Evaluating Session-Aware Admission-Control Strategies to Improve the Profitability of Service Providers

Narjess Ayari and Denis Barbaron
Orange Labs - France Telecom R&D
2, Avenue Pierre Marzin, 22307 Lannion, France
{narjess.ayari,denis.barbaron}@orange-ftgroup.com

Laurent Lefèvre
INRIA RESO - LIP (UMR 5668 CNRS, ENS, INRIA, UCB) Université de Lyon, 46, allée d'Italie - 69364 LYON, France - {laurent.lefevre}@ens-lyon.fr

Abstract- **In this paper, we investigate an economically motivated session-aware admission-control model for Internet servers subjected to future NGN session-oriented services. Our fundamental observation is that it is sometimes desirable to reject new sessions from some customers, so that others may be completed and thereby generate some revenue for the operator. Conversely, most of the already existing admission-control schemes, which operate either at the packet level or at the request level, may interrupt life-service sessions at any time during their lifespan. In this work, we evaluate novel service-oriented session-aware admission-control strategies for controlling the acceptance of the offered network traffic to an Internet server. Conducted simulations target popular VoIP service. Simulation results demonstrate that responsive session-aware admission control improves the service provider's benefit and offers better QoS to subscribers. Particularly, we show that session-aware responsiveness provides network QoS in terms of completion of the service sessions independently of their durations.**

I. Introduction

When overloaded, a processing server does not have sufficient resources to provide the service to all clients. It contributes to the poor QoS through sustaining heavy request queuing delays and prolonged processing time. Since a poor perceived performance is a foremost impediment to the success of any service, an operator needs to provide an acceptable QoS for the admitted network traffic. Therefore, not only does a processing server require enough processing resources, its resources also need to be prevented from overloading.

The concept of admission control is a well-known means to prevent a server or a network route from overloading. It consists of regulating the acceptance of the offered network traffic according to the usage of the controlled resources. Most state-of-the-art researches on that topic advocate session-oblivious mechanisms where dropping requests pertaining to an accepted session can occur at any time during the session's lifespan. From an operator's perspective, this means that the server, which seems to sustain a high throughput, is in reality wasting its resources on failed or on interrupted sessions.

Conversely, the concept of customer session is of particular interest for many applications. For commercial web sites, for instance, a customer sends a number of HTTP requests to the web site during a single session. For the web site to be successful, it is important that as many customers as possible complete their sessions, since those customers may generate some revenue. Voice-over-IP is another emerging application requiring improved QoS. Firstly, from an economically-motivated point of view, rejected or interrupted calls are not charged to customers, while resulting in an increasing number of angry clients. Secondly, they cause the operator's processing resources to be wasted. Since the overload of a call server or an intermediate gateway counts among the potential reasons for the interruption of ongoing calls, admission control is very desirable to guarantee an acceptable QoS for the admitted multiple flow-based VoIP calls.

Most of the already investigated admission-control mechanisms have focused on packet-level or on flow-level admission control [1,2,3,4]. These strategies are particularly suitable for single flow-based sessions where a single subscriber session spans over a single flow over time.

In [7], we advocated an admission control architecture that uses deep packet inspection to provide session awareness to routers subjected to multiple flow-based NGN services. We focused on an engineering point of view and described the design of the architecture's building blocks. In this paper, we investigate a novel economically-justified session-aware admission-control paradigm to improving resource usage and profitability of session-oriented servers. Our fundamental observation is that it is desirable to reject new sessions from some customers, so that others may be completed successfully, and thereby generate some revenue for the service provider. Through simulations, we demonstrate that responsive session-aware admission control improves the service provider's benefits and contributes to providing better QoS to subscribers. Particularly, we show that session-aware responsiveness provides network QoS in terms of completion of the service sessions independently of their durations.

The remainder of this paper is organized as follows. In section II, we review some research works on admission control for Internet servers. Section III presents the mechanism that we advocate to control the offered session-based network traffic to an Internet server. Section IV is devoted to the performance evaluation of our proposal. Simulation results show that our approach provides the operator with better profitability and contributes to the completion of the offered

sessions independently of their duration. Finally, section V concludes the work and outlines interesting future directions.

