

**AC ALGORITHMS IN AQUILA QoS IP NETWORK**

Christof Brandauer<sup>1</sup>, Wojciech Burakowski<sup>2</sup>, Marek Dąbrowski<sup>2</sup>,  
Berthold Koch<sup>3</sup>, and Halina Tarasiuk<sup>2</sup>

<sup>(1)</sup>Salzburg Research, Salzburg, Austria

<sup>(2)</sup>Warsaw University of Technology, Warsaw, Poland

<sup>(3)</sup>Siemens AG, Munich, Germany

e-mail: cbrand@salzburgresearch.at, {wojtek, mdabrow5, halina}@tele.pw.edu.pl,  
Berthold.Koch@icn.siemens.de

**Abstract**

This paper introduces different AC (Admission Control) algorithms implemented in the multi-service AQUILA QoS IP network [4]. They are aimed at regulating volume of submitted traffic inside network services designed for handling real-time streaming VBR (Variable Bit Rate) and elastic (more precisely, produced by greedy TCP sources) flows. For streaming flows, the MBAC (Measurement Based AC) scheme assuming well known Hoeffding bound [2] formula for assessment sufficient link capacity for guaranteeing given packet loss ratio is involved. This approach demands aggregate mean bit rate measurements and peak bit rate declarations. For elastic flows, two alternative AC algorithms are discussed, both targeted for assuring requested TCP throughput. First one follows the Token Bucket Marking (TBM) concept while the second assumes setting of advertised receiver TCP window size, allowing us to maintain the ideal TCP behaviour (lossless packet transfer). In the paper, a short overview of each of the above mentioned AC algorithms is provided. The simulation and experimental results showing the effectiveness of particular AC algorithms are also included.

**Keywords:**

QoS IP network, admission control, streaming and elastic traffic

**1. INTRODUCTION**

The most promising solution for QoS IP network is the DiffServ architecture [1]. One of the proposals for such a network, developed inside the AQUILA IST European project [3], is an enhancement of generic DiffServ architecture by adding new functionalities for admission control and resource management as well as by defining a new set of network services.

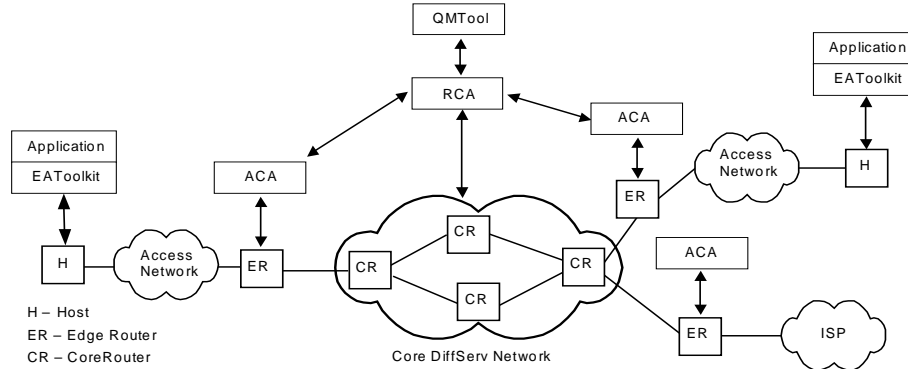


Figure 1: General architecture of AQUILA QoS IP network.

The general AQUILA architecture is depicted on Fig.1. It assumes that the AC agent (ACA) is located at each Edge Router (ER) for assuring that traffic submitted to the core network does not exceed given volume of capacity determined by allocated AC limit, being a part of total link capacity between the ER and CR (Core Router). The value of AC limit is allocated by the Resource Control Agent (RCA), situated on the top of the network, and could be changed depending on the network traffic conditions. The call handling scenario is the following. A user, who wants to make a connection, uses the end-user application toolkit (EAToolkit) for sending a request to the ACA, containing traffic contract parameters. The ACAs (on ingress and egress sides) admit this call only when capacity required for serving traffic generated by new call and calls being in progress with assumed QoS (Quality of Service) objectives is not greater than AC limit.

