The Design, Modeling and Simulation of Switching Fabrics for an ATM Network Switch

Dmitriy Molokov, Muhammad Shaaban,
James Heliotis, Andreas Savakis.
Rochester Institute of Technology.
Department of Computer Engineering.

Keywords: ATM, communication, switch, fabric, traffic.

ABSTRACT

The requirements of today's telecommunication systems to support high bandwidth and added flexibility brought about the expansion of Asynchronous Transfer Mode (ATM) as a new method of high-speed data transmission. Various analytical and simulation methods may be used to estimate the performance of ATM switches. Analytical methods considerably limit the range of parameters to be evaluated due to extensive formulae used and time consuming iterations. They are not as effective for large networks because of excessive computations that do not scale linearly with network size. One the other hand, simulation-based methods allow determining a bigger range of performance parameters in a shorter amount of time even for large networks. A simulation model, however, is more elaborate in terms of implementation. Instead of using formulae to obtain results, it has to operate software or hardware modules requiring a certain amount of effort to create. In this work simulation is accomplished by utilizing the ATM library -- an object oriented software tool, which uses "software chips" for building ATM switches. The distinguishing feature of this approach is cut-through routing realized on the bit level abstraction treating ATM protocol data units, called cells, as groups of 424 bits. The arrival events of cells to the system are not instantaneous contrary to commonly used methods of simulation that consider cells as instant messages. The simulation was run for basic multistage interconnection network types with varying source arrival rate and buffer sizes producing a set of graphs of cell delays, throughput, cell loss probability, and queue sizes. The techniques of rearranging and sorting were considered in the simulation. The results indicate that better performance is always achieved by bringing additional stages of elements to the switching system.

1 INTRODUCTION

The purpose of this work is to evaluate the performance of switching fabrics for ATM network switches. Simulation seems to be the only effective way to observe the cooperation of the sequence of elements in an ATM switch and to predict its behavior under selected conditions. Since there are several factors affecting the switching fabric behavior and many parameters need to be evaluated, such an evaluation cannot be made analytically. The most appropriate way of simulation is software simulation.

Different approaches were involved in the simulation of switch components and particularly switching fabrics. The common way of simulating networks, as many sources suggest, is to make use of a synchronous network, in which at each clock cycle messages move from stage *i* to stage *i*+1. This method was proposed in [1]. Networks are usually constructed of "software chips" and the organization of the simulation model is object oriented. The examples from sources [1] and [4] served as the starting point in the creation of the simulation tool implemented in C++ called "ATM library".

This discrete time simulation software package makes it possible to construct models of all the ATM switch components and to organize communication among them, thus reproducing the ATM switch operation. The ATM library was carefully elaborated, adapting many useful features of "Arch library" [5] used for simulation of computer architectures. The ATM library simulates switching elements, queues, connectors, and two modules for interfacing with input sources and output locations, as components of an ATM switch. The interfacing modules called sockets substitute the input and output modules by reading the data from the input files and writing it to the output files. An input socket is also able to accept traffic from dynamic traffic generators based on [4].

This simulation tool models a synchronous network, where time is measured by clock cycles equal to the intervals, during which one bit propagates between two stages of a network. Worm-hole or cut-through routing of ATM cells represented by streams of bits makes a cell arrival event not instantaneous, thus extending one arrival to a time slot equal to 425 clock cycles. Hence, the traffic arrival rate has an upper bound. Common simulation models treat message arrivals as instantaneous events, which mimics store-and-forward routing. The simulation results with non-instantaneous arrivals are only comparable with the results obtained analytically or by usual way of simulation when very large number of simulation traces is

K. Elleithy et al. (eds.), Advances in Computer, Information, and Systems Sciences, and Engineering, 275–281.

generated to make one time slot infinitely small relatively to the overall simulation time.

