# Session Based Access Control in Content Delivery Networks in Presence of Congestion

Novella Bartolini*
University of Rome ”La Sapienza”, Italy
novella@di.uniroma1.it

Emiliano Casalicchio
University of Rome ”Tor Vergata”, Italy
casalicchio@ing.uniroma2.it

Imrich Chlamtac
The University of Texas at Dallas
chlamtac@utdallas.edu

## Abstract

*To ensure probabilistic guarantees on quality of service in Content Delivery Networks (CDN), an access control support is needed that takes into account a proper differentiation of requests and performs session based decisions, managing different types of services and different service phases. In this paper we introduce a CDN architecture with access control capabilities at session aware access routers. We formulate a Markov Modulated Poisson Decision Process for access control that captures the heterogeneity of multimedia services and the variable availability of resources due to the network congestions that characterize a non-dedicated network environment. The structural properties of the optimal solutions are studied and considered as the basis for the formulation of heuristics that perform close to the optimal policy.*

## 1. Introduction

Content Delivery Networks (CDN) [17] are based on a placement of server replicas and requests redirection that guarantees resource availability, service quality and proximity of content to the user. However in many circumstances it is impossible to estimate the amount of resources required to fulfill the requests of service. Flash crowds and unpredictable link congestions could cause a critical performance degradation of some servers leaving few resources available for CDN services.

Typical CDN services consist in a sequence of temporally and logically related requests issued by the same client, forming a *session* [8, 5, 4]. The session concept must be at the basis of any access control mechanism in CDN as pointed out in [8, 4], and if a session has been admitted, its successive requests should also be admitted, especially during critical phases in which more revenue could be gained or lost.

Aim of this paper is to investigate the performance of session based access control policies in a CDN non-dedicated environment, with possible congestions. The problem of access control to web and application servers has been studied in literature concerning the case of web servers and server farms where the admission policy is performed by a dispatcher and the network links between the dispatcher and the content servers are dedicated. We consider a CDN architecture in which admission control procedures are performed by the access routers.

In [4, 7, 8] the problem of http session admission control is analyzed considering a single web server architecture and the effects of congestion are not considered. In [6] a single server architecture is analyzed, and the congestion on the links between the client and the server is also taken into account by the admission control policy by adjusting the rate of accepted requests to the target server according to the measured performance. In [12, 16] a distributed web site is considered where a dispatcher performs an access control based on short-time prediction of traffic requirements.

Unlike our work, the here mentioned schemes cannot be applied to a CDN environment in which servers are replicated and geographically distributed. In such a scenario, the pool of available servers is most of the time a subset of the servers known by the access routers. The absence of some servers is detected by the measurement activity performed by the access routers, in presence of overloads or congestion on the non dedicated response paths. Without loss of generality, we limit our analysis to two typical CDN services: informational web access and e-commerce

* The work of N. Bartolini has been partially supported by the WEB-MINDS Project, under the FIRB program

0-7695-2233-5/04 $20.00 © 2004 IEEE

requests and transactions. Like in [4, 8] we formulate models of service sessions that will be used to represent the lifetime of an accepted request. We propose a stochastic decision process to optimize performance parameters like the probability of disrupting an ongoing session due to lack of resources, the probability of successful session completion, that is the probability that a session is terminated due to the client's will, and the probability that a request is blocked at its first attempt of access by the admission control mechanism. The proposed decision model is based on a Markov Modulated Poisson Process (MMPP) [9] of the service session that captures the bursty nature of data and packetized traffic [18, 14], typical of multimedia network applications accessible through CDN. The congestion of the non-dedicated links is modelled by a Markov modulated dynamic of servers vacation. A structural analysis of the optimal solution is conducted to study the behavior of the optimal access policy.

To overcome the scalability limits of the analytical solution, the results of the structural analysis of the optimal policy are used to construct possible heuristics to be applied by the access routers. Performance comparisons between the heuristics and the optimal policies are also given by means of simulations.

The novelty of the proposed model resides in its applicability to multiple servers scenarios in non dedicated network environment, with heterogeneous sources of congestion.

In Section 2 we introduce two typical CDN types of service and their related Markov modulated phase model. In Section 3 a congestion model is introduced to take into account the presence of external traffic on the non dedicated links between the routers and the replica servers. In Section 4 we introduce a Markov Modulated Poisson Decision Process and related revenue optimization problems. In Section 5 we conduct a structural analysis of the optimal admission control policy and on the basis of this analysis we propose some heuristics. In Section 6 performance comparisons among optimal policies and heuristics are given.

## 2. Session models of CDN services

In [7, 11, 14, 18] it is suggested that MMPPs can be used to approximate or predict the burstiness of the input of a web server. We assume exponential arrivals of service requests, while the session duration is modelled by means of an alternation of idle and active phases, following a Markov modulated process of the lifetime of a service session.

In the simplest cases it is desired to differentiate between two classes of clients *premium* and *basic*, such that the premium clients receive better service than basic clients in case of overload. As premium class we introduce an e-commerce service with dynamic http requests, while as a basic class we consider a typical informational web session with static http requests. For simplicity, only two types of service are considered in this papers, although many other types of service could be considered like streaming or interactive games and others, and more complex service models can be studied by using the same methodology.

