A publicar nas actas do SPECTS'2001 – International Symposium on Performance Evaluation of Computer and Telecommunication Systems, Orlando, Florida, USA, 15-19 Julho, 2001

SIANET Route: PNNI Routing Module of an IPv6-over-ATM Network Simulator

Sérgio Duarte[1], Jorge Sá Silva[2], Nuno Veiga[3] and Fernando Boavida[2]

Affiliation and Address:

Centro de Informática e Sistemas da Universidade de Coimbra

Departamento de Engenharia Informática

Polo II - Pinhal de Marrocos

P-3030 Coimbra

PORTUGAL

Tel.: +351 39 790000 Fax.: +351 39 701266 Email: sduarte@ipg.pt

ABSTRACT

This paper presents the the main features of the routing module of a discrete event-driven simulator called SIANET, which can be used for the design and analysis of Internet Protocol version 6 (IPv6) over Asynchronous Transfer Mode environments for private networks. The simulator, which was constructed in the scope of the MEDIA project (MARS Extensions Developed for IPv6 over ATM) encompasses the Private Network to Network Interface (PNNI) specification developed by the ATM Forum. After a brief description of the simulator, the routing module and the signalling protocol are discussed. The paper finishes with sample simulation results obtained with the presented simulator.

Keywords: ATM, IPv6, PNNI, Routing, MARS.

[1] Guarda Polytechnic Institute
[2] Computer Department – Coimbra University
[3] Leiria Polytechnic Institute

1. INTRODUCTION

Current and future applications as voice, data and video make use of different upper-layer technologies. Consequently network technologies need to support them. Asynchronous Transfer Mode [1] (ATM) has generally been regarded as the ultimate network technology, which can integrate voice, data and video services, for private and publics LANs, MANs and WANs. With the tremendous growth of the Internet and the reluctant deployment of public ATM Networks, the future role of ATM seems to be less clear than what it used to be. The growth of multimedia-based applications on the Internet is immense, especially as current applications offer not only typical data services but also real time voice and video services. This is causing the Internet to enter what was hitherto the typical target market of ATM. Furthermore the Internet Society is quite drastically changing their policy of shared resources and free usage, and actively investigating how to introduce resource reservation and charging support in the Internet to provide better support for multimedia applications and service providers.

The dominance and evolution of IP based networks in LAN and WAN areas, especially with the IP version 6 [2](IPv6) proposal, has also led to proposals for ATM deployment that considerably differ from the traditional view of public telecom operators, such as using ATM only as a high speed transmission system.

IPv6, as the new Internet Protocol, was developed to solve some of the shortcomings of IPv4. In fact, the major IPv6 improvements are a larger address space, to fight the address space exhaustion, better routing - in particular with source routing capabilities and multicast routing by using address scopes - auto configuration features, multicasting capabilities, QoS support and security capabilities.

While discussion goes on about which technology - IP or ATM - will be best for an integrated services network, some people, including the authors, address their integration, studying ways to develop efficient IPv6 over ATM [3, 4, 5] networks.

ATM networks are connection oriented. This fact implies the need for ATM specific signalling protocols and addressing structures, as well as protocols to route ATM connection requests across the ATM network.

This work presents the conception and the development of the routing part of a discrete event-driven simulator, called SIANET (the abbreviation for Simulator of Ipv6 over Atm NETworks), that can be used for the design and analysis of IPv6-over-ATM environments for private networks. Figure 1 presents a view of SIANET's graphical user interface. SIANET was developed in the scope of the MEDIA project, which proposes and explores a number of extensions to existing IP-over-ATM multicasting solutions, that make use of the advantages offered by IPv6 and by the User Network Interface version 4.0 specifications [6].

