AN INTEGRATED SOFTWARE ENVIRONMENT FOR THE SIMULATION OF ATM NETWORKS

M.Ajmone Marsan, A.Bianco, C.Casetti, C.F.Chiasserini,
A.Francini, R.Lo Cigno, M.Mellia, M.Munafò
Dipartimento di Elettronica, Politecnico di Torino
Corso Duca degli Abruzzi, 24 – 10129 Torino – Italy
e-mail: lastname@polito.it

Keywords: TLC Networks, ATM, Simulation Tools.

Abstract

This paper describes the rationale and the implementation of a simulation environment for the analysis of ATM networks. The simulation environment is based on the integration of two software tools that operate at different time scales, and thus permit the computation of different types of performance metrics. The two chosen time scales are named “call-level” and “cell-level”. The call-level time scale is the one at which the relevant events mainly concern the user access to communication services, hence the establishment and release of connections. The cell-level time scale instead is the one at which the relevant events mainly concern the handling of ATM cells and of the PDUs that are transferred by means of them.

1 Introduction

The phenomena of interest for the analysis and design of ATM (Asynchronous Transfer Mode) networks occur at a number of time scales, that differ by many orders of magnitude. At the low end of the range we find the bit (or symbol) times on the highest data rate channels in the network; at the high end we find the times between equipment failures, or between topological changes induced by network planning activities. The former often are of the order of 1 ns, while the latter typically are of the order of months or years.

The analysis and design of ATM networks most frequently concentrates on two different time scales only, that we name “call-level” and “cell-level”, respectively. At the call-level time scale the relevant events mainly concern the user access to communication services, hence the establishment and release of connections. The cell-level time scale, instead, is mainly concerned with the handling of ATM cells and of the PDUs (Protocol Data Units) that are transferred by means of them.

The integrated simulation environment described in this paper comprises different simulation tools for the two time scales. The call-level simulator is named ANCLES (Atm Networks Call LEvel Simulator), whereas the cell-level simulator is named CLASS (Cell Level Atm Services Simulator). ANCLES allows the estimation of call blocking probabilities, as well as the assessment of the effectiveness of routing and CAC (Connection Admission Control) algorithms. CLASS provides estimates of performance parameters like PDU and cell loss probabilities, PDU and cell delay distributions, as well as the assessment of the effectiveness of traffic management techniques such as shaping, policing, and the algorithms embedded within the ABR (Available Bit Rate) ATM transfer capability.

The rationale behind the design of the integrated simulation environment is briefly explained in the following. First of all, consider that the simulation of cell-level dynamics is much more CPU-intensive than the simulation of the call-level dynamics, for a fixed period of evolution of the ATM network. Moreover, the simultaneous observation of the call and cell-level dynamics (besides being generally not feasible) may be quite wasteful, since not all the call-level configurations produce interesting results at the cell level. In particular, periods in which the load imposed by users on network resources is low, surely result in good
performance, so that running endless simulations only to discover that low network loads yield negligible cell loss probabilities is hardly wise. Finally, averaging the cell-level performance over periods of high and low network load would hide the worse performance obtained by users during high-load periods, on which the network design should instead concentrate.

A more sensible approach might consist in a fast exploration of the network dynamics at the call level as long as the individual loads of resources remain “low”, proceeding with a detailed investigation of the cell-level behavior only when the configuration produced by the call-level dynamics induces a “high” load of some network resources (of course, the definition of what are low and high loads is quite critical; we shall return to this point later on). Thus, in our integrated simulation environment, ANCLES provides a fast exploration of the call-level dynamics, generating connection configurations, and the resulting resource load patterns; when a critical configuration is found, the analysis is continued with CLASS at the cell-level, generating the estimates of the performance parameters for that particular connection configuration. After these are obtained with sufficient precision, the simulation control is returned to ANCLES, and the exploration of the network dynamics continues at the call level until a new critical connection configuration is encountered. A pictorial representation of this mode of operation is shown in Fig. 1.

When the control of the simulation experiment is transferred from ANCLES to CLASS, the connection configuration is frozen, so that the computation of performance estimates through CLASS is a steady-state analysis, that does not consider the interactions between the call and cell-level dynamics. By so doing an approximation is certainly introduced, but this decoupling of the dynamics at the two time scales is exactly what is necessary to obtain a drastic reduction of the CPU time requirements. As a matter of fact, a partial coupling of the two dynamics may be accounted for in the definition of what a critical connection pattern is: indeed, besides defining a threshold for the average load of the network resources, it is also possible to impose a threshold on the holding time of the critical configuration, so as to avoid a steady-state analysis (with the transfer of millions of cells) for connection configurations that only last for few tens or hundreds cell times.