We refer to *informational web* requests as to traditional browsing through a site composed by static pages only. A session of this type of service will consist of few phases, possibly traversed many times. The session starts with a browsing phase A and follows alternating think phases B with browsing phases. During think phases the user spends some time thinking before deciding which next request to issue and the session does not consume resources with the exception of session identifiers. Figure 1 shows the session model for informational web services. The index "1" is used to differentiate this type of service from the second one, e-commerce, that will be introduced later. We assume exponential arrivals with average rate $\lambda_A^1$. When a request arrives, it enters the phase A in which a http request is issued to one of the servers, selected by the access router. Phase A requests are characterized by a resource consumption $b_A^1$ of 1 unit, $b_A^1 = 1$, and the single phase duration is exponentially distributed with average $\mu_A^1$. When the client gets the response, the request enters phase B in which the user thinks about the next request to issue. The think phase is characterized by null resource utilization: $b_B^1 = 0$, with the exception of a session identifier, while the average phase duration $\mu_B^1$ depends on the user behavior and is the completion rate of think phases $\mu_{\mathtt{t}}$. Both in phase A and in phase B there is a probability that the client voluntary terminates the session. To represent phase transitions and voluntary terminations, we introduce the transition probabilities between phases: $\pi_{AB}^1$, that is the probability of having a transition from phase A to phase B, and the opposite transition $\pi_{BA}^1$. New session arrivals only occur in phase A, meaning that a new informational web session is only started when a related http request is issued. Real traces of an informational site can be used to tune the values of the involved parameters. Since the particular choice of the site trace is not particularly meaningful for our purposes, we consider average val-

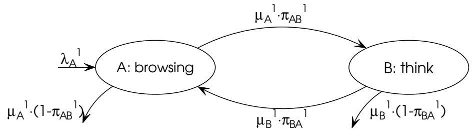

$\mu_A^1 \cdot \pi_{AB}^1$
$\lambda_A^1$
A: browsing
B: think
$\mu_A^1 \cdot (1-\pi_{AB}^1)$
$\mu_B^1 \cdot \pi_{BA}^1$
$\mu_B^1 \cdot (1-\pi_{BA}^1)$

**Figure 1. Session model of the informational web type of service.**

ues of different traces: $\lambda^1_A = 50 \;\; sec^{-1}$, $\mu^1_A = 100 \;\; sec^{-1}$, $\mu^1_B = \mu_{\mathtt{t}} = 0.05 \;\; sec^{-1}$, $\pi^1_{AB} = 0.95$ and $\pi^1_{BA} = 0.6$.

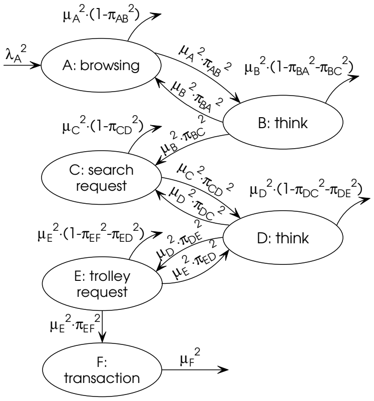


**Figure 2. Session model of the e-commerce type of service.**

During an *e-commerce* session, many phases can be considered in which dynamic http requests are issued. Figure 2 shows the e-commerce session phase model. As seen in [5, 4] the e-commerce session starts in a browsing phase `A` in which the client issues an http request to enter the site. After some back and forth between the browsing phase `A` and the think phase `B`, if the client is interested in a particular product, it enters the search request phase `C`, in which a query is submitted to the e-commerce database and a dynamic html page is produced with the related results. After another phase of thinking, represented by phase `D`, the client may decide to issue a new query to the database, going back to phase `C`, or to put some products in the trolley, entering phase `E`. Once entered phase `E` the session is considered very critical because its interruption due to congestion or overload potentially causes a profit loss. In all phases the client may willingly terminate the session. Phase `B` and `D` represent think phases with null resource consumption, that is $b^2_B = b^2_D = 0$, where the session in the other phases consumes a single resource unit, that is $b^2_A = b^2_C = b^2_E = b^2_F = 1$. Realistic parameters can be considered as follows: $\lambda^2_A = 10 \;\; sec^{-1}$, $\mu^2_A = 100 \;\; sec^{-1}$, $\mu^2_B = \mu^2_D = \mu_{\mathtt{t}} = 0.05 \;\; sec^{-1}$, $\mu^2_C = 0.333 \;\; sec^{-1}$, $\mu^2_E = 1 \;\; sec^{-1}$ and $\mu^2_F = 0.2 \;\; sec^{-1}$. Table I shows the transition probabilities between phases. As before, traces taken from real sites can also be used.

| $\pi_{**}$ | A | B | C | D | E | F | `exit` |
|---|---|---|---|---|---|---|---|
| A | 0 | 0.8 | 0 | 0 | 0 | 0 | 0.2 |
| B | 0.3 | 0 | 0.6 | 0 | 0 | 0 | 0.1 |
| C | 0 | 0 | 0 | 0.5 | 0 | 0 | 0.5 |
| D | 0 | 0 | 0.2 | 0 | 0.5 | 0 | 0.3 |
| E | 0 | 0 | 0 | 0.5 | 0 | 0.3 | 0.2 |
| F | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

**Table 1. Phase transition probabilities for an e-commerce site**

## 3. Congestion model

At the basis of CDN design, together with the replica server placement policy, is an active/passive measurement support that enables the access router to select the best suited replica among the set of replica servers it has knowledge about [13, 2, 19]. Congestion on the non-dedicated links between the access routers and the replica servers has an impact on the resource availability, making sometimes impossible the use of even unloaded servers. This means that even though the access router has knowledge of a set of fully replicated servers, for a total of $C$ resource units, every time a request arrives, it selects the replica server from a restricted *available pool* that is the set of servers that have enough free capacity and can be reached through a non congested link. A link is reputed to be under congestion effect if its expected round trip time is increased by an intolerable latency. The response time of a resource at the end of a congested link $T_{\mathtt{c}}$ is thus greater than the response time measured in non congested situations $T_{\mathtt{nc}}$. We repute that congestion level is intolerable when the observed response time $T_{\mathtt{obs}}$ is increased by more than a fixed percentage value $\alpha_{\mathtt{c}}$, that is when: $T_{\mathtt{obs}} = T_{\mathtt{c}} \geq T_{\mathtt{nc}}(1 + \alpha_{\mathtt{c}})$, where the value of $\alpha_{\mathtt{c}}$ is selected according to the required QoS levels.

