TAP: Architecture for Trusted Transfers in ATM Networks with Active Switches [1]

Prof. José Luis González-Sánchez (*) and Dr. Jordi Domingo-Pascual (**)
(*) University of Extremadura. Escuela Politécnica de Cáceres Avda. Universidad S/N. (10.071) Cáceres (Spain)
Phone: +34 927 257 259 Fax.:+34 927 257 202 E-Mail: jlgs@unex.es
(**) Polytechnic University of Catalonia. Campus Nord, Modul D6, Jordi Girona 1-3 (08034) Barcelona (Spain)
Phone: +34 93 401 6981 Fax: +34 93 401 7055 E-Mail: jordi.domingo@ac.upc.es

Abstract

Significant research efforts are currently centered on effective mechanisms of cell loss control and congestion control for ATM networks. This paper presents TAP (Trusted and Active PDU transfer); a new architecture and protocol for ATM networks that provide assured transfers to a set of privileged VPI/VCI. The architecture proposes a new Extended AAL type 5 (EAAL-5), manages the privileged connections and offers an improvement in the performance when network connections cause some cell loss by taking advantage of the idle time in the traffic sources to do the retransmissions of CPCS-PDU-EAAL5. We propose a new and trusted protocol that uses NACK mechanisms (using backward RM cells), and which is supported by active ATM switches equipped with Dynamic Memory to store Trusted native EAAL-5 PDUs (DMTE). The active architecture presents the Resource Management Agent (RMA) to manage the retransmissions of PDUs between active switches, and the Early Packet Discard Agent (EPDA) to recover PDUs internally within the active switches. Several simulations demonstrate the effectiveness of the mechanism that recovers the congested PDUs locally at the congested switches with better goodput in the network. Also, the senders are alleviated of negative end-to-end retransmissions. The simulations, using ON/OFF sources, analyse point-to-point, and also point-to-multipoint connections using objects, threads, synchronizations, and distributed processes implemented in Java language.

1 Introduction and related work

ATM technology is characterized by its excellent performance and by offering to the user the possibility to negotiate QoS (Quality of Service) [1] parameters such as throughput, delay, jitter and reliability. Reliability in ATM networks is provided by the Header Error Control (HEC) field of 8 bits in the header of the ATM cells and by the Cyclic Redundancy Check (CRC) in the Common Sublayer-Protocol Data Unit (CS-PDU). Error control in ATM networks is performed end-to-end by the terminals. The main problem is that a single cell loss causes a reassembly CRC error at AAL5 level which in turn leads to a retransmission a of a complete PDU (i.e., IP datagram).

ATM networks experience three types of errors [2-5]: cell losses due to congestion in switches; corruption of data portions due to bit errors, and switching errors due to undetected corruption of the cell header.

We note that congestion is by far the most common type of error, and here is where we want to improve the trusted transfers with TAP.

Current literature describes three basic techniques to achieve reliability: ARQ (Automatic Repeat Request), FEC (Forward Error Correction) and hybrid mechanisms of ARQ in combination with FEC.

ARQ [6] is a mechanism based on retransmissions of data that were not correctly received due to some of the problems cited above. ARQ offers two variants and both require the transmitter and receiver to exchange information on the state of the transmission. In a variant, the receiver sends positive acknowledgement (ACK) messages back to the sender even when it has successfully received the packets. This is the classical mechanism to improve reliability in unicast transmissions. In the second variant of ARQ the receivers send negative acknowledgement messages (NACK) back to the sender only when they have lost data. FEC [7-11] is an important alternative to ARQ whose operation principle is to encode the packets in the transmitter with redundant information so that it is possible to reconstruct the original packet by reducing, or even eliminating, the retransmissions and the negative effect of implosion.

(1) This work is sponsored in part by the CICYT under Grant No. TEL99-1117-C03-03 and by the Junta of Extremadura (Education and Youth Council) and the European Social Fund Register No. BRV98101.

While ARQ adds latency (due to the cost of NACK) and implosion, FEC adds overhead and thus the redundant code added by this method is useless when the network is experiencing congestion. Hence, ARQ may not be suitable for applications with requirements of low latency, and FEC performs worse in networks with low bandwidth or which experiences frequent congestion. In our architecture we adopted ARQ with NACK (using RM cells) to alleviate the effect of implosion. Support for reliable multicast cannot be based on retransmissions from the source. In TAP, the intermediate active nodes carry out the retransmissions.