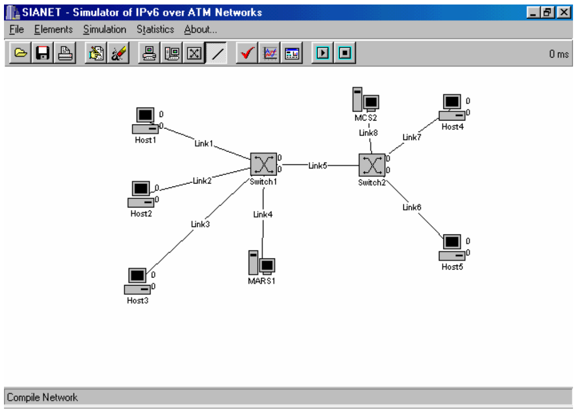


Fig. 1 – SIANET Graphical User Interface

This study gives special attention to the routing module of the simulator, which implements the Private Network to Network Interface (PNNI) Specification [7] Version 1.0 recommended by the ATM Forum. Since ATM networks are connection oriented, a route for a call between two end systems is derived through a routing protocol. After choosing a route, the network invokes its signalling protocol to establish the call on the route. Once the call is established, ATM cells corresponding to the call are switched from de input port to the required output port at the switches along the route.

The source code of the simulator was written in the C++ programming language, with the Borland C++ Builder [8] tool, on a Microsoft Windows platform. The goal of this paper is to give an overview of SIANET and more precisely show the routing and signalling protocol implementations. The second section of this paper provides a general description of the simulator. The third section describes the routing protocol implementation. Section 4 describes the signalling protocol implementation. Section 5 provides an example of the use of SIANET.

2. THE SIANET SIMULATOR

This simulator was specifically developed for the design and analysis of IPv6 over ATM. SIANET has a modular architecture (fig.2), in order to allow easy addition of functionality and to increase the potential applicability of this simulation tool. This tool allows the analysis of network traffic derived from different Internet applications (such as ftp, telnet, HTTP, audio and video). These applications use the IPv6 protocol that, in turn, uses ATM. The coexistence of IPv6 with ATM is achieved by using the Classical IP over ATM paradigm [9](CLIP). In addition to the implementation of the base modules, we are currently developing new modules oriented to the study of multicast layer 3 over ATM systems, specifically with the implementation of the Multicast Address Resolution Server [10](MARS) client and server modules, the implementation of Multicast Servers (MCS), and the new IPv6 over ATM sub layer.

As we said, the simulator has a number of modules: a network module, a graphical user interface (GUI), an event controller, an engine, and a routing module. These modules interact with each other as shown in figure 2.

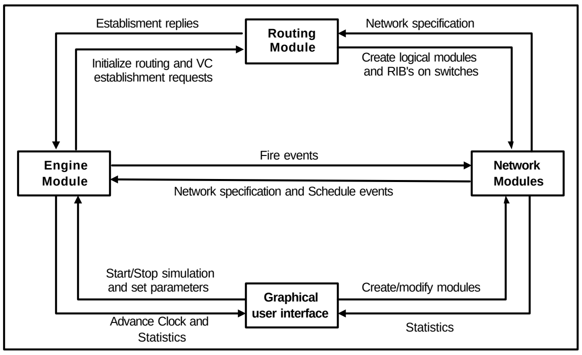


Fig. 2 – Simulator architecture

The network module contains the network components, and is divided into four components: ATM switches, hosts (ATM end systems), ATM links and Internet applications.

These applications generate traffic according to their types (CBR, VBR, ABR, UBR), and they run in ATM end systems (hosts). ATM switches implements the cell switching between links. A link module is a representation of an ATM link, and can be physical or logical. Finally, a host is an ATM end system representation that is responsible for originating call requests, according to the needs of applications, and for the transmission and reception of ATM cells. The network module schedules events to be processed by the engine module, provides the network specification to the engine and routing modules, and generates statistics.

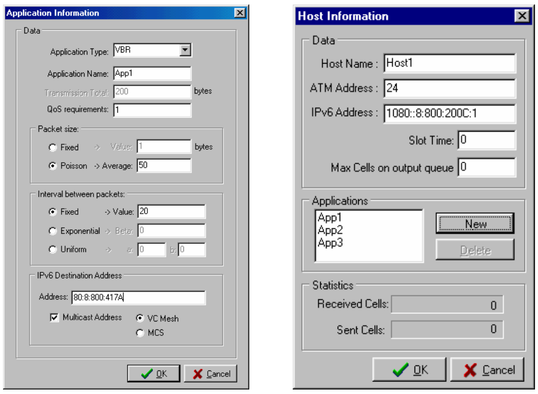


Fig. 3 – SIANET parameter specification examples

The Graphical User Interface is a user-friendly tool that simplifies the creation of the network topology, the creation of network modules and the specification of their parameters (fig. 3). The GUI provides buttons to start and stop the simulation, display the clock simulation time, and display statistical results. At present, the simulator provides statistics concerning the parameters identified in Table 1.

