ELSEVIER

Information Processing Letters 65 (1998) 179–182

Information Processing Letters

The Bandwidth Allocation Problem in the ATM network model is NP-complete

Sundararajan Vedantham [1], S.S. Iyengar *

Department of Computer Science, Louisiana State University, Baton Rouge, LA 70803, USA

Received 1 July 1997; revised 1 December 1997
Communicated by S.G. Akl

Abstract

We show that the Bandwidth Allocation Problem in the ATM network model is NP-complete. Based on that inference we suggest using the Genetic Algorithm technique to select a subset of calls from the set of incoming call requests for transmission, so that the available network bandwidth is utilized effectively, thus maximizing the revenue generated while preserving the promised QoS. © 1998 Elsevier Science B.V.

Keywords: ATM; NP-complete; Genetic algorithms; Computational complexity

1. Introduction

The Asynchronous Transfer Mode (ATM) network model has been evolving as the standard for future networking that is expected to carry voice, real time video and a large volume of still images in addition to the growing volumes of computer data. ATM networks are predominantly expected to be implemented using optical fiber with very low data error rate and guaranteed Quality of Service (QoS). It works on the assumption that the required bandwidth for transmission will be available throughout the connection time; the QoS deteriorates drastically when the bandwidth requirements are not met by the network. In realistic situations, a large number of audio, video and data calls with varying traffic characteristics being initiated by different sources simultaneously, can be vying for the available network bandwidth. Depending upon the design of the network, several different parameters that define the characteristics of the source traffic (such as peak rate of information generation, acceptable delay, data loss, etc.) are communicated to the network during the call initialization procedure. This process is similar to lifting a telephone receiver and dialing a number we wish to reach. The network needs to understand the requirements and determines if there is enough bandwidth available from the source to destination(s) to complete the call properly.

Depending upon the bandwidth requirements and the rigor with which the network is expected to ensure a high QoS, the revenue generated by different calls may vary widely. Other than the revenue, there can be several factors (like call content, source and destination, etc.) making one call more important than another. Under such circumstances, the network needs to allocate the available bandwidth in the most efficient fashion. This process is called the *Bandwidth Alloca-*

* Corresponding author. Email: iyengar@bit.csc.lsu.edu.
[1] Email: sundar@bit.csc.lsu.edu.

PII S0020-0190(98)00226-3

tion Problem or BAP. In this paper we first prove that the bandwidth allocation problem in ATM networks is NP-complete. Then we show that the genetic algorithm paradigm can be effectively used to handle the problem.

2. BAP is NP-complete

In this section, we pose the Bandwidth Allocation Problem (BAP) in the ATM network model and prove that it is NP-complete. The problem is defined as follows:

Problem. Given that $N_1, \ldots, N_K$ connections of classes $1, \ldots, K$ (traffic classified under one class possesses similar cell flow characteristics and QoS requirements) are offered to a link of bit rate B, can the network select enough calls to utilize the available bandwidth fully and generate the maximum possible revenue while the QoS requirements are satisfied for all classes? [1].

Theorem 1. *BAP is* NP-*complete.*

Proof. We propose to prove the problem NP-complete by the following two steps:

(1) We first show that BAP belongs to the class NP.
(2) Then we show that the Knapsack Problem, well known to be NP-complete, is a restricted version of the BAP and this simple reduction can be carried out in polynomial time.

We initially redefine the BAP stated above more precisely.

Problem Instance: $N_1, \ldots, N_K$ connections of classes $1, \ldots, K$, bandwidth requirement B_k and revenue generation value R_k for each connection class, a bit rate B, and a revenue goal R.

Question: Can the network select enough calls generating revenue $\geqslant R$ and total bandwidth requirement $\leqslant B$ presuming QoS is always satisfactory?

It is easy to see that BAP $\in$ NP since a nondeterministic algorithm designed to solve the problem has to simply guess a collection of transmission requests that will be accommodated in the available channel with bit rate B and verify the following in polynomial time:

- check whether the selected calls fit with in the available bandwidth.
- Check whether the generated revenue is more than R.
- Check if the QoS was satisfactory.

The Knapsack Problem instance is defined as follows [2].

Problem instance: A finite set U, a "size" $s(u) \in Z^+$ and a "value" $v(u) \in \mathbb{Z}^+$ for each $u \in U$, a size constraint $B \in \mathbb{Z}^+$, and a value goal $L \in \mathbb{Z}^+$.

Question: Is there a subset $U' \subseteq U$ such that $\sum_{u \in U'} s(u) \leqslant B$ and $\sum_{u \in U'} v(u) \geqslant L$?

Now we can show that the Knapsack Problem is a special case of BAP. The finite set U defined in Knapsack is the set of incoming calls, $s(u)$ is B_k, $v(u)$ is R_k, Knapsack size is B, value goal L is R. This simple reduction, known as Restriction, can be carried out in constant time. This proves that the BAP problem as defined above is NP-complete. □

Since the BAP is proven to be NP-complete, there can be no polynomial time deterministic algorithm that solves the problem optimally. So in the next section we attempt using Genetic Algorithm as a tool to develop a solution that may be close to the optimal solution.

