This is the published version


Available from Deakin Research Online

http://hdl.handle.net/10536/DRO/DU:30005039

Reproduced with the kind permission of the copyright owner

Copyright: 2003, International Society for Computers and Their Applications
A Simulation Model for Video Traffic Performance via ATM over TCP/IP

Morshed U. Chowdhury, Yiqing Tu, and Thao Pham
School of Information Technology,
Deakin University, 221 Burwood Highway, Burwood, Victoria, Australia 3125
Email: {muc, ytu}@deakin.edu.au

Abstract: Although TCP has emerged as the standard in data communication, the introduction of ATM technology has raised numerous problems regarding the effectiveness of using TCP over ATM networks, especially when video traffic performance is considered. This paper presents a simulation model for transmission performance of video traffic via ATM over TCP/IP. The interactivity between TCP/IP and ATM, generation of MPEG traffic and evaluation of traffic performance are implemented in the model. The design and implementation details of the model are carefully described. The experiments conducted using the model and experimental results are briefly introduced, revealing the capability of our model in simulating network events and in evaluating potential solutions to performance issues.

Index Terms: MPEG video, ATM, TCP/IP, network simulation.

I. Introduction

In the last few years, the Transmission Control Protocol (TCP) has emerged as the standard in data communication. However, the introduction of the Asynchronous Transfer Mode (ATM) technology has raised many questions regarding the effectiveness of using TCP over ATM networks. Recent studies have shown that TCP/IP, when implemented over ATM networks, is susceptible to serious performance limitations [5].

To locate and address the network performance issues, simulation is a vital tool. A simulator, if designed properly, can exploit the complicated interaction between TCP and ATM. Gurski and Williamon [4] presented a simulation model as a part of the ATM-TN network to study the TCP performance on ATM. The work of Shah [12] developed a simulation tool based on Network Simulator Version 1 (NS1), facilitating the establishment of TCP traffic on top of the ATM network. Nevertheless, NS1 has been developed and updated into Network Simulator Version 2 (NS2), and numerous new features with network standard have been built into it. As a result, Shah's model is not compatible with these new features and is not suitable for simulating the network events under the latest standard.

Currently, a major part of the network traffic will be produced by multimedia sources like tele-conferencing terminals and video-on-demand servers. Moreover, the transmission of real time traffic has an extremely strict time delay and delay variability requirements that must be met [8]. A variety of traffic characterization techniques have been presented for the purpose of modelling and reproducing the needed traffic source [9, 10 and 11]. Based on this, efficient and accurate network planning tools will facilitate network designers and engineers to access and predict traffic performance of the variable volume of video trace over the underlying networks.

This paper presents a simulation model for video traffic performance over TCP/IP and ATM networks. The structure of the system is described, and the modification and adaptation to existing models based on NS2 is provided in detail. The experiment conducted using the model and the experimental result is briefly introduced, showing the capability of our model in simulating network events and in evaluating potential solutions to performance issues.

2. Background on TCP/IP and ATM

TCP is a transport-layer protocol designed to provide reliable, error-free transmission of data between two hosts. TCP, which is a connection-oriented protocol, transmits application data in units of bytes called segments. The longest permitted length of segment data is called the Maximum Segment Size (MSS). TCP protocol provides error control, flow control and congestion control through a variety of techniques.

Asynchronous Transfer Mode (ATM) is a method by which fixed-length units called cells are individually transmitted to a destination. Each ATM cell is 53 bytes in length. ATM is based on statistical multiplexing of packets so that it is able to support a wide assortment of services including voice, video and data all over the same network.

Currently, large amounts of network traffic are transmitted via TCP/IP over ATM. However, a number of studies show that certain features of ATM and the way in which TCP interacts with ATM may cause considerable degradation to TCP throughput. One of the problems is the effect of ATM cell loss on TCP/IP. Other performance issues involve ATM
cell tax, an ATM connection setup delay effect of large ATM MTU [5].

There has been a lot of work that measures network performance on TCP over ATM network [4, 12]. Most of the work investigates the problem of CBR traffic in terms of network performance. On the other hand, when video traffic is delivered over the network, the interaction between TCP and ATM may not only influence network performance, but also have a tremendous impact on the application level. Therefore, we put more emphasis on application performance in this paper.

