Towards Multi-Service Traffic Shaping in Two-Tier Enterprise Data Centers

Yesid Jarma Alviz, Marcelo Dias de Amorim, Yannis Viniotis

Towards Multi-Service Traffic Shaping in Two-Tier Enterprise Data Centers

Yesid Jarma*, Marcelo Dias de Amorim*, and Yannis Viniotis†,‡

*UPMC Sorbonne Universités, Paris, France
†North Carolina State University, Raleigh, NC, USA
‡IBM Software Group, Research Triangle Park, NC, USA

***Abstract*—In Enterprise Data Centers (EDC), service providers are usually governed by Client Service Contracts (CSC) that specify, among other requirements, the rate at which a service should be accessed. The contract limits the rate to no more than a number of service requests during a given observation period. In two-tier setups, a cluster of Service-Oriented Networking (SON) Appliances form a pre-processing tier that accesses services in the service tier. SON Appliances *locally* shape the flow of requests to enforce the *global* rate defined in the CSC. Off-the-shelf SON Appliances present architectural limitations that prevent them from being used to efficiently perform traffic shaping in the presence of *multiple* service hosts. In this paper, besides identifying these limitations, we provide two contributions in this field. First, we introduce a SON Appliance architecture fit for *multi-service* traffic shaping. Second, we propose and validate an algorithm for *multipoint-to-multipoint* service traffic shaping in two-tier EDCs. We show via simulation that our approach solves the multipoint-to-multipoint service traffic shaping problem while pushing the system to its maximum capacity.**

I. Introduction

The Internet changed the way business is conducted worldwide. To remain competitive, businesses have been implementing information technology support for business processes over the years. The current trend is to have applications located in Enterprise Data Centers (EDCs), in which computing operations are able to switch over between machines in a transparent way, maintaining user sessions, application availability, and access to data resources. In this context, Service-Oriented Architectures (SOA) have become the main solution for the integration of applications and technologies in the business domain. SOA can be implemented by dint of different technologies such as Web Services (WS), Enterprise Service Bus (ESB), and middleware appliances. The latter are frequently referred to as Service-Oriented Networking (SON) Appliances, which consist on specific hardware that provides dedicated operations such as accelerated XML processing, functional offloading, service integration, and intelligent routing [1].

Modern EDCs are usually deployed following a two-tier architecture, where border routers located at the edge of the Enterprise network send client requests to a cluster of SON Appliances on the first tier in order to get access to service and data clusters forming the second tier. Access to services is governed by Client Service Contracts (CSCs) dictated by Service-Level Agreements (SLAs), which aim at protecting EDC resources. CSC define the maximum rate at which a service may be accessed in terms of a *number of requests during an enforcement period* defined by IT administrators. In a two-tier setup, SON Appliances are responsible for limiting the access (i.e., controlling the traffic) to application servers in order to protect them from being unduly overwhelmed. This also allows better satisfying business goals and meeting customer's performance level expectations.

Several approaches, using both static and dynamic credit-based strategies, have been developed in order to enforce the rate specified in the CSC [2], [3]. Nevertheless, these solutions have so far only considered the *multipoint-to-singlepoint* case where a cluster of SON Appliances *shapes* service traffic toward a *single* service instance. Moreover, current off-the-shelf SON Appliances present architectural limitations that prevent them from efficiently performing traffic shaping in the presence of *multiple* service hosts.

In this paper, we identify the need for implementing multiple exit queues at each SON Appliance when these are used to access multiple service instances. We propose an algorithm for *multipoint-to-multipoint* service traffic shaping in two-tier EDCs that explores the communication capabilities to dynamically adapt to the changes in the global state of the system. We show via simulation that our approach, when combined with the strategic use of one output queue per each service instance, effectively solves the multipoint-to-multipoint service traffic shaping problem and pushes the system to its maximum capacity. In summary, the contributions of our work are:

- We identify the need for a queuing management scheme that is more adapted for scenarios where multiple appliances access concurrently multiple services.
- We propose an algorithm for shaping request traffic towards several services when the number of output queues is the same as the number of services.
- We validate our algorithm via extensive simulations and show that our approach is able to push the system to its maximum capacity while respecting the service contracts.