II. Background and Related Works

Admission-control mechanisms for Internet servers have almost the same objectives as admission-control mechanisms for the core network. The latter typically involve a gateway that estimates the offered network load by measuring the queue length of its backlogged packets. The admission-control policy uses this value to regulate the acceptance of the incoming packets. The Drop Tail algorithm describes the simplest form of packet discarding. It suggests dropping incoming packets when the router's buffer space becomes full. This idea is obviously applicable to the admission control of requests made to Internet servers. However, its major drawback is that it rejects aggressively and arbitrarily the incoming traffic while approving similarly different classes of traffic. Both the Early Random Drop (ERD) and the Random Early Detection (RED) approaches [1,2] addressed Drop Tail's deficiencies. They suggested to prevent the router's congestion rather than simply reacting to it. In essence, the idea is to prevent the backlog saturation by anticipating the rejection of the incoming packets as soon as the backlog queue's length goes beyond a predefined threshold. A packet will be dropped with a probability computed according to the router's backlog queue length. Randomization tends to ensure that all classes of traffic will suffer the same loss rate. Chen and Mohapatra [3] applied the ERD approach to regulate the acceptance of web server requests. They applied a double-threshold-based admission control on the application server's listen queue. Requests of lower priority are rejected with a higher probability as soon as the server's utilisation exceeds the first threshold. All requests are rejected when the second threshold is reached. This approach was shown to be effective for controlling differentiated services, mainly in terms of queuing delays between lower and higher traffic-priority classes. However, it has one major drawback: the application server's queue length does not necessarily represent a good indicator of the server's experienced load. Abdelzaher *et al*. [4] assumed a linear regression method which estimates the impact of the handled requests on the system's utilization. They used a linear feedback control-theory which admits an appropriate number of requests while keeping the system utilization bounded. However, they didn't consider any further constraint on the handled traffic. Lee *et al*. [5] focused on the web server and assumed the knowledge of the request arrival rate as well as the maximum waiting time for each incoming traffic class. They suggested two admission-control approaches. The first approach maximizes the potential profit of the service provider, while the second one admits as many clients as possible into the controlled web server. Carlstrom and Rom [6] presented an architecture and some algorithms for optimizing the performance of web services. They advocated an analytical admission-control model, to maximize the service provider's objective. Their key idea was to break down a given web session into stages each having specific requirements and transition probabilities. The server must be aware of this structure so as to provide QoS-aware processing.

III. Proposal of Session-Aware Admission-Control Strategies for Internet Servers

A. Problem statement and motivations

Fundamental to our proposal is the characterization of the operator's reward as the mean monetary equivalent due to the blocking, the completion and the interruption of the offered sessions over a long time scale. On the basis of economical arguments, we define the operator's reward as the mean pay-off that encompasses the following three performance criteria.

Firstly, in an ideal situation when the Internet server does not suffer overload, an accepted session terminates normally. The consequence of that good functioning is user satisfaction, which translates into a positive image of the operator regarding competition. On the contrary, when the server experiences overload, any newly offered session will be blocked, resulting in bad user perception of the QoS provided by the system. Finally, when the server runs close to its capacity and is overloaded, the already active sessions are most likely to be interrupted.

Based on the operator's point of view, we assume that interrupting existing sessions is even worse than blocking new sessions, since interrupted calls are not charged to customers, while resulting in an increasing number of angry clients and causing the operator's processing resources to be wasted.

Let us associate to the successful completion, rejection, and interruption of a session an equivalent monetary value that describes a per-session gain contribution to the total operator's reward, denoted by R_c, C_b, and C_i, respectively. If N_c, N_b, and N_i stand for the mean number of completed, rejected, and interrupted (respectively) sessions per unit of time, then the operator will try to maximize the following objective function,

$$R := R_c N_c - C_b N_b - C_i N_i , \qquad (1)$$

The decision variables that the operator would play on to maximize this objective R are the acceptance probabilities of the newly offered sessions, denoted below by (p).

B. Proposed session-aware admission control

From the above analysis, it turns out that the optimal network traffic acceptance policy would accept an offered session to the server if and only if the reward resulting from accepting the session is more important than the penalty resulting from rejecting it. This policy is expected to maximize the operator's profitability by maximizing the number of completed sessions under heavy load.