For now, four types of packet flows requiring QoS guarantees have been recognized as typical in Internet. They are the following: (1) streaming constant bit rate (e.g. VoIP), (2) streaming variable bit rate (e.g. video applications), (3) elastic, produced by greedy TCP or TCP-like sources (e.g. FTP), and (4) elastic, non-greedy TCP sources (e.g. home banking). In this spirit, four QoS network services (NS) have been defined and implemented in AQUILA: Premium CBR (PCBR) for traffic (1), Premium VBR (PVBR) for traffic (2), Premium Multi Media (PMM) for traffic (3), and Premium Mission Critical (PMC) for traffic (4). Each network service is optimised for handling adequate type of packet flows with guaranteed QoS. For this purpose, specific traffic handling mechanisms support given network service, including admission control algorithm. In addition, Standard Service for best effort traffic is also provided.

In this paper we present the AC algorithms associated with PVBR and PMM services. For PVBR, the MBAC scheme with Hoeffding bound formula for assessment of required link capacity satisfying given packet loss rate is involved. This approach demands aggregate mean bit rate measurements on the link and per

flow peak bit rate declarations. For PMM, two alternative AC algorithms are discussed, both aimed at assuring target TCP throughput. The first one follows the Token Bucket Marking (TBM) concept while the second assumes setting of advertised receiver TCP window size, in this way allowing us for maintaining the ideal TCP behaviour (lossless packet transfer).

The paper organisation is as follows. Section 2 gives overview of the MBAC method for PVBR service and includes numerical results showing its effectiveness. The AC algorithms for PMM service are shortly described in section 3. The numerical results illustrating their usefulness are also included in this section. Finally, section 4 concludes the paper.

## **2. MEASUREMENT BASED AC SUPPORTING PVBR SERVICE**

The PVBR service is designed for handling streaming VBR traffic with target QoS objectives defined as low packet losses and low packet delay. For example, a candidate for using this service is real-time video, like video-conference or live streaming video. For meeting the above QoS requirements, the traffic submitted inside PVBR should be served with relatively high priority and packets should be allowed for waiting in buffer by short time only. As a consequence, this leads to handle the PVBR traffic in separate way in the router by submitting it to a dedicated buffer of limited size. In addition, high priority for such a queue should be assigned in the scheduler governing access to the outgoing link.

One can find some similarities between the PVBR service in AQUILA QoS IP network and the rt-VBR service in ATM. The recognised solution for admission control in the case of rt-VBR is to use REM (Rate Envelope Multiplexing) scheme and to apply a DBAC (Declaration Based AC) approach for regulating submitted traffic. Note that DBAC assumes the user declarations referring to the peak and sustained (as an assessment of the mean) bit rates for calculating the required link capacity, expressed in the form of effective bandwidth e.g. using Lindberger formula [2]. However, this approach gives satisfactory results only when the declared sustained rate is close to the mean rate. Such method was tested at the beginning phase of AQUILA project [3]. Unfortunately, it appeared that it is rather difficult task for a user to precisely specify a priori a proper value of mean bit rate. Remark, that this value is policed and incorrect declarations could cause undesirable packet dropping. In addition, even if it is possible to make correct declarations in the case of stored video, it is not possible to be done in the case of e.g. live video. As a consequence, a user will have rather tendency to over-declare the sustained rate, leading to non-effective bandwidth utilisation. The above motivated introducing MBAC approach instead of classical DBAC. By applying effective MBAC algorithms, one can expect the following profits:

- To simplify the traffic declarations; usually it is difficult for the user to specify accurate parameters other than the peak bit rate;
- To take into account the volume of submitted traffic to the network. This should result in better network utilization (finally, leading to more accepted flows), since in some cases the carried traffic could vary from declaration;
- To capture the stochastic nature of the user traffic more accurately than it is possible with DBAC, which assumes traffic description by deterministic parameters.