The sections of the paper are grouped as follows. The second section gives the overview of the basic switching methods used in the ATM traffic management, followed by the description of the switching part of the simulation model in section 3 and the traffic generation part in section 4. The fifth section offers the simulation results of the various network architectures, and the last sixth section summarizes the results of this work.

2 SWITCHING FABRICS AND TECHNIQUES OVERVIEW

Modern communication systems are built from transmission links interconnected with routers. Routers rearrange the connections among the links according to a packet or a circuit switching regulation. Particularly, ATM standard accepts switching of cells preserving their order within routers. A router consists of a set of input and output port controllers connected with an interconnection network. In order to establish a path within a router, called switch in the ATM standard, a sequence of commutations is performed in the switching elements forming the interconnection network.

Various types of network architectures exist. A network may be blocking or non-blocking. Blocking networks may deny requests to set up new connections due to conflicts with already established connections. Non-blocking architectures are further divided into strict sense non-blocking -- always allowing establishing a new connection, wide sense non-blocking -- preventing conflicts through proper policy of allocating the connections, and rearrangeable non-blocking -- giving the way for a new connection by rearranging already existing paths.

A banyan network presented in Figure 1 is a blocking architecture. This type of architectures is the simplest and the most scalable comparatively to all other types. All the architectures of banyan type are subject to internal conflicts at the intermediate stages that substantially reduce their performance. To avoid this, lots of alternatives to banyan networks reducing contention in the fabric, i.e. set of switching elements, are used. One group of alternatives is completely interconnected networks, such as knockout or crossbar tree with each input connected to each output. Another group is fully connected networks where every element is connected to every element in the contiguous stage. The network with the largest number of switch-boxes is the crossbar switch. It eliminates all internal conflicts. Other techniques of conflict reduction are network replication, link dilation, and extended general shuffle. Conflicts at the output stage between cells directed to the same output port cannot be eliminated.

Of the variety of conflict reducing techniques, in this work the methods of rearranging and sorting are investigated. Usual multistage interconnection networks

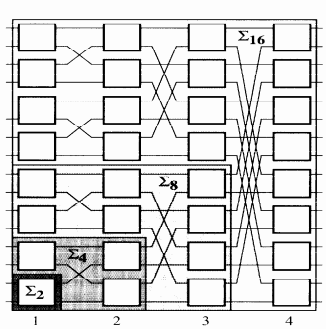

Fig 1. An example of banyan network -- SW - banyan [3].

provide only a single path between each input port and each output port, which leads to internal conflicts. Rearrangeable networks have alternative ways to route cells. For example, Benes rearrangeable network simulated in this work is derived from omega network by the horizontal extension technique discussed in [3]. According to this technique, m additional stages are incorporated into the available banyan network architecture of n stages to increase the number of paths from any input to any output from 1 to 2^m, where m is no greater than n-1. But even with rearranging, the number of paths may not be sufficient enough to eliminate all the internal conflicts as opposed to sorting. The sorting techniques are even more effective and sometimes allow complete eliminating of internal conflicts, although at even higher cost. The simulation model discussed in the next section evaluates performance of these techniques.

3 SIMULATION APPROACH

The simulation methodology in this work is based on the ATM library. This tool allows evaluating the time in clock cycles required to pull a certain amount of traffic through an ATM switch. It is possible to obtain the average and the maximum queue sizes in all the objects. Additionally, the absolute maximum queue size and the overall average queue size are determined for all the objects of the same type: Queue's and SwElement's separately. These two classes are further defined in this section. The other evaluated parameters are the number of conflicts in all the switching elements, the total number of lost cells in all objects, and the cell loss probability. For each output port, the minimum, average, and maximum cell delay is determined, and for all output ports, the absolute maximum and minimum delays and the overall average delay are calculated. The ATM library uses as input data either artificial traces or traffic produced by traffic generators discussed in section 4. UML diagramme of the ATM library without the traffic generation part is in Figure 2. The ATM library has a class called BaseClass from which