Dynamic Admission Control for Improving Utilization of Parlay X Gateway

Evelina Pencheva[1] and Ivaylo Atanasov[2]

Abstract - **The paper presents a model of Parlay X gateway used for evaluation of its performance. The model considers the distributed architecture of the gateway and applies mechanisms for admission control and load balancing to prevent from overloading A new admission control algorithm is proposed to improve quality of service and node gateway utilization. Simulations are used to evaluate the behavior and the benefits of the proposed algorithm.**

Keywords – admission control, load balancing, web-based traffic, performance evaluation

I. INTRODUCTION

Parlay X is technology that allows open access to network functions. The access is provided through application programming interfaces which hide for the application developers the underlying network specifics and complexity. Third party applications can access network information and control functions by interface method invocation instead of programming network protocols. The "intelligence" is concentrated in a Parlay X gateway which provides interfaces towards applications and "talks" the control protocols toward the network.

In the context of service architecture, a common problem is overload. To avoid congestion usually an admission control mechanism is used. In [2] the authors propose a load control mechanism aimed at supporting constraints imposed by the distributed Parlay X gateway architecture. The mechanism uses preliminary defined threshold values to predict the load and decide if the message should be accepted or rejected. A paper that treats overload control for distributed web-based applications is [3]. The authors suggest a control algorithm that self-configures a dynamic constraint on the rate of incoming new sessions in order to guarantee the fulfillment of the quality requirements specified in service level agreement (SLA). In [4], a staged event-driven architecture is proposed which decomposes a complex, event-driven application into a set of stages connected by queues avoiding the high overhead associated with thread-based concurrency models, and decouples event and thread scheduling from application logic. The proxy-based overload control for web applications presented in [4] is based on measurements of metrics such as response time, throughput, and resource utilization.

[1]Evelina N. Pencheva is with the Faculty of Telecommunications, TU-Sofia, 7 Kl. Ohridski blvd, 1000 Sofia, Bulgaria, E-mail: enp@tu-sofia.bg

[2]Ivaylo I. Atanasov is with the Faculty of Telecommunications, TU-Sofia, 7 Kl. Ohridski blvd, 1000 Sofia, Bulgaria, E-mail: iia@tu-sofia.bg

This paper presents a model of Parlay X gateway providing interfaces for access to location information. The model considers the distributed gateway architecture and applies dynamic control strategy for admission of incoming application traffic. The aim is to optimize the gateway utilization considering the current application demands.

The paper is structured as follows. First in the rest of the paper, we present related work where a dynamic model for the token bucket algorithm is used in packet networks. In Section III, we present a model of Parlay X gateway and in Section IV we describe the traffic model and define node utilization function. In Section V, the dynamic control with feedback for traffic policing is presented. Section VI describes simulation parameters and Section VII discusses the obtained numerical results. At the end, we present our conclusion mentioning issues for future work.

II. RELATED WORK

A token bucket (TB) is well known mechanism used for admission control and packet filtering. In [1], the authors present a dynamic model for the TB algorithm. In a general traffic model, the traffic is observed at regular time intervals $[t_{k-1}, t_k)$ for k=1, 2, …, K. The observation periods are equal and small enough during which only one packet may arrive. The input traffic from several traffic sources is multiplexed in an access node. To prevent the node from overloading, admission control mechanism is used. The incoming traffic of each source is policed by a TB. Usually, the rate of token accumulation is constant in each observation interval. The authors present a balance equation for the state of the i-th TB and express the conforming and non conforming traffic. They suggest a feedback control strategy for dynamic change of the number of tokens transmitted to the TB per time unit. The dynamic control grants tokens according to traffic source demands.

In [2], the overload control in Parlay X gateway providing interfaces for 3rd party call control is studied. An algorithm based on the priority and utility function is proposed. However, the rejection of new service messages in terms of priority will sometimes lead to over-control, which results in part of the resources being idle.

Based on research in [1] and [2], we suggest a model of Parlay X gateway providing access to location information for 3rd parties. The model is used to evaluate the traffic load of the gateway. Our model considers the distributed architecture of Parlay X gateway with a number of processing nodes called PX-M converters. A PX-M converter converts the application request into Mobile Application Part (MAP) request and MAP response into application response. The converters work in parallel. The traffic load is balanced between PX-M

converters using Round Robin algorithm. Model includes admission control implemented by a number of TBs. We apply adaptive control to token accumulation rate to optimize the gateway utilization and to improve the quality of service.