Table 1 – SIANET obtainable results

Network Module	Parameter
ATM Switch	Number of sent cells
	Number of received cells
	Number of dropped cells
	Mean queue lengths
Hosts	Number of sent cells
	Number of received cells
	Mean queue lengths
ATM link	Available Bandwidth (%)
	Mean queue lengths
Applications	Connection time
	Transfer time
	Call establishment time

Results are displayed in a graphical form (fig.4) during the simulation run. This can be very useful, because it enables the user to instantly monitor de output statistics during the simulation. At the end of the simulation, the simulator generates a report with all the measured values concerning all network modules in numerical form.

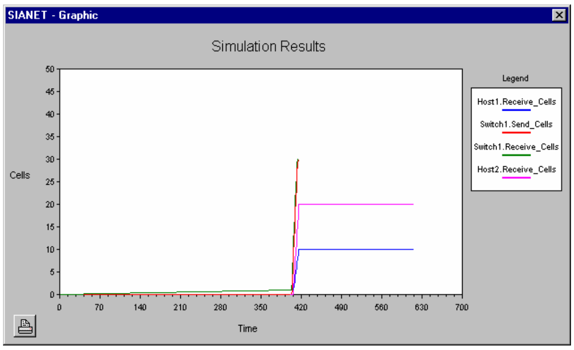


Fig.4 - Graphical output example

The engine module is the core of the simulator. It is responsible for the simulation control, performs the data conversion between layer stacks and converts packets into cells. The engine fires the routing module initialisation, fires events to the network module, receives events from other modules and provides statistics to be displayed by the GUI. The engine simulation control includes an event controller that is one of the most important sub-modules. The network is treated as a discrete-event system, where the events occur in specific points in time, and nothing of interest for the system occurs between these points. The state of the network is determined by the state of its components. For example, the state of a link can be represented by its available bandwidth. An event is an activity that can change the network state, and is the basic unit of the discrete-event simulation. The function of the engine event controller is to provoke the execution of events, maintain and update the events list, and maintain the general clock of the simulator. The events list contains the set of events to be executed in a chronological order.

The routing module provides an implementation of the Private Network to Network Interface (PNNI) specification version 1.0, and is composed of two components:

- A PNNI virtual circuit routing protocol, which is used to route the signalling requests through the ATM network.
- A PNNI signalling protocol, used to relay ATM connection requests within the network, between de source and destination UNI.

The routing module creates the logical links and logical switches. It is responsible for the routing hierarchy construction, and for the switches

Routing Information Base (RIB) initialisation. This module is specifically described in the next section of this paper.

The simulator was validated by using results obtained in two independent testbeds described in [11, 12].

3. THE ROUTING MODULE

The routing module implements the Private Network to Network Interface (PNNI) specification from the ATM Forum.

The routing protocol is divided into two modules: RIB (Routing Information Base) and path calculation algorithm. The RIB contains information concerning the connectivity and state of the links and switches, and their reachability in the network. The path calculation algorithm calculates routes based on the information contained in the RIB. The routing module supports hierarchical routing. With this type of protocol, switches maintain detailed information concerning neighbouring links (in the same peer group, as we will explain in detail in this document) and summarized information about switches and links in other peer groups.