II. Background

A. Considered system

We consider an enterprise data center deployed as a two-tier logical system architecture as the one shown in Fig. 1. This system is composed of the following entities:

- *Border routers*. These are the first entry points of the system. They are responsible for terminating customers' TCP connections, assembling XML formatted requests,

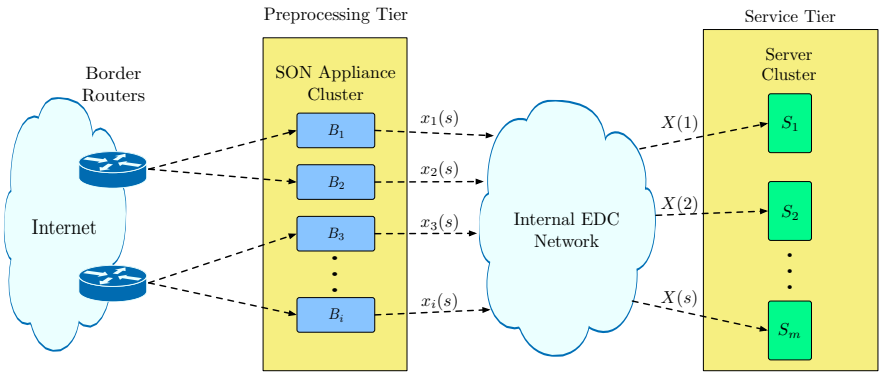


Fig. 1. Considered two-tier system architecture. Customers outside the EDC must interact with the *preprocessing tier* (Tier 1) in order to get access to the *service tier* (Tier 2).

and forwarding them to the *preprocessing tier*. Also, they are eventually in charge of distributing the service load among the appliances without any deep-content inspection.

- *Preprocessing tier*. The main building block of the preprocessing tier is the SOA middleware. The preprocessing tier is in charge of performing operations such as accelerated XML processing, functional offloading, service integration, and intelligent routing.
- *Service tier*. It is composed of clusters of service servers that can be application servers or storage servers. This entity processes the bulk of service requests. It also specifies the rate at which the services may be accessed or Service Access Requirement (SAR).

B. Off-the-shelf SON Appliances

One of the major issues that has prevented a wider adoption of SOA is performance [4]. Indeed, as the time needed to parse an XML document can take up to a few minutes [5], the response time of a service instance is potentially large. To better satisfy business goals, service providers use SOA middleware that provides accelerated XML processing called Service-Oriented Networking (SON) Appliances [1], [6].

SON Appliances can implement a number of functions, which include functional offloading, service integration, and intelligent routing [7]. In addition to providing these functions, in two-tier EDC setups, SON Appliances may also be responsible for controlling the rate at which client requests are sent to the service hosts. We refer to this problem as *service traffic shaping*.

C. Multipoint-to-point service traffic shaping

Typically, a service instance is accessed from a single SON Appliance; therefore, the traffic from the gateway to the service host follows a *point-to-point* pattern. A single entry point provides the advantage of simplified service access management. Furthermore, since point-to-point traffic shaping is a well-studied problem in the networking space, well-known solutions from packet/ATM networks can be applied [8], [9], [10]. Nevertheless, in two-tier EDC deployments, the problem is fundamentally different. In classic packet/ATM networks, the resource protected by the shaping function is typically link bandwidth and buffer space, the units of which are precisely defined and measurable. Service Level Agreements (SLAs) are standardized by industrial bodies and CSCs are very well defined. In contrast, in EDC setups, the resource protected by the shaping function is CPU processing power on the service instance, which varies based on the type, size, and content of the request documents. Moreover, in this context, *CSC contracts are neither precisely defined nor standardized.*