The congested resources will be removed from the available pool until the measured state of the link goes back to normal conditions. The active sessions that are being processed from congested servers are migrated to non congested servers. If all requests are busy or congested and a newly congested replica server is in the middle of processing a session, it is removed from the available pool but is allowed to complete the elaboration of the current phase with an increased service time. If no more free and non congested resources are available at the epoch of the next phase transition, the session will be abruptly terminated.

We model congestion on the non dedicated links by means of the random variable $x_{\mathtt{c}}$, $(0 \leq x_{\mathtt{c}} \leq C)$. The value of this variable is governed by two possible events:

*congestion arrival* on a resource (a resource departs from available pool): it happens with negative exponential distribution, with average rate $\mu_{\mathtt{AP}}(C - x_{\mathtt{c}})$;

*congestion termination* on a resource (a resource goes back to the available pool): it happens with negative exponential distribution with average rate $x_{\mathtt{c}}\lambda_{\mathtt{AP}}$.

The state of the model can be described through an $(N+1)$ dimensional vector $\mathbf{x} = (\mathbf{x}^N, x_{\mathtt{c}})$, where the vector

$\mathbf{x}^N$ represents the phase occupancy of both the types of service: two phases for the web informational service and six phases for the e-commerce service, therefore $N = 8$ phases. The vector $\mathbf{x}^N$ is made up as follows: $x_1$ and $x_2$ represent the number of ongoing informational web sessions in phase A and B respectively, and $x_3, \ldots, x_8$ will instead represent the number of ongoing e-commerce sessions in phases A, . . ., F, respectively. The transition rates will also be defined according to the same convention used in the enumeration of the state variable. Therefore $\lambda_1 \triangleq \lambda^1_{\mathtt{A}}$ and $\lambda_3 \triangleq \lambda^2_{\mathtt{A}}$. Analogous enumeration will be used to define the outgoing rates $\mu_1, \cdots, \mu_8$, in place of $\mu^1_{\mathtt{A}}, \cdots, \mu^2_{\mathtt{F}}$ and for the capacity requirements $b_1, \cdots, b_8$, in place of $b^1_{\mathtt{A}}, \cdots, b^2_{\mathtt{F}}$.

Since the capacity of the replicated servers is limited to $C$ and the number of session identifiers is limited to $C^{\mathtt{ID}}$, $0 \leq \sum_{i=1}^{N} b_i x_i \leq C$ and $0 \leq \sum_{i=1}^{N} x_i \leq C^{\mathtt{ID}}$, while $0 \leq x_{\mathtt{c}} \leq C$. We assume that the replica servers are homogeneous. Therefore we can state that the congestion affects all classes and phases (with the exception of think phases) with latency effects that are proportional to the number of congested resources and to the occupancy level of the various classes. If $\beta_{\mathtt{c}}(\mathbf{x}) \triangleq x_{\mathtt{c}} / \sum_{i=1}^{N} b_i x_i$ is the percentage of congested resources in state $\mathbf{x}$, the congestion will affect $b_i x_i \beta_{\mathtt{c}}(\mathbf{x})$ resources for phase $i$ requests. Notice that if phase $i$ is a think phase, no congestion latency must be taken into account since there is no resource consumption with the only exception of session descriptors.

The average phase completion rate of phase $i$ is

$$\overline{\mu}_i(\mathbf{x}) \triangleq (1 - b_i)\mu_{\mathtt{nc}i} + b_i(\beta_{\mathtt{nc}}(\mathbf{x})\mu_{\mathtt{nc}i} + \beta_{\mathtt{c}}(\mathbf{x})\mu_{\mathtt{c}i}), \quad (1)$$

where $\mu_{\mathtt{nc}i}$ is the phase $i$ completion rate $\mu_i$, as described by the session lifetime models seen in section 2, and $\mu_{\mathtt{c}i} = \mu_i / (1 + \alpha_{\mathtt{c}})$.

## 4. A Markov modulated decision process for session based access control

Without loss of generality, we only consider two types of service where each ongoing session is modulated, among different phases of resource consumption and idle thinking times, thus creating a MMPP of services.

The state of the process is a vector of $N + 1$ components, and the state space can be defined as follows: $\Lambda = \{(x_1, \ldots, x_{N+1}) : \sum_{i=1}^{N} b_i x_i \leq C; \sum_{i=1}^{N} x_i \leq C^{\mathtt{ID}}; x_{N+1} \leq C; x_i \geq 0, i \in \{1, \ldots, N\}\}$.

We summarize the events that cause the dynamic of the process with the related rates. Arrivals in session initiating phases $i$ occur with rate $\lambda_i$ where $\lambda_i = 0$ if $i \neq 1, 3$ (phases 1 and 3 are the initial phases of sessions for the informational web and e-commerce type of service respectively).

