# SIMULATING MULTICAST TRANSPORT PROTOCOLS IN ESTELLE[1]

*A re-usable IP network module*

Justin Templemore-Finlayson
*Intitut National des Télécommunications*
*Evry, France*
*templemo@hugo.int-evry.fr*

Eugen Borcoci
*University POLITEHNICA of Bucharest,*
*Bucharest, Romania*
*eugenbo@first.elcom.pub.ro*

Key words: Multicast Transport Protocols, Formal Description Techniques, Estelle, Formal Simulation, Component re-use.

Abstract: State of the art multicast research points to a future architecture in which multicast transport protocols provide end-to-end control over the IP multicast datagram service, similar to TCP/IP. However, many aspects of these protocols are still to be resolved, and are active areas of research. We present a simulation architecture for applying the Formal Description Technique Estelle to the development of multicast transport protocols. A key component of this architecture is the IP network (IPN) module, which is the focus of this paper. The IPN module reproduces complex medium behaviours specific to multicast transport protocols and not provided by existing IP medium models, including: IGMP services, network heterogeneity, address-based routing, and multicast delivery. The sophistication of the IPN module presents a realistic simulation environment for the formal investigation of multicast transport protocols. The module has in addition been designed to be independent of the upper-layer protocol being investigated, and as such presents a re-usable component for the simulation of any IP-based protocols.

[1] This work has been partially supported by the Copernicus/IDEMCOP project.

---

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-0-387-35533-7_26

Tommaso Bolognesi and Diego Latella (Eds.), *Formal Methods for Distributed System Development.*

# 1. INTRODUCTION

Multicast is a key technology to the future of the Internet. Multicast protocols enable *group communication* - communication with more than two participants - which increases the functionality of the Internet without significant negative effect on network resources [10].

Applications which can benefit from a native multicast protocol include electronic file distribution, collaborative environments and multimedia streaming, to name but a few. In addition, important network-related functions will migrate to multicast: network announcements, replication, mirroring and discovery protocols can all benefit from multicast.

The basis of multicast applications will most likely be a TCP/IP - like stack, in which multicast transport protocols provide end-to-end reliability and services using the IP multicast datagram service [7].

Design of multicast transport protocols is ostensibly simple when a native IP multicast service is present, but it has proved difficult to create protocols that scale well on a heterogeneous wide area network, such as the Internet. The growing body of multicast research has also shown that multicast transport protocols, especially reliable protocols, can pose a threat to the overall functioning of the Internet [10], as multicast transport protocols must co-operate with each other, and TCP. These factors make it important to understand the operation of these protocols in the Internet, prior to their deployment.

In this paper we present a simulation architecture for applying the Formal Description Technique (FDT) Estelle [1] to the analysis of multicast transport protocols. This FDT provides a rigorous method for communication protocol design and analysis by simulation, which can help our understanding of multicast transport protocols.

**A key element of this architecture is an Estelle IP network (IPN) medium module.**

Creation of a medium module is a common technique for simulating Estelle-based protocols, but the IPN module includes substantial novelty through its support of multicast transport protocols: The module provides multicast delivery, and Internet Group Management Protocol (IGMP) services, and emulates heterogeneous wide area network behaviour. These characteristics are essential for realistic simulation of multicast transport protocols.

In addition, the IPN module was designed and implemented completely independently of upper-layer protocols. It presents a definitive, re-usable component for Estelle-based simulation of any transport or application protocols, unicast or multicast, based on IP.

To the author's knowledge, the IPN module is the first multicast-capable medium module to be developed for a Formal Description Technique. Such medium models have been developed during research using informal simulators, namely ns and Opnet, but none have been completely separated from the upper-layer protocol, and are therefore not useful to the broader simulation community.

Section 2 presents the IP Network module requirements, including the service interface which is a key element of re-use.

Section 3 describes the internal functioning of the module in detail.

Section 4 presents simulation experiments of the Reliable Multicast Transport Protocol (RMTP-II) [9],[12], a proposed multicast transport protocol for the Internet. A simulation architecture for RMTP-II using the IPN module is presented, and the use of the IPN module to produce network behaviours is discussed.