We are particularly interested in the SAR definition that, in general, follows the following format: "Limit the rate to a service provider to no more than X requests per second with an observation/enforcement period of T seconds", where an enforcement period is a time interval during which the *aggregate* of requests sent to the service host by all the appliances cannot exceed $X \times T$. Note that, in this particular case, since "requests" are defined in units of XML documents, CPU processing time at the service instance is not known exactly. Furthermore, this SAR does not include additional requirements such as a maximum burst size. On the other hand, in traditional networks, the parameters specified in SLAs, for example, include, in addition to an average rate, a peak rate (which is the maximum rate at which packets can be sent in a short time interval), and a burst size (a limit for the number of packets to be transmitted in a short time interval).

Another fundamental difference is that in the classic networking environment, traffic shaping has local scope, since traffic is in the form of a single connection. In a two-tier EDC environment, as the one under consideration in this manuscript, service customers access either a *single* or *multiple* service instance(s) from *multiple* entry points. We refer to this problems as *multipoint-to-point* and *multipoint-to-multipoint service traffic shaping*, respectively. The existence of multiple entry points may be dictated by policy (e.g., the presence of multiple security zones) or performance requirements (e.g., clusters of SON Appliances); the desired effect is "global" shaping. *The challenge is therefore to globally enforce the CSC contract by taking local actions at each entry point.*

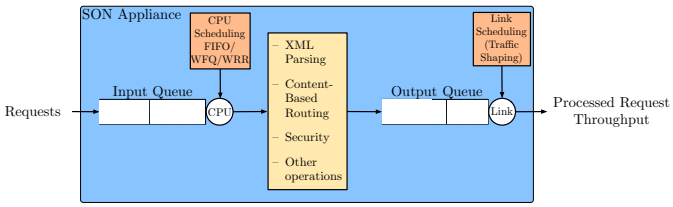


Fig. 2. Internal architecture of off-the-shelf SON Appliances.

III. SON Appliances and Traffic Shaping

A. From multipoint-to-point to multipoint-to-multipoint shaping

As mentioned in Section II, we consider an enterprise data center deployed as a two-tier logical system architecture. Fig. 1 shows in more details the components of each tier. In contrast to existing approaches, we specifically consider, for this body of work, the case where multiple SON Appliances access concurrently multiple service hosts and must shape traffic towards the latter in order to protect them from being unduly overwhelmed. We also consider that each different service host defines its own SAR and that all of the appliances in the cluster are able to process requests for all service instances. In this context, the key challenge is to enforce *each global per-service* CSC contract by taking *local actions* at each appliance.

We start by formalizing the per-service SAR specified by the CSC. Let $x_i(s)$ be the number of requests appliance i is allowed to send to service host s (for the remainder of this paper, we will refer to this value as i's credits for s). As per the SAR, the preprocessing tier must guarantee that the cumulated number of requests sent by all the appliances in the cluster towards service s must respect:

$$\sum_{i=1}^{B} x_i(s) \leq X(s) \times T. \tag{1}$$

To the best of our knowledge, there is no known published strategy for guaranteeing the specification described in Eq. 1. Consequently, in production environments, the simplest solution used nowadays is to apply a static, homogeneous policy referred to as Manual and Static Allocation (MSA). This policy assigns the same rate to each appliance at all times:

$$x_i(s) = \left\lfloor \frac{X(s) \times T}{B} \right\rfloor, \qquad \forall i \in [1, \ldots, B]. \tag{2}$$

MSA, although simple, is quite inefficient as it only provides satisfactory performance when the incoming traffic rates at the appliances are identical. In practice, this is hardly the case as there is no a priori knowledge on the rates at which the preprocessing tier will receive requests from the clients. Moreover, even though the border routers perform some load balancing, since the delays required for request preprocessing are highly heterogeneous, the rate at which the appliances are ready to send documents to the service instances does not follow a uniform law. Therefore, a number of appliances may hold queued requests while others remain idle. As a consequence, the system is unable to exploit its maximum capacity.

