AcTMs (Active ATM Switches) with TAP (Trusted and Active PDU Transfers) in a Multiagent Architecture to Better the Chaotic Nature of TCP Congestion Control[1]

José Luis González-Sánchez[1], Jordi Domingo-Pascual[2], and João Chambel Vieira[3]

[1] University of Extremadura. Escuela Politécnica de Cáceres Avda, Universidad S/N. (10.071) Cáceres (Spain)
jlgs@unex.es
[2] Polytechnic University of Catalunya, Campus Nord, Modul D6, Jordi Girona 1-3 (08034) Barcelona (Spain)
jordi.domingo@ac.upc.es
[3] University Moderna of Portugal, Polo de Beja Rua Marquês de Pombal, 1 (7800-067) Beja (Portugal)
jccv@umoderna.pt

Abstract. TAP (Trusted and Active PDU transfers) is a distributed architecture and a protocol for ATM networks that provides assured transfers to a set of privileged VPI/VCI. Our AcTMs (Active ATM switch) model supports the trusted protocol. This research also offers an attractive solution to the chaotic nature of TCP Congestion Control. Several simulations demonstrate the effectiveness of the mechanism that recovers the congested PDU locally at the congested switches with better end-to-end goodput in the network. Also, the senders are alleviated of NACK and end-to-end retransmissions.

1 Introduction

The ATM technology is characterized by its good performance with the different traffic classes and by its negotiation capacity of the QoS (Quality of Service) [1] parameters. Congestion causes the most common type of errors, and it is here that our work is intended to offer guaranteed transfers through our TAP (Trusted and Active Protocol) architecture. TAP adopts ARQ (Automatic Repeat Request) with NACK (Negative Acknowledgement) using RM (Resource Management) cells to alleviate the negative effect of implosion. The intermediate active nodes are responsible for local retransmissions to avoid end-to-end retransmissions. We have implemented a modification of EPD (Early Packet Discard) as a means of congestion control that we have called EPDR (Early Packet Discard and Relay) in order to alleviate the effect of congestion and PDU fragmentation. Currently, congestion control is delegated to

[1] This work is sponsored in part by the Regional Government of Extremadura under Grant No. 2PR03A090 and by the CICYT under Grant No. TIC2002-04531-C04 Advanced Mobile Services (SAM).

P. Lorenz and P. Dini (Eds.): ICN 2005, LNCS 3421, pp. 266 – 274, 2005.
© Springer-Verlag Berlin Heidelberg 2005

protocols that solve it with end-to-end retransmissions such as TCP. This is an easy technique to implement at high speeds by simplifying the switches and routers, but the whole network is overloaded with the retransmissions and this does not offer protection against egoist sources.

At present, the ATM networks are used as a technology to support all kinds of traffic, with predominance of TCP/IP protocols. Therefore we present the advantages that our mechanism of congestion retrievals can offer, not only for the native ATM traffic, but also for the traffic generated by TCP/IP sources. A considerable amount of research has intended to integrate two technologies as different as ATM and TCP/IP; however, their integration offers [2] poor results in the behavior of TCP throughput over ATM. While ATM is a connection-oriented technology of switched cells of 53 octets and uniform size, TCP and IP are based on routing mechanisms of segments and datagrams of variable size. These characteristics cause a very negative effect on the throughput when the TCP segments cross ATM switches with buffer size less than the window size of TCP. This causes loss of cells and retransmissions due to timeout. Moreover, the loss of only one cell causes the loss of a TCP segment at the receiver of the communication that will request the retransmission to the source, which must resend the whole segment and not only the lost cell.

Firstly, we shall comment on the general characteristics of TCP. Section 3 presents the TCP characteristics over ATM and the next section propose the use of TAP in an IPoverATM scenario and the evaluation of TAP performance. Our conclusions are presented at the end of this paper.

2 TCP Congestion Control Can Be Improved

The TCP protocol is a set of algorithms that sends packets to the network without any previous reservation, but can react if any event appears. Within this set of algorithms, the Congestion Control algorithm and the Loss Segment Retrieval algorithm are the most important.