## 2. THE IP NETWORK MODULE REQUIREMENTS

The IPN module is a completely re-usable Estelle module which emulates a wide area, multicast-capable IP network.

In Estelle, it is common practice to specify a system environment using a medium module when simulating a system. This provides the user with control over the environment in which the protocol is being tested. A medium module is usually configured with the static resources of the network, and interactive simulation allows the provocation of dynamic network situations and events.

The IPN module is based on the *generic medium module*, an Estelle medium module successfully used in for simulation of several large protocols (e.g. SSCOP [2], OSI-TP [5], XTP [6]).

The generic medium module is a generic unicast/broadcast transfer medium which produces network events for packet loss, duplication and re-ordering. It models a homogeneous network in that network event probabilities and channel bit rate for each sender-receiver pair are the same. It provides transfer of variable-size packets in unicast and broadcast modes, with transmission delay proportional to the packet size.

Within the EDT simulator, it is possible to change the configured event probabilities during simulation, thereby reproducing dynamic network conditions.

This model has limited applicability for simulation of multicast transport protocols, as it lacks a multicast service and does not model the heterogeneous behaviour exhibited by wide area networks. The IPN module reproduces the behaviour of the generic medium module for an IP network,

and adds a native IP multicast service and heterogeneous behaviour between different sender-receiver node pairs in the network.

The key elements of the new IPN module are :

- IP unicast and multicast addressing
- unicast and multicast packet delivery
- IGMP services
- heterogeneous network behaviour

The IPN module multicast model faithfully reproduces that of IP multicast.

Heterogeneous network behaviour is specified by defining event probabilities for each individual pair of nodes connected to the IPN module. The following events can be configured probabilistically:

- packet loss
- packet duplication
- packet re-ordering
- transmission delay

In addition, where the generic medium module requires editing of the internal Estelle code of the module for re-use, the IPN module is encapsulated, and entirely configured through a standard external interface, described in the following section. This configuration by the higher-layer protocol does not add any complexity to the protocol specification, as it is simply a formalisation of the IP network interfaces.

Note that the IPN module simplifies the generic medium module by using a transmission delay independent of packet size. This is a conscious design choice which enables us to encapsulate and re-use the IPN module without change. If desired, transmission delays based on packet-size can be re-introduced by modifying the IPN module.

## 2.1 IPN module interface

The IPN module implements a simplified interface of actual IP. This interface permits upper-layer entities to dynamically change their relationship with the communication medium, within a static Estelle system structure. This service interface is depicted in Figure 1.

The *bind* service presents the corresponding IP/UDP function to start listening on a given network address. It eliminates the need for static configuration of the interfaces at the IPN module : the parameterised address (*my_address*) is used to perform packets routing and delivery.

The *packet* service performs multicast or unicast delivery (subject to non-deletory network events), based on the destination address (*dest_addr)* specified. The IPN module recognises IP multicast addresses and performs multicast or unicast delivery appropriately. The IPN module does not

examine the *PDU* field. By using an external variant record *PDU_type* containing the upper-layer PDU definitions, the IPN is able to transfer any *PDU* without reconfiguration.

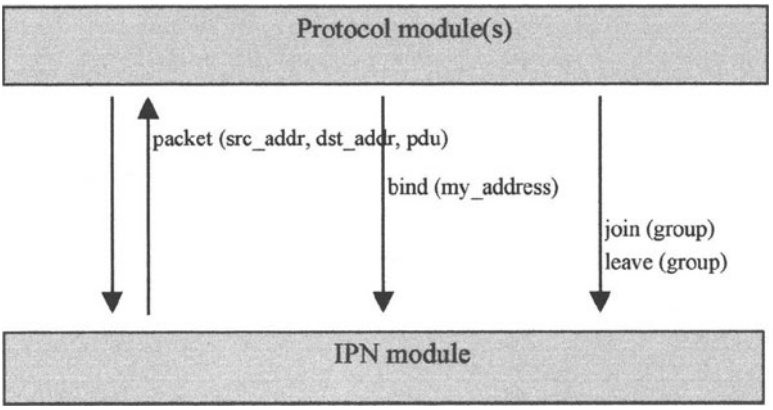


Figure 1. IP Network (IPN) service interface

The *join* and *leave* services correspond to the basic IGMP services for subscribing to and unsubscribing from a multicast group [7].