In order to keep the complexity of the PNNI simulator module as low as possible, the following assumptions were made:

- Traffic characteristics: we consider data traffic, but we don't consider network errors.
- Network hierarchy:
  - ✓ The simulator only supports two levels of routing hierarchy: the lowest level and one higher level;
  - ✓ A simplified address format version is used, instead of the format specified in PNNI 1.0 specification, composed of a number with the following format: pgxxx, where pg is the peer group identifier, and xxx is the switch identifier. In the higher level the switch identifier is 000 (zero).
  - ✓ The construction of the routing hierarchy is carried out before the simulation starts;
  - ✓ The peer group leader election is made before the simulation starts;
- Only two types of packets in the PNNI module are considered:

- ✓ PTSP packets – The packet that switches use to advertise their local information to all other switches in the network;
- ✓ Signaling packets – These packets are triggered by applications that request a call, and are used to establish a call between two ATM end systems.

- The Routing Information Bases of ATM switches are initialised with the network state topology before the simulation starts.
- The Call Admission Control (CAC) algorithm is simplified and just verifies if the outgoing link has enough bandwidth to support the call without violating the QoS guarantee provided to existing calls.
- The Generic CAC (G-CAC) algorithm is a standard algorithm that any switch can use to estimate the reserved bandwidth by a CAC algorithm in another switch, and is not implemented in this application.
- We consider that all network modules are always active during simulation runs.

3.1. A HIERARCHICAL ROUTING PROTOCOL

In the hierarchical routing protocol, the network is divided into logical networks called peer groups (PG). PG's are composed of a set of logical nodes, connected by logical links. In the lowest level each logical node represents one physical switch, and links also represent physical links. A peer group is represented in the following level of the hierarchy by one logical switch, elected as Peer Group Leader (PGL). The PGL is responsible for the sending of PNNI Topology State Packets (PTSP) to switches of the higher hierarchic level. These relations can be seen in the figure 5.

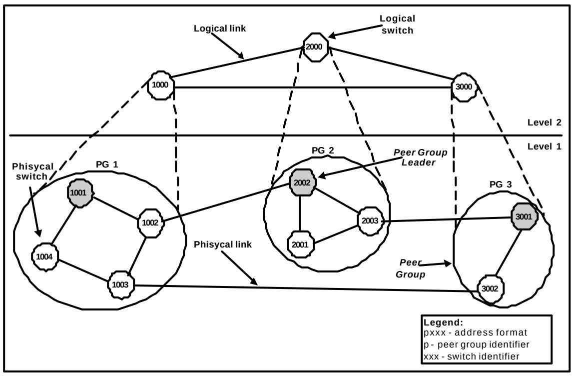


Fig. 5 – SIANET two-level routing hierarchy

Figure 5 shows an example of a two-level routing hierarchy in SIANET, with three Peer Groups (PG) in the first level (1, 2 and 3). For a node in PG 1, all nodes in PG 2 appear as a single logical node identified by 2000, the identification of the logical node representation of the PG in the higher level. Similarly, the entire PG 3 appears as a logical node identified by 3000. In level 2 we have logical switches and logical links that are logical representations of the switches and links in the lower level.

In the case of the SIANET simulator, the network hierarchy is defined before the start of simulations, and as we consider that switches and hosts are always active the structure of the hierarchy doesn't change. For the same reason, the PGL election for all peer groups is made only once, prior to the beginning of simulations.

3.2. ROUTING INFORMATION BASE

In SIANET, the RIB contains information regarding the connectivity and state of the links and switches in the network. The state of links and switches is specified by metrics and attributes. A metric is quantitative and can be used to define the cost of a link during route calculation, while an attribute is qualitative and it can be used to determine whether or not one link or switch can be used for route calculations. In SIANET we use the total bandwidth, the available bandwidth and the administrative weight, a cost attributed to each link, as metrics for route calculations.

The RIB of each switch is constructed before the beginning of the simulations, with the metrics and attributes described above. RIB's are updated through PTSP's (PNNI-Topology State Packets) flooding. PTSP are packets that travel through the network with information concerning the state of links and switches. PTSP's are sent on a periodic basis or when one switch detects significant changes in its local information (for instance, a change in the available bandwidth).