3. Background on MPEG Traffic

At the moment, the MPEG coding scheme is widely used for any type of video application. This MPEG family of standards includes MPEG I, MPEG II and MPEG4. In addition MPEG7 is under development. Because MPEG4 is aimed to provide high video quality at relatively low bit rates and has the capability of real-time adaptive encoding, it has become most suitable for Internet applications among other schemes. In this paper, we focus on the transmission performance of video data streams of MPEG4 type.

A variety of techniques has been used to model network traffic, including autoregressive moving average (ARMA) models, Bernoulli process models, Markov chain models, neural network models, self-similar models, spectral characterization, transform-expand-sample (TES) models, traffic flow models, and wavelet models [6, 11]. It has been determined that not only short range but also long range dependencies can be found and should be described in video traffic [2].

One of the above-mentioned approaches is based on TES models [7], which are suitable for modelling VBR traffic. Using the TES modelling approach, data that closely match given observations of a time series is generated, and both the matching of the histogram and the matching of the autocorrelation are implemented.

4. NS2- A simulation tool

Our simulation model is based on the tool of Network Simulator (Version2) (NS2). NS2 is an object-oriented, discrete event driven network simulator. NS2 is especially useful for simulating various IP networks. Firstly, it implements network protocols and various traffic source behaviours. Secondly, router queue management mechanisms and routing algorithms are realized. In addition, other protocols used in a LAN environment are also implemented in NS2, such as multicasting and MAC layer protocols.

NS2 [1] is an OTCL (Object-oriented version of TCL) script interpreter, which consists of a simulation event scheduler, network component object libraries and libraries for building networks. When NS2 is used to simulate the network activities, an event scheduler is initiated at first, then, the network topology is set up, using existing network objects and the functionalities for setting up the network.

After the network topology is set up and the simulated process is performed, the simulation result is produced. The result is generally one or more text-based output files that contain detailed simulation data. The data can be used for further analysis, or it can be input to a graphic simulation display tool, called Network Animator (NAM), to be visualized.

NS2 has an initial version called NS1. Compared with NS1, NS2 has several additional characteristics: i). complex objects are decomposed into simple ones that are readily used; ii). The object-oriented version of TCL is employed instead of TCL and iii). The interface code to the OTCL interpreter is separate from the main simulator.

5. Adapted Simulation Model

5.1 Overview

Our simulation model is an extension of NS2. It supports the two-way TCP protocol, and ATM connection can be easily established based on it. A traffic source can be generated according to any given video trace encoded in MPEG4 format. In this way, the simulation of video traffic transmitting over TCP/IP and ATM networks can be performed using our simulator.

The whole model consists of three layers, ATM layer, TCP layer and Video traffic layer from top to bottom. The design and implementation details are given as follows.

5.2 Network simulator for TCP

TCP is implemented in detail in NS2. Congestion control is realized by a slide window protocol, and error control by checking and retransmitting. For compatibility of our model with the applications by other people, no change is made to any classes in NS2, but some classes are derived from original classes as a part of our model. Some modifications are made as follows: 1. The current node address is recorded in the packet while the packet is forwarded, so that the previous hop address can be obtained at any time. This information is essential for ATM routing purposes; 2. functions are added to obtain the IP address of the particular node; 3. functions are added to get the bandwidth of an adjacent link.
5.3 Extension for ATM protocol

Although NS2 provides substantial functionalities in building up a range of network protocols, there is yet no implementation of the ATM protocol. Therefore, the implementation of ATM is the main concern in our work, which includes two related aspects: defining of the network topology and the packet forward mechanism.

Effectively defining the ATM network topology is taken into consideration so that network components including ATM link and ATM switch can be developed in terms of class ATM Link and ATM Switch. However, independency of ATM routing from TCP routing is impossible if these various components are implemented separately. Therefore, another class of ATM Network is employed to represent the complete ATM network. ATM Network is derived from the class of TCP Node so that the whole ATM network appears to the TCP traffic as just a single node. The ATM Network object can retrieve all objects from ATM Switches and ATM Links. Objects of ATM Nodes and ATM Link can be added arbitrarily to the ATM Network, and this last is implemented as a matrix of links. Moreover, external TCP node can be configured to attach to the ATM Network, and through it further link to one of the ATM switches.