# 3. IPN MODULE FUNCTIONAL DESCRIPTION

## 3.1 Implementing the IPN service interface at the IPN-user protocol

The IPN service interface is defined in an Estelle channel definition, reproduced in Figure 2. To use the IPN module as a medium in a simulation, a protocol needs to extend this channel definition with its own PDU definitions.

There is a one-to-one correlation between the interactions defined in the channel, and the services described in Section 2.1.

Customisation of the channel to carry specific PDUs simply requires the definition of the `PDU_type` as a union record with a case for each PDU.

```
channel IPN_access (PROVIDER,USER);
  by USER:
    join_group (group:network_address_type);
    leave_group (group:network_address_type);
    bind (my_address:network_address_type);
  by PROVIDER,USER:
    pdu (src_addr:network_address_type;
         dst_addr:network_address_type;
         pdu:PDU_type);
```

Figure 2. The Estelle channel defining the IPN module service interface.

## 3.2 Packet addressing and delivery

The IPN module accepts packets with three fields: *source address*, *destination address*, and *protocol data unit.*

The source and destination address are defined as UDP addresses consisting of an IP address and port number. The IPN module examines the *destination address* IP address to determine the destination host(s). The UDP port numbers are preserved for the upper-layer protocol or an intermediate UDP module to handle.

The choice of whether to use unicast or multicast delivery is based on the *destination address* specified in a packet. IP multicast reserves Class D addresses (224.0.0.0 to 239.255.255.255) for multicast. The IPN module recognises these multicast addresses and delivers using multicast as appropriate.

## 3.3 Heterogeneous network behaviour

The *generic medium module* presents a medium with homogeneous transmission and event characteristics. It allows the user to configure the following characteristics of the network:

- packet loss probability
- packet duplication probability
- transmission delay based on packet length

These values are shared for all host pairs in the simulated network.

The IPN module extends this model by allowing these characteristics to be configured per host pair connected to the medium. This flexibility enables the heterogeneous conditions of a wide area network to be modelled.

The IPN module makes a simplification to the trasfer delay modelling, calculating delay between two hosts using a normal distribution function

around a specified mean delay. Transfer delay is therefore independent of packet length. Measuring the packet length in Estelle would require knowledge of the upper-layer protocol data unit structure at the IPN module, which conflicts with the goal of encapsulation of the IPN module. A protocol designer may of course alter the IPN module to introduce packet-size dependent delays.

These medium characteristics are accessible during interactive simulation. In tool-sets such as EDT [3],[4], the characteristics of the path between two or a group of hosts can be changed to produce situations such as local congestion or single link failures. Construction of medium scenarios as described in [6] can be used to create a realistic Internet environment in which to test a multicast transport protocol.

## 3.4 Internal architecture

The internal architecture of the IPN module is depicted in Figure 3. The module operates in the following way:

1. A packet placed on the medium by the $i^{th}$ host is received at interaction point SAP[*i*].
2. The time of receipt is recorded (*timeIn*) and stored with the packet in the Input list.
3. If the packet *dst_addr* field is a not Class D address, i.e. unicast, the destination interaction point, SAP[*j*] is looked up in the address tables.
4. The transfer events *loss* and *duplication*, and the *transmission delay* for the host pair (*i*, *j*) are generated. If the packet is duplicated, an independent transmission delay is generated for it.
5. Any packet not lost is placed in the Output list. The scheduled output time is stored in *timeOut* and the destination interaction point index *j* in *dstHost*.
6. If the incoming packet *dst_addr* was a Class D address, i.e. multicast, the SAP of each host joined to that multicast group is looked up in the address tables, and steps 4 and 5 are performed for each host.
7. An output scheduler periodically increments a time counter and searches the Output list for packets whose *timeOut* equals the current time. The referenced packet (*src_addr*; *dst_addr*; *pdu*) stored in the Input list is then output on SAP[*dstHost*].
8. A packet is removed from the Input list when there are no more references to it in the Output list.