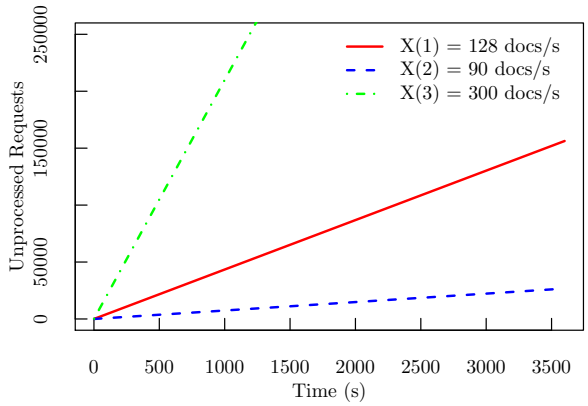


Fig. 3. Number of requests left unprocessed overtime when using a static credit allocation scheme together with a single output FIFO queue for multipoint-to-multipoint service traffic shaping.

B. Architectural shortcomings of off-the-shelf SON Appliances

The main building block of the preprocessing tier in the considered two-tier EDC is the set of off-the-shelf SON Appliances (see Section II-B). Fig. 2 shows the main internal components of a SON Appliance. Inbound XML formatted requests are put in an entry queue where a CPU scheduler allocates the necessary resources for parsing requests and performing other operations as authentication and validation. Once a request is processed, it is placed in an output queue, which follows a FIFO service discipline, before being sent to the correct service host. At this stage, the appliance is responsible for enforcing each per-service CSC.

The current architectural design of off-the-shelf SON Appliances makes them unfit for efficiently shaping traffic towards multiple different service hosts. Indeed, because of the use of a single FIFO output queue, as soon as the lowest per-service SAR is fulfilled, when a request for a service which no longer has credits reaches the front of the queue, it blocks all requests behind it even if there are credits left for other services. This shortcoming has major impact on the efficiency of the system, as it will be shown later in this section.

C. Arguments towards new algorithms

As mentioned before, the use of a single FIFO output queue severely limits the performance of SON Appliances when shaping traffic towards multiple service hosts. To illustrate this, we undertook a series of simulations, where a cluster of six SON Appliances access concurrently a cluster of three service hosts. We define a different SAR for each service host. Fig. 3 shows the impact of using a single FIFO exit queue together with MSA over time. As per Definition **??**, the optimal algorithm would leave no unprocessed requests overtime. Nevertheless, after only an hour of simulated time, over 150,000 requests for Service 1, around 3,000 for Service 2, and over 700,000 for Service 3 have been left unprocessed. Clearly, the qualitative shortcomings of both MSA and off-the-shelf

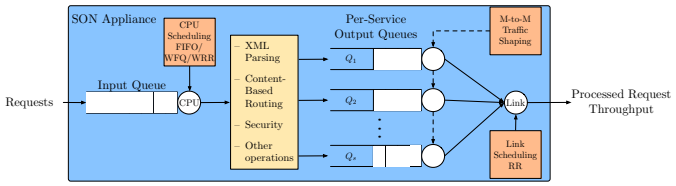


Fig. 4. Proposed internal architecture of SON Appliances for multipoint-to-multipoint service traffic shaping.

appliances severely hampers the system. As a consequence, the system is unable to exploit its maximum capacity. *Given the costs of implementing EDCs and issues inherent to their provisioning, it is imperative to design efficient algorithms that optimize the overall utilization of the system.*

IV. Toward multipoint-to-multipoint service traffic shaping

A. Appliance requirements

In order to properly perform service traffic shaping in two-tier EDC setups with multiple service hosts, we propose some simple architectural changes to current off-the-shelf SON Appliances. First, we propose the use of a single output queue for each service present in the service tier. Second, because there are now several output queues accessing concurrently a single output link, we propose the use of a simple Round-Robin scheduling algorithm for sharing the link resource among the output queues. Fig. 4 depicts the proposed internal architecture.