As we have seen, the main research to offer reliability is inspired in mechanisms such as redundant code (like CRC or FEC) that introduce important overhead in transmission and affect the throughput negatively. These methods solve the problem of retrieval of the corrupted cells, but they cannot resolve another undesirable, and more frequent problem, such as congestion and packet fragmentation in switches due to congestion which causes the effective throughput to be degraded. The most commonly proposed congestion control schemes to improve throughput and fairness, while minimizing delay in ATM networks, are the Random Cell Discard (RCD), Partial Packet Discard (PPD), Early Packet Discard (EPD) [12]; Early Selective Packet Discard (ESPD), Fair Buffer Allocation (FBA) and Random Early Detection (RED). TAP uses EPD to alleviate the effect of congestion and packet fragmentation.

ATM Adaptation Layer type 5 (AAL-5) has been developed to support transfers of non assured frames of data user, where the lost and corrupted Common Part Convergence Sublayer Service Data Unit (CPCS-SDU) cannot be solved with retransmission [4]. We propose EAAL-5 as an extended and enhanced native AAL-5. EAAL-5 is part of TAP that supports service assured with retransmissions and is also compatible with native AAL-5. In this paper we propose a mechanism to take advantage of the idle periods in the data sources to retransmit the Common Part Convergence Sublayer PDU (CPCS-PDU) of EAAL-5.

Active, open and programmable networks is a new technical area [13-17] to explore ways in which network elements may be dynamically reprogrammed by network managers, network operators or general users to accomplish the required QoS and other features as customized services. This offers attractive advantages, but also important challenges in aspects such as performance, security or reliability. Hence, this is an open issue for research and development in customized routing and protocols, whether to move the service code (placed outside the transport network) to the network's switching nodes. The literature in this field studies several mechanisms to obtain advantage from active nodes. Paper [13] is an excellent review of this new research field. A network is active if there are active nodes in its multicast distribution trees with the capacity to execute the user's programs, and also if it implements mechanisms of code propagation. Many of the advantages of active protocols are achieved by installing active nodes at strategic points. Concepts such as active networks, protocol boosters or software agents are proposed and developed for IP networks; however the proposals are insufficient for ATM networks and reference [14] is an example of this recent research in ATM. In congested networks all the agents then locally redistribute the capacity as appropriate.

We bring active characteristics to TAP through hardware mechanisms and software techniques. TAP architecture includes the RMA agent (Resource Management Agent) to manage the retransmission requests between peer active switches, and the EPDA agent (Early Packet Discard Agent) to recover PDU from the DMTE.

We have simulated and studied point-to-point, point-to-multipoint and multipoint-to-multipoint transfers combined with the active switch and other non-active ATM switches to constitute a VPN (Virtual Private Network). Java Development Kit V1.2.1 has been the language and environment used to implement TAP due to the special characteristics offered by Java.

Section 2 describes TAP architecture. In sections 3 and 4, we present our prototype of an active switch that support the architecture. Section 5 describes and outlines our work currently in progress to enhance the TAP protocol. Finally we offer some concluding remarks in section 6.

2 General description of TAP architecture

AAL-5 was proposed [4] to reduce overhead introduced by AAL3/4. **Fig. 1** compares CPCS-PDU format of native AAL-5 and EAAL-5 with all its fields. As we can see, the tail of PDU has 4 fields. The CPCS-UU (User-to-User indication) field is used for the transfer of CPCS user to user information. The CPI (Common Part Indication) octet is used to align 64 bits to the CPCS-PDU tail.

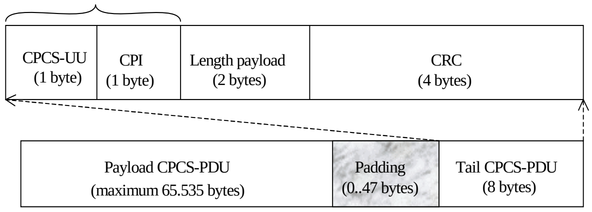

Fig. 1 CPCS-PDU AAL5 and EAAL5 format

TAP utilizes these two octets as the PDU sequence number, which is assigned end-to-end by the EAAL5 user. The CRC is used as in AAL-5 to detect bit errors in the CPCS-PDU. The value of CRC is calculated including all the fields of the CPCS-PDU. The sequence number in PDUs is preserved end-to-end to avoid recalculating the CRC and modifying the tail of the CPCS-PDUs.

ARQ [6] is a technique of retransmission used in TAP protocol. To implement NACK we use the standard Resource Management (RM) cells, without fixed frequency but generated when a switch is congested. This is to alleviate the negative overhead effect due to a fixed number of RM cells that will waste bandwidth.

