INFINITE QUEUE – BASED CONNECTION ADMISSION CONTROL IN ATM NETWORKS

H. Abu Zaied

Comp.Inf. Sys.Dept. , Faculty of IT, Philadelphia University, Jordan
E-mail: Hhhabz@yahoo.com , H_zaied@philadelphia.edu.jo

INTRODUCTION

The information transfer technology of today is passing through a critical curve. Since ISDN took place with commercial introduction, a lot of standards for the broadband network have been defined. It was agreed then that ATM would be the transfer mode for the future broadband ISDN. Connection Admission Control (CAC) in ATM networks depends on a set of factors such as the characteristics of traffic and the network resources. In traditional CAC the traffic characteristics must be checked and the required network resources (i.e. bandwidth) must be checked also before the judgment of admission is madeTo release the network from the user incorrect declaration of his traffic, it was accepted to measure the number of arriving cells to enable CAC to make decisions correctly. But unfortunately this policy is advantageous only for peak cell rate declaration. In addition to the previously mentioned disadvantages the wasted time spent in checking the network resources capability for handling the waiting arriving call. To avoid the mentioned disadvantages we propose a new CAC that combines both user and measurement declaration for traffic specifications. The proposed strategy will confirm the traffic specifications after receiving it in the virtual link, which is embedded in the node (in the auxiliary memory).

VIRTUAL CONNECTION ADMISSION CONTROL

So far a new arriving call must wait till the CAC checks the availability of a sufficient bandwidth to handle the waiting call at its requested Quality of Service (QoS) while maintaining the agreed of already established connection in the network. This check or inquiry action of routing tables for all the outgoing links at the source node and the next node takes a lot of time and represents a load for the processing unit. The link total channel capacity is the main limiter in accepting the arriving calls, the call must wait at the entry of the node unless it passes the bandwidth availability check. Can we make use of the spent time in checking the availability of the resources and setting up the call to be transmitted over the available link? We aim at bypassing the limit of link total channel capacity momentarily. Instead of waiting at the entry of the node it is better to make use of the waiting time (which is wasted) in preparing the call for transmitting. This idea can be realized by installing an auxiliary storing entity to represent the physical link, so the physical link is replaced by a virtual link but with ultra high capacity. This virtual link will be divided in the same manner as the physical link; the VPI's will be the same except that the VP's bandwidth of the virtual link will much greater than those of the physical link. VC's bandwidth (size) will be the same for both links but their quantities are different (we apply the path and service separation). Consequently we devised a shadow link such as the physical link with the same number of VP's but with larger bandwidths and with greater number of VC's of the same bandwidth. The proposed virtual link is hidden in the node in an auxiliary memory. The virtual link will be divided into segments; each of them is assigned to serve a specified class of traffic, so the number of these segments equals the number of traffic classes. The virtual link (auxiliary memory) will be divided among these segments. Each traffic class will be assigned a portion of memory according to its assigned bandwidth in the physical link and can be defined by the following relation: $C_{vi} = C_v \cdot C_{Ti} / C_T$, where: C_{vi} - represents the memory size for traffic class numbers *I* in Mbits; C_v - represents the total size of the auxiliary memory (virtual link) in Mbps; C_T - represents the total channel capacity for the physical link in Mbps; C_{Ti} - represents the assigned bandwidth for traffic class number *I* in Mbps.

Now we divide the physical link into a number of virtual channels (VC's). As each VC can carry only one call, so the total number of VC_i for traffic class number *i* will be the maximum allowable number of calls for this class. This number VC_i depends on the assigned bandwidth and peak bandwidth for the specified class, mathematically we can write the following equation: $VC_i = C_{Ti} / P_i$. During the network operation we will have one of the following conditions: 1) All VC_i are loaded and so we call them busy VC_{ib}; 2) All VC_i are unloaded and so we call them free VC_{if}; 3) Some of them are busy others are free.