At present, the version of the integrated simulation environment is not quite as automated as described above. This is not due to any technical problem or difficulty. Rather, our present opinion is that it is advisable to let the network designer enter the loop and decide whether the (costly) detailed simulation with CLASS of the configurations produced by ANCLES is actually worthwhile. Thus, a typical simulation experiment starts with the execution of ANCLES, that produces a number of critical configurations, generating for each one of those a CLASS input file. The network designer then decides which simulation runs to activate.

Moreover, it is important to emphasize that each critical configuration generated by ANCLES may correspond to several CLASS simulations, as shown in Fig. 1. This is due to several reasons: first, a call-level traffic description provided by ANCLES can be
mapped onto several different cell-level traffic models. Second, several network parameters are meaningful only at the cell level and do not influence the call-level performance; thus they can be taken as parameters to compare different architectural choices at the cell level. Finally, several different ATM transfer capabilities can be chosen as an alternative to provide the specified QoS (Quality of Service) to the users. It is a network designer choice to determine which set of cell-level simulation runs are most interesting.

A rather sophisticated statistical analysis is performed on the data collected by the two simulators. Simulation experiments are automatically stopped when the desired user-specified accuracy is reached on a selected set of performance parameters. The software automatically determines the duration of the initial simulation transient period, collects the measured data, computes aggregate performance figures, and provides an estimate of the accuracy reached by the results, using the “batch means” technique.

2 The Call Level Simulator

ANCLES is an event-driven asynchronous simulator, devoted to the simulation of the call-level dynamics in ATM networks. ANCLEs is aimed at the comparison and assessment of different routing algorithms and connection admission control (CAC) techniques. It considers as relevant performance metrics the throughput of the whole network, the load carried by each link, and the call blocking probability, computed both for the entire network and user-by-user.

The network models simulated by ANCLEs are mainly composed of instances of three basic entities:

- **ATM switches**, that perform the routing functions and implement the CAC algorithms;
- **channels**, that accommodate the information transfer between either adjacent nodes or user-node pairs;
- **users**, that act as sources and sinks for the traffic flowing through the network.

Users drive the simulation in ANCLEs; they collect connection requests from their associated call generators and forward them to the network, while acting as destinations for the calls coming from remote users.

At the beginning of a simulation session, ANCLEs acquires the simulation experiment description including: i) the network topology in terms of a weighted graph connecting nodes, channels, and users, the weights representing the channel data rate; ii) the traffic relations among network users, and the statistical characterization of the sources; iii) the parameter that fully specify each instance of the different entities; iv) the selection of the CAC technique and routing algorithm that must be adopted during the simulation run; v) a number of options concerning both the network operations and the simulation session management; vi) the performance indices to be measured and statistically analysed; vii) the thresholds to be used for the identification of critical network configurations (to be possibly later simulated with CLASS), as well as the maximum number of configurations to be stored.

Each time a user generates a connection request, the routing function is executed in order to identify a convenient end-to-end path; for each path being considered, this requires the execution of the CAC function on each link belonging to the path, in order to check whether sufficient resources are available; if enough resources are available, they are allocated to the call, and the call is accepted. If no end-to-end path is found to route the incoming call in agreement with its QoS requirements, the call is dropped. Obviously, when an active call terminates, the corresponding resources are released.

As a result of the simulation run, ANCLEs computes and records on specific files the data gathered from the observation of the network dynamics. The information that is normally generated by a simulation run includes the call blocking probabilities at both the network and the user levels, the average network throughput, the average channel loads, the results of the statistical analysis in terms of accuracy and confidence interval, and the CPU time consumed by the simulation experiment.

In addition, ANCLEs can record one or more critical network configurations, that may be later simulated with CLASS. Since the maximum number of recorded configurations is fixed at the beginning of the simulation run, ANCLEs progressively substitutes the old configurations with the new ones, if the latter are more critical than the former. Thus, the resulting collection comprises the most critical configurations observed during the entire experiment.