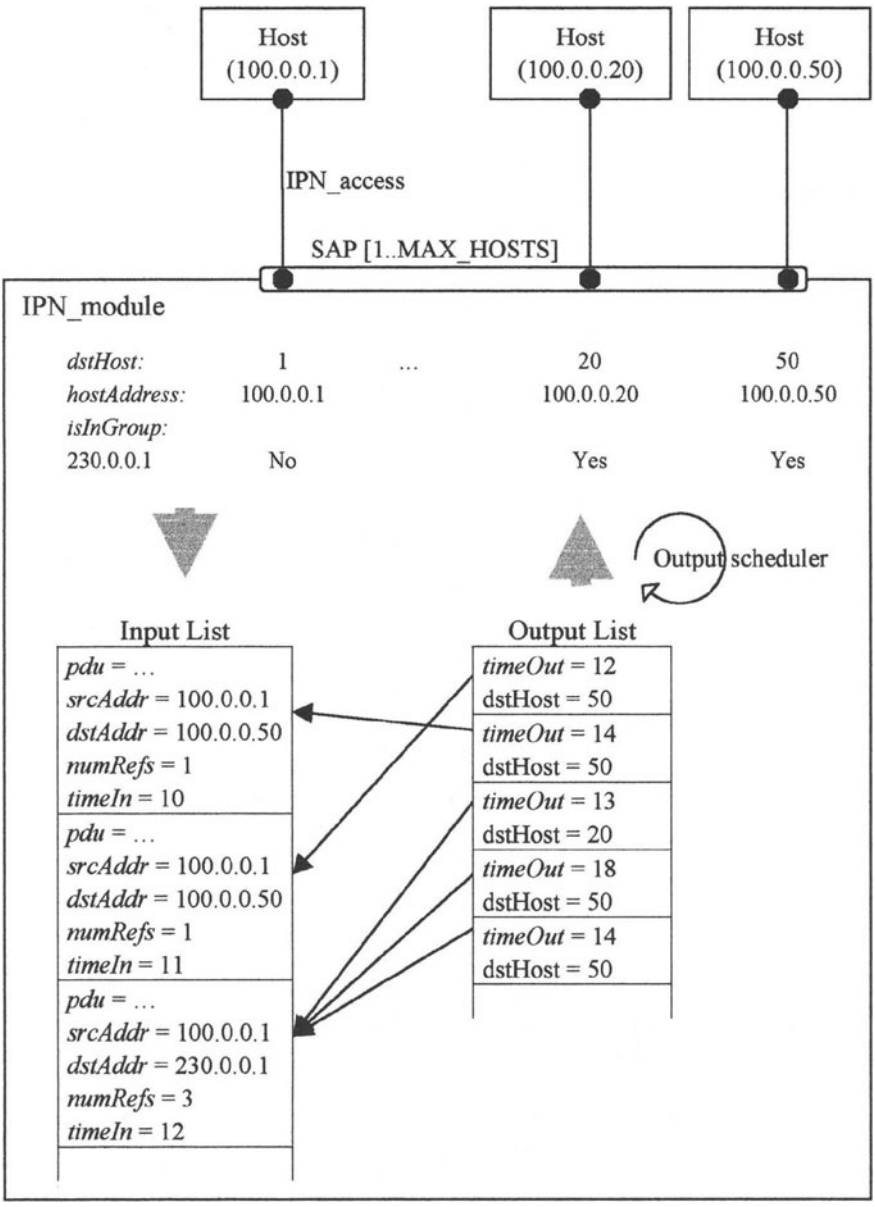


Figure 3. IPN module internal architecture

Within this scheme, re-ordering occurs when a packet to be routed between two hosts experiences a shorter delay than a previous packet scheduled between those two hosts.