The first case describes the congestion regime, when the link is saturated against this type of traffic. The second case represents the idle condition, when this type of traffic has a loss of flow due to a certain reason. Finally the third case which represents the normal operation condition, but this case may have two subclasses: the first one when $VC_{ib} > VC_{ib}$ and we say that the network closes to the congestion state. The second one when $VC_{ib} < VC_{if}$ and we say that the link is slightly loaded. In each node exists a routing table, which contains complete details about all the outgoing links from this node. The item in this table is the number of free VC's, VC_{if}. The link with the maximum number of free virtual channels $VC_{if\max}$ will be chosen to carry the new call.

ROUTING STRATEGY

Routing with a low degree of precision looks like the opposite picture of admission. The admitted calls have to be discharged (drafted) via a certain routing policy. As we mentioned previously the routing function is synonym to the Gate function in the transistor. The proposed routing strategy is affected by a set of factors such as the proposed bandwidth allocation, connection admission strategies and the proposed strategy for weighting the network links. In the presented strategy we are concerned with maximum number of free virtual channels among the outgoing links of the sending node only. The router will scan the routing tables in all the outgoing links from the node and consequently it will choose the link with the maximum number of free virtual channels $VC_{if\max}$ to handle the new arriving call. Actually after receiving the calls in VCAC, sorting them and accommodating them in the assigned segment for the specified traffic class they will be numbered by the same number of VP in which they will be carried later on the physical link. Thanks to the proposed bandwidth allocation (path and service separation) all VC's will have the same bandwidth size, so no need to confirm the ability of the free VC, VC_{if} for carrying the arriving call, because all VC_i have the same bandwidth size for the specified class traffic. So if we have one VC_{if} we can set the new call on it immediately. According to the proposed strategies we bypassed the CAC decision for the call to wait until the availability of sufficient bandwidth check is carried out. We also bypassed the inquiry of the successor nodes and we became concerned only with the outgoing links of the sending node and finally we bypassed the ability of the available bandwidth to carry the new call. Routing tables hidden in the node are the eyes of the routing algorithm. In these tables there are the required information for the proposed routing strategy, such as the number of free VC's, VC_{if} for each traffic class, the busy VC's, VC_{ib} to give a complete picture of the link. The number of free virtual channels is of critical importance, because it tells the draft CAC how much bandwidth is available in the considered link. The proposed strategy aims at employing the link with highest number of VC_{if}. The first gain can be yielded from this strategy is that the load is dispatched uniformly all over the network. Another advantage of this policy is the prevention of congestion occurrence while there is some room in any of the successor links. It is very easy job for each node to look for the loaded link among its links only and it will not be worry about its successors due to the virtual link embedded in the each node. The virtual link is a good solution for link capacity shortage, because it releases the call from waiting the CAC to check the availability of sufficient bandwidth and the router to carry out a local and global check through the network. The gained time was used to prepare the call for transmission on the available link. We must not forget to say that even if the call is prepared for transmission but no available links it will be rejected.

CONCLUSIONS

In this paper, we introduced a new connection admitting strategy, which aims at minimizing the call setup delay by eliminating the checks of the physical resources before admitting the arriving call. Experimental and simulation results show that the proposed Virtual Link CAC is better than the DLCP in terms of accepted and rejected calls. The reallocation mechanism built in the simulator is a multipurpose tool.

REFERENCES

[1] H. Abu Zaied.An Early Congestion Sensing Probe for ATM Networks.Proc. of International Conference on Information Technology and Sciences, ICITNS 2003,October 19th -21st , Amman, Jordan, p.110-113.

[2] H. Abu Zaied and A. Dalala'h. A New Call Admission Control Function for an ATM Backbone. Proc. of The 2000 Arab Conference on Information Technology, ACIT"2000, Oct. 31st – Nov. 2nd, Zarka, Jordan, p.88 - 93.

[3] H. Abu Zaied, Victor Besliu, A. Dalala'h and M. S. Obaidat. Dynamic Near Shortest Path routing algorithm. – The Proceedings of the 2001 International Symposium on Performance Evaluation of Computer and Telecommunication Systems, SPECTS 2001, July 15-19, Orlando, Florida, p. 415-422.