The MBAC method selected for the PVBR service is the algorithm based on Hoeffding bound [6, 15]. It is relatively easy to implement since it assumes the measurements of aggregate mean bit rate only. For this purpose, a special measurement module is implemented as part of the ACA agent. The number of bytes transferred on the outgoing link is collected by polling the router in predefined fixed periods. The implemented method for estimating the mean bit rate is based on the sliding window algorithm.

The applied MBAC approach assumes that to the system of capacity  $C$  with  $N$  running flows, each with declared peak rates  $h_i$  ( $i=1, \dots, N$ ), and measured aggregate mean rate  $M$ , new flow with declared peak rate  $h_{new}$  is admitted only if:

$$h_{new} + M + \sqrt{\frac{-\ln P_{loss}}{2} \left( \sum_{i=1}^N h_i^2 + h_{new}^2 \right)} \leq C, \quad (1)$$

where  $P_{loss}$  is the target packet loss probability, e.g  $10^{-4}$ .

Fig.2a and 2b, corresponding to link capacities 10Mbps and 100Mbps respectively, show the effectiveness of the discussed MBAC method comparing to two DBAC methods, one based on effective bandwidth estimation using Lindberger formula and the second assuming a peak rate allocation. The tested traffic patterns corresponded to MPEG4 video traces with the peak and mean rates equal to 0.94 and 0.135 Mbps [16]. As it was expected, the MBAC overcomes DBAC only when the declared sustained rates are essentially far (say more than 1.5 times) from the mean rates. One can obtain better results by using per flow measurements but this is hard to get in the AQUILA architecture assuming AC in ingress and egress sides. While it can easily be done on the ingress side, per-flow measurements on the egress side require classification of flows leaving from the network and this is currently not captured in the AQUILA architecture.

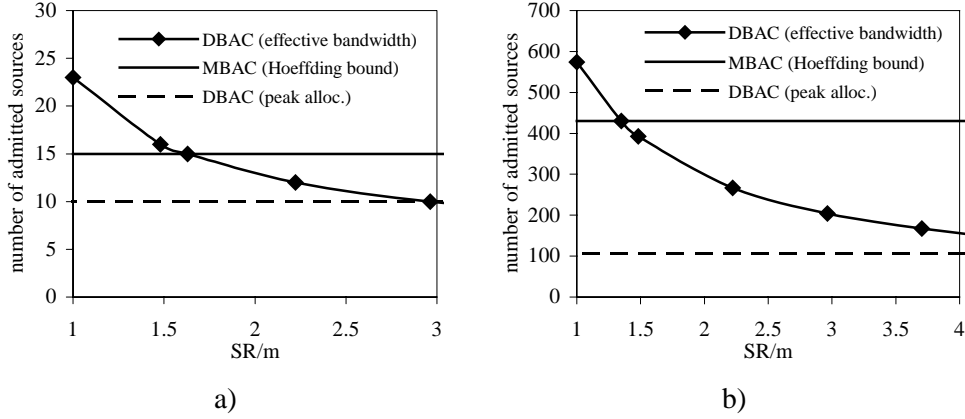


Figure 2: Comparison of efficiency of MBAC and DBAC schemes on a link with capacity a) 10Mbps and b) 100Mbps.

### 3. AC FOR PMM SERVICE

The PMM service is aimed at the efficient handling of greedy TCP flows requiring guarantees with respect to the target requested rate ( $R_{req}$ ). The example PMM application is (large) file transfer (FTP), when a user wants to transfer a file within a satisfactory time interval.

The TCP traffic control algorithms employ a sliding window based mechanism. Basically, TCP transmission is controlled through a send window which limits the maximum amount of data that may be outstanding, i.e. sent but not acknowledged. The size of the send window changes over time due to the flow and congestion control algorithms. The average rate of a TCP sender ( $R_{avg}$ ) is given by  $R_{avg} = W_{avg} / RTT_{avg}$ , where  $W_{avg}$  is the average size of the send window and  $RTT_{avg}$  is the average round-trip-time (RTT) for that TCP connection.  $W_{avg}$  depends on the packet drop probability and thus on the degree of congestion.  $RTT_{avg}$  depends on several delay components (transmission, propagation, queuing, processing). Due to the volume based control and the fact that the  $R_{avg}$  depends on varying factors that are hard to estimate, the goal of guaranteeing TCP rates is not easily achieved.