The rest of this section is devoted to describing the structural properties of the acceptance strategies that we advocate to optimize the operator's global reward *R*. We believe that in a more realistic and precise model, the admission control decision should depend on the load of the system [7, 8], which could, for instance, be measured in terms of number of packets pending in the server's network buffers.

Firstly, when the load experienced by the server is under a first threshold denoted by T_1, the incoming traffic is accepted and forwarded to the next processing entity. Once the server's load exceeds the first threshold T_1, any incoming traffic holding a request for the establishment of a new session is dropped with a probability *p* computed as a function of the server's instantaneous load. In the remaining of this paper, we refer to *p* as the probability of rejection of a new session. When the server's load exceeds the second threshold denoted by T_2, only the traffic pertaining to the already established sessions is admitted. This rule is introduced mainly to avoid the interruption or the QoS degradation of the already established sessions. Particularly, we will show that this rule is necessary to reduce discrimination against long term sessions. Indeed, short lived conversations typically have a higher chance to be completed normally. When the server runs very close to its maximal capacity denoted by *C*, the offered traffic is dropped, until some processing resources are released. For improved responsiveness, an offered datagram is dropped with a probability *p* derived as a function of the instantaneously measured server's load denoted by (l_s), as follows,

$$p = \begin{cases} 0, & if(l_s \le T_1) \\ f(l_s), & if((NewSession) and (T_1 < l_s \le T_2)) \\ 1, & if((NewSession) and (T_2 < l_s \le C)) \\ 1, & if(l_s > C) \end{cases}, \qquad (2)$$

In the following subsection, we describe the advanced simulations conducted to evaluate the advocated admission-control strategies.

IV. Performance Evaluation

To evaluate the proposed admission-control strategies, we have implemented a generic simulator which consists of a processing server with a limited buffer-sized listen queue, a set of generators of clients, sessions, requests and feedbacks, as well as a set of estimators useful to maintain real-time information about the server's load as well as the rates it achieved at session- and request-level. We aim at showing that responsive session-aware admission control improves the operator's benefit as well as the QoS provided to subscribers by improving its session-level rate. Session-level rate will be defined below as the number of successfully completed sessions per unit of time.

A. Simulation model overview

We use a discrete event-based simulation model to describe the interaction between the requests' arrivals to the system and their departures from it. In our model, we define a session as a sequence of individual requests that are generated for the duration of the session. A new session is produced by a session generator according to specified input parameters such as the client's identifier, the time at which the session was initiated, its original and current durations, its source throughput, and more. A session is completed normally when the total number of the related requests are completed successfully. For a given simulation run, the client generator generates *N* sources of traffic each corresponding to one client. Every client is able to generate sessions at a source rate δ. The request generator issues session's requests according to specific session's parameters such as the inter-request think-time and the number of retries in case of a request loss.

The processing server is characterised by its capacity and service-processing time. The service-processing time is assumed to be proportional to a request's size.

C. Performance Metrics

We aim to study the impact of different admission-control policies, session-aware and session-oblivious, on the operator's achieved profitability. Profitability has been typically estimated by measuring the servers' throughput. Actually, a server's throughput is defined as the number of requests or packets that are processed per unit of time. Usually, a server running close to its edge capacity is considered satisfactory.

In this paper, we define the useful throughput or the success rate as the number of successfully completed sessions per unit of time. We believe that this metric provides a truthful estimation of the operator's benefit, since a server that sustains a high request-level rate may actually be wasting its resources on failed or on interrupted sessions.

In our simulations, we considered a realistic traffic, with similar traffic patterns as voice CBR-like traffic. Table 1 below summarizes the parameters of the model used to generate the offered sessions [see Table 1].

Table 1 - Traffic Profile Model for Voice CBR Conversations.

Model	Distribution
Packet Inter Arrival Times	Constant
Session Inter Arrival Times	Exponential
Session Duration	Exponential
Packet Size	Constant

Simulation results presented below average the output of ten simulation runs, made up of new arrivals of client sessions.

D. Simulation Scenarios and Methodology

In this subsection, we detail our study of the effect on the achieved useful throughput of the different advocated measurement-based admission-control strategies. We simulate the behaviour of an admission-control-enabled processing server over a simulation duration of 100 seconds. A client initiates a conversation with an average talking time that obeys an exponential distribution. Session duration is varied over the simulation run and we have considered medium- and long-term sessions with exponentially distributed durations. The traffic rate is set to 50 packets per unit of time.