Phase terminations happen at average rate $x_i \overline{\mu}_i$, where $\overline{\mu}_i$ is given by equation (1).

Resources abandon the available pool with rate $(C - x_{N+1})\mu_{\mathtt{AP}}$ while congestion terminates and the resources go back to the available pool with rate $x_{N+1}\lambda_{\mathtt{AP}}$.

A decision support is added to the process by defining a decision space and related costs and profits. A decision is an $N$ dimensional vector $\mathbf{a}$. Since we do not want the system to intentionally interrupt a session unless there are no more resources available, we only consider accept/reject decisions at the beginning of a new session. After being accepted, a session is kept alive as long as there are available and non congested resources. The indicators $a_i$ denote the admission, with value 1, or the denial of service, with value 0, of class-$i$ new session requests. Therefore $a_i$ is null if the phase $i$ is not the initial phase of any type of service (in our model $a_i = 0$ for $i \neq 1, 3$). The space of decision is $\mathcal{A} = \{\mathbf{a} = (a_1, \ldots, a_N) : a_i \in \{0, 1\}\}$. We refer to $\mathcal{S}$ as to the set of all feasible couples of vectors of the kind (*state,decision*).

The process we are describing is not uniform and the dwell time in each state is both state and decision dependent. The process can be uniformized [10, 15] at any rate $\Gamma$ that exceeds the maximum outgoing rate from any state (see [3] for details).

Let $\tilde{p}^{\mathbf{a}}_{\mathbf{xy}}$ denote the uniformized transition probability from state $\mathbf{x} = (\mathbf{x^N}, x_c)$ to state $\mathbf{y} = (\mathbf{y^N}, y_{\mathtt{c}})$ if the decision $\mathbf{a}$ is taken and $(\mathbf{x}, \mathbf{a}) \in \mathcal{S}$. We use $\mathbf{e_i}$ to denote the identity vector. The values of $\tilde{p}^{\mathbf{a}}_{\mathbf{xy}}$ are described below.

- New session request in starting session phase $i$, for any starting session phase $i$, (in the considered scenario $i = 1, 3$): $y_{\mathtt{c}} = x_{\mathtt{c}}$ and $\mathbf{y}^N = \mathbf{x}^N + \mathbf{e_i}$.

$$\tilde{p}^{\mathbf{a}}_{\mathbf{xy}} = \lambda_i a_i / \Gamma, \text{ where } a_i = 0 \text{ if } \sum_{k=1}^{N} b_k x_k + x_{\mathtt{c}} < C. \quad (2)$$

- Transition from phase $i$ to phase $j$: $y_{\mathtt{c}} = x_{\mathtt{c}}$ and $\mathbf{y}^N = \mathbf{x}^N - \mathbf{e_i} + \mathbf{e_j}$.
  A transition towards a think phase is always permitted. A transition towards and active phase is instead allowed, provided that enough free resources are available, only if there is no congestion or if the congestion doesn't affect the resources required to fulfil the phase transition request, that is when $\sum_{k=1}^{N} b_k x_k + (b_j - b_i) + x_{\mathtt{c}} \leq C$. We introduce the set $\mathcal{I}_i(\mathbf{x})$ as the set of phases $j$ that cannot be reached by a session coming from phase $i$ due to the problem of congestion. The set $\mathcal{I}_i(\mathbf{x})$ is empty if there is no congestion and phase transitions are always possible.

$$\tilde{p}^{\mathbf{a}}_{\mathbf{xy}} = x_i \frac{\overline{\mu}_i}{\Gamma} \pi_{ij}, \text{ if } j \notin \mathcal{I}_i(\mathbf{x}), \quad (3)$$

$$\mathcal{I}_i(\mathbf{x}) = \{j : b_j > C + b_i - (\sum_{k=1}^{N} b_k x_k + x_{\mathtt{c}})\}. \quad (4)$$

- Session termination in phase $i$: $y_{\mathtt{c}} = x_{\mathtt{c}}$ and $\mathbf{y}^N = \mathbf{x}^N - \mathbf{e_i}$.

The event of a session termination may occur for two reasons. A session can be terminated by the system due to congestion while attempting a phase transition towards an active phase $j$, that is when $j \in \mathcal{I}_i(\mathbf{x})$, with rate $x_i\overline{\mu}_i\pi_{ij}/\Gamma$. A session can also be voluntary terminated by the user at the end of phase $i$ with rate $x_i\overline{\mu}_i\pi_{ij}(1-\sum_{j=1}^{N}\pi_{ij})/\Gamma$. Therefore the overall probability of having a phase-$i$ termination is

$$\tilde{p}^{\mathbf{a}}_{\mathbf{xy}} = \frac{x_i\overline{\mu}_i}{\Gamma}(1 - \sum_{j \notin \mathcal{I}_i(\mathbf{x})} \pi_{ij}). \tag{5}$$

- Congestion arrival: $\mathbf{y}^N = \mathbf{x}^N$ and $y_{\mathtt{C}} = x_{\mathtt{C}} + 1$. Non congested resources may exit from the available pool due to congestion on the non-dedicated links with rate $\mu_{\mathtt{AP}}(C - x_{\mathtt{C}})/\Gamma$. Thence

$$\tilde{p}^{\mathbf{a}}_{\mathbf{xy}} = \frac{\mu_{\mathtt{AP}}(C - x_{\mathtt{C}})}{\Gamma} \text{ if } x_{\mathtt{C}} < C. \tag{6}$$