When congestion is detected, EPD discards a PDU. Then the EPD Agent searches for the discarded PDU in the DMTE. If this PDU is not in the local DMTE, the RM Agent of the active node generates a RM cell which is transmitted backwards to the upstream active switch indicating the sequence number of the discarded PDU. The RM must also contain the VPI/VCI to identify the connection that has experienced discard problems. This mechanism requires to relate the sources of traffic with theirs ports E/S to alleviate the effect of equal values in VPI/VCI at differents ports. Octets 22 to 51 of the RM cell store the identifier Port/VPI/VCI/PDUid of PDUs requested (see **Table 1**).

Field	Octet	Bit(s)	Description
Header	1-5	All	RM-VPC:VCI=6 and PTI=110; RM-VCC:PTI=110
ID	6	All	Protocol identifier
DIR	7	8	Direction
BN	7	7	BECN Cell
CI	7	6	Congestion Indication
NI	7	5	No increase
RA	7	4	Request/Acknowledge
Reserved	7	3-1	Reserved
ER	8-9	All	Explicit Cell Rate
CCR	10-11	All	Current Cell Rate
MCR	12-13	All	Minimum Cell Rate
QL	14-17	All	Queue Length
SN	18-21	All	Sequence Number
Reserved	22-51	All	Identifier Port/VPI/VCI/PDUid
Reserved	52	8-3	Reserved
CRC-10	52 53	2-1 All	CRC-10

Table 1 Fields and their position in RM-cells [18]

When the RM cell arrives at an active switch, the TAP searches the solicited PDU and, if it is still in DMTE, then the PDU is retransmitted, as long as the idle time for the connection is sufficient.

When a NACK (RM cell) arrives at a non-active switch this only processes the RM cell and resends it to its neighboring switch in the direction of the closest upstream active switch. The non-active switches do not have DMTE to retrieve PDUs and their function is only to send (or resend) PDUs forward to their destination and also send NACKs (RM cells) backwards to the active nodes.

To conclude this point we emphasize that TAP cannot offer complete reliability, but assures and recovers an important number of PDUs that otherwise would be lost by congestion. The mechanism also guarantees that there are no end-to-end retransmissions but between active peer nodes. The retransmission mechanism generates unordered PDUs at the receiver. The protocol offers two kind of service. The first one named SEQ (sequential) sorts the PDUs and when it detects a sequence failure, assumes that the PDU is lost and leaves the retrieval to protocols of upper layers (i.e. TCP). The second type of service is the unordered (connectionless) and does not do any kind of sorting. These two services are offered by the proposed EAAL-5.

Fig. 2 plots the TAP architecture including the DMTE memory and the agents. **Fig. 2** shows the prototype of TAP with two active switches. The trusted sources generate their flow that arrives at the DMTE and which is processed to the output buffer that multiplexes the cells to the second active switch.

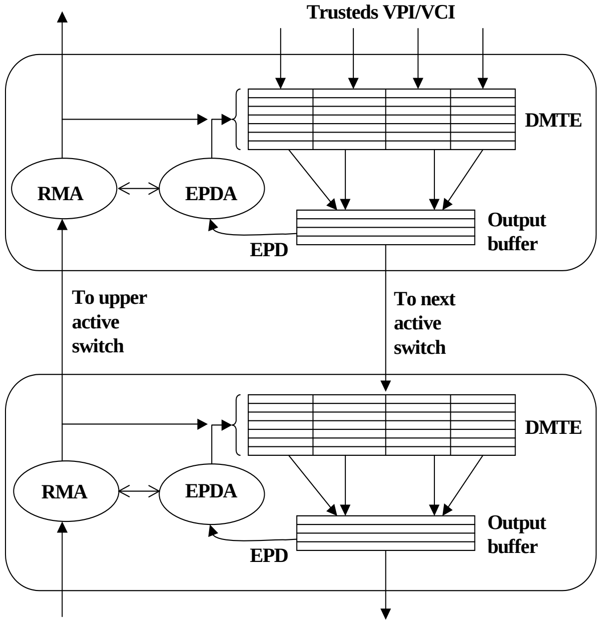

Fig. 2 TAP architecture

3 DMTE (Dynamic Memory of Trusted EAAL-5 PDUs)

The architecture of the ATM switch is similar to an output buffered switch with VC-merging capabilities. The DMTE is the key module. It behaves as a common shared memory and VC-merge buffer at the same time.