Firstly, we consider a server deploying Request-Aware admission Control (RAC), where dropping requests pertaining to the same session occurs at any time during the session's lifespan. The request-aware strategy that we consider responsively drops the offered packets on the basis of a probabilistic model, as follows,

$$p = \begin{cases} 0, & if(load \le T_1) \\ \left\| \dfrac{load - T_1}{capacity - T_1} \right\|, & if(T_{1<load \le C}) \\ 1, & otherwise. \end{cases}, \qquad (3)$$

Secondly, we consider a server deploying Session-Aware admission Control (SAC). When the server runs under a heavy load, the offered packets pertaining to the already established sessions are given a higher priority compared to those holding requests for establishing new sessions.

In the first variant of this model, which is depicted as the ON/OFF Session-Aware admission Control (ON/OFF SAC), the dropping probability p of the offered new sessions is derived as follows,

$$p = \begin{cases} 0, & if\left(load \le T_1\right) \\ 1, & if\left(\left(NewSession\right) and \left(load > T_1\right)\right) \end{cases}, \quad (4)$$

However, the second variant of this strategy advocates a responsive Session-Aware admission-Control model (RESP SAC), where new sessions are dropped with a probability p computed as a function of the server's current load as follows,

$$p = \begin{cases} 0, & if\left(load \le T_1\right) \\ \left\| \dfrac{load - T_1}{capacity - T_1} \right\|, & if\left(\left(NewSession\right) and \left(load > T_1\right)\right) \end{cases}, \quad (5)$$

In the following subsection, we will be studying the effect of session awareness on the achieved average useful throughput for different session durations. Next, we will turn our attention to studying the effectiveness of the model described by equation (2) in fairly competing sessions independently of their duration. For this purpose, we will consider two simulation scenarios:

1) Generating homogeneous sessions over the same simulation run, having a duration that ranges from medium- to long-term.
2) Generating mixed sessions over the same simulation run. In order to focus on the fairness property, we have chosen that medium- and long-term conversations be equitably generated over the simulation time.

E. Simulation Results and Discussion

Figure 1 illustrates the effect of session unawareness on the achieved average throughput compared to the achieved average useful throughput for the simulation run [Fig. 1]. The generated sessions are medium-term. Session-unaware admission control is deployed either by involving no admission control (No AC) or by deploying request-aware admission control (RAC).

As we could have expected, when the server is oblivious to admission control or when it deploys request-aware admission control, the achieved average useful throughput has lower values compared to the achieved average throughput. Particularly, when the server is oblivious to admission control (No AC), while it seems to fully satisfy the clients' demands, only nearly 14 % of the offered sessions are successfully completed compared to the completion of nearly 17 % of the offered sessions on average, when responsive request-aware admission control (RAC) is deployed on the server. Indeed, dropping the offered packets to the server at any time during the session's lifespan does not efficiently prevent session interruptions. This observation confirms our economically-justified point of view, advocating that request-aware admission control does not deterministically contribute to improving the operator's profitability and the QoS perceived by clients.

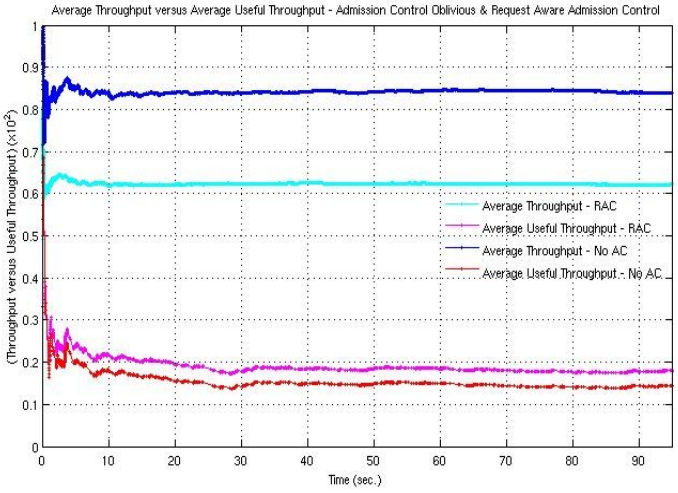

Fig. 1. Effect of Session-Unaware Admission Control on the Average Useful Throughput - Medium-Term Sessions & Single-Threshold-based RAC (75 % of the Server's Capacity).