In the following subsections we concisely overview the types of users available in the present version of ANCLEs, as well as the routing algorithms and CAC techniques that are presently supported by the tool. It must be emphasized, however, that the modular software structure is designed so as to quite easily allow the expansion of the library of available algorithms.
2.1 Users

During the simulation experiment, users generate connection requests and releases. Each connection request is associated with the identifier of the destination user (chosen according to the traffic relations specified for the experiment), as well as a statistical description of the bandwidth requirements.

Both the connection request interarrival times and the call durations are modeled as random variables with appropriate distributions (negative exponentials are normally used; but other types of distributions can be trivially inserted).

The types of users presently available in ANCLes differ in their statistical characteristics:

- **CBR** users generate Constant Bit Rate (CBR) connection requests. The cell and bit rates associated with these connections are constant throughout their whole lifetime.

- **ON-OFF VBR** users generate ON-OFF Variable Bit Rate (VBR) connection requests, modelling the behavior of sources through a two-state (OFF and ON) Markov model. The source using the connection is either idle, during the OFF period, or transmitting a constant rate cell stream, during the ON period.

- **Uniform VBR** users generate VBR connection requests assuming that the cell rate of the corresponding source varies with time according to a uniform distribution between two fixed values.

- **Video VBR** users generate VBR connection requests assuming that the cell rate of the corresponding source can only take three distinct values, according to a suitable probability distribution.

- **UBR** users generate Unspecified Bit Rate (UBR) connection requests; the only available traffic descriptor is the connection Peak Cell Rate (PCR), that remains constant throughout the connection lifetime, and is determined by a uniform distribution between two fixed values.

**Equivalent bandwidth** Each connection request is assigned an equivalent bandwidth, whose value depends on its peak and mean cell rate (PCR and SCR), on the link capacity (C) and on the negotiated cell loss rate (CLR). The equivalent bandwidth associated with a call can thus change from link to link, if links have different capacities.

The CAC algorithm is quite simple: each link is associated with an equivalent bandwidth accumulator, whose content is added to the equivalent bandwidth of the incoming call. If the resulting sum is greater than the link capacity, the call request is rejected, otherwise it is accepted on that link.

**Peak and measured bandwidth** This CAC algorithm uses only the PCR of the new connection, but exploits traffic measurements to determine the available bandwidth on each channel.

The new connection request is considered acceptable on the link if the sum of the PCR of the new connection and of the connections already supported by the link does not exceed the link capacity. If the previous step was not successful, the measured traffic is examined; the new connection is accepted if the following condition is verified:

\[ \mu_t + k^* \sigma_t + \text{PCR}_{\text{new}} \leq \rho C \]

where \( \mu_t \) and \( \sigma_t \) are, respectively, the average and the standard deviation of the load observed on the link during the measurement period, \( \rho \) is a protection factor applied to the link capacity (\( 0 < \rho \leq 1 \)) in order to make the CAC algorithm safer, and \( k^* \) is the weight applied to the standard deviation of the measured traffic.

**QoS-aware Peak bandwidth** The last CAC algorithm presently included in ANCLes aims at admitting connections with different QoS requirements using a very limited amount of information. Specifically, three QoS classes are considered:

- **Class 1**: with stringent CLR (Cell Loss Rate) and CDV (Cell Delay Variation) requirements;
- **Class 2**: with stringent CLR requirements, but no need for CDV guarantees;
- **Class 3**: with no need for guarantees on either CLR or CDV.

Both CBR and VBR connections can request admission as either class 1 or class 2, depending on their...
QoS requirements. UBR connections instead normally request admission as class U.

The CAC itself is quite easily described:

- Class 1 and 2 connections are accepted if \( \sum_{c} PCR + \sum_{c} MfCR \leq \alpha C \) where C is the link capacity and \( \alpha \) is a protection coefficient \( (\alpha \leq 1) \). MfCR stands for Modified Cell Rate, and is an expression characterizing VBR sources through a cell rate value between PCR and SCR, according to a linear modification factor \( \gamma \); i.e.: \( MfCR = SCR + \gamma (PCR - SCR) \). Obviously, for CBR sources \( MfCR = PCR \).

- Class U calls are accepted if: \( \sum_{cU} PCR \leq \beta \left( C - \sum_{CBR,c} PCR - \sum_{VBR,c} 1/2 SCR \right) \)

  where \( \beta \) is a bandwidth utilization coefficient, that can be greater than 1.