The main function of this memory is to store temporally the PDUs in the active switch after they have been transmitted to the output buffer, so that they can be requested in retransmissions. The TAP protocol keeps a copy of each PDU that arrives at the active switch (while VC merging is performed). Several PDUs are stored for each privileged connection. When a complete PDU arrives at the DMTE it is "copied" completely into the output buffer. Round Robin mechanisms are applied to guarantee fairness at the output buffer. The PDU remains in the DMTE until free space is needed for storing a new PDU.

Due to the big size of PDU-AAL-5 (up to 65,535 bytes), and the potential high number of connections (VPI/VCI), the size of the DMTE may be excessive. This is the reason why we limit the number of stored PDUs for each connection, and furthermore, only support a reduced number of privileged VCIs with trusted transfers.

The TAP accesses the DMTE through an index consisting of the port number, the PDUid (which corresponds to the UU and CPI fields in AAL5) and the VPI/VCI, and we have implemented different mechanisms to optimize the management and storage of PDUs.

While a PDU from the DMTE is being retransmitted, if there is an incoming PDU with the same VPI/VCI and no free space is available in the DMTE the incoming PDU is discarded. It is similar to a loss caused by the lack of VC-merge buffers.

4 Agents in active nodes

All the networks have a great variety of hardware elements (switches, routers, bridges, brouters, hubs, end systems, etc.) with well-known functions (switching, routing, bridging, congestion and flow control, QoS guarantee, running of applications, etc.). The current networks are communication channels that transfer packets into end systems using the above cited hardware. But also there is new research to provide the hardware elements with high performance by using technical software. This enhances the network with active characteristics (active networks) since the hardware elements compute, change and operate the packets and they can also transfer and propagate code. Therefore, an active network is a programmable network that allows code to be loaded dynamically into network nodes at run-time. The literature on active networks studies several mechanisms to obtain advantage from active nodes. Paper [13] discusses two proposals to build active networks: the idea of programmable switches and the concept of capsules. It also presents some of the most interesting architecture for active networks such as ANTS, ARM, SRM, Switchware, Active Bridge, etc. However, the proposals are insufficient for ATM networks and reference [14] is an example of this recent research in ATM. The paper discusses mobile software agents used to implement robust operation and maintenance functions in ATM networks. The agents have a role similar to that of OAM cells in the ATM standards; they are transmitted between control entities at regular intervals using predefined resources. The difference between the mobile agents and the OAM cells is that they can contain code as well as state information.

The ATM performance management is one of the most challenging network management problems, and article [17] describes the application of software agent technology to this problem and proposes a new architecture and supporting technology for network management. For example, this work describes how the network may be partitioned so that each agent is responsible for managing VPCs on a single link. In congested networks the agents then all locally redistribute the capacity as appropriate.

Also, a delegated agent may be used to execute tests that diagnose a problem in a device and then reconfigure it. For some problems, the recovery routines may involve co-ordination and even delegation of additional agents. Papers [19,20] describe some aspects in distributed management by delegation, and also, several delegated agents for management functions of the network elements in order to automate the monitoring, analysis, and control of these devices.

The QoS adaptation is another field of ATM technology for which the agents can offer attractive features. Paper [21] suggests the use of QoS monitoring agents in the ATM switches as well as in the end systems. This interesting article reviews the ATM service classes, the traffic parameters and the QoS parameters and proposes agents for QoS adaptation and monitoring, in a way that allows automatic recovery, if possible, from all the QoS violations.

There is no consensus on deciding when a network is active. There are two great tendencies: a network is active if it incorporates active nodes with the capacity to execute a user's program, or if it implements mechanisms of code propagation. The TAP architecture is active in both trends, because it provides active nodes at strategic points that implement an active protocol to allow user's code to be loaded dynamically into network nodes at run-time. Also that provides support for code propagation in the network thanks to RM cells.

The ATM switch of our model network is an output buffered switch that just reads VPI/VCI information of arriving cells and forwards them to the corresponding output port. But we equipped this switch with active hardware and software techniques to achieve our objectives.The TAP architecture uses two agents to perform the following functions. The Early Packet Discard Agent (EPDA) controls congestion based in EPD. This agent monitors the output buffer and when the occupancy is above the threshold, discards any new incoming PDUs (packets). The last EAAL5 cell contains the VPI and VCI in the header and the PDUid in the trailer of the EAAL5-CPCS-PDU. The complete PDU is discarded as in EPD but the information about the VPI, VCI and PDUid is used to generate a request for the retransmission of this PDU. If the requested PDU is still in the local DMTE it may be recovered and resent to the output buffer. Otherwise, the requests must be forwarded to the upstream active switch.