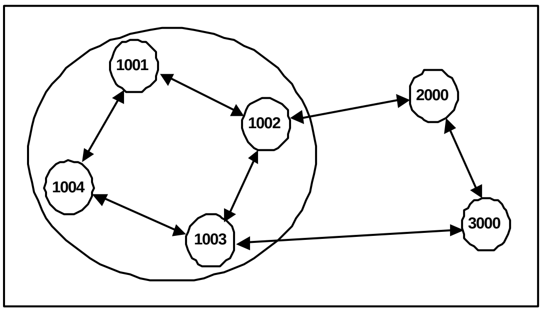


Fig. 6 – Network connectivity as viewed by node 1001

At each switch, the RIB holds detailed information about the switches and links in its own PG and maintains summarized information about the switches and links that belong to different PGs. In figure 6 we can see the RIB information as viewed by node 1001, for the sample network of figure 5. As we can see, node 1001 has detailed information in its RIB about network connectivity in its own peer group, but switches that belong to other peer groups appear just like one switch. For node 1001 the logical nodes 2000 and 3000 represent all the switches that belong to the peer groups 2 and 3, respectively.

3.3. ROUTING COMPUTATION ALGORITM

As in the PNNI routing protocol, the simulator uses source routing to calculate the route between source and destination. After receiving a call establishment request from one host, the source switch calculates the path to the destination switch. When the switch calculates the path it does not use links that do not have enough bandwidth to support the call. To know if one link has capacity to support the call, the switch uses the CAC (Call Admission Control) algorithm, which just checks if the outgoing link has enough bandwidth to support the connection and if the QoS parameters are not violated. As we just consider the cell transfer delay as QoS parameter, the delay of the outgoing link is verified. To calculate the path, the source switch uses the Dijkstra's shortest path calculation algorithm [13]. For the link cost utilization, the simulator uses the following equation:

$$\text{Link cost} = A + \frac{C}{(1-U_{link})}$$

Where A is the administrative weight value attributed to each link, C is a constant, and the U_{link} is the bandwidth utilization (in relation to the maximum bandwidth) of the link.

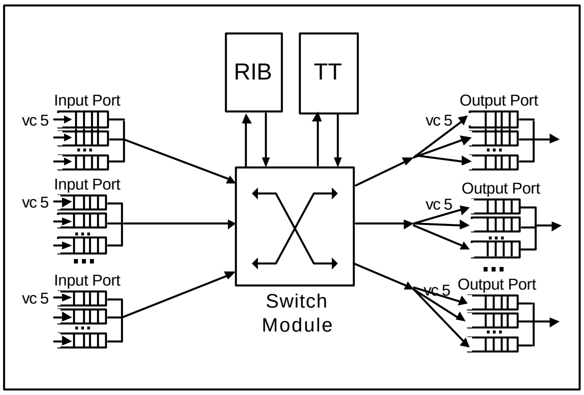
4. THE SIGNALLING PROTOCOL

The signalling protocol is used to establish a connection between two ATM end systems, on the route chosen by the source switch. In SIANET, the signalling protocol is implemented in the routing module.

Table 2 – Signalling packet fields

Name	Description
callNumber	Call sequence number
packetType	type of packet (SETUP, REJECT, ACCEPT, RELEASE...)
direction	packet direction (up - source to destination, or down – destination to source)
end_addr	host ATM end address
atm_tos	ATM type of service (CBR, VBR-rt, VBR-nrt, ABR, UBR)
traffic	traffic descriptors (uses only max cell rate)
QoS	QoS requirements (uses only cell transfer delay)
Route	Designated transit list (DTL) stack
NRA	Number of rerouting attempts
Block_links	Blocking link list

When a host makes a call request, it submits a signalling packet to the source switch. The SIANET signalling packet has the fields shown in table 2. The call sequence number is used to prevent that one switch processes a signalling packet twice. The packet type field is initially set to SETUP and changed to other states, as we will see later. The direction field is used to know if the packet is on the source/destination call direction or the reverse. The end_addr field is a simplified ATM host address with the format pgxxxhhh, where pg is the peer group identifier, xxx is the switch identifier to which the host is attached, and hhh is the host identifier. Atm_tos is the field that defines the call type of service request. The traffic descriptor field defines the call maximum cell rate request. Subsequent versions of the simulator will implement other traffic descriptors like mean cell rate and mean burst length. The Quality of Service field defines the cell transfer delay. Route field contains the switches in route to the destination. The NRA is a field that is incremented each time a CAC algorithm fails in a switch on the route, and is used to limit the number of rerouting attempts. The block_links field is the list of links that cannot be used for route calculation because



they do not guarantee the resource reservation of the call parameters request.