### 2.3 Routing Algorithms

The routing algorithms currently implemented in ANCLES result from adaptations of those illustrated in [1] and differ for the criteria followed in the selection of the paths to be tested with the CAC function. Whatever criterion is used to order the paths, a primary path is always identified for each pair of users: this primary path is always tested first by the CAC function. All other viable paths are referred to as secondary paths.

**Single path routing** Only the primary path is considered to be an admissible end-to-end path for the incoming call. If the CAC function returns a negative answer, even for just one link in the path, the connection is not accepted.

With this algorithm, the resources needed to support each call are minimized, but it is not possible to adapt the routing strategy to the actual distribution of the traffic inside the network.

**Controlled alternate routing** With this algorithm, if the call cannot be accommodated along the primary path, the secondary paths are checked, according to their order based on the number of hops. A control mechanism is introduced when considering a secondary path: the links that the secondary path does not share with the primary are assumed to provide a smaller bandwidth than their actual capacity; the bandwidth reduction sets aside an amount that depends on the average load that these links should support when included in primary paths.

This approach encourages the selection of the primary path, allowing the use of a secondary path only when the load currently carried on the links forming the secondary path is quite low.

**Uncontrolled alternate routing** In this case, no capacity reduction affects the links included in the secondary paths. The network is thus more vulnerable to the waste of resources that could stem from the spreading of connections over secondary paths.

### 3 The Cell Level Simulator

CLASS is a software tool for the simulation of ATM networks at the level of abstraction adequate for the observation of the dynamics relating to cells and cell bursts. Hence, CLASS permits the computation of estimates of several performance metrics referring to either individual cells or groups of cells; typically, a group of cells may correspond to either a peak of traffic, or one PDU of a higher-layer protocol. For a rather detailed description of a previous version of CLASS, the reader is referred to [2]: in this section we provide only a synthetic description of the tool, with particular emphasis on the new features now available.

CLASS is a synchronous time-driven simulator, where the time unit is the minimum cell transmission time in the network (i.e., the cell transmission time on the channel with the highest data rate), and where the simulation time is progressively increased by one time unit, executing all the operations resulting from the events scheduled during the time unit.

The performance metrics most frequently estimated with CLASS are the loss probabilities referring to cells and higher layer PDUs of a given VC (Virtual Channel), the average load of channels, the end-to-end throughput of cells and PDUs on a given VC, the cell and PDU delays (including averages, variances, percentiles, and distributions, of access delays as well as end-to-end delays) along a VC.

The network models simulated by CLASS are composed of instances of the same three basic entities used in ANCLES, that now however implement different functionalities, according to the different time scale:

- **ATM switches**, that perform the cell switching functions and implement the traffic management algorithms relative to nodes;

- **channels**, that transfer cells between either adjacent nodes or user-node pairs;
• users, that act as sources and sinks for cell flows, and implement the traffic management algorithms relative to users.

Since the main motivation for the development of CLASS originally was the investigation of the traffic management techniques for the provision of connectionless services in ATM networks (the acronym CLASS originally stood for ConnectionLess Atm Services Simulator), a wide number of models of traffic sources and traffic management algorithms are available; instead, the description of switches and channels is rather abstract and simplistic.

Channels are described as point-to-point links with constant propagation delay and zero error probability. Furthermore, the propagation delay is forced to be a multiple of the simulation time unit.

Switches are assumed to be non blocking with output queuing, and fixed processing delay. This means that cells that arrive at an input interface are brought to the output interface with no loss and constant delay. Each output interface comprises one transmitter, and several buffers, for traffic flows at different priorities (buffers are served with a strict priority scheme). Losses may occur due to the finiteness of the output buffers.

A wide variety of traffic sources can be used to define the ATM network workload model. They range from simple, abstract sources, that can be quite useful for the model debugging phase, and for the comparison with analytical models, to very detailed representations of the transport protocols and of the applications above them.

The abstract source models include:

• CBR cell flows with given PCR,
• ON-OFF VBR cell flows with given PCR and SCR,
• Bernoulli cell generation processes,
• Poisson PDU sources; PDUs can have assigned distribution of the number of cells resulting after segmentation,
• Mobile Poisson PDU sources.