WAN heterogeneity is also experienced. Packets travelling between the same two hosts experience varying delays, and probabilistic loss or duplication. Packets sent between different host pairs may experience radically different conditions. Furthermore, packets sent to a group will experience different conditions for each different receivers within that group.

The use of separate Input and Output lists makes the IPN module scalable to a large number of connected hosts: If a packet is multicast to a group with one thousand members, only a single copy of the packet is kept (in the Input list), and the memory requirements for storing a thousand references to that packet are very small.

### 3.4.1 Example

In Figure 3 the current state of an IPN module gives examples of unicast delivery with re-ordering, and multicast delivery to multiple receivers are given. For brevity, we refer to each host by its last byte only (i.e. 100.0.0.1 becomes simply 1).

In this example, the current network configuration shows host 1 bound to SAP[1], host 20 bound to SAP[20] and host 50 bound to SAP[50]. Hosts 20 and 50 are also all members of the multicast group 230.0.0.1.

All the packets in the Input list have been transmitted by host 1. The first two packets in the list are destined for host 50. None have been lost or duplicated (there is a single reference to each in the Output list), but the transition delay for the first packet is 4 time units, while for the second, transmission delay is only 1 unit. As the second packet was received only one time unit after the first, they will be scheduled for output in the reverse order, thus re-ordered.

The third packet will be multicast, as it has a Class D destination address (230.0.0.1). Entries for each member of the group (hosts 20 and 50) are inserted into the output table. Loss, duplication and transmission delays are calculated separately for each receiving host. None are lost, but the packet destined for host 50 has been duplicated, and each of the resulting three packets in the Output list are scheduled to be output at different times.

# 4. USE OF THE IPN MODULE TO SIMULATE THE RMTP-II PROTOCOL

The IPN module has been developed within a project to formally specify and validate the Reliable Multicast Transport Protocol-II (RMTP-II) [12]. An Estelle specification of the RMTP-II has been written [11], and is in the process of being simulated. The IPN module is a vital component in these simulations, enabling the wide area network behaviour experienced by this protocol to be reproduced.

## 4.1 The RMTP-II protocol and specification

The Reliable Multicast Transport Protocol (RMTP-II) defines a transport layer protocol for the reliable multicasting of data from one or relatively few senders, to a large group of receivers. RMTP-II uses the IP multicast service to distribute data, and IP unicast to exchange control traffic between protocol nodes.

RMTP-II provides a multicast equivalent of TCP: a sender-reliable, connection-oriented multicast service. The main challenge to be met by the protocol is the scalable processing of control traffic. RMTP-II solves this by constructing a tree of protocol nodes rooted at the sender and spanning all the receivers. Control traffic from receivers is aggregated as it flows towards the sender, thus avoiding packet implosion.

The Estelle specification of RMTP-II is implementation-oriented, which will be used to generate a working implementation compatible with other RMTP-II implementations.

## 4.2 Application of the IPN module

Use of the IPN module in the simulation of the RMTP-II protocol does not require any static configuration of the IPN module. However, the RMTP-II protocol must be designed to correctly use the IPN module interface. The following conventions must be observed in the RMTP-II protocol :

1. The channel `IPN_access` (see Section 3.1), must be defined in the specification module, and the type `PDU_type` declared as a variant record containing all the RMTP-II PDUs.
2. A `bind` interaction output is added to the RMTP-II protocol modules initialisation part in order to identify themselves to the IPN module.

3. Packet handling at the protocol modules must be implemented using IP addressing and the interaction definition formats of the channel `IPN_access`.
4. `Join` and `leave` interactions must be explicitly output to subscribe to / unsubscribe from multicast groups.

As the RMTP-II specification is an implementation-oriented specification, which has to deal with underlying IP protocol details, these steps did not add any complexity to the specification, simply syntactic changes.

In the following we present two examples of simulations of the E-RMTP protocol. In these simulations, the IPN module is used in conjunction with a network behaviour scenario to reproduce dynamic WAN conditions in which to test the protocol.