3. Genetic Algorithm

There are several computational paradigms in existence today that provide heuristic solutions to problems that are difficult to analyze deterministically. Simulated annealing, Neural networks, Genetic Algorithm (GA) are some such techniques. Each paradigm boasts certain virtues and strengths that may be very useful in handling a set of hard problems. In general, simulated annealing works well with large size problems that does not require on-line solutions. Compared to GA, neural networks may be harder to implement and modify frequently as the selection criteria for calls changes continuously. So we present a Genetic Algorithm based solution here. Plausibly, a new heuristic algorithm designed to handle BAP specifically might compete well with the Genetic Algorithm in providing an effective solution for BAP.

Table 1
Comparison of performance

Type of algorithm	Data	Audio	Video	Bandwidth used	Revenue
Hand computation	0	50	0	100%	$150
Greedy Algorithm 1	100	0	0	100%	$100
Greedy Algorithm 2	0	0	25	100%	$125
Genetic Algorithm	8	34	6	100%	$140

3.1. Call selection

We consider a situation where a router has to select and accept a subset of incoming calls for transmission so as to maximize the use of the available bandwidth. The maximization is defined by the value of revenue generated and the percentage of available bandwidth utilized by the calls that are accepted. We assume that there are more incoming calls than the available bandwidth.

We classify the incoming calls into Type 1 (Data), Type 2 (Audio) and Type 3 (Video) calls. They require 1, 2 and 4 units of bandwidth respectively. The revenue generated by each type of call is $1, $3 and $5 per call for data, audio and video respectively. The total bandwidth available for transmission is 100 units. Calls vying for the bandwidth are a mix of all three types. We developed two greedy algorithms that allocate bandwidth to the incoming calls based on a specific criteria. We compared the revenue generated under these schemes to the revenue generated by the calls selected as per the genetic algorithm solution. Results consistently showed that the allocation made by the genetic algorithm is better than the allocation made by the greedy algorithm for the situations we analyzed.

The genetic algorithm can adapt to the variations that occur in the types of incoming calls during different times of the day or month and modify the selection criteria accordingly so that the utilization of the bandwidth is close to optimum most of the time. The down side is the extra computational resources, both time and computing power, required to execute the GA. But given a reasonably powerful processor, time requirement for even thousands of GA iterations can be kept under a few milliseconds thus making it suitable for real time scenario. Alternatively, the execution of GA can be carried out off-line and the results can be used to modify the selection criteria of a simpler algorithm running in the foreground to improve the call selection process in successive iterations.

We used SUGAL (SUgarland Genetic ALgorithm) [3], a Genetic Algorithm software package to study the various possibilities. Table 1 presents the bandwidth utilization and revenue generated information for solutions provided by two different greedy algorithms, genetic algorithm and a brute-force optimum solution for a sample problem instance. We note that the solution provided by the genetic algorithm is better than the ones provided by the greedy algorithms.

4. Analysis of results

We used a simple example above to illustrate the concept, so that the actual optimum solution could be computed easily by brute force. We can see that the bandwidth utilization is 100% in all the cases. The Greedy Algorithm 1 tried filling up the available bandwidth with as many number of calls as possible, resulting in 100 data calls getting accepted generating $100 revenue. Greedy Algorithm 2 paid attention only to the cost of the calls. Since video calls produce the maximum revenue per call, it chose to accept 25 video calls, generating $125 revenue. While doing so, it fails to note that audio calls, while producing lower revenue per call compared to video calls, generate better revenue since their bandwidth requirement per call is lower than video calls.

Genetic Algorithm on the other hand, cannot only handle such subtleties adeptly but can also adapt very well to changing revenue generation parameters. In the static case we discuss, it out performs the greedy algorithms as we can observe from Table 1. It gave a solution accepting 8 data, 34 audio and 6 video calls generating $140 with 100% bandwidth utilization. Even in more complicated situations, depending upon the resulting value of the evaluation function, it can change the priority given to different calls. For example, depending upon the fee structure, it may give priority to one call type in the day time and another at night or accept more broadcast calls compared to one-to-one calls, etc. The attractive feature here is that no periodic intervention will be required to reprogram the algorithm since the genetic algorithm can use the successive revenue generated figures to adapt itself sufficiently.

5. Conclusions

We showed that the problem of effective bandwidth utilization in the ATM network model is NP-complete. We also demonstrated that the Genetic Algorithm paradigm can be used to solve the problem. If the calling terminal is forced to pay differing charges for different links it uses during a transmission, there will be another version of the BAP that is even more complicated. Variations in charge could be in terms of fee paid, congestion encountered, delay in call setup, etc. depending upon the resource availability on the network. Allocating bandwidth taking such criteria under consideration could be equally difficult, if not more. But even such cases can be handled using GA, since it does not require much intervention as evaluating conditions change.

References

[1] J.-Y. Le Boudec, The asynchronous transfer mode: A tutorial, Comput. Networks ISDN Systems 24 (1992) 279–310.

[2] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, Freeman, San Francisco, 1979.

[3] A. Hunter, SUGAL User Manual v20, http:// osiris. sund.ac.uk/ahu/sugal/home.html, 1995.