III. Model of Parlay X gateway

The incoming traffic is generated by Service providers (SPs) which have contracts with network operator. The contract defines constraints that have to be fulfilled. The constraints include the peak and average number of application requests that should be accepted per time unit, and the maximum delay between application request and response. To be able to fulfill the constraints and to avoid congestion at the PX-M converters an Admission Control/Load Balancing (AC/LB) mechanism is used. The aim with load balancing is to distribute uniformly the load between the converters. The admission control rejects the nonconforming requests. When a SP sends a request to the Parlay X gateway, it is received by the LB/AC. The admission control, which is modeled by TB, decides whether to accept or reject the request. If the request is accepted, the load balancing decides to which PX-M converter has to be forwarded. Each accepted request has to be answered. The PX-M converter maps the request onto MAP message. Each of the PX-M converters is modeled as a single FIFO buffer with limited size. The buffer size is restricted by the maximal delay between a request and its response.

The modeled PX gateway system consists of n SPs and m PX-M converters as shown in Fig.1.

We assume that all PX-M converters have same capacity C. The Round Robin algorithm is used for load balancing because of the equal capacity of the PX-M converters. Each SP might include several applications but as the contracts are agreed between the SPs and the Parlay X gateway, the number of applications is not important. A single SP is a traffic generator to the LB/AC as far as each SP is connected to its own LB/AC.

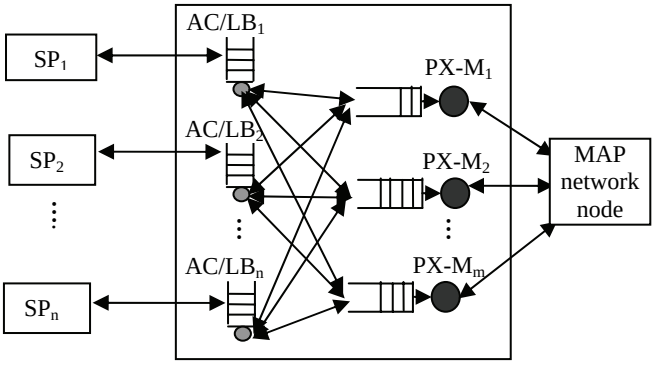

Fig.1 Model of distributed Parlay X Gateway architecture

IV Traffic model

The traffic generated by each SP is a random process observed at time t_k, as shown in Fig.2.

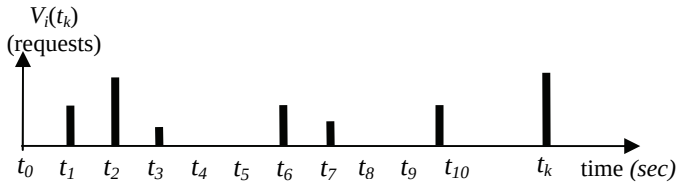

Fig.2 The traffic model of the i-th SP

The status of the token buckets is given by a vector valued function $\mu \equiv (\mu_1, \mu_2, \ldots, \mu_n)'$ of dimension n . The vector valued function μ is described by a system of equations where the dynamic token accumulation for i-th TB is given by:

$$\mu_i(t_k) = \mu_i(t_{k-1}) + [\min(u_i(t_k), T_i - \mu_i(t_{k-1}))] - V_i(t_k).I\{ V_i(t_k) \le \mu_i(t_{k-1}) + [\min(u_i(t_k), T_i - \mu_i(t_{k-1}))]\} \quad (1)$$

The indication function $I(S)$ is used to represent the truth of the statement S where $I(S) = 1$ if the statement S is true, or $I(S) = 0$ otherwise.