In contrast, figure 2 illustrates the average achieved useful throughput, when session-aware admission-control strategies are deployed on the server, compared to when responsive request-aware admission control is used by the server [Fig. 2].

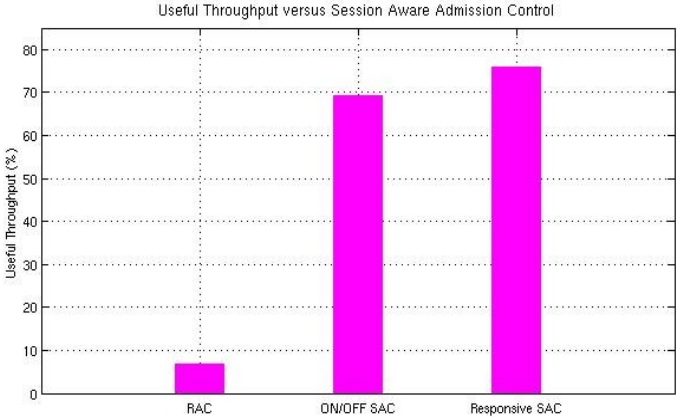

Fig. 2. Effect of Session-Aware Admission Control on the Average Useful throughput - Medium-Term Sessions & Single-Threshold-based Admission Control (75 % of the Server's Capacity).

Firstly, we can see that when responsive session-aware admission control is used, nearly 76 % of the offered sessions are completed successfully, for a threshold value of 75 % of the server's capacity. The choice of the threshold value is justified by further simulation results where we observed that, under the above assumptions, the best useful throughput is provided when using a threshold of 75% of the server's capacity. Figure 3 below describes the observed average useful throughput versus the threshold, assuming medium-term sessions and a responsive session-aware admission control [Fig. 3].

On the other hand, we can see that responsive session-aware admission control outperforms the basic session-aware

admission-control model (ON/OFF SAC) by completing, on average, 8.83 % more offered calls.

It is also worth pointing out that the gain achieved when using responsive session-aware admission control (RESP SAC), compared to when using responsive request-aware admission control (RAC), is the completion of nearly 69 % more offered sessions, on average, under the same previously considered assumptions.

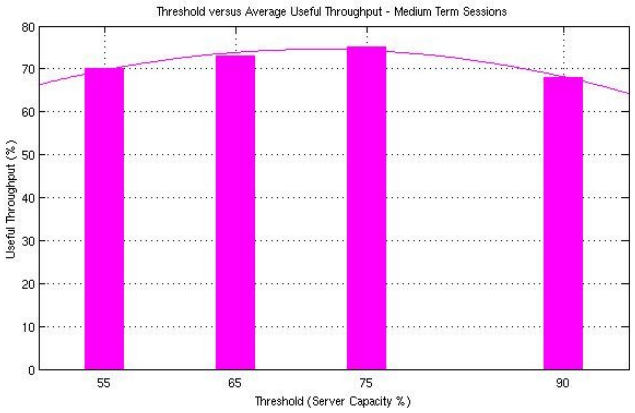

Fig. 3. Average Useful Throughput versus Threshold - Medium-Term Sessions & Responsive Session-Aware Admission Control.

Next, we focus on comparing the effectiveness of single- and double-threshold-based session-aware admission-control strategies. Figure 4 focuses on revealing the gain achieved by a session-aware admission-control-enabled server [Fig. 4]. It compares the achieved average useful throughput when the server uses either single-threshold-based responsive session-aware admission control (Single SAC) or double-threshold-based session-aware admission control (double SAC). The durations of the generated traffic are homogeneous over simulation runs, meaning that all generated sessions are either medium- or long-term over a given simulation run. The threshold value of the used Single SAC policy is set to 75 % of the server's capacity. The used double SAC involves a first threshold value set to 75 % of the server's capacity and a second threshold value set to 90 % of the server's capacity.