As for packet forwarding, the first problem is the design of the routing algorithm, including the existing TCP routing implemented by NS2, which is outside of the ATM network, and the new development of the ATM routing method.

The TCP routing algorithm can remain unchanged outside of the ATM network since the ATM network appears as a single node, which is just pointed to. However, some information needs to be recorded to facilitate the routing inside of the ATM network. The routing logic is decided by the ATM network itself independently. Each packet enters the ATM network through an incoming TCP node, and exits it through an outgoing TCP node. The address of these two nodes can be obtained using TCP routing logic, and they are then recorded in the packet head to facilitate ATM routing.

With the information provided by the TCP protocol, routing inside of the ATM network can be easily realized. The ATM Network has access to all the links and ATM switches so that it can obtain the corresponding attached ATM switch according to the TCP node address. After that, routing is carried out inside of ATM switches according to ATM routing logic. In our system, the simple routing logic, mini-path is implemented. Though, other routing algorithms could be easily plugged into the model.

Another problem related to packet forwarding is packet segmentation and reassembling. On one hand, ATM Network objects are responsible for breaking packets down to cell level and sending them to the specific switch. For synchronizing cell transmission, sending time is calculated and the event is scheduled then. On the other hand, once it arrives at its last hop of the ATM switch and is ready to be reassembled into TCP traffic, each cell is directed to an object of AAL. Each object of ATM Switch has a set of AAL objects, which corresponds to each of the remaining switches. The AAL object checks if the cells arrive at the switch in sequence. Also, when the last cell of the packet arrives, the cell number of this cell and the total cell are compared to find out if part of the packet has been lost. In case any cell of one packet is lost, various cell discard algorithms can be adopted.

One of these discarding algorithms simply discards all the received cells in the packet. This simple discarding approach is very effective when used in a plain ATM network. However, since TCP protocol has strict error control mechanisms, any cell loss will be followed by the retransmission of the whole packet, which leads to bandwidth waste and further network congestion. In particular when it is video traffic which is transformed, unnecessary retransmission may occur. The experimental results concerning this will be presented in Section VI.

5.4 Video Traffic Generation and Performance Evaluation

A traffic generator in our experiment produced the adopted MPEG traffic. The generator utilizes an empirical real trace of video frames generated by the MPEG4 encoder, and then the needed traffic is generated adopting a model using the TES. In this way, the generated traffic closely matches the statistical characteristics of the real video trace [7]. Our generated video traffic, as a type of real video trace, includes three types of frames, which are I frames (intra-coded frames), P frames (predictive coded frames) and B frames (bidirectional or interpolated predictive coded frames). Each video frame is sent by the MPEG4 sender at equal intervals but in various sizes, because each frame is at a different compression rate.

There are two parameters used to control the traffic generation. The first parameter, initial seed, results in the variants of traffic trace, while the second parameter, transmission rate, determines the sending speed of video traffic [7]. In our experiment, two traffic traces are generated, with initial seed and rate 0.5 and 1, 0.1 and 2 respectively.

Statistics are gathered in the application level to evaluate traffic performance. Firstly, linkage between source and sink node in application levels are generated during topology establishment. Secondly, the sink actually has no detailed information of received packets apart from packet size. Instead, because information of sending packets is put into a queue at the source node, the queue at the source node can
be retrieved, and the details about packets, including sending time and frame number, can be collected. Therefore, the sending and receiving times of each frame are obtained, and the transmission delay of each frame can be easily calculated.

6. Experiment and Result

6.1 Experiment Setup

![Figure 1. Simulation network topology](image)

With the above-mentioned established model, a set of preliminary experiments has been carried out. The performance of the TCP/IP and ATM networks regarding MPEG video traffic is investigated with various parameter configurations. The relevant network parameters that have been studied for traffic performance of ATM and TCP/IP networks include bandwidth of the TCP link, buffer size of ATM switch and Maximal Segment Size (MSS) of the TCP protocol.