- Congestion termination: $\mathbf{y}^N = \mathbf{x}^N$ and $y_{\mathtt{C}} = x_{\mathtt{C}} - 1$. Congested resources may return to normal conditions and become available again to serve requests, with rate $\lambda_{\mathtt{AP}}x_{\mathtt{C}}/\Gamma$. Thence

$$\tilde{p}^{\mathbf{a}}_{\mathbf{xy}} = \frac{\lambda_{\mathtt{AP}}x_{\mathtt{C}}}{\Gamma} \text{ if } x_{\mathtt{C}} > 0. \tag{7}$$

- Dummy transitions from each state to itself: $\mathbf{y} = \mathbf{x}$. This transitions are added to the chain of the original, non uniform process, in agreement with the uniformization procedure:

$$\tilde{p}^{\mathbf{a}}_{\mathbf{xy}} = \frac{1}{\Gamma}\{\Gamma - [\sum_{i=1}^{N}(\lambda_i a_i + x_i\overline{\mu}_i) + (C - x_{\mathtt{C}})\mu_{\mathtt{AP}} + x_{\mathtt{C}}\lambda_{\mathtt{AP}}]\}. \tag{8}$$

The transitions that are not listed above have null probability.

## 4.1. Profits and losses during session lifetime

Aim of this section is to give a formulation of a cost/profit function that associates penalties and incomes to state, events and decisions. The admission control will decide whether a new session request should be admitted or not. If the new session request is rejected, a rejection penalty will be paid. Phase transitions are not subject to the admission control, and all the subsequent phases of an admitted session are admitted provided that enough non congested resources are available. If there are no available resources to complete a session the system will incur an interruption penalty usually higher than the rejection penalty. On the other hand, if a session is successfully completed, that is the user willingly terminates the session, more luckily with a purchase, a profit is gained.

We introduce the penalty $H_{\mathtt{EC}}$ to be paid by the system for the denial of service to an e-commerce request. A penalty $H_{\mathtt{IW}}$ is incurred by the system when an informational web request is refused, where $H_{\mathtt{EC}} > H_{\mathtt{IW}}$.

A second type of penalty relates to the interruption of an ongoing session. This is not decision related, but is in most cases due to an underestimation of the congestion problem or of the load situation. In the case of the informational web type of service, none of the phases is particularly critical. The transitions from phase A to phase B are always admitted since they bring the system from an active processing phase to a think phase. The opposite transition from phase B to phase A is instead permitted only if there are available and non congested resources, otherwise a penalty $H_{\mathtt{TA\_IW}}$ is incurred.

In the case of the e-commerce type of service, the phase transitions A-B, C-D and E-D, are always admitted since they are directed towards a think phase. Transitions B-C and D-E are not considered very critical and if the session is interrupted during these transition the system will incur a penalty $H_{\mathtt{TA\_EC}}$ that is less than the penalty $H_{\mathtt{AA\_EC}}$ that the system will pay in case of interruption of the session during the very critical transition from phase E to phase F.

In order to make the system accept a new session only if it is likely to guarantee continuity of service until the end of the session, the penalties for the denial of service will be lower than any phase interruption penalty.

Apart from the listed transition related costs, we consider a state related cost, to which we refer as $H_{\mathtt{BC}}$, that is paid as long as the system persists in a busy and congested state.

We now introduce the profits associated to the successful completion of a session. A profit $V_{\mathtt{IW}}$ is gained when the user terminates an informational web session. For what concerns the e-commerce type of service, if the user terminates the session with a purchase, a profit $W_{\mathtt{EC}}$ is gained, while if there is no purchase a lower profit $V_{\mathtt{EC}}$ is obtained for the successful termination of the session.

Based on the costs and profits here introduced, we define the uniformized cost function $r_{\mathtt{cost}}(\mathbf{s}, \mathbf{a})$ and the uniformized reward function $r_{\mathtt{rew}}(\mathbf{s}, \mathbf{a})$ (see [3] for more details).

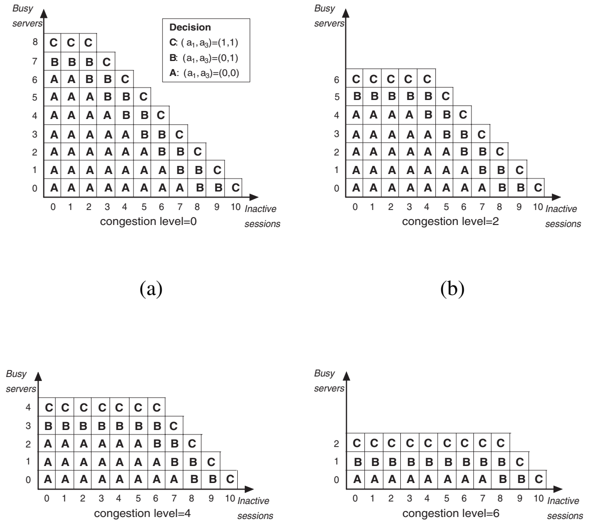
The Linear Programming (LP) formulation associated with our decision process for the minimization of the average cost is:

$$\begin{array}{ll}
\min \sum_{(\mathbf{s},\mathbf{a}) \in \mathcal{S}} [r_{\mathtt{rew}}(\mathbf{s},\mathbf{a}) - r_{\mathtt{cost}}(\mathbf{s},\mathbf{a})] & \cdot x_{\mathbf{sa}} \\
x_{\mathbf{sa}} \geq 0 & (\mathbf{s},\mathbf{a}) \in \mathcal{S} \\
\sum_{(\mathbf{s},\mathbf{a}) \in \mathcal{S}} x_{\mathbf{sa}} = 1 & \\
\sum_{\mathbf{a} \in \mathcal{A}} x_{\mathbf{ja}} = \sum_{(\mathbf{s},\mathbf{a}) \in \mathcal{S}} \tilde{p}^{\mathbf{a}}_{\mathbf{sj}} x_{\mathbf{sa}} & \mathbf{j} \in \Lambda
\end{array} \tag{9}$$

where $x_{\mathbf{sa}}$ is the probability for the system to be in state s and at the same time to take decision $\mathbf{a}$. The problem (9) can be solved by means of value iteration and the corresponding optimal solution will been named OPT.