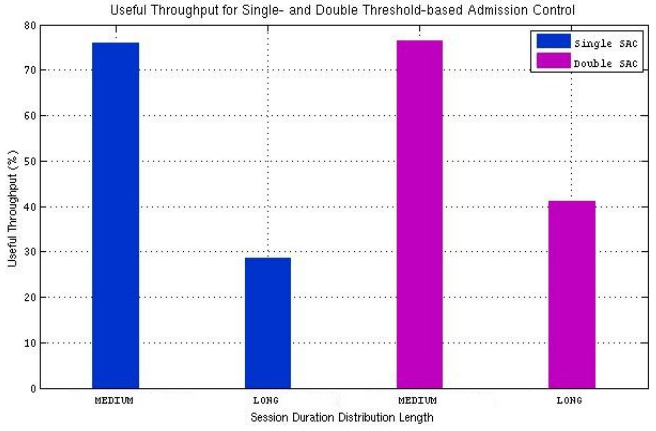

Fig. 4. Percentage Histogram of Completed Calls for Different-Admission Control Strategies – Homogeneous Generated Traffic.

As we can see, when the offered sessions are medium-term, double-threshold-based admission control performs slightly better than single-threshold-based admission control. Indeed, only 0.52 % of performance gain is obtained, on average. However, when the offered sessions are long-term, double-threshold-based session-aware admission control offers a performance gain of nearly 44 % more completed sessions, compared to when single-threshold-based policy is used. This result confirms with intuition, since this model is actually designed to suit long-term sessions.

Let us now examine the average useful throughput when single- and double-threshold-based session-aware admission-control strategies are performed on mixed traffic. Medium-term and long-term sessions are equitably generated over the simulation runs. Figure 5 plots the percentage histogram of the achieved average of completed calls in that case [Fig. 5]. The first threshold is set to 75 % of the server's capacity for both strategies. The second threshold value is set to 90 % of the server's capacity for the Double SAC strategy.

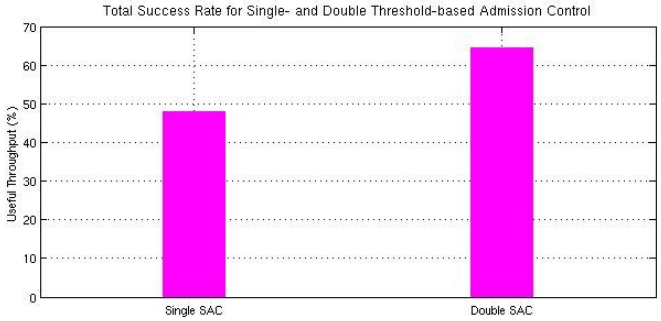

Fig. 5. Percentage Histogram of Completed Calls for Different Admission-Control Strategies - Mixed Traffic.

We can globally see that double-threshold-based session-aware admission control (Double SAC) contributes to improving the average success rate by completing nearly 25 % more offered calls, compared to when single-threshold-based admission control (Single SAC) is used.

Figure 6 highlights the contribution of long-term sessions' success rate in this achieved total average useful throughput. It plots the percentage histogram of the average completed calls as a function of session duration for both single- and double-threshold-based session-aware admission control [Fig. 6].

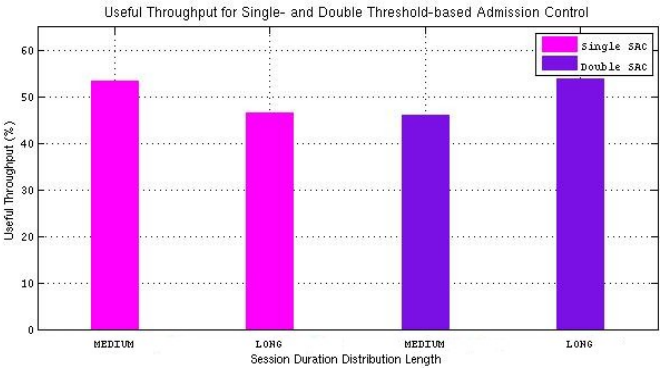

Fig. 6. Percentage Histogram of Completed Calls for Different Admission-Control Strategies - Mixed Generated Traffic.