An AC scheme for TCP flows that has been investigated by many authors, e.g. in [8,14], assumes to reject a new connection by dropping SYN and ACK SYN segments in the case of network congestion. For instance, congestion can be identified when the number of waiting packets in the queue exceeds a predefined threshold value. It was shown that this approach could guarantee a fair share of link capacity between running TCP connections but without possibility of rate differentiation between the flows. A list of proposed AC algorithms for elastic

traffic and based on some declarations is presented in [15], but none of them is explicitly targeted for guaranteeing the requested rate for TCP.

For the PMM service two AC algorithms are implemented. Each of them operates per TCP flow and is of declared based type. They assume that a user, before establishing TCP connection, submits its request to the network. The traffic contract specifies the target requested bit rate ( $R_{req}$ ). Furthermore, on the basis of the  $R_{req}$  and information about RTT of the TCP connection, the user declarations are mapped into the form of single token bucket parameters, say rate ( $R$ ) and bucket size ( $BS$ ), constituting input parameters for AC decision. First of proposed AC is based on the token bucket marking (TBM) while the second assumes ideal TCP behaviour by setting appropriate value of advertised window size.

### **Token bucket marking and differential dropping**

One approach to the task of assuring TCP rates is to employ TBM mechanism at the network ingress and to differentially (depending on their marking) drop packets inside the network. We investigate here the feasibility of such an approach under the assumption that the token bucket is configured with a static parameter set, which is computed from a flow's requested rate.

It is well known that the aggressiveness of TCP flows is indirectly proportional to both  $RTT_{avg}$  and  $W_{avg}$ . Aggressiveness denotes the ability of a flow to grab bandwidth from the network. In the PMM service class where flows may request different rates and potentially have different RTTs there will thus generally be a competition between flows of different aggressiveness. The fundamental idea of an approach with TBM / differential dropping is to counteract the inequalities of TCP aggressiveness and to trim each flow to its requested rate. It is essential that a flow does not exceed that rate because the extra amount of bandwidth taken by that flow would be missed by the other concurrent flows inside that service class. The TBM is able to push-back aggressive flows and to support the weak flows by marking less/more packets as in-profile.

In the literature there is an accurate model of TCP sending behavior in a network with token bucket marking at the ingress and differential dropping in the core network [12]. This model looks very promising as it provides closed-loop formulae for the computation of the required token bucket rate/size in order to achieve a target sending rate. We assume here an under-subscribed scenario where – according to the model - it's always possible to achieve the requested rate. The model provides a theoretical solution for assuring TCP rates and we thus evaluate its practical applicability.

The model in [12] requires (amongst other parameters) knowledge of the packet drop probability for out-of-profile packets ( $p_2$ ) and the flow's average round-trip-time  $RTT_{avg}$ . The token bucket rate  $R$  that is required to achieve a desired sending rate  $R_{req}$  is computed as:

$$R = \begin{cases} R_{req} - \frac{3}{2R_{req} p_2 RTT_{avg}^2} & R_{req} \leq \frac{3W}{2RTT_{avg}} \\ \frac{4}{3} R_{req} - \frac{3}{2RTT_{avg} \sqrt{2}} \sqrt{Z + \frac{1}{p_2}} & R_{req} > \frac{3W}{2RTT_{avg}} \end{cases}, \quad (2)$$

where  $W = \sqrt{2(BS + 1/p_2)} + 2\sqrt{2BS}$ , and  $BS$  is the token bucket size.

After the computation of the token rate  $R$  the amount of resources needed for that request can be determined. For those flows, where  $R > R_{req}$ , the flow requires at least an available bandwidth of  $R$  in order to obtain  $R_{req}$ . For those flows, where  $R \leq R_{req}$ , an amount of  $R_{req}$  must be available.