The EPDA is coordinated with the Resource Management Agent (RMA) to request retransmission of the PDU to the peer active switches. The function of the RMA agent is to generate native RM cells that are transmitted backwards to the upstream active switch. Non-active switches will recognize the RM cells as TAP RM cells and will not take any action upon them; they will simply forward the RM cells.

When an RMA agent receives an RM cell it takes on the task of looking for the requested PDU in the DMTE memory using /Port/VPI/VCI/PDUid as the index. If the PDU is still there it means that the connection cell flow presents an idle period and the PDU may be recovered. We should recall that PDUids are assigned end-to-end for each VCC and there is not any change for misinterpretation of the requested PDU. If the cell flow of the connection is dense (very short idle periods between successive PDUs) then the new incoming PDUs will use the DMTE and the “old” PDUs will be removed. We are currently working on the use of RM cells as a transport mechanism to carry out code propagation between active nodes. This code contains instructions to optimize the retransmision of PDUs in multipoint connections. The RMA agents utilize these instructions to inspect the distribution tree at width providing better goodput in retransmissions.

Another important point is that RMAs do not perform a protocol in the classical way; that is, there is only one opportunity to recover a PDU. No sliding windows or timeout retransmissions are used in this proposal.

5 Performance Evaluation

Previous work [22] has presented and demonstrated the good behavior of RAP protocol over TAP architecture. We have simulated several software techniques to introduce active characteristics in switches. These mechanisms control and manage the privileged VCI and we will also offer an active mechanism to retrieve PDUs querying neighboring active switches and to search optimized paths when a PDU is retransmitted.

The simulation allows us to define the congestion probability in transmitters, receivers and each ATM switch. When a node is undergoing congestion, it then requests the retransmission of the corresponding PDU. We use negative acknowledgements (NACK) to demand the PDU.

The simulation also permits the user to introduce variable values such as ON/OFF traffic source parameters, the number of receivers or the number of non-active switches.

In our simulation to analyse ATM cell loss we have used ON/OFF (bursty) traffic sources. The ON/OFF model [23,24] is used to characterise ATM traffic per unidirectional connection. **Fig. 3** shows this model as a source which either actively sends (ON state) CSCP-PDU-AAL-5 data for some time t_{on} at a traffic rate R or PCR (Peak Cell Rate) or is silent (OFF state) producing no cells for some time t_{off}.

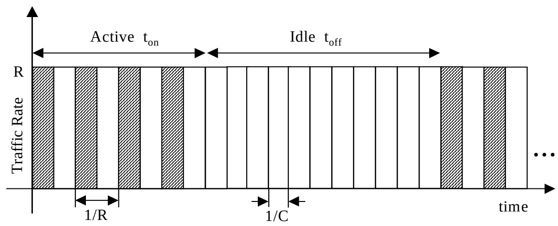

Fig. 3 Cell pattern for a single ON/OFF source

Table 2 shows the maximum and minimum source traffic descriptors used in our simulation. We utilize a process that switches between an idle (silent) state, and the active state (sojourn time) which produces an average fixed rate of cells (between 64 Kbits/s to 25 Mbits/s) grouped in PDUs of 1,500 bytes. During the ON states this process generates cells at a cell arrival rate R.

Source traffic descriptor	Parameter	Minimun	Maximum
Bandwidth Source	BS	64 kbit./s.	25 Mbit./s.
Cell arrival rate	R or PCR	167 cells/s.	65,105 c./s.
Cell inter-arrival time	1/R	6 ms.	15 μs.
Bandwidth link	BL	155.52 Mbps.	622 Mbit/s.
Cell slot rate	C or CSR	353,208 cell/s.	1,412,648 cell/s.
Service time per cell	1/C	2.83 μs.	0.70 μs.
Active time period	t_{on}	0.96 s.	1 s.
Mean number of cells in an active state	C_{on}	160 cells	65,105 cells
Time in idle state	t_{off}	1.69 s.	2 s.
Mean number of empty slots in idle states	C_{off}	596,921 cell slots	2,825,296 cell slots