Figure 1 shows the network topology used for our simulation experiments. This topology is intended to represent a simple but typical ATM wide area network. Two TCP sources connect to the TCP sink through ATM switches and ATM links. Each source sends traffic frames and the sink sends only acknowledgements. A simple single-stage output buffered switch is tested in our experiment. In addition, for simplicity, cells passing through the ATM switch are only subject to variable queuing delay.

A frequently adopted measure of end-to-end protocol performance in existing research work is effective throughput because the main focus of these studies is the performance at network level. Nevertheless, when MPEG traffic is taken into consideration, this type of video traffic is very sensitive to cell delay. To measure the traffic performance of MPEG, frame delay is employed in our paper as a reflection of cell delay so that transmission performance is measured in the application level. In addition to frame delay, delay variation is also an important indicator for measuring picture jitter of real video traffic. In summary, both frame delay and delay variation are calculated in our experiment as performance metrics.

6.2 Experiment 1

The first experiment studied the behaviour of the ATM/TCP model for different configurations of the link bandwidth. The ATM bandwidth was fixed as mentioned and switch buffer size set to 100. At first, bandwidths of ingoing and outgoing TCP links were both set to 3 Mbps. Then, the bandwidth of the outgoing TCP links only was reduced to 2 Mbps.

The simulation results in terms of the average frame delay and delay variation of two sources are given in Table 1. It seemed that the latter bandwidth situations performed as well as the former. Nevertheless, according to our statistical figures, in the former situation where TCP link bandwidths were both 3 Mbps, 286 and 283 frames were transferred altogether by the two sources throughout the whole simulation period. In contrast, in the latter situation when the bandwidths were 3 Mbps and 2 Mbps, the two sources stopped sending packets after Frame 10 and Frame 34 respectively. This interruption may be due to the transmission failing to recover from congestion by an effective mechanism in the situation of dramatically mismatching of ingoing and outgoing TCP links.

6.3 Experiment 2

At this stage, the second situation studied in Experiment 1 was further examined. Based on the second situation, the buffer size of each ATM switch increased to 200 and 300 cells. As a result, the transmission of video frames kept on going throughout the whole period of simulation under these two settings, and 286 and 283 frames were transferred by the two sources respectively. As pointed out in Section VB, the performance declined significantly when bandwidth of ingoing and outgoing links did not match under the setting of a switch buffer size of 100. However, the experimental result of this section reveals that the network performance could be improved by increasing buffer size, although the link bandwidth was still unbalanced.

The impact of buffer size of the ATM switch on the traffic performance is given in the upper part of Table 2, which is the result of the above-mentioned experiment. Furthermore, from Row 2 and Row 3 of the table, it is seen that frame delay and frame delay variation remained the same when buffer sizes increased from 200 to 300. This result suggests that once bandwidth and buffer had been allocated properly so that the transmission continued smoothly, additional
buffers allocated to the switch do not help to reduce frame delay further.

The situation with TCP link bandwidths of 3 Mb/s and 2 Mb/s was also tested. Row 4 to Row 6 of Table 2 provides the experimental result. According to the figures, the same conclusion can be drawn under the condition of balanced bandwidth of ingoing and outgoing links.

6.4 Experiment 3

The last experiment studied the performance of ATM/TCP models for different MSS. Here, the buffer size of the ATM switch was set to 200 and bandwidths of TCP links were 3 Mb/s and 2 Mb/s. The MSS parameters of 9140 bytes and 512 bytes were respectively tested. As seen in Table 3, large and small MSS led to similar simulation results in terms of frame delay and its variation.

6.5 Result Analysis

The figures from our experiment show that, firstly, the mismatching in bandwidth of ingoing and outgoing TCP links will result in the interruption of traffic transmission, due to the transmission failing to recover from network congestion, especially in the case of VBR traffic. Secondly, on the basis of the first result, traffic performance can be improved to some extent by increasing buffer size, compensating for the problem resulting from unbalanced link bandwidth. Finally, performance under different configurations of MSS are compared in terms of frame delay and its variation. As a result, large and small MSS led to similar simulations. A contradiction exists between our result and the conclusion that smaller MSS results in low transmission efficiency given in [5]. This is likely because system performance is measured by frame delay and its variation in application level in our work, rather than by effective throughput in network level as in [5].