## 4.3 Simulation experiments with RMTP-II and the IPN module

Figure 4 presents the RMTP-II simulation architecture in Estelle, including the IPN module. In Figure 4 the RMTP-II protocol is defined by the collection of shaded modules, the *Sender Node (SN)*, the *Top Node (TN)* and a group of *Receiver Nodes (RN)*. Each of these performs a specialised function in the protocol.

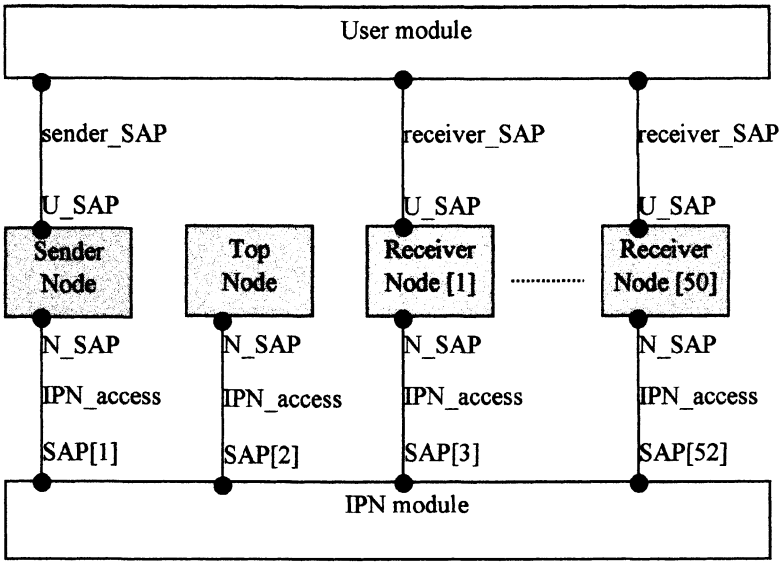


Figure 4. RMTP-II simulation architecture

The SN performs data transmission, the RN data reception, and the TN acts as a co-ordinator between the group members. The *IPN module* provides the IP wide area network environment, and the *User module* simulates sender and receiver user behaviour.

Simulations are performed by a method of guided simulation, used in the validation and performance evaluation of other complex Estelle specifications in [5] and [6].

In our simulations of RMTP-II, we use this technique to define independent user and medium scenarios, which enables us to test different protocol behaviours (stimulated by the user scenario) under different network conditions (stimulated by the medium scenario).

In the following we present an example simulation of RMTP-II using the IPN module. Note that this simulation does not show us anything new or unexpected about the RMTP-II protocol, apart from offering a formal confirmation of a well-known phenomenon. However, these measurements would not be possible without the IPN module. They demonstrate how the IPN module can be used to simulate the complex network conditions experienced by multicast protocols, which can be used to search for solutions to problems faced by multicast transport protocols.

### 4.3.1 Investigating the "crying-baby" problem

The "crying-baby" problem [10] describes the decrease in throughput observed by a member of a group due to the presence of a slower receiver, or "crying-baby", in that group. This is a common example of the difficulties faced by multicast protocols in a heterogeneous environment.

In this experiment, we create a crying-baby medium scenario for RMTP-II by gradually increasing the error rate observed between the *Sender Node* and an arbitrary *Receiver Node* (the "baby"), while maintaining constant error rates between the sender and other receivers.

We measure the effect of this scenario on protocol throughput by recording and calculating the average of transfer times seen at each receiver. In this experiment, the transfer time is an indicative value of the number of retransmission cycles required to deliver all the packets in a data stream.

Figure 5 shows two results. Firstly the expected rise in transfer time to the crying-baby. Secondly and more interestingly, it shows how the average transfer time experienced by receivers closely shadows that of the crying-baby. This is the expected result in the case of RMTP-II, which uses a flow control window that only advances once all receivers in the group have acknowledged a packet.

The simulation therefore confirms the observation that throughput for the RMTP-II protocol is only as great as the throughput of the slowest receiver in the group.