The conforming traffic is also given by a vector valued function of dimension n, which we denote by $G \equiv (G_1, G_2, \ldots, G_n)'$, where

$$G_i(t_k) = V_i(t_k).I\{ V_i(t_k) \le \mu_i(t_{k-1}) + [\min(u_i(t_k), T_i - \mu_i(t_{k-1}))]\} \quad (2)$$

The vector of nonconforming traffic is given by $R \equiv (R_1, R_2, \ldots, R_n)'$, where

$$R_i(t_k) = V_i(t_k) - G_i(t_k) \quad (3)$$

To formulate the status of each PX-M converter, we have to consider the load balancing mechanism. With $\sigma_i(t_k)$ we denote the state of the i-th LB that is the PX-M converter to which the conforming application request has to be forwarded, and it is given by:

$$\sigma_i(t_k) = 1 + (\sigma_i(t_{k-1}) + G_i(t_k)) \bmod m \quad (4)$$

The state of i-th LB models the control variable which depends on its previous value, the conforming traffic for the period of observation, and finally the count of converters. The state reflects the robust Round Robin principle of balancing.

With $H_{ij}(t_k)$ we denote the part of the conforming traffic distributed by the i-th access controller to the j-converter. It is given by:

$$H_{ij}(t_k) = \lfloor G_i(t_k))/m \rfloor + I[j \le \sigma_i(t_k)] \quad (5)$$

The conforming traffic forwarded to the j-th PX-M converter is given by:

$$F_j(t_k) = \sum_{i=1}^{n} H_{ij}(t_k) \quad (6)$$

The status of the j-th PX-M converter is given by the size of the queue at the PX-M converter denoted by $q_j(t_k)$ and waiting for service, that is, for onward transmission to the network. This variable is governed by the following equation:

$$q_j(t_k) = \max [q_j(t_{k-1}) - C_j(t_k - t_{k-1})/2, 0] + \min\{F_j(t_k), Q_j - \max [q_j(t_{k-1}) - C_j(t_k - t_{k-1}), 0]\} \quad (7)$$

where with Q_j we denote the buffer size of the j-th PX-M converter, and C_j is the capacity of the j-th PX-M converter. Each conforming application request has to be answered, so we assume that the capacity is measured as requests per second (the capacity for responses is the same). The first component in the right hand side of the above expression describes the leftover traffic at time t_k after the PX-m converter has sent the converted request to the network during the period $[t_{k-1}, t_k)$. The second component represents the

traffic accepted from all load balancers during the same period. The traffic accepted by the *j*-th PX-M converter is given by the smaller of the available (empty) space in the buffer and the sum of the conforming traffic.

It is expected that the *j*-th PX-M converter may not be able to convert all forwarded application requests because of its buffer size limitation Q_j and capacity limitation C_j. Thus the traffic lost at the *j*-th PX-M converter is given by:

$L_j(t_k) = F_j(t_k) - \min\{F_j(t_k), Q_j - \max[q_j(t_{k-1}) - C_j(t_k - t_{k-1}), 0]\}$ (8)

From this equation we may define Parlay X gateway utilization as:

$$\eta = \frac{1}{C(t_K - t_0)} \sum_{k}^{K} \left(\sum_{i=1}^{n} V_i(t_k) - \left(\sum_{i=1}^{n} R_i(t_k) + \sum_{j=1}^{m} L_j(t_k) \right) \right) \quad (9)$$

Another measure of Parlay X gateway efficiency is the average throughput which is intimately related to the gateway utilization as defined above.

$\Delta = C\eta = \frac{1}{(t_K - t_0)} \sum_{k=0}^{K} \Delta(t_k)$, where $\Delta(t_k)$ denotes the throughput during the period $[t_{k-1}, t_k)$.

V. Objective function and load control

The objective function for the network provider is given by:

$$J(u) = \sum_{j=1}^{m} \sum_{k=0}^{K} \alpha_j(t_k) L_j(t_k) + \sum_{i=1}^{n} \sum_{k=0}^{K} \beta_i(t_k) R_i(t_k) + \sum_{j=1}^{m} \sum_{k=0}^{K} \gamma_j(t_k) q_j(t_k) \quad (10)$$

where $u = (u_1, u_2, \ldots, u_n)'$ is the control vector that appears in the dynamic model of the TBs.

The parameters $\{\alpha_j, \beta_i, \gamma_j, i = 1...n, j = 1...m\}$ are nonnegative functions of time assigning relative weights given to various losses. The first parameter imposes a penalty on losses in the *j*-th converter, the second parameter imposes a penalty on lost traffic at the admission control for the *i*-th TB and the third parameter is an approximate measure of waiting time or delay at the *j*-th PX-M converter before being served. The problem is to find a control policy that minimizes this function.