Fig. 7 – SIANET switch structure

In SIANET, the ATM switch implementation has the structure shown in figure 7. Each switch has a set of input and output queues associated with the switch links. The queues are created when one VC is established, and each queue is associated with a VC. In the simulator we implement a special queue associated with each switch link, established after the beginning of the simulation, that is reserved for signaling and is identified by VC=5. Switches also have a translation table (TT), that contains information about all Virtual Channels (VC) that are active in this switch, namely the input VPI/VCI and input port identification, and output VPI/VCI and output port identification. Each switch also has a RIB, created before the beginning of the simulation, that is updated through PTSP's flooding. A switch uses the RIB information to calculate the path between itself and the destination switch, which contains information about the available bandwidth, the administrative weight value and the mean delay of each link.

In SIANET, when a switch receives a signalling packet one of three things can occur:

- The switch is a source switch or the packet arrives from another PG (figure 8) - The switch analyses the signalling packet and if the direction is UP and the packet type is SETUP then the switch attempts to find a route.

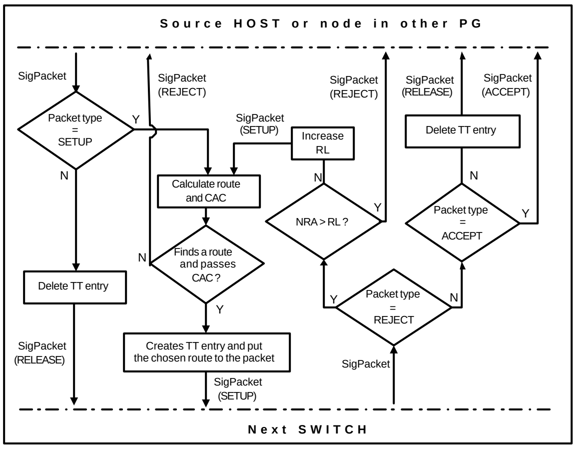


Fig. 8 – Source switch algorithm

If it has success, a TT entry is created at the switch and it sends the packet to the next switch on the route; otherwise, it changes the packet type to REJECT and the direction to DOWN and sends the packet back to the source host or to the previous PG; if the packet type is RELEASE then the corresponding TT entry is deleted, and it sends the packet to the next switch. If the packet direction is DOWN then if the packet type is REJECT it verifies if number of rerouting attempts exceeds the rerouting limit (this value is a simulation global parameter) and if is true it sends the packet to the source host or to the previous PG, otherwise increments the value of NRA and attempts to find a new route as we described above; if the packet type is RELEASE then the corresponding TT entry is deleted, else the packet is ACCEPT and in both cases it sends the packet to the source host or to the previous PG.

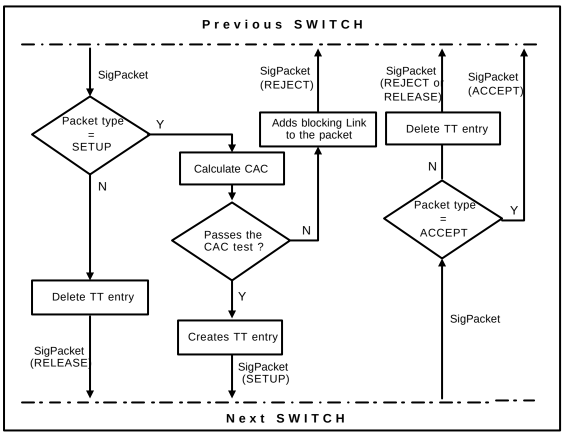
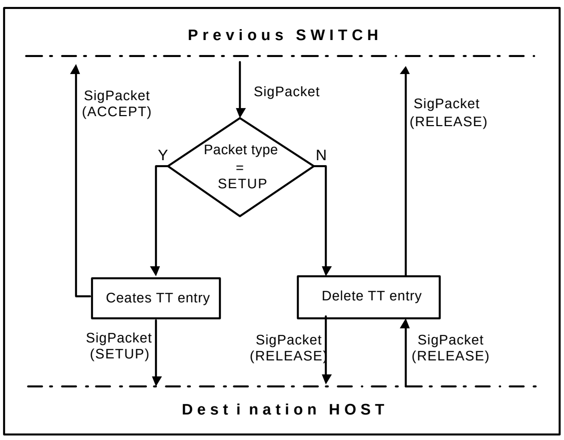