A source TCP fixes the amount of data to be sent by using the CWND window, and transmits a window of segments for each RTT. The TCP adjusts the size of this window depending on the conditions of the network. Thus, the size of CWND increases to twice the segments for each ACK received in Slow Start algorithm, and increases by 1/CWND for each ACK received in the Congestion Avoidance algorithm. CWND increases exponentially while the size is less than SSTHRESH (using the Slow Start algorithm that progressively increases the number of segments (1, 2, 4...) when the ACKs are received). When the size of CWND is equal to the SSTHRESH, the congestion control of Congestion Avoidance works. Thus, the CWND window increases linearly by 1/CWND for each ACK.

The Slow Start algorithm is used by TCP to check the unknown capacity of the network and the amount of segments that it can support without congestion. When congestion is imminent, TCP passes the control to Congestion Avoidance which changes to a lineal increase of CWND until the congestion is detected.

We should point out that the TAP protocol solves the problems of loss that affect the decrease in size of the congestion window and also the subsequent retransmission of end-to-end losses. Thus, the source will not be obliged to reduce and to adjust its

rate of transmission all the time, and also, when congestions appear, these are solved locally in the affected nodes.

Equation (1) calculates the transmitted bandwidth (BW) and expresses the throughput in the network and calculates the performance of TCP after all previous simplifications. *MSS* is the maximum segment size of TCP; *K* is a constant term and we can estimate the random packets lost with a *P* constant probability assuming that the link delivers *1/P* consecutive packets. Paper [3] presents other references with several approximations to the value of the *K* constant regardless of its value that is always less than 1.

$$BW = \frac{MSS}{RTT}\frac{K}{\sqrt{P}} \tag{1}$$

We can reorganize (1) and, considering *RTT* and *MSS* as constants, and *W* as the window size used by TCP, we will obtain the average loss rate *P* in equation (2),

$$P = \frac{0{,}75}{W^2} \tag{2}$$

Equation (2) can be understood as if the network discards a percentage of segments independently of actions that have been performed by the source. So, this describes how the source can react.

The loss probability *P* determinates the throughput of the TCP source, as the (2) previous expression intuitively indicates. When the loss probability increases, the throughput decreases logarithmically to achieve a lineal evolution. The negative logarithmic slope represents the fall of the throughput in the network when this is experiencing loss of packets. The problem is that TCP duplicates the intervals of retransmission times between successive loss of packets.

3 TCP over ATM

The research in the throughput evaluation of TCP over ATM is divided into three main groups [4]: 1) those research papers studying the dynamism of TCP; 2) those analyzing the throughput of ATM; and 3) those observing the interaction between TCP windows and the mechanisms of congestion control of the ATM layer. Although the throughput evaluation of TCP over ATM has been the objective of several research papers, the proposals only solve particular problems such as the fragmentation of TCP, the buffers required, the interaction between congestion schemes of TCP and ATM, and the degradation of TCP. There are a lack of proposals to solve all or even some of these problems. Our research has looked into these aspects and offers a MAS (MultiAgent System), optimizing the goodput with an improvement of entry queues. Moreover, an accurate policy of buffer management is used through the delegation of activities in agents that constitute the MAS.

Reference [2] presents the study of congestion in TCP networks over ATM and shows how the TCP throughput also falls when the discard of cells at ATM switches begins. The low throughput obtained is due to the waste of the bandwidth at congested links that transfer packets of corrupted cells; that is, packets with some dropped cells. Other research papers have demonstrated that TCP over UBR, with

EPD suffers a considerable degradation of fairness, and also need a big buffer size, even if there are few connections.

The literature [5, 6] also describes other ways to avoid the degradation of throughput at TCP sources over UBR. In order to do this, the discard of ATM cells is disconnected when there is congestion. So, the timeouts of TCP are avoided although they are the main cause of fall at TCP throughput, and also the periods of congestion are reduced, avoiding the big delay experienced with the fast retransmission algorithm of TCP before the source receives the duplicate ACK.