## 5. Structural analysis of the optimal admission policy and heuristics formulation

An analytical study of the properties of the optimal policy has not been conducted due to the high dimensionality of the Markovian process. Nevertheless the value iteration method has been adopted to obtain the optimal policy in some significant cases. The purpose of this analysis is to obtain clues for the formulation of possible heuristics to be adopted in more general scenarios where the analytic methodology could not scale.



| Busy servers | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 8 | C | C | C | | | | | | | | |
| 7 | B | B | B | C | | | | | | | |
| 6 | A | A | B | B | C | | | | | | |
| 5 | A | A | A | B | B | C | | | | | |
| 4 | A | A | A | A | B | B | C | | | | |
| 3 | A | A | A | A | A | B | B | C | | | |
| 2 | A | A | A | A | A | A | B | B | C | | |
| 1 | A | A | A | A | A | A | A | B | B | C | |
| 0 | A | A | A | A | A | A | A | A | B | B | C |

(a)

| Busy servers | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 6 | C | C | C | C | C | | | | | | |
| 5 | B | B | B | B | B | C | | | | | |
| 4 | A | A | A | A | B | B | C | | | | |
| 3 | A | A | A | A | A | B | B | C | | | |
| 2 | A | A | A | A | A | A | B | B | C | | |
| 1 | A | A | A | A | A | A | A | B | B | C | |
| 0 | A | A | A | A | A | A | A | A | B | B | C |

(b)

| Busy servers | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 4 | C | C | C | C | C | C | C | | | | |
| 3 | B | B | B | B | B | B | B | C | | | |
| 2 | A | A | A | A | A | A | B | B | C | | |
| 1 | A | A | A | A | A | A | A | B | B | C | |
| 0 | A | A | A | A | A | A | A | A | B | B | C |

(c)

| Busy servers | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | C | C | C | C | C | C | C | C | C | | |
| 1 | B | B | B | B | B | B | B | B | B | C | |
| 0 | A | A | A | A | A | A | A | A | B | B | C |

(d)

**Figure 3. Optimal policy ($C = 8$, $C^{\texttt{ID}} = 10$)**

Figure 3 shows the behavior of the optimal policy when the traffic parameters are those described in section 2 for each class of service, the number of available resources is $C = 8$ and $C^{\texttt{ID}} = 10$, the costs are $H_{\texttt{EC}} = 10.000$, $H_{\texttt{IW}} = 5.000$, $H_{\texttt{TA\_EC}} = 11.000$, $H_{\texttt{AA\_EC}} = 100.000$, $H_{\texttt{TA\_IW}} = 6.000$, $H_{\texttt{BC}} = 100$ and the rewards are $V_{\texttt{IW}} = 5.000$, $V_{\texttt{EC}} = 9.000$ and $W_{\texttt{EC}} = 110.000$. The number of inactive sessions that is indicated on the x-axis of these figures, represents the number of sessions in thinking phase during observation.

Though not generalizable, figure 3 shows a structure of the optimal policy that holds in most of the analyzed scenarios. In most cases the optimal policies consists in reserving resources (computational capacity and identifiers) to the high priority customers (e-commerce stream of requests). The amount of reserved resources strictly depends on the traffic congestion.

Numerical results show a similar behavior of the optimal policy in many different scenarios. For this reason we considered the following heuristic (HEU) that mimics the behavior of the optimal policy (OPT). The HEU policy reserves $K_{\texttt{rss}}$ units of server capacity and $K_{\texttt{rsID}}$ session identifiers to the high priority stream of session activation requests.

We refer to $x_{\texttt{t}}$ as to the number of ongoing sessions in the think phase, $x_{\texttt{t}} \triangleq \sum_{i=1}^{N}(1-b_i)x_i$, while the number of sessions in active phases is $x_{\texttt{busy}} \triangleq \sum_{i=1}^{N} b_i x_i$, and $x_{\texttt{c}}$ is the number of congested units of server capacity. We define the following threshold values: $T^{\texttt{s}} \triangleq C - K_{\texttt{rss}}$ and $T^{\texttt{ID}} \triangleq C - K_{\texttt{rsID}}$.