On this basis we can deduce a simple admission control rule that can be considered as a special case of peak rate allocation. The resources required by a single flow are expressed by the greater value of the token rate  $R$  and the requested rate  $R_{req}$ . The inequality in (3) ensures that the bandwidth required by the aggregate stream submitted to PMM is smaller than  $\rho_{PMM}$  times the capacity  $C$  reserved for this service, where  $\rho_{PMM}$  is an over-allocation factor to keep some safety margin.

$$\sum_{i=1}^{N_{PMM}} \max(R_i, R_{req,i}) \leq \rho_{PMM} \cdot C \quad (3)$$

The number of flows in the PMM (including the new one if being admitted in this service) is denoted by  $N_{PMM}$ .

The main problem of the above sketched approach lies in the estimation of input parameters needed for the TBM, especially the average round-trip-time  $RTT_{avg}$  and the drop probability for out-profile packets  $p_2$ . We leave aside the discussion of finding a reasonable estimation for  $RTT_{avg}$ . Instead we focus on  $p_2$  because it seems impossible to find an accurate estimation for this parameter. It would require a constant drop probability, independent of the number of congested routers in the flow's path and the level of congestion. However,  $p_2$  depends on the portion of out-profile packets in relation to the total number of packets arriving at a router ("out-share"). This out-share in turn depends on (i) the size of requested rates and (ii) on what portion of the capacity that is reserved for the PMM service, is currently allocated to other flows.

Dependence (i) is due to the fact that flows with lower bandwidth requests produce more out-of-profile packets than flows with higher bandwidth requests; this general effect is reinforced if the token bucket rate is computed according to (2). Dependence (ii) is due to the fact that greedy TCP flows will fully utilize the available capacity - independently of the requested rate.

Interestingly, it is not possible to make a worst-case estimation of  $p_2$ . In the one case, if  $p_2$  is estimated too low, the sending rates of the TCP flows are over-

estimated and the resulting token rates are too small. For those requests where  $R > R_{req}$ , the  $\max(R, R_{req})$  is smaller than the amount of bandwidth that is really needed by that flow. Thus, in general, too many flows would be admitted.

In the other case, if  $p_2$  is estimated too high, the resulting sending rates of the TCP flows are under-estimated and the computed token rates are too high. This again leads in general to a situation where too many flows are admitted because the real TCP rates are higher than the ones used for the admission control algorithm.

Consequently, the difficulty of estimating the parameter  $p_2$  leads to the problem of incorrect admission control. It is thus very questionable if providing TCP rate guarantees is feasible on the basis of TBM with a *static* configuration and differential queue management. This motivates the investigation of adaptive markers. One such approach has been published quite recently [17].

### AC algorithm with setting advertised window size

The proposed AC algorithm operates also per TCP flow and is of declared based type. On the basis of the  $R_{req}$  and information about round trip time (RTT) of the TCP connection, the user declarations are mapped into the form of single token bucket parameters, rate ( $R$ ) and bucket size ( $BS$ ), constituting input parameters for AC decision. The  $R$  value corresponds to the  $R_{req}$  while the  $BS$  value depends on the required maximum TCP window size ( $W_{req}$ ). The  $W_{req}$  is determined by minimum and average values of the RTT, say  $RTT_{min}$  and  $RTT_{avg}$ , as well as the maximum transfer unit ( $MTU$ ). The discussed AC algorithm guarantees that the received target throughput is stable and close to the requested bit rates for all admitted TCP connections, even if they differ in  $R_{req}$  as well as RTT values.

The starting point for AC algorithm constitutes the connection demands, which are expressed in the form of:

- the target requested rate,  $R_{req}$  [bps],
- and, the information about minimum RTT,  $RTT_{min}$  [s], for the connection.

Now, on the basis of  $(R_{req}, RTT_{min})$ , the next step is to dimension parameters of associated token bucket,  $R$  and  $BS$ . For calculating the  $R$  [bps] and  $W_{req}$  [bits] values we use the following approximate expression (see [18]):

$$R = R_{req} \frac{RTT_{avg}}{RTT_{min}} \cong R_{req} + \sqrt{A / (R_{req} + \sqrt{A / (R_{req} + \sqrt{A / R_{req}})})}, \quad (4)$$

$$\text{where } A = \frac{R_{req}^2 MTU}{2RTT_{min}} \quad (5)$$

The  $BS$  [bits] is done by:

$$BS = \frac{W_{req} * (L_{in} - R)}{L_{in}} \quad (6)$$



$L_{in}$  occurring in (6) denotes the bit rate of the link from host to Edge Router (see Fig.1).