The detailed source models essentially include the adaptation of the actual code of two transport protocols: TCP and XTP. Both protocols can be forced to a greedy behavior, corresponding for example to a large file transfer. In addition, TCP can also behave like in a client-server scenario, mimicking the operations of a series of WWW document transfers. The latter option, combined with Pareto distributions for both the file size and the interval between two page transfer provides a source model that exhibits self-similarity [3].

The TCP source model available in CLASS is a direct derivation of the original BSD 4.3-reno C code; the actual source code was downloaded, and modified so as to let it efficiently run above the simulation kernel. Since the BSD 4.3-reno release is not tailored to high-speed networks, slight modifications were necessary to adapt it to the ATM environment.

The traffic management algorithms modeled in CLASS include traffic shaping and policing according to the GCRA (Generic Cell Rate Algorithm) definition [4], the WFQ (Weighted Fair Queuing) scheduling algorithm [5, 6] within switches, as well as a very detailed implementation of the ABR transfer capability [4]. In particular, the implementation of ABR permits the Relative Rate Marking (RRM) as well as the Explicit Rate Marking (ERM) operation; in both cases a number of different switch algorithms are available.

3.1 The Wireless Access Loop

The approaches for the provision of services to mobile users through ATM networks are far from being standardized; the definition of a wireless access model in CLASS was thus based on the research work currently performed within several international projects. The reference model adopted in CLASS for wireless ATM (W-ATM) assumes the full integration of mobile terminals within B-ISDN (Broadband Integrated Services Digital Network); thus the radio interface is packetized and optimized for the size of ATM cells. The protocol stack within the mobile terminal comprises a standard ATM access, up to the AAL (Atm Adaptation Layer), while base stations implement the protocol stack only up to the ATM layer.

Consistently with the aims of CLASS, the model of the mobility services is developed at the cell level, disregarding important aspects of the radio interface. The radio channel is assumed to be fully reliable, although with variable delay, accounting, for instance, for the use of ARQ (Automatic Retransmission reQuest) algorithms.

The main aspect of mobile services that can be observed through CLASS concerns the management of network handovers, i.e., of procedures for the rerouting of the ATM connection due to the mobile terminal movement from one cell to another. Several approaches were proposed in the literature for this task. The approach adopted in CLASS is a make-break incremental re-establishment of the path: the VC route
is modified only in the minimal portion close to the mobile terminal, while the old portion of the connection is closed only after the new portion has become available, in order to ensure loss-free handovers.

The handover protocol model implemented in CLASS is based on in-band signalling and grants the integrity of the information flow (no cell misordering), being thus suitable also for data transmission. The details of this protocol were described in [7, 8]. Different protocols assuming an incremental re-establishment of the connection can be easily introduced in CLASS.

Mobile terminals in CLASS roam through a predefined set of cells, each one covered by a radio base station, following a mobility pattern defined by the user in probabilistic terms. When a handover procedure cannot be completed because of the lack of resources, the corresponding connection is not dropped; rather, the failure is recorded, and the handover is re-scheduled after a random delay.

The call blocking probability metrics for a W-ATM scenario can instead be obtained with ANCLEs. Models of mobile terminals are currently being implemented in ANCLEs; they share the same description of the mobility pattern, as well as the same macroscopic characteristics of the mobile user models in CLASS.

4 Conclusions

The rationale, the capabilities, and the implementation of a simulation environment for the analysis of realistic ATM networks were described. The simulation environment is based on the integration of two software tools, named ANCLEs and CLASS, that operate at the call-level and at the cell-level time scales, respectively.

The separate investigation of network dynamics at different time scales allows a drastic reduction in CPU time requirements, hence the performance analysis of realistic network configurations, as well as the possibility of choosing the observation intervals most suited to the visualization of the phenomena of interest, whose impact would vanish if averaged over too long intervals.

The development of an integrated set of simulation tools, from which several different dynamic behaviors at different time scales can be generated, permits a unique and consistent description of the ATM network, and guarantees the consistence of the results generated at the different time scales.

A further step on the decomposition of the analysis at different time scales can be fruitfully applied to the development of an ATM network planning tool, where the addition of a planning level can account for structural modifications of the network.

CLASS and ANCLEs are public domain software; information about the tools, as well as instructions to obtain the source codes can be found on the Web at: http://hp0tlc.polito.it/class and http://hp0tlc.polito.it/ancles.

References