We denote with $W(t_k)$ the resources that have to be distributed between SPs in time t_k as a sum of requests that will be processed by all converters and the available places in the converter buffers:

$$W(t_k) = \sum_{j=1}^{m} (\tau . C_j + (Q_j - q_j(t_k))) \quad (11)$$

The control policy for tokens granted to different SPs considers their current demands and it is defined by:

$$u_i(t_k) = ug_i(t_k) + \frac{V_i(t_k)}{\sum_{i=1}^{n} V_i(t_k)} . (W(t_k) - \mu_i(t_k)) \quad (12)$$

The left side of the equation (12) presents the tokens that will be granted to *i*-th SP for the interval $[t_k, t_{k+1})$. The first component in the right side is the number of token that the network operator has been guaranteed to provide to the *i*-th SP, and the second component represents the part of all available resources that are distributed proportionally to the demands of the SPs.

VI. Simulation parameters

To demonstrate the usefulness and effectiveness of the model, we implement the model and the adaptive TB control algorithm in Java. We have used Java IDE Eclipse.

The simulation duration is 600 *s*. During simulation, statistic data are gathered at intervals of 1 *s*.

In the simulation 8 SPs and 4 PX-M converters are used. The capacity of the Parlay X gateway is 800 requests per second which is equally distributed between the converters.

We assume that 2 of the SP generate priority traffic with higher peak and guaranteed rates in comparison with the rest 6 SPs. The peak rate for the high priority traffic is 130 requests per second and the guaranteed rate is 100 requests per second. The peak rate for the low priority traffic is 100 requests per second and the guaranteed rate is 75 requests per second. The behavior of each SP is modeled by Markov Modulated Poisson Process (MMPP) as the arrival process in the context of web services [5]. New application requests are generated according to four-state MMPP, where mean rates for the higher priority SPs are 0, 50, 90, 130 requests per second and for the low priority SPs the mean rates are 0, 30, 60, 100. Changes between different states are uniformly distributed and occurred according to Poisson process with mean 4 seconds. The SP traffic is policed by 8 TBs whose conforming outputs are multiplexed between 4 PX-M converters. The token rate is equal to the guaranteed rate and the bucket size is 30 for high priority SP and 25 for low priority SP. Initially $\mu_i(t_0) = T = 30(25)$, $q_i(t_0) = 0$.

The observable intervals $[t_k - t_{k-1})$ are equal (100 *ms*). The processing time for a single request/response in a converter is 5 *ms*. Given the constraint of 100 ms between given request and its response, the maximum waiting time in the buffer in each direction is *n*.5 = 45*ms* where with *n* we denote the buffer size (9 requests/responses). We assume that the networks delay is constant (1 *ms*) in order to correlate the time between request and its response spent in the gateway.

When choosing values for the relative weights given to various losses, we stress on losses in the gateway where the conformed traffic is lost due to the converter overload. We choose $\alpha_j(t_k) = 10, \beta_i(t_k) - 1, \gamma_j(t_k) = 1, i = 1...n, j = 1...m$ for all t_k.

VII. Numerical results

We define evaluation function *Z* which represents the ratio of the gateway utilization in case of constant rate of token accumulation (without control) to the throughput in case of adaptive rate of token accumulation (with control). The evaluation function *Z* is presented by $Z = \frac{\eta_{static}}{1 + \eta_{dynamic}}$.

Fig.3 shows the changes of the evaluation function *Z* in time. The gateway utilization in case of adaptive control is higher than the utilization without control. The average throughput gain is 8.378827%.

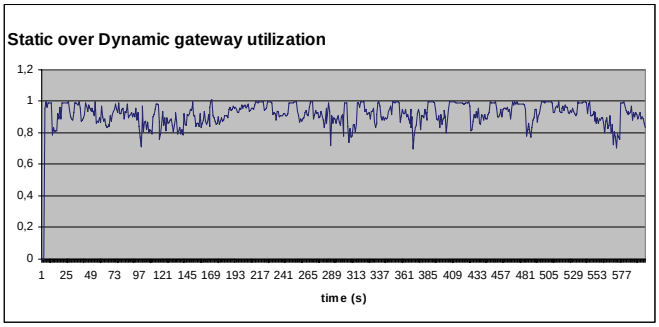

Fig.3. Ratio of gateway throughput for constant and adaptive token rates

To assess the losses, we define evaluation function *Y* which represents the ratio of the losses in case of static token rate to the losses in case of dynamic token rate. The evaluation function *Y* is presented by $Y = \frac{J_{static}}{1 + J_{dynamic}}$.

