 (2.92 sec.)	535826 (18.29 sec.)	471060 (33.12 sec.)
Selected Server	517845 (2.91 sec.)	513146 (18.45 sec.)	463223 (33.50 sec.)

As expected gains are obtained when adopting a multi-hour approach and a selected server strategy: (i) in the selected server case there is a 9.7% gain between multi-hour and uni-hour; (ii) selected server is superior to fixed server, 1.7% in multi-hour and 4.2% in uni-hour; (ii) there is a gain of 13.5% between uni-hour / fixed server and multi-hour / selected server. We note that, although the results can vary significantly with the network scenario, in general considerable gains are obtained by resorting a multi-hour approach and selected server strategy. To appreciate the effect of statistical multiplexing we have converted the VBR service into a CBR service with the same peak bit rate. In the multi-hour case, the cost increased to 4251850.

In [7] we tested the performance of the dimensioning procedure. The results reproduced in Figure 2 show that, in the uni-hour case, the dimensioning procedure can design small networks (up to 30 nodes) in less than one minute and medium sized networks (up to 50 nodes) in less than four minutes. In the multi-hour approach used two different time periods,

and as such, the required time is approximately two times larger than the time required by the uni-hour approach. These results were obtained using a 200 MHz Pentium MMX with 128 MB of RAM running MS Windows 98 operating system. Figure 2 also shows the duality gap (best and work case calculated over 10 runs) which is given by $(S-B)/B$, where S is the best solution and B the theoretical lower bound, for the uni-hour case. The duality gap is a measure of the quality of the solution since the optimal solution is always between the obtained solution and the theoretical lower bound. Results show that the duality gap improves for networks with higher number of nodes. This behavior is very important since for small networks, there are standard linear programming algorithms that calculate the optimal solution for the dimensioning problem.

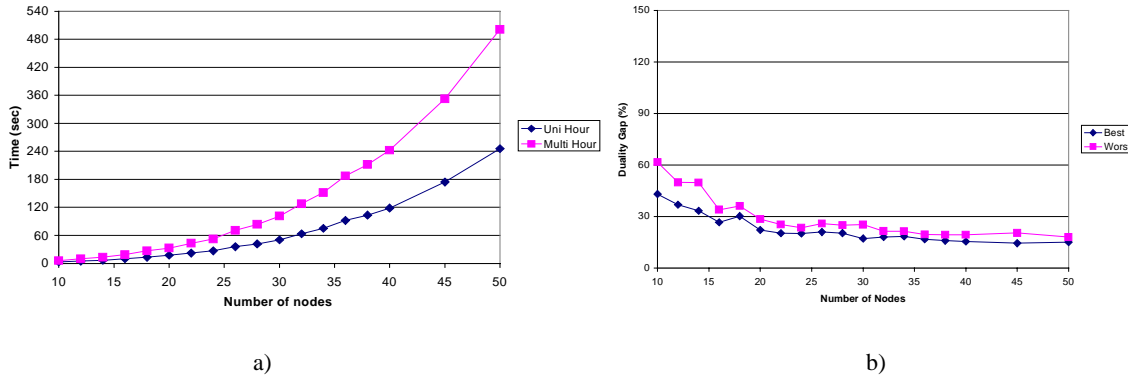


Figure 2 – Performance of the design procedure in terms of a) speed and b) duality gap.

5 References

- [1] R. Valadas, “Dimensioning and Resource Management of ATM Networks”, *Proc. 7th IFIP/ICCC Conference on Information Networks and Data Communications - INDC’98*, Aveiro, Portugal, June 15-17, 1998, pp. 209-220.
- [2] T. Bauschert, M. Perske, L. Turczyk, “Advanced Broadband Network Planning with SICAP-ATM”, *Proc. 8th International Telecommunications Network Planning Symposium – Networks’98*, Italy, 1998, pp. 593-598.
- [3] K. Ramakrishnan, D. Mitra, J. Morrison, “VPN DESIGNER: A Tool for Design of Multiservice Virtual Private Networks”, *Proc. 8th International Telecommunications Network Planning Symposium – Networks’98*, Italy, 1998, pp. 153-158.
- [4] D. Medhi, “Multi-Hour, Multi-Traffic Class Network Design for Virtual Path-Based Dynamically Reconfigurable Wide-Area ATM Networks”, *IEEE/ACM Transactions on Networking*, Vol. 3, No. 6, 1995, pp. 809-818.
- [5] J. Beasley, “Lagrangian Relaxation”, *Modern Heuristic Techniques for Combinatorial Problem*, Ed. by Colin Reeves, Blackwell Scientific publications, 1993.
- [6] M. Held, P. Wolfe, H. Crowder, “Validation of subgradient optimisation”, *Mathematical Programming*, vol. 6, pp. 62-88, 1974.
- [7] A. de Sousa, R. Valadas, L. Cardoso, A. Duarte, “ATM Network Dimensioning for Mixed Symmetrical and Asymmetrical Services with Dynamic Reconfiguration in a Multi-Network Provider Environment”, *Proc. Third IFIP Workshop on Traffic Management and Design of ATM Networks*, 1999, pp. 5/1-5/15.