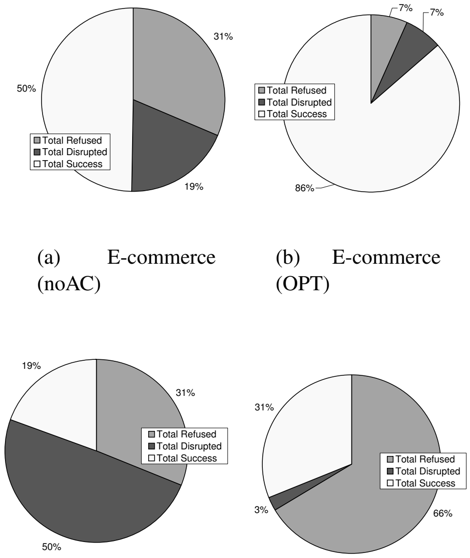
The HEU policy can be formulated in the following way:

- If $x_{\texttt{busy}} < \min\{T^{\texttt{ID}} - x_{\texttt{t}}; T^{\texttt{s}} - x_{\texttt{c}}\}$ take decision $(a_1, a_3) = (1, 1)$, that is give service to both streams of requests.
- If $\min\{T^{\texttt{ID}} - x_{\texttt{t}}; T^{\texttt{s}} - x_{\texttt{c}}\} \leq x_{\texttt{busy}} < \min\{C^{\texttt{ID}} - x_{\texttt{t}}; C - x_{\texttt{c}}\}$ take decision $(a_1, a_3) = (0, 1)$, that is give service only to e-commerce requests.
- If $x_{\texttt{busy}} \geq \min\{C^{\texttt{ID}} - x_{\texttt{t}}; C - x_{\texttt{c}}\}$ take decision $(a_1, a_3) = (0, 0)$, that is no new session can be admitted due to lack of available resources.

## 6. Simulative comparisons

In this section we analyze the effects of the policies introduced in section 5. Simulations are conducted on the OPNET simulator [1]. We provide performance comparisons among the optimal policy (OPT) and the heuristics (HEU) with different choices of the threshold parameters. A trivial policy, consisting in doing nothing to improve performance, will be named noAC, and will be used as a benchmark for comparisons. With the noAC policy both streams of session activation requests are treated alike and no discrimination is done between service classes. Figure 4 points out the effect of the optimal admission control policy in a scenario where $C = 8$, $C^{\texttt{ID}} = 10$, and where the traffic parameters of the informational web class of requests are $\lambda_A^1 = 30 \ \ sec^{-1}$, $\mu_A^1 = 100 \ \ sec^{-1}$, $\mu_B^1 = \mu_{\texttt{t}} = 0.05 \ \ sec^{-1}$, $\pi_{AB}^1 = 0.95$ and $\pi_{BA}^1 = 0.6$, and for the e-commerce class of requests $\lambda_A^2 = 30 \ \ sec^{-1}$, $\mu_A^2 = 100 \ \ sec^{-1}$, $\mu_B^2 = \mu_D^2 = \mu_{\texttt{t}} = 0.05 \ \ sec^{-1}$, $\mu_C^2 = 0.333 \ \ sec^{-1}$, $\mu_E^2 = 1 \ \ sec^{-1}$ and $\mu_F^2 = 0.2 \ \ sec^{-1}$.

Figure 4 shows that the introduction of the optimal control policy has a negative impact on the informational web stream because there is an increase in the blocking probability of new session activation requests. Although there is a high blocking probability of new requests, the informational web stream of requests also has a benefit in a reduced session disruption probability and in an increased successful completion probability. The reduced amount of infor-



(a) E-commerce (noAC)

(b) E-commerce (OPT)

(c) Informational web (noAC)

(d) Informational web (OPT)

**Figure 4. Probability of successful termination of e-commerce and informational web requests.**

mational web requests that gains access to the CDN service when an admission control policy is applied, is managed with better performance than in the case with no admission control. The high priority stream, that is the e-commerce stream of requests, encounters an increased performance both in terms of reduced blocking probability and in terms of increased successful termination probability when the optimal admission control policy is applied.

Since finding the optimal policy is computationally intensive, the heuristic HEU is considered and its behavior is analyzed with two different choices of the threshold values. We name HEU1 the heuristic HEU when $K_{\tt rss} = 4$ units of server capacity and $K_{\tt rsID} = 5$ session identifiers are reserved to the high priority stream of session activation requests, while we name HEU2 the heuristics where $K_{\tt rss} = 2$ and $K_{\tt rsID} = 3$.

The performance of the admission control policies can be measured from the service provider point of view in terms of revenues. As global index to measure the revenues from the provider point of view we use the average value of the reward function ($W = \sum_{(\mathbf{s},\mathbf{a})\in\mathcal{S}}[r_{\tt rew}(\mathbf{s},\mathbf{a}) - r_{\tt cost}(\mathbf{s},\mathbf{a})]x_{\mathbf{sa}}$). The average reward function may be positive or negative depending on the considered scenario and on the values of penalties and profits associated to the actions. Figure 5 points out the very little difference between OPT and HEU2 in terms of average reward function. HEU2 is in fact the heuristic that mostly mimics the optimal policy of figure 3.

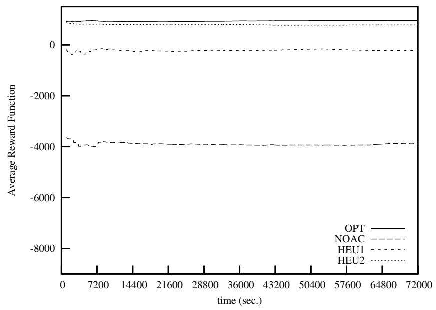


**Figure 5. Average reward function**

Analyzing the performance both from the service provider and from the client point of view the analysis of the rate of successful termination of e-commerce requests at the transaction phase is also very significative. The trend of this measure is shown in figure 6 where the HEU2 policy shows a good approximation of the behavior of the OPT policy.