Table 2 Source traffic descriptors ON/OFF

Also periodically the source generates empty time slots. We use in all examples a C or CSR (Cell Slot Rate) of C=353,208 cell/s since our network model uses 155.52 Mbit/s links. When the cell arrival rate R is less than the cell slot rate C, there are empty slots during the active states as we can see in **Fig. 3**.
The cell inter-arrival time 1/R is the unit of time for the ON state, and the mean duration in the active state is,

$$t_{on} = \left(\frac{1}{R}\right) * C_{on},$$

Also, the mean duration in the silent or idle state is,

$$t_{off} = \left(\frac{1}{C}\right) * C_{off},$$

Empirical studies [23] demonstrate that $t_{on} = 0.96$ s. and $t_{off} = 1.69$ s. and we use these values in the simulation, although we have used other values to analyze its effect over TAP. We use these and other formulae to implement the sources of simulations.
We now report results from the simulation of the TAP protocol. This section shows several scenarios which we have used in the simulation to analyse the performance of TAP. Note that we have varied some of these parameters to analyse the behaviour of the TAP when it changes the scenario and the source traffic descriptors as we show in this section.
Fig. 4 shows a basic network configuration consisting of 3 active ATM switches. **Fig. 4** plots the flow of PDUs, and we can see the DMTE, requests of retransmissions and other parameters.

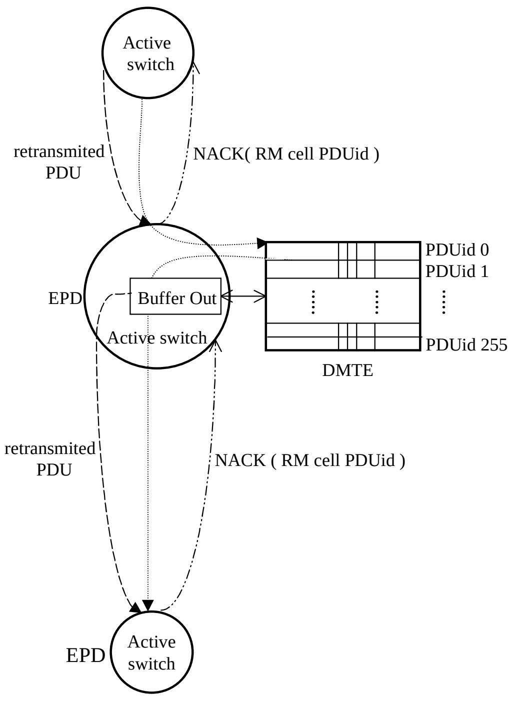

Fig. 4 Simulation network for trivial case of TAP

Common parameter values for the majority of simulations are the following:

- *Source traffic: 64 Kbps. (PCR=167 cells/sec.).*
- *PDUs AAL5 size: 1,500 bytes (32 cells/PDU).*
- t_{on}*=0.96 s.;* t_{off}*=1.69 s.*
- *Each of the links is full duplex with a bandwidth capacity of 155.52 Mbps.*
- C_{on}*=PCR*0.96=160 cells;* C_{off}*=CSR* 1.69= 596,921 cell slots.*

Fig. 5 shows the results of varying PCR between 60 and 2,000 cells per second. As we can see, when the arrival rate is low, the number of retrieved PDUs increases. We can see now the number of NACKs not sent (not retransmitted PDUs) is greater when the PCR value increases. In this way, the network is not over-charged with useless retransmissions.

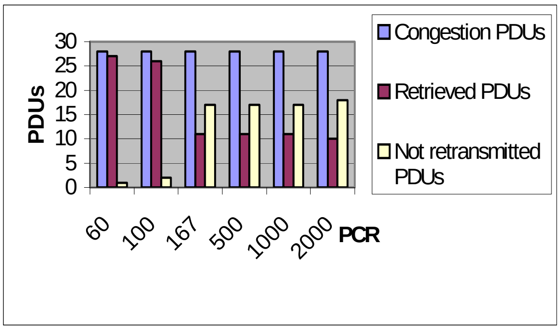

Fig. 5 Effect of PCR variation

In this simulation we fixed the data source that transmits 750 Kbytes; congestion probability = 10^{-3}. For PCR=167 cells/s.; t_{on}=0.96 s.; and t_{off}=1.69 s. and total PDUs discarded by congestion 56; retrieved PDU via TAP 11, and 17 PDUs are not requested. When PCR reaches 60 cells/s. the performance is optimized (27 retrieved PDUs out of 28) since all the lost PDUs are retrieved and there are no DMTE failures (all the requested PDUs are in the DMTE).
Another scenario consists of *1* source node, *1* active ATM switch, *n* non-active switches and *1* destination node. When a NACK arrives at a non-active switch, this also transfers the RM cell to the next switch. When the RM arrives at the active switch this uses the DMTE to retransmit the requested PDU. This scenario is the same as above, only the number of non-active switches varies. In this configuration we have simulated the protocol with several non-active switches and the results obtained show no changes. Only the delay in transmissions varies due to propagation times, but the index of retrieved PDUs is maintained as we have already shown.
Fig. 6 presents a point-multipoint configuration consisting of *1* source node, *1* active ATM switch, *n* non-active switches and *n* destination nodes. This is equal to the above basic scenario, only the number of destination nodes in multipoint connections varies. At

present we are working to achieve multipoint connections to TAP. If we consider the above results we can see intuitively that the total delay will change. Also the amount of DMTE memory required increases in active switches to manage the VPI/VCI of n connections. **Fig. 6** illustrates an example with 3 receivers (D1, D2 and D3) where D1 and D3 request a retransmission that is sent to the active node through the n non-active switches.