Fig. 9 – Intermediate switch algorithm

- The switch is an intermediate switch (figure 9) - The switch analyses the signalling packet and if the direction is UP and the packet type is SETUP then the switch activates its CAC algorithm and if the request passes the CAC then a TT entry is created in the switch, and it sends the packet to the next switch on the route; if the CAC fails, the switch changes the packet type to REJECT and the direction to DOWN, adds the link identification to the blocking link list and sends the packet back to the previous switch; if the packet type is RELEASE then the corresponding TT entry is deleted and it sends the packet to the next switch. If the packet direction is DOWN then if packet type is ACCEPT then it sends the packet to the previous switch, else the packet type is RELEASE or REJECT and the corresponding TT entry is deleted and the switch sends the



packet to the previous switch.

Fig. 10 – Destination switch algorithm

- The switch is the destination switch (figure 10) - If the packet direction is UP then if the packet type is SETUP a TT entry is created at the switch and it creates a copy of the signaling packet. The switch then sends one packet to the destination host and changes the packet type of the other packet to ACCEPT and the direction to DOWN, and sends the packet back to the previous switch. If the packet type is RELEASE then the corresponding TT entry is deleted, and the switch sends the packet to the destination host. If the packet direction is DOWN and it is a RELEASE packet, the corresponding TT entry is deleted and the switch sends the packet to the previous switch.

5. SAMPLE SIMULATION AND RESULTS

In order to illustrate the use of SIANET, this section presents a sample simulation and some of its results. The simulator was used to evaluate and compare the performance of hierarchic routing as opposed to flat routing. Two scenarios were simulated, based on the network shown in Figure 11.

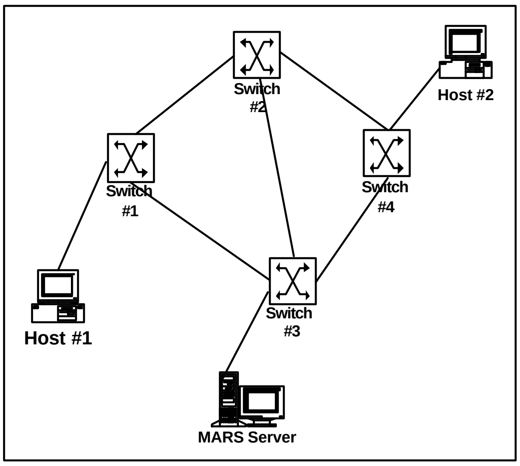


Fig 11 – Network under study

In both scenarios, the network is composed of four switches, two hosts and one MARS server (Multicast Address Resolution Server). The difference between the scenarios is that in scenario #1 all the switches belong to the same PG, and so hierarchic routing is not used, whereas in scenario #2 the first switch belongs to one PG (pg 1) and the others belong to another PG (pg 2). Thus, scenario #2 uses hierarchic routing. Both scenarios consider CBR traffic only, each call requiring a bandwidth of

1Mb/s. Link propagation delays are considered negligible.

In addition to evaluating the impact of hierarchic routing, the simulation study also intended to identify the influence of the number of active connections on the call set up delay. Thus, several simulation runs were made with different numbers of established connections, for each of the scenarios under study. Figure 12 presents the results. Note that each point in the graphics represents the average of ten independent runs.