With all these differences, and as ATM is a protocol placed under the TCP transport layer, solutions are required to solve the throughput problems due to the integration of these different technologies. These solutions propose changes at ATM switches inside the network; or the implementation of new extensions for TCP; or perhaps, specialized protocols for nodes placed at the limits of the ATM network and the TCP network. Our TAP protocol solves these problems by working inside the network, with hardware (active ATM switches) and also software (multi-agent system with TAP protocol) mechanisms. All this configures the whole TAP architecture.

4 Advantages of TAP and Evaluation of TAP Performance

We propose EAAL-5 layer (Extended AAL-5), specifically designed for data communications over ATM. At TCP over ATM, the datagrams are transferred to the data field (payload) of EAAL-5, as we can see in Fig. 1 which shows the stack protocols of TCP over ATM. The TAP architecture is active, because it provides active nodes at strategic points that implement an active protocol to allow the user's code to be loaded dynamically into network nodes at run-time. TAP also provides support for code propagation in the network thanks to the RM cells.

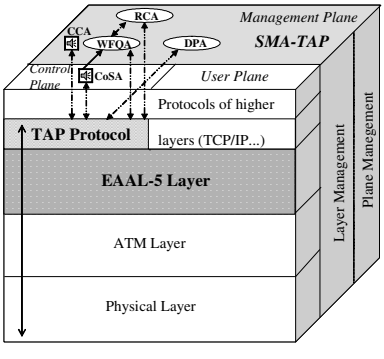

Fig. 1. TAP-MAS architecture

In Fig. 1 the TAP-MAS architecture is shown, focusing on the way to incorporate the modifications we propose for the ATM architecture in our proposals. Notice that the position of the MAS is part of the management plane because it provides new mechanisms to manage the ATM resources.

In [7] we have presented the architecture based on TAP-MAS, constituted by software agents and equipped with a DMTE dynamic memory. We have also implemented the PQWFQ (PDU Queues PDU based on Weighted Fair Queuing) algorithm to apply fairness at sources. Also, the EPDR algorithm manages the buffer congestion and avoids PDU fragmentation. The general motivation of this work is to find solutions to alleviate this negative problem of end-to-end retransmissions.

One of the most interesting aspects of the AcTMs is the implementation of the EPDR (Early PDU Discard and Relay) algorithm on the switch buffer. We propose this algorithm to retransmit the PDU between adjacent AcTMs and to avoid the existence of fragmented PDU in the network. In order to reduce the fragmentation, the algorithm controls the size of the input buffer threshold in each switch. Fig. 2 shows the input buffer of the AcTMs with the sizes controlled by the EPDR algorithm.

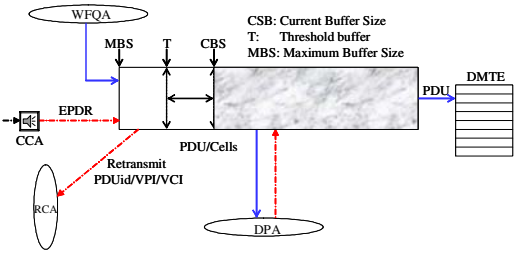

Fig. 2. Buffer of the AcTMs with EPDR

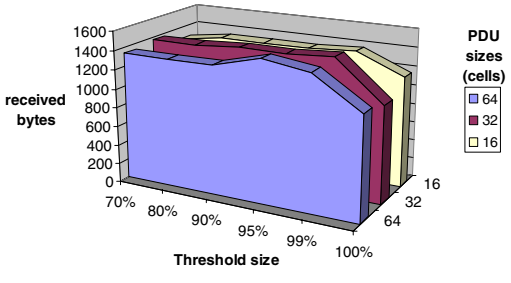

Fig. 3. Correctly received bytes in receiver R-EAAL-5

Fig. 3 shows the effect of the size of the threshold value of the EPDR algorithm. This study has been made of different threshold values and different PDU sizes. We have observed that the EAAL traffic generated by the ATM privileged sources is better than the AAL traffic of ATM sources without GoS. This is because less EAAL PDU are lost than AAL, and that avoids the retransmissions of higher layer protocols. The number of EAAL-5 PDU arriving rightly to the R-EAAL-5 sink, depending on the threshold size and the PDU size is shown. When the PDU size is minor, more PDU are received. The fluctuations are due to the in-flight cells at the links.