As we can see, double-threshold-based session-aware admission control privileges the fair completion of the offered sessions independently of their duration. Indeed, the second threshold aims at a kind of protection of long-term sessions under overload. We observe that more than 13.69 % more long-term calls, established through the server, have satisfactory quality. In contrast, this percentage drops to 2 % completed calls, when responsive request-aware admission control is used. As well, since the double-threshold-based admission control is giving a higher priority to long-term calls, it completes less medium-term calls.

In realistic use-cases, calls range from short to long term. If we define QoS as the guarantee of completing more sessions independently of their duration, hence, based on the above simulation results, we can say that the session-aware admission control that we have advocated in section III contributes to having better QoS provided to subscribers.

V. Discussion

In the previous section, we investigated session-aware admission control as the means to efficiently prevent the overload of Internet servers while maximizing the operator's profitability. We compared the average achieved useful throughput for the different considered scenarios involving the generation of either mixed or homogeneous sessions in terms of durations. As we have shown, session-aware admission control significantly surpasses standard strategies. For a satisfactory operator's revenue, session-aware admission control improves the success rate by more than 69 % compared to standard request-aware admission control when the server deals with medium-term offered sessions. On the other hand, we have shown that using responsive double-threshold-based admission control is beneficial to increase the performance of servers processing long-term sessions. A performance gain of nearly 44 % more successfully completed long term calls is noticed when double-threshold-based session-aware admission control is used, compared to when using single-threshold-based session-aware admission control. Finally, we showed that in more realistic scenarios, where the calls' durations range from short to long term, double-threshold-based session-aware admission control contributes to the completion of more sessions independently of their duration.

VI. Conclusion and Future Works

Service-aware network management is a key issue for the current and future NGN service-oriented networks. Particularly, providing session-aware QoS at the access, the core and the edge of the current Internet are hot topics for service operators and network providers. In this paper, we focused on the contribution of Internet servers to improving the QoS user experience. Based on an economically-justified service model, we derived and evaluated a novel session-aware admission-control model for Internet servers subjected to session-oriented services. Simulations have considered realistic voice CBR-like traffic. We showed that responsive session-aware admission control improves the providers' benefit as well as the perceived QoS by subscribers. Particularly, we showed that responsive double-threshold-based session-aware admission control adapts to more realistic use-cases by guaranteeing the completion of sessions independently of their duration.

Significant perspectives to this work include extending the advocated session-aware model to handle more QoS metrics, such as the client's category and other composite QoS metrics, as well as addressing the QoS of highly variable Internet traffic, for example, by using advanced forecasting techniques to improve the stability pattern of admission-control decisions under overload.

VII. References

[1] S. Floyd and V. Jacobson, "Random Early Detection for congestion avoidance", Proceedings of the IEEE/ACM Transactions on Networking, Aug. 1993, pp. 60-64.

[2] E. Hashem, "Analysis of random drop for gateway congestion control", Technical report MIT-LCS-TR-467, Laboratory of Computer Science, 1989, pp. 60.

[3] X. Chen and P. Mohapatra, "Providing differentiated services from an Internet server", Proceedings of the 8th IEEE Transactions on Computer Communication and Networks, 1999, pp. 214-216.

[4] T. Abdelzaher, K. Shin and N. Bhatti, "Performance guarantee for Web server end-systems: a control theoretical approach", Proceedings of the IEEE Transactions on Parallel and Distributed Systems, 2002, pp. 60-131.

[5] S. Lee, J. Lui, Y. Yau, "Admission control and dynamic adaptation for a proportional delay diffserv-enabled web server", Proceedings of the IEEE SIGMETRICS'02, 2002, pp. 60.

[6] L. Cherkasova and P. Phaal, "A mechanism for peak load management of commercial web sites", Proceedings of the IEEE Transactions on Computers, 2002, pp.61.

[7] N. Ayari, D. Barbaron, L. Lefèvre, and P. Primet, "A Session Aware Admission Control Scheme for next Generation IP Services", Proceedings of the 5th IEEE Consumer Communications and Networking Conference (IEEE CCNC 2008), Las Vegas, USA, January 2008.

[8] N. Ayari, D. Barbaron, L. Lefèvre, P. Primet, "SARA: A Session Aware Infrastructure for High Performance Next Generation Cluster-based Servers", The Australian Telecommunication Networks and Application Conference(ATNAC07), Christchurch, New Zealand, 2nd-5th December 2007.