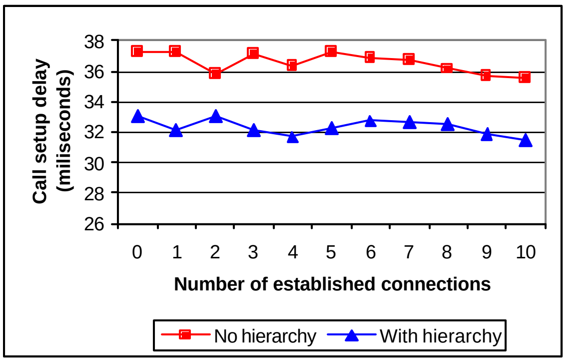


Fig. 12 – Simulation results

The figure shows that the call set up delay in scenario #1, that is, without hierarchic routing, is roughly 10% higher then the call set up delay in the case of hierarchic routing. Additionally, the simulation also showed that there is no relation between the call set up delay and the number of established connections. In fact, the existence of other connections should only influence call set up only when the bandwidth utilization exceeds 70% of the total bandwidth, because in this case the CAC algorithm starts rejecting connection requests and, consequently, the time to successfully establishing a connection increases.

6. CONCLUSIONS

This paper presented a simulation tool, SIANET, for IPv6-over-ATM networks, with emphasis on the PNNI routing and signalling module. SIANET is a modular simulation tool written in C++ that runs on Microsoft Windows platforms and that allows the easy creation, modification and study of network models. The tool can be used to obtain graphical as well as numerical results, and supports ATM Forum's Private Network to Network Interface specification. Future developments will extend the tool in order for it to support point-to-multipoint call and connection control, additional ATM traffic control and more ATM QoS features. The inclusion of new features is made easy by the modular nature of the simulator.

Acknowledgement

The work presented in this paper is partially financed by the Portuguese Foundation for Science and Technology, FCT.

7. BIBLIOGRAPHY

[1]Clark, Martin P., "*ATM Networks - Principles and Use*", John Wiley & Sons ltd, 1996

[2] Deering, S and Hinden, R., "*Internet Protocol Version 6 (IPv6) Specification*", RFC 1883, XEROX PARC, Ipsilon Networks Inc, December 1995

[3] S. Giordano, R. Schmid, R. Beeler, H. Flinck, J.Y. Le Boudec, "*IP and ATM: current evolution for integrated services* ", Technical Report , DI-EPFL, Lausanne, Switzerland, September 1999

[4]Armitage, G., et. al., "*IPv6 over Non-Broadcast Multiple Access (NBMA) networks*", Internet Draft <draft-ietf-ion-ipv6-01.txt> Lucent Technologies, Brigth Tiger Technologies, Digital, March 1998

[5] Armitage, G., Schulter, P. and Jork, Markus, "*IPv6 over ATM Networks*", Internet Draft <draft-ietf-ion-ipv6-atm-02.txt> Lucent Technologies, Brigth Tiger Technologies, Digital, April 1998

[6] ATM Forum, "*ATM User-Network Interface (UNI) Specification version 4.0*", ATM Forum Specification af-sig-0061.000, url: ftp://ftp.atmforum.com/, Jully 1996

[7] The ATM Forum, "*ATM Private Network-Network Interface Specification*", Version 1.0, March 1996

[8] Reisdorph, Kent and Henderson, Ken, "*Teach yourself Borland C++ Builder* ", Sams Publishing, Borland Press, December 1995

[9]Laubach, M and Halpern J., "*Classical Ip and ARP o*

ver ATM", RFC 2225, Com21, Inc., Newbridge Networks, Inc., April 1998

[10] Armitage, G., "*Support for Multicast Over UNI 3.0/3.1 based ATM Networks*", RFC 2022, Bellcore, November 1996

[11] Veiga, Nuno, Sá Silva, Jorge, Duarte, Sérgio, Boavida, Fernando, "*Performance measurement of an IPv6-over-ATM Platform*", Technical report, CISUC, University of Coimbra, November 2000

[12] Farraposo, Silvia, "*Estudo do encaminhamento dinâmico em redes ATM*" (Study of Dynamic Routing in ATM Networks), M.Sc. thesis, University of Coimbra, September 2000 (Portuguese).

[13] Stallings, William, "*High-Speed Networks: TCP/IP and ATM design principles*", ISBN 0-13-525965-7, Prentice-Hall, Inc. Upper Saddle River, NJ, 07458, 1998