Fig.4 shows the changes of the evaluation function *Y* in time. The losses in case of adaptive control are less than the losses without control. The average ratio is 5.42972.

Fig.5 shows the ratio of measured delays (the time between request and its response) in case of static and dynamic rate of tokens. As it can be seen, the average delay in case of no control of token rate is less than the average delay in case of adaptive control. This is because of the control strategy which accepts more requests, which means that the buffers before the converters are fuller in comparison with the static token rate. The average delay without control is 49 *ms*, and with applying control is 49,5 *ms*. The measured maximum delay is less than 100 *ms*, as has to be expected due to the limited buffer size.

VIII. CONCLUSION

The paper presents a model for evaluation of the traffic load of a Parlay X gateway with distributed architecture. The numerical results can be used for setting values of quality of service parameters agreed with network operator. The suggested algorithm for adaptive control improves the gateway performance.

The proposed model is suitable for transaction traffic pattern that means that the requests do not correlate each other. Our future work will consider the session traffic pattern where several requests are related to a session.

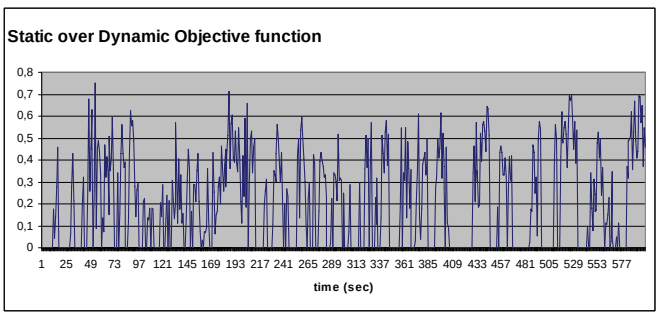

Fig.4. Ratio of losses for constant and adaptive token rates

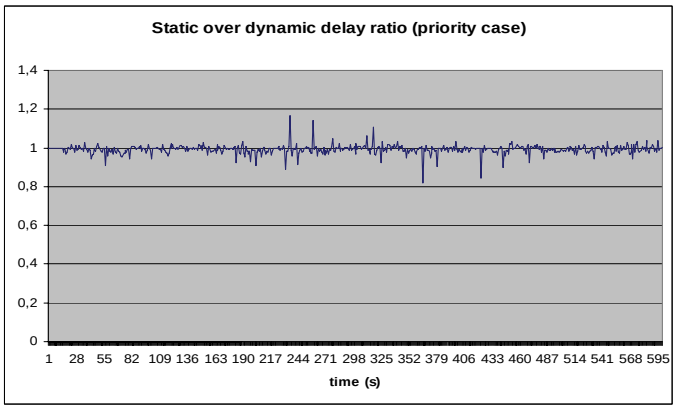

Fig.5 Ration between delays without and with control of the token rates

ACKNOWLEDGEMENT

The work is conducted under the grant of the Project DO 02-135/2008 funded by Bulgarian Ministry of Education and Science.

REFERENCES

[1] Ahmed, N., Wahg, Q., Barbosa, O., Systems Approach to Modeling the Token Bucket Algorithm in Computer Networks, Mathematical problems in Engineering, 2002, vol. 8(3), pp.265-279

[2] Anderson, J., Kihl, M., Sobirk D., Overload Control of a Parlay X Application Server, Proc. of SPECTS'04, pp.821-828, 2004.

[3] Bartolini, N., Bongiovanni, G., Silvestri S., Self- through self-learning: Overload control for distributed web systems, Computer Networks, 53, 2009, pp.727–743

[4] Mathur, V., Dhopeshwarkar , S., Apte, V., MASTH Proxy: An Extensible Platform for Web Overload Control, http://www2009.org/proceedings/pdf/p1113.pdf, 2009

[5] Muscariello, M., Mellia, M., Meo, M., Marsan, M., Lo Cigno, R., Markov models of internet traffic and a new hierarchical MMPP model, Computer Communications vol. 28 , issue 16, 2005, pp. 1835-1851