Admission control rules are the following. Let us assume that for serving traffic belonging to the PMM service is allocated the link capacity  $C$  with associated buffer size  $B$ , counted in number of packets. We admit new flow for PMM, which is characterized by  $(R_{new}, BS_{new})$  if the following conditions are satisfied:

$$\sum_{i=1}^{N_{PMM}} R_i + R_{new} \leq C \quad (7)$$

$$\sum_{i=1}^{N_{PMM}} BS_i + BS_{new} \leq B, \quad (8)$$

where  $N_{PMM}$  denotes the number of running TCP flows.

Below we show some numerical results showing the effectiveness of discussed AC algorithm. The capabilities of the AC for getting QoS differentiation are illustrated by results corresponding to TCP connections differing in requested bit rates. The presented experiment corresponds to the single bottleneck network topology (see Fig.3) with the link loaded up to the link capacity, equals 2 Mbps and 4 TCP connections differ in requested rate. The TCP connections are of greedy type, sending packets of constant size,  $MTU=1500$  bytes. The  $RTT_{min}$  is the same for each connection.

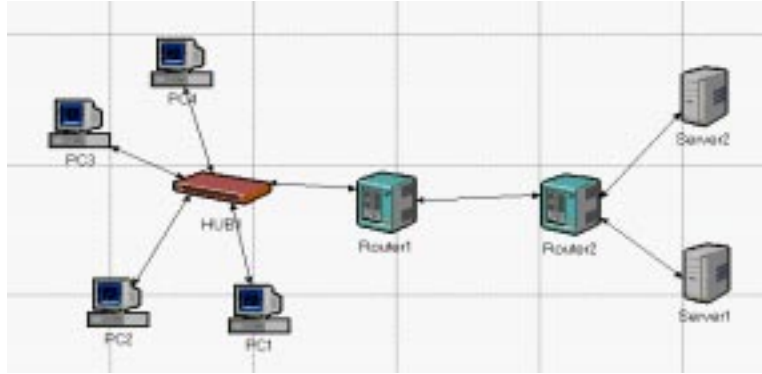


Figure 3: Network topology.

Table 1 shows the results of received throughput (with confidence interval 95%) for each of 4 TCP connections. The results were obtained by simulation in OPNET with TCP Reno implementation. One can observe that the received throughput is bit greater than the requested rate. This result is satisfactory since we can guarantee for a user the TCP throughput close to requested rate. Notice, that in this case the throughput is also close to the goodput, since no packet losses occur (assumed ideal TCP behaviour).

Table 1: Received TCP throughput for each of TCP connections;  $C=2$  Mbps,  $RTT_{min}=0.1$  sec

TCP connections	R (kbps)	$R_{req}$ (kbps)	$RTT_{avg}$ (calc)	$W_{req}$ [bytes]	BS [bits]	$RTT_{avg}$ (sim)	Throughput (kbps) (sim)
PC1-Server1	400	288.022	0.1388	5000	38400	0.1161	333.352 – 355.608
PC2-Server1	400	288.022					
PC3-Server2	600	455.593	0.1122	7500	56400	0.1161	503.456 – 528.064
PC4-Server2	600	455.593					

#### 4. SUMMARY

In the paper we presented the implemented AC algorithms in AQUILA QoS IP network. They correspond to variable bit rate streaming and TCP-controlled traffic. For the streaming traffic, the measurement-based approach is investigated. For the TCP-controlled traffic, the two AC are investigated, one based on token bucket marking approach and the second leading to ideal behaviour of TCP (no packet losses) thanks to appropriate setting of advertised TCP window size. Different aspects of applicability of these algorithms were discussed. Exemplary simulation results confirming the expectations are also included.