It is interesting to analyze the obtained results when the EPDR control does not act on the buffer. The Fig. 4 is obtained with the threshold at 100%. The EAAL traffic

shown is significantly better than AAL traffic. The AcTMs-9 switch discards all the incorrect PDU coming from the S-EAAL-5 source. This is one of the tasks of the CoSA agent. While the corrupted AAL PDU go on transiting in the network making unnecessary use of resources, the correct PDU corresponding to privileged EAAL connections with GoS are discarded by the next active switch.

Moreover, the number of correct received PDU with EAAL traffic is higher than the received ones with AAL traffic. This is not due to the retransmissions because with the 100 % threshold an AcTMs switch cannot retransmit requests.

The previous figures have shown how the threshold value does not affect the number of PDU that arrive at a sink, except for 100% threshold. However, we know that congestions are produced and cells are lost in some PDU. For this reasons we now study the influence of the EPDR threshold in the appearance of the congestions and retransmit request of EAAL-5 PDU. Fig. 5 shows the discarded cells in i.e. AcTMs-9 switch due to discarded cells by EPDR or congestions. The cells are discarded depending on the PDU size and the buffer threshold.

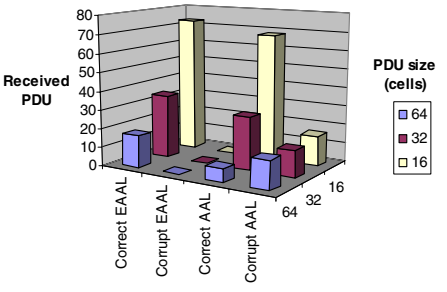

Fig. 4. Corrupt and corrects received PDU with 100% threshold

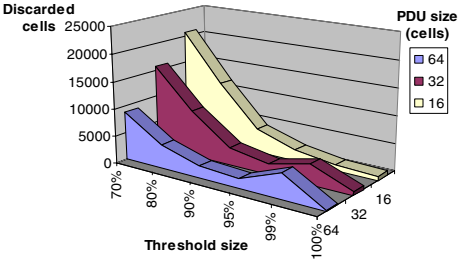

Fig. 5. Discarded cells depending on the PDU size

When the threshold per-cent increases, the number of discarded cells decreases, but only until a limited value. For PDU of 64 and 32 cells with a 99% threshold on a buffer of 1000 cells, the number of discarded cells increases. This is due to the EPDR algorithm behaviour. If the filling level of the buffer is higher than the threshold when

the first cells of a PDU arrive, the whole PDU is discarded. For this reason when a threshold level is minor, the probability of discarding a PDU is higher.

When the difference of size between the threshold and the buffer size is very close to the PDU size, it increases the probability of discarding a cell because it does not fit in the buffer. This is corroborated because fewer cells are discarded (with a threshold close to 99%) as the PDU size decreases.

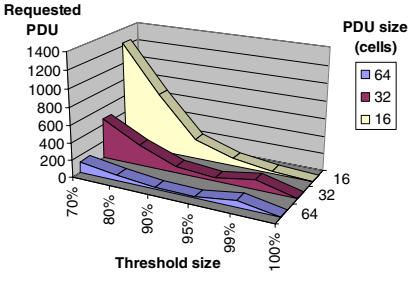

Fig. 6 Retransmitted PDU by the AcTMS 4 to the AcTMs 6

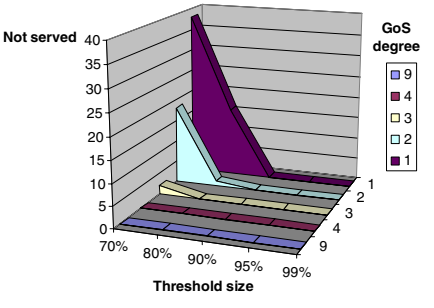

Fig. 7. Non-served retransmissions

Fig. 6 shows the retransmissions requested by an active switch (i.e. AcTMs-6) to a previous active switch (i.e. AcTMs-4). The effect of the threshold on the PDU discarding is observed. The difference between discarding with PDU of 16, 32 or 64 cells is because there are more PDU of 16 cells than PDU of 64 cells, although the number of sending bytes is the same. It is also observed that there are more PDU of 32 cells discarded than PDU of 64 with a 99% threshold, and this is because the retransmissions of the PDU of 32 are faster due to the lower limited number of cells.