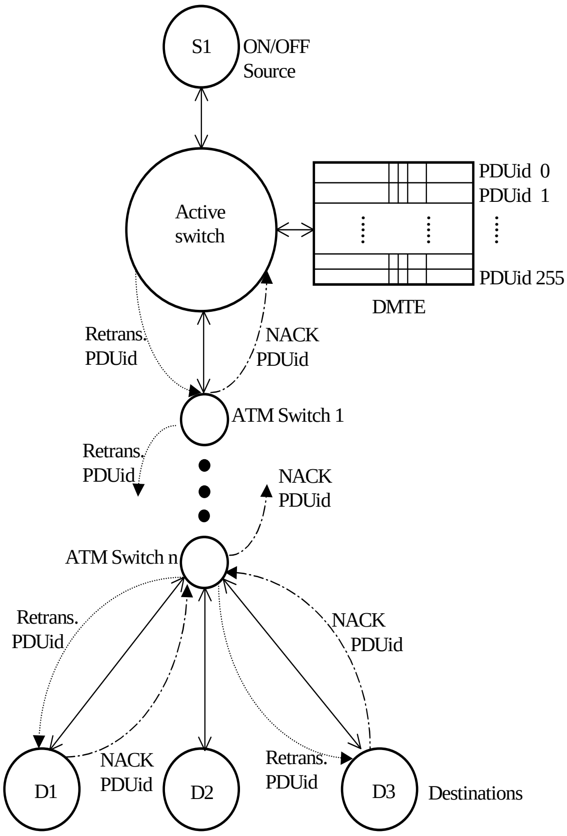

Fig. 6 VPN with point-to-multipoint transfers

However, we shall now describe several aspects on which we are working to achieve better *goodput*.

We will consider other source traffic descriptors such as SCR (Sustainable Cell Rate) and MBS (Maximum Burst Size). With these parameters we can characterize the traffic better. Also, like most applications used in the TCP protocol for transmission data in frame based structures, we are working to implement the Guaranteed Frame Rate (GFR) [25] service class to provide a minimum service guarantee to UBR, VBR and ABR services. To support GFR we will simulate sources with a Minimum Cell Rate (MCR) guarantee for a given MBS and Maximum Frame Size (MFS). TAP will offer guarantee with the GFR service class that is able to distinguish eligible and non-eligible frames and also discards cells properly. We are currently working to enhance the architecture including other intelligent agents to characterize the traffic and their class of service.

6 Summary

In this paper we have presented TAP as the architecture for an active protocol that can take advantage of suitably equipped active ATM switches. TAP manages a set of privileged VCI to improve trusted connections when the switches are congested. To achieve these active characteristics we use an active ATM switch with DMTE, a dynamic memory that temporarily stores PDUs of each privileged VCI. We have verified that it is possible to retrieve an important number of PDUs only with DMTE and a reasonable additional complexity of the active switches supported by two software agents. The retransmission mechanism is based on ARQ with NACK that generates RM cells to request PDUs. Our simulations demonstrate that the intuitive idea of taking advantage of silent states in ON/OFF sources is true. Thus we can achieve better performance and QoS in ATM networks.

We are also working to simulate multipoint-to-multipoint connections to analyze the behavior of TAP in all types of scenarios.

7 Literature

[1] R. Steinmetz, and L. C. Wolf, "Quality of Service: Where are We ?," *Fifth International Workshop on Quality of Service IWQOS'97*, pp.211-221, 1997.

[2] Recommendation I.361, "B-ISDN ATM Layer Specification," *ITU-T*, 11/1995.

[3] Recommendation I.363.1, "B-ISDN ATM Adaptation Layer Specification, Type 1," *ITU-T*, 08/1996.

[4] Recommendation I.363.5, "B-ISDN ATM Adaptation Layer Specification, Type 5," *ITU-T*, 08/1996.