#### 5. REFERENCES

- [1] S. Blake et al., *An Architecture for Differentiated Services*, RFC 2475, December 1998.
- [2] J. Roberts, U. Mocci, J. Virtamo, Final Report COST 242, *Broadband network teletraffic: Performance evaluation and design of broadband multiservice networks*, Lectures Notes in Computer Science 1155, Springer, 1996.
- [3] A. Bak, W. Burakowski, F. Ricciato, S. Salsano, H. Tarasiuk, *Traffic handling in AQUILA QoS IP network*, Quality of Future Internet Services, Lecture Notes in Computer Science 2156, Springer 2001.
- [4] AQUILA Project Consortium, Deliverable D1302, *Specification of traffic handling for the second trial*, <http://www.ist-aquila.org>, December 2001.
- [5] W. Burakowski et al, *AQUILA network architecture: first trial experiments*, to appear in Special Issue of Journal of Telecommunications and Information Technology, Warsaw 2002.
- [6] R.J. Gibbens, F.P. Kelly, *Measurement-based connection admission control*, 15<sup>th</sup> International Teletraffic Congress, June 1997.

- [7] W. Burakowski, M. Dąbrowski, *Multiservice QoS IP network: architecture and practical verification in pilot instalation*, Telecommunication Review, No. 5/2002 (in Polish).
- [8] M. Benameur, S. Ben Fredj, F. Delcoigne, S. Oueslati-Boulahia, and J. W. Roberts, *Integrated Admission Control for Streaming and Elastic Traffic*, Quality of Future Internet Services, Lecture Notes in Computer Science 2156, Springer 2001.
- [9] H. ElAarag, M. Bassiouni, *Performance evaluation of TCP connections in ideal and non-ideal network environments*, Computer Communications 24 (2001), pp. 1769-1779.
- [10] S. Floyd, V. Jacobson, *Random Early Detection Gateways for Congestion Avoidance*, IEEE/ACM Transactions on Networking, August 1993.
- [11] J. Padhye, V. Firoiu, D. Towsley, J. Kurose, *Modeling TCP Throughput: A Simple Model and its Empirical Validation*, Proc. ACM SIGCOM'98, August 1998.
- [12] S. Sahu, P. Nain, D. Towsley, C. Diot, V. Firoiu, *On Achievable Service Differentiation with Token Bucket Marking for TCP*, Proc. ACM SIGMETRICS'00, Santa Clara, CA, June 2000.
- [13] T. Ziegler, C. Brandauer, S. Fdida, *A quantitative Model for the Parameter Setting of RED with TCP traffic*, The Ninth International Workshop on Quality of Service (IWQoS'2001), Karlsruhe, Germany, June 2001.
- [14] R. Mortier, I. Pratt, C. Clark, and S. Crosby, *Implicit Admission Control*, IEEE Journal on Selected Areas in Communications, Vol. 18, No. 12, December 2000.
- [15] F. Brichet, M. Mandjes, M.F. Sanchez-Canabate, *Admission control in multiservice networks*, Proceeding of the Mid-Term Seminar COST 257, 1999, Villamora, Portugal.
- [16] F.H.P. Fitzek, M. Reisslein, *MPEG-4 and H.263 video traces for network performance evaluation*, Technical report TKN-00-06, Technical University Berlin, Dept. of Electrical Eng., Germany, October 2000.
- [17] Y. Chait, C.V. Hollot, V. Misra, D. Towsley, H. Zhang and C.S. Lui, *Providing Throughput Differentiation for TCP Flows Using Adaptive Two-Color Marking and Two-Level AQM*, Proceedings of the INFOCOMM'2002 Conference, San Francisco 2002, USA, [http://www.ecs.umass.edu/mie/labs/dacs/Two-Color\\_Marking.htm](http://www.ecs.umass.edu/mie/labs/dacs/Two-Color_Marking.htm)
- [18] W. Burakowski, H. Tarasiuk, *Admission control for TCP connections in QoS IP network*, submitted for the INFOCOMM'2003 Conference