Next, the influence of the GoS grade in the goodput of the EAAL connections is studied. The GoS grade is a parameter set to privileged EAAL connections that is negotiated with the active switches in the established connections process. This parameter represents the number of EAAL-5 PDU requested to the active switches to store in their DMTE memory in order to be able to serve the retransmissions. The results of these simulations are shown in Fig. 7 where PDU of 16 cells have been used

in the AcTMs-6 switch. We studied the number of non-served retransmissions for different grades of GoS in EAAL connections depending on the threshold. Non-served retransmissions are those that cannot serve either the AcTMs-4 switch or the AcTMs-1 switch because there is not a copy of the PDU in the DMTE. This occurs when the required PDU has been substituted by another more recent one because there is not enough vacant space in the DMTE. Fig. 7 shows how the number of non-served PDU decreases quickly when the grade of GoS increases.

To manage the buffer and the input queues at each AcTMs switch we have implemented the PQWFQ (PDU Queues based on Weighted Fair Queuing) algorithm as part of the WFQ agent at TAP-MAS. This algorithm achieves a fair treatment of the PDU that arrive at AcTMs switches. We must treat the PDU of connections with GoS (Guarantee of Service) as privileged traffic.

These and other simulations demonstrate that the TAP architecture takes advantage of the AcTMs switches. We have verified that it is possible to retrieve an important number of PDU only with DMTE and a reasonable additional complexity.

5 Conclusions

In protocols of transport layer such as TCP over ATM, a packet is discarded by the network when one or several cells are lost, and the destination node requests the whole retransmission of the corrupted or lost packet. We have demonstrated through simulations the degradation experienced by the throughput of TCP. We have also studied how this falls logarithmically when the probability of loss of the ATM cells increases. The TAP protocol makes the retransmissions locally, and this decreases the loss probability and affects the throughput of TCP that alleviates the delays due to the end-to-end RTT. These simulations have demonstrated that the addition of active switches improves the throughput of a congested ATM network. Also the goodput of a privileged set of sources with GoS is increased. We conclude that the ATM network improves with the addition of the AcTMs switches.

References

1. Janusz Gozdecki, Andrzej Jajszczyk, and Rafal Stankiewicz, “Quality of Service Terminology in IP Networks,” IEEE Communications Magazine, (March 2003).
2. Romanow, A. and Floyd, S., “Dynamics of TCP traffic over ATM networks,” *IEEE JSAC*, pp. 633-641, (1995).
3. Matthew Mathis, Jeffrey Semke, Jamshid Mahdavi, and Teunis Ott, “The Macroscopic behavior of the TCP Congestion Avoidance Algorithm,” *Computer Communications Review of ACM SIGCOMM*, vol 27, n. 3, (1997).
4. K. Djemame, and M. Kara, “Proposals for a Coherent Approach to Cooperation between TCP and ATM Congestion Control Algorithms,” *Proceedings UKPEW’99,* (1999).
5. Hongqing Li, Kai-Yeung Siu, Hong-Yi Tzeng, Ikeda, C., and Suzuki, H., “A simulation study of TCP performance in ATM networks with ABR and UBR services,” *Proceedings IEEE INFOCOM’96*, pp. 1269-1276, (1996).

6. Shunsaku Nagata, Naotaka Morita, Hiromi Noguchi, and Kou Miyake, "An analysis of the impact of suspending cell discarding in TCP-over-ATM," *Procceedings IEEE INFOCOM'2000*, pp. 1147-1156, (2000).
7. José Luis González-Sánchez, Jordi Domingo-Pascual, and Alfonso Gazo Cervero, "Robust Connections for TCP Transfers Over ATM Through an Active Protocol in a Multiagent Architecture," Proceedings 10th *IEEE IEE ICT'2003*, pp. 830-836 (2003).