7. Conclusion

This paper presents a simulation model for MPEG traffic transmission over TCP/IP and ATM network. The components of the model are described in detail. The simulation tool, NS2, was utilized as a basis of our model, and was extended to further involve functionalities ranging from the ATM protocol, MPEG traffic generation to the traffic performance evaluation. A preliminary performance study using this model was conducted to assess the impact of network configuration regarding MPEG traffic performance. The experimental result is also briefly presented in the paper.

According to our preliminary results, inappropriate configuration may lead to a dramatic traffic performance decline in our simulated situation. On the other hand, extra resource allocation, such as redundant bandwidth, does not necessarily facilitate performance improvement. Thus, our model will help network engineers to simulate the network environment and identify network bottlenecks.

In addition, our model is able to be further extended and adapted. First of all, other parameters, such as TCP window size and timer parameters could be chosen and be optimized for TCP and ATM traffic optimized for particular traffic. Secondly, new connection admission schemes and new cell discarding algorithms for ATM switches are promising to take best advantage of network resources. Thirdly, different schemes for generating traffic and the various approaches to evaluation of traffic performance need to be further developed. Therefore, our simulation model will also be a valuable tool for evaluating potential solutions to various performance issues for ATM and TCP/IP.

References

[7]. A. Matrawy, I. Lambadaris, and C. Huang, MPEG traffic modeling using the transform expand sample methodology, Proc. of the 4th IEEE International Workshop on Network Appliances (IWNA4), Gaithersburg, MD, 2002.
[8]. M. Pearson and J. Cleary, Current techniques for measuring and modelling ATM traffic, Tech. report, Department of Computer Science, the University of Waikato, 1996
Table 1. Impact of Link Bandwidth on Traffic Performance

<table>
<thead>
<tr>
<th>Bandwidth of Ingoing TCP Links (Mbps)</th>
<th>Bandwidth of Outgoing TCP Links (Mbps)</th>
<th>Switch Buffer Size (Cell)</th>
<th>Source 1</th>
<th>Source 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Frame Delay (s)</td>
<td>Frame Delay Variation(s)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.1002</td>
<td>0.0300</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>400</td>
<td>0.1037</td>
<td>0.0302</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>100</td>
<td>0.1037</td>
<td>0.0302</td>
</tr>
</tbody>
</table>

Table 2. Impact of ATM Switch Buffer Size on Traffic Performance

<table>
<thead>
<tr>
<th>Bandwidth of Ingoing TCP Links (Mbps)</th>
<th>Bandwidth of Outgoing TCP Links (Mbps)</th>
<th>Switch Buffer Size (Cell)</th>
<th>Source 1</th>
<th>Source 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Frame Delay (s)</td>
<td>Frame Delay Variation(s)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.1037</td>
<td>0.0302</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>100</td>
<td>0.1043</td>
<td>0.0301</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>200</td>
<td>0.1043</td>
<td>0.0301</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>300</td>
<td>0.1002</td>
<td>0.0300</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>100</td>
<td>0.1002</td>
<td>0.0300</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>200</td>
<td>0.1002</td>
<td>0.0300</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>300</td>
<td>0.1002</td>
<td>0.0300</td>
</tr>
</tbody>
</table>

Table 3. Impact of MSS on Traffic Performance

<table>
<thead>
<tr>
<th>Bandwidth of Ingoing TCP Links (Mbps)</th>
<th>Bandwidth of Outgoing TCP Links (Mbps)</th>
<th>MSS (byte)</th>
<th>Source 1</th>
<th>Source 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Frame Delay (s)</td>
<td>Frame Delay Variation(s)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.1037</td>
<td>0.0302</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>9140</td>
<td>0.1056</td>
<td>0.0366</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>512</td>
<td>0.1056</td>
<td>0.0366</td>
</tr>
</tbody>
</table>