[5] Recommendation I.371.1, "Traffic Control and Congestion Control in B-ISDN," *ITU-T*, 06/1997.

[6] Stephan Block, Ken Chen, Philippe Godlewski, and Ahmed Serhrouchni, "Design and Implementation of a Transport Layer Protocol for Reliable Multicast Communication," *Université Paris, http://www.enst.fr/~block/srmtp*, 1998.

[7] Jörg Nonnenmacher, M. Lacher, M. Jung, E. Biersack, and G. Carle, "How bad is Reliable Multicast without Local Recovery?," *INFOCOM'98. Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies, Proceedings IEEE,* Vol. 3 pp. 972-979, 1998.

[8] Dan Rubenstein, Sneha Kasera, Don Towsley, and Jim Kurose, "Improving Reliable Multicast

Using Active Parity Encoding Services (APES)", *Technical Report 98-79, Department of Computer Science, University of Massachusetts*, July, 1998.

[9] Luigi Rizzo, "On the feasibility of software FEC," *http://www.iet.unipi.it/~luigi*, Universita di Pisa, 1997.

[10] Ernst W. Biersack, "Performance Evaluation of Forward Error Correction in an ATM Environment," *IEEE Journal on Selected Areas in Comm*., vol. 11, No. 4, pp. 631-640, May. 1993.

[11] H. Esaki, G. Carle, T. Dwight, A. Guha, K. Tsunoda, and K. Kanai, "Proposal for Specification of FEC-SSCS for AAL Type 5," *Contribution ATMF/95-0326 R2, ATM Forum Technical Committee*, October 1995.

[12] Maurizio Casoni and Jonathan S. Turner, "On the Performance of Early Packet Discard," *IEEE Journal on Selected Areas in Communications*, Vol. 15, no 5, pp. 892-902, Jun. 1997.

[13] David L. Tennenhouse, Jonathan M. Smith, W. David Sincoskie, David J. Wetherall, and Gary J. Minden, "A Survey of Active Network Research," *IEEE Communications Magazine*, pp.80-86, January 1997.

[14] David A. Halls and Sean G. Rooney, "Controlling the Tempest: Adaptive Management in Advanced ATM Control Architecture," *IEEE Journal on Selected Areas in Communications*, Vol. 16, Nº 3, pp. 414-423, April 1998.

[15] G. Pujolle and D. Gaïti, "ATM Flow Control Schemes Through a Multi-Agent System," *International Conference on Information Engineering'93, Proceedings of IEEE*, Volume 1, pp. 455-459, 1993.

[16] Jiann-Liang Chen, Ying-Ping Yu, and Shinfeng Lin, "Exploiting Multi-Agent Scheme for Traffic Management in ATM Networks," *Intelligent Control (ISIC), Proceedings, IEEE*, pp. 594-599, September 1998.

[17] Jim Hardwicke and Rob Davison, "Software Agents for ATM Performance Management," *Networks Operations and Management Simposium NOMS 98, IEEE*, Volume 2, pp. 313-321, 1998.

[18] The ATM Forum Technical Committee. Traffic Management Specification Version 4.0 af-tm-0056.000, April 1996.

[19] German Goldszmidt, and Yechiam Yemini, "Delegated Agents for Network Management," *IEEE Communications Magazine*, Volume 36, 3, pp. 66-70, March 1998.

[20] German Goldszmidt, and Yechiam Yemini, "Distributed Management by Delegation," *IEEE*, pp. 333-340, 1995.

[21] F. Nait-Abdesselam, N. Agoulmine, and A. Kasiolas, "Agents Based Approach for QoS Adaptation In Distributed Multimedia Applications over ATM Networks," *IEEE International Conference on ATM, ACATM-98*, pp. 319-326, 1998.

[22] José Luis González-Sánchez and Jordi Domingo-Pascual, "RAP: Protocol for Reliable Transfers in ATM Networks with Active Switches," *Technical report http://www.ac.upc.es/recerca/reports/INDEX1999DAC.html*, November 1.999.

[23] J. M. Pitss, "Introduction to ATM. Design and Performance," *Ed. Wiley*, 1997.

[24] A.L. Roginsky, L.A. Tomek and K.J. Christensen, " Analysis of ATM Cell Loss for Systems with on/off Traffic Sources," *IEE Proc. Commun.* Vol. 144. No. 3, pp. 129-134, June 1997.

[25] Norbert Vicari and Robert Schedel, "Performance of the GFR-Service with Constant Available Bandwidth," *INFOCOM'99 Proceedings, IEEE*, pp. 567-574 vol.2, 1999.