B. A multipoint-to-multipoint service traffic shaping algorithm

As the next step, we propose a credit-based algorithm that relies on the notion of *enforcement subperiod* [2]. The enforcement period is divided into K subperiods. During each subperiod, the algorithm measures the number of requests that were processed and forwarded, and current queue sizes for each service host, and adapt its sending rate for the next subperiod by assigning *credits* to each appliance. One *credit* allows an appliance sending one processed request to a service host.

At the beginning of each enforcement subperiod each appliance starts the credit allocation scheme by calculating its own per-service weight. First, the n-th request in the queue for service s is assigned a weight w_n. The weight of a request is inversely proportional to its size and depends directly on the number of measurement subperiods used. For simplicity, we make the assumption that, on average, the processing time of a request is proportional to the length (size) of the request.[1] Therefore, large requests, which take longer to process, will have smaller weights. The weight of appliance B_i for service s, during subperiod k is the sum of the weights of all the requests in the output queue for service s:

[1]In reality, the average processing time is proportional to length of the requests (e.g., due to parsing the entire XML document for checking well-formedness) as well as other factors, like the actual content of the XML document.

$$W_{B_i}(s,k) = \sum_{n=1}^{Q} w_n. \tag{3}$$

Once each appliance calculates its own weight, it determines the number of per-service credits it is allocated during the next subperiod under a weighted strategy:

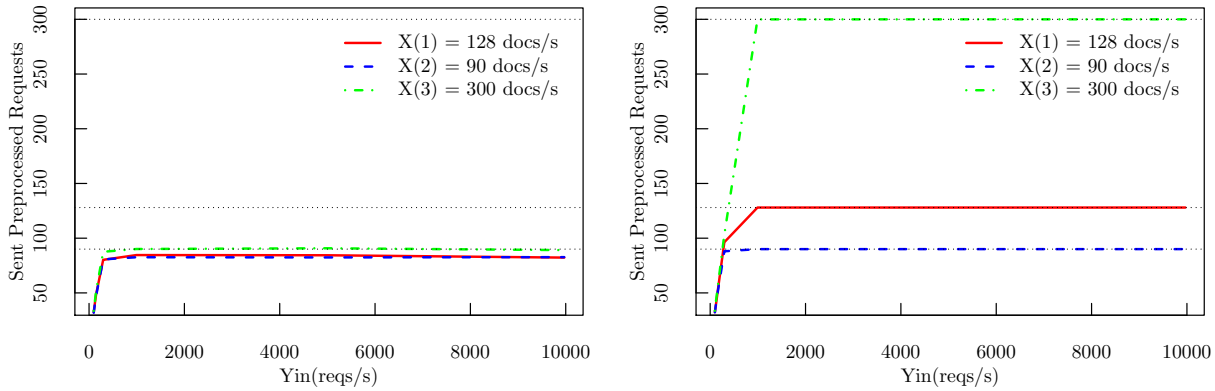
$$x_i(s,k) = \left\lceil D(s) \times \frac{W_{B_i}(s,k)}{\sum_{n=1}^{B} W_n(s,k)} \right\rceil, \tag{4}$$

where $D(s)$ is the number of preprocessed requests for service s, W_{B_i} is the weight of appliance i, and W_n is the aggregate of the weights of all appliances in the cluster. Note that an approximation function (in this case, a *ceiling* function) is necessary, as the CSC specifies an *integer* number of documents to be sent to the service tier. By using ceiling function the maximum per-service allowed rate might be exceeded. In order to tackle this issue, appliances enter a "lottery" in which they exchange random generated numbers amongst them, and the appliances with the lowest numbers are "penalized" by having one of their credits taken away from them, depending on the number of credits that are exceeding the per-service CSC. This exchange of random values can be done both in a centralized or distributed manner. By design choice, in this paper we opt for the distributed way. Moreover, to reduce the possibility of conflicts between appliances (i.e., two or more appliances generating the same number), random numbers should be chosen in a range much larger than the number of appliances in the cluster.