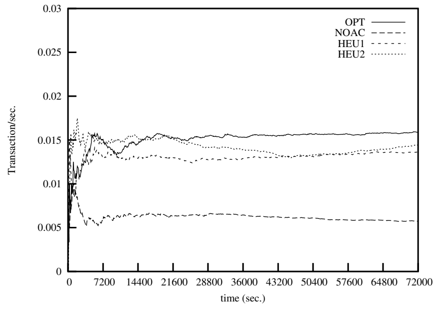


**Figure 6. Rate (requests/seconds) of e-commerce session successful termination with a transaction.**

Figures 7 and 8 show that the OPT policy performs best in terms of e-commerce request blocking probability but with a significant loss in terms of informational web blocking probability. The HEU2 policy shows a good trade-off between the blocking probabilities of the two streams of requests, with a very low blocking probability of e-commerce requests and an acceptable level of blocking probability of the low priority stream of informational web requests.

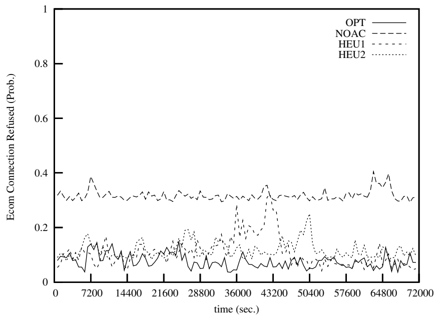


**Figure 7. E-commerce request blocking probability**

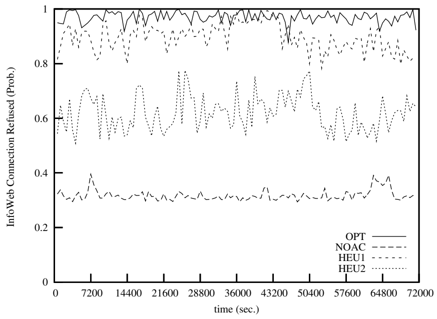


**Figure 8. Informational web request blocking probability**

## 7. Conclusions

This paper addresses the problem of access control in content delivery networks based on service classification and prioritization. Two session models are introduced to study typical services. A content delivery network is modelled as a service center where the presence of non dedicated links subject to external traffic is modelled as a Markov modulated process of server vacation. The problem of session based access control is analyzed as a decision problem, that yields an optimal solution that shows in most cases the behavior of a congestion-dependent reserved resources policy. Though computationally heavy, the study of the structure of the optimal policy gives suggestions on the formulation of possible heuristics. The proposed heuristics are studied by means of simulations showing the possibility of a trade-off solution between prioritizing the e-commerce stream of requests and compromising the traffic of informational web services, thus improving both the user's perceived quality and the service provider's revenue.

## References

[1] Opnet technologies inc. *http://www.opnet.com.*

[2] G. Agarwal, R. Shah, and J. Walrand. Content distribution architecture using network layer anycast. *Proc. of IEEE WIAPP 2001.*

[3] N. Bartolini, E. Casalicchio, and I. Chlamtac. Session based access control in content delivery networks. *TR-WEBMINDS-23, http://web-minds.consorzio-cini.it/activities/index.php*, 2004.

[4] J. Carlstrom and R. Rom. Application aware admission control and scheduling in web servers. *Proc. of IEEE INFOCOM 2002.*

[5] H. Chen and P. Mohapatra. Session-based overload control in qos-aware web servers. *Proc. of IEEE INFOCOM 2002.*

[6] X. Chen and J. Heidemann. Experimental evaluation of an adaptive flash crowd protection system. *http://www.isi.edu/div7/publication_files/tr-203-573.pdf.*

[7] X. Chen, P. Mohapatra, and H. Chen. An admission control scheme for predictable server response times for web accesses. *Proceedings of WWW 2001.*

[8] L. Cherkasova and P. Phaal. Session based admission control: a mechanism for peak load management of commercial web sites. *IEEE Trans. on Computers*, 51(6), 2002.

[9] W. Fischer and K. Meier-Hellstern. The markov-modulated poisson process (mmpp) cookbook. *Performance Evaluation*, 18(2), 1993.

[10] D. P. Heyman and M. J. Sobel. *Stochastic Models in Operations Research.* McGraw-Hill, 1984.

[11] A. Iyengar, M. Squillante, and L. Zhang. Analysis and characterization of large-scale web server access patterns and performance. *Proc. of WWW 1999.*

[12] M. Kihl and N. Widell. Admission control schemes guaranteeing customer qos in commercial web sites. *Proc. of NetCon 2002.*

[13] K. Mase, A. Tsuno, Y. Toyama, and N. Karasawa. A web server selection algorithm using qos measurement. *Proc. of ICC 2001.*

[14] R. Morris and D. Lin. Variance of aggregated web traffic. *Proc. of IEEE INFOCOM 2000.*

[15] H. C. Tijms. *Stochastic modelels. An algorithmic approach.* John Wiley & Sons, 1994.

[16] A. Verma and S. Ghosal. On admission control for profit maximization of networked service providers. *Proc. of WWW 2003.*

[17] D. C. Verma. *Content Distribution Networks.* John Wiley & Sons Inc., 2002.

[18] T. Yoshihara, S. Kasahara, and Y. Takahashi. Practical time-scale fitting of self-similar traffic with markov-modulated poisson process. *Telecommunication Systems*, 2001.

[19] E. Zegura, M. Ammar, Z. Fei, and S. Battacharjee. Application-layer anycasting: a server selection architecture and use in a replicated web service. *IEEE/ACM Trans. on Networking*, 8(4), 2000.