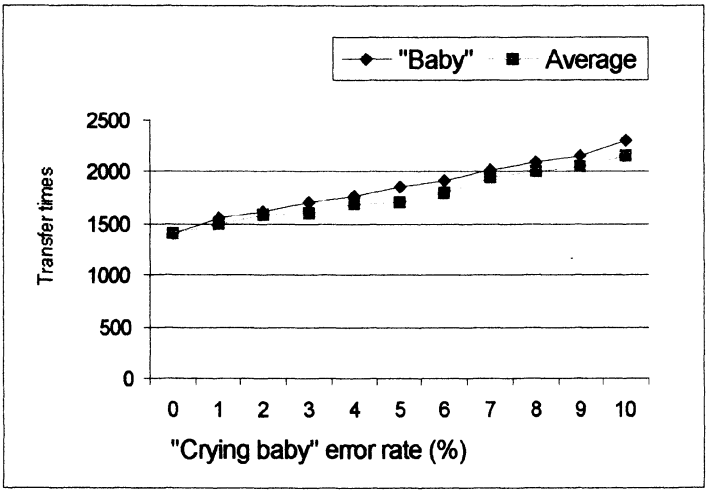


Figure 5. RMTP-II "crying-baby" simulation experiment.

## 5. CONCLUSIONS

Realistic validation and performance evaluation of a protocol in Estelle requires a realistic model of the environment in which that protocol is designed to operate.

The IPN module presents a Estelle module which reproduces the complex behaviour of a wide area, multicast-capable IP network. The module is self-contained and re-usable, and as such it presents a "ready-to-use" IP network medium, which relieves the protocol designer from writing such a medium module themselves.

The IPN module has been used as a medium module in the Estelle simulations of a multicast transport protocol, RMTP-II. A simulation example presented in Section 4 shows that the IPN module is capable of reproducing the complex heterogeneous environment which these protocols experience. These simulations are part of an ongoing project to formally investigate the many problems faced by Internet multicast transport protocol designers, and in which the IPN module plays an important role.

# REFERENCES

[1] ISO/IEC, 1989. Information Processing Systems. Estelle - A Formal Description Technique based on Extended State Transition Model, IS 9074.

[2] E Borcoci, S Budkowski, 1999. Traffic capability of the ATM-SSCOP protocol - Simulation Study, *Proceedings of the European Simulation Multiconference 1999*, Warsaw, Poland.

[3] S Budkowski, 1992. The Estelle Development Toolset, *Computer Networks and ISDN Systems Journal*, Special Issue on FDT Concepts and Tools, Vol.25, No.1.

[4] S Budkowski et al., 1998. *The Estelle Development Toolset*, Institut National des Télécommunications, Evry, France. http://www-lor.int-evry.fr/edt/

[5] O Catrina and S Budkowski, 1998. *Integration of subtransactions in OSI-TP. Validation of the Estelle specification using EDT*, Research Report, Institut National des Télécommunications, France, December 1998.

[6] O Catrina and E Borcoci, 1996. Estelle specification and validation of XTP 4.0, *Copernicus Project (COP62) Deliverable for Workpackage 2, Task 2.1*, University of Bucharest, 1996.

[7] S Deering, 1989. *Host Extensions for IP multicasting, RFC 1112*, August 1989.

[8] A Mankin, A Romanow, S Bradner, V Paxson. *IETF Criteria for Evaluating Reliable Multicast Transport and Application Protocols, RFC 2357*, June 1998.

[9] P Sanjoy, K Sabnani, JC Lin and S Bhattacharya, 1997. Reliable Multicast Transport Protocol (RMTP), *IEEE Journal on Selected Areas in Communications*, Vol. 15, No. 3.

[10] B Quinn and K Almeroth, 1999, *IP multicast applications : Challenges and Solutions*, Internet Draft, 26 February 1999.

[11] J Templemore-Finlayson, 2000, *An Estelle Specification of the Reliable Multicast Transport Protocol II*, Research Report N° 00001-LOR, Institut National des Télécommunications, France, January 2000.

[12] B Whetten, M Basavaiah, S Paul, T Montgomery, N Rastogi, J Conlan and T Yeh. *Reliable Multicast Transport Protocol - II*, Internet Draft, April 1998.