C. Simulation results

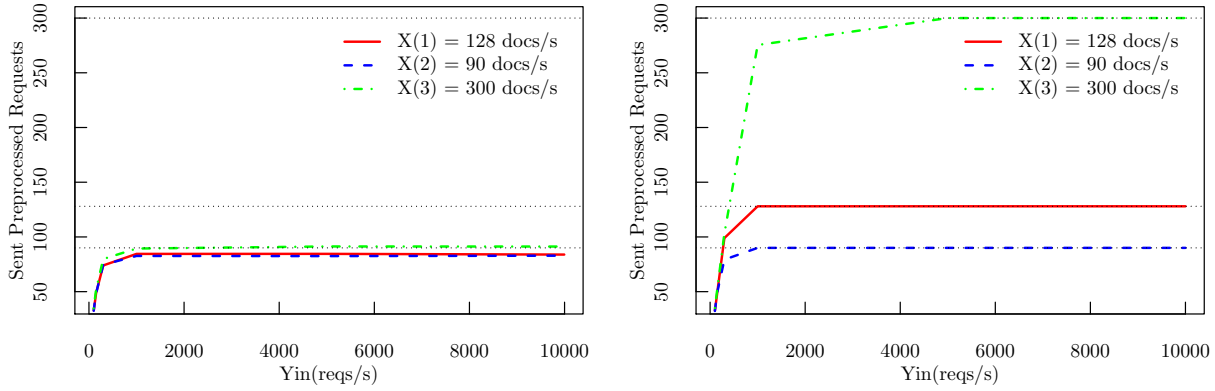
To study the performance of our multipoint-to-multipoint shaping approach, we undertook a series of simulations. To this end, we used the OMNeT++ Discrete Event Simulation System [11]. The OMNeT++ library controls the simulated time and the concurrent execution of the code running on each one of the simulated SON Appliances. All appliances run the same code. The algorithms are written in C++ and are event driven.

We modeled client service requests as Poisson processes. The *average* input rate to the system, noted as Y_{in}, is chosen as a fixed value unknown to the SON Appliances; and is varied to verify CSC compliance for all input rates. Representative results are shown in Fig. 5 and are discussed later in this section. We also simulated bursty traffic using a Poisson Pareto Burst Process (PPBP) model [12]. Burst arrivals are modeled as Poisson processes with a duration sampled from a Pareto distribution. Representative results are shown in Figs. 6 and 7. For all simulations, we assume that all SON Appliances are able to process requests for all service hosts. We also assume that the processing rate of each document at each appliance varies and depends directly on request sizes. Previous works have explored the responsiveness of credit-based algorithms [2], [3]. Results show that for $T = 1$ and $K = 40$, the algorithm achieves a reasonable responsive



(a) MSA. (b) Multipoint-to-multipoint approach.

Fig. 5. Comparison of the performance between a static allocation approach and our multipoint-to-multipoint approach for different input rates to the system.



(a) MSA. (b) Multipoint-to-multipoint approach.

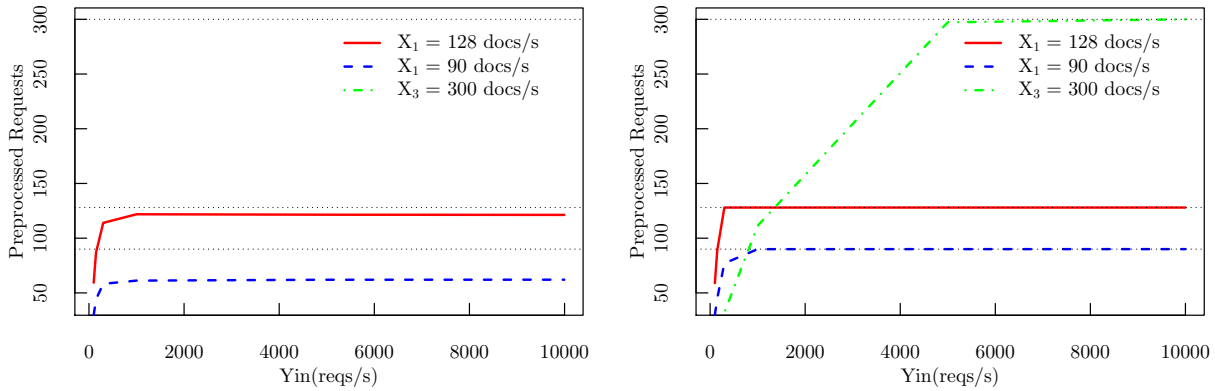
Fig. 6. Comparison of the performance between a static allocation approach and our multipoint-to-multipoint approach for different input rates to the system under bursty traffic.

behavior. Nevertheless, in a real deployment scenario, the choice of the length of an enforcement period rests at the discretion of an IT administrator. The number of subintervals should then be chosen accordingly. Therefore, for all the presented simulations, we have set $T = 1$ second, $K = 40$, $B = 6$, and $S = 3$. All data points shown on the curves represent an average over 50 runs.

Our first analysis aim verifying if our approach is able to solve the multipoint-to-multipoint traffic shaping problem. In Fig. 5, we explore the performance of both our approach and MSA during an enforcement period under poisson traffic. This figure shows the number requests sent to each service host during one enforcement period, as a function of the input rate. The horizontal dotted lines shows the values of $X(s) \times T$. For sending rates much lower than the lowest contract ($X(2) = 90$ requests/s), both schemes perform equally, as expected. However, for sending rates over $X(2)$, our algorithm outperforms MSA. Indeed, because of the use of a single FIFO output queue, as soon as the lowest CSC is fulfilled, when a request for a service which no longer has credits reaches the front of the queue, it blocks all requests behind it even if there are credits left for other services. On the other hand, our approach pushes the system to its maximum by processing and sending exactly $X(s)$ requests per observation period to each respective service host.

The next two analyses center around the adaptability of our algorithm under non-uniform traffic. In Fig. 6, we explore the performance of our algorithm as a function of the input (bursty) traffic. The figure shows the number requests sent to each service host during one enforcement period, as a function of the bursty input rate. The horizontal dotted lines show the values of $X(s) \times T$. Even with bursty traffic, our algorithm is able to comply with the per-service CSCs.

In Fig. 7 we explore the performance of both our approach and MSA during an enforcement period under bursty traffic. However, for this set of results the request traffic was unevenly distributed among the services as follows: 60% of the generated requests were bound towards Service 1, 30% towards Service 2, and 10% towards Service 3. The figure shows the number requests sent to each service host during one enforcement period, as a function of the input rate. The horizontal dotted lines shows the values of $X(s) \times T$. As expected, for sending rates much lower than the lowest contract ($X(2) = 90$ requests/s), both schemes perform



(a) MSA.

(b) Multipoint-to-multipoint approach.

Fig. 7. Comparison of the performance between a static allocation approach and our multipoint-to-multipoint approach for different input rates to the system under bursty traffic when the traffic is unevenly distributed among services: (60% S(1), 30% S(2), 10% S(3).

equally. However, for sending rates over $X(2)$, we evidence a particular behavior. For MSA, the number of sent requests towards Service 1 closely approaches its contract ($X(1) = 128$ requests/s). Nevertheless, the traffic towards Services 2 and 3 is reduces dramatically in comparison to simulations where the traffic was distributed evenly among the services. Since more requests are being sent towards Service 1, the credits for this service will be rapidly consumed. In consequence, requests going to Services 1 and 2 will be blocked at the exit queue. On the other hand, we can evidence that our approach can dynamically adapt to this kind of scenarios. As a result contracts for all three services are completely fulfilled, thus pushing the system to be used to its maximum capacity. Note that, in the case of Service 3, the contract is attained around $Y_{in} = 5,000$ requests/s. This is due to the actual number of requests being sent towards Service 3.

V. Summary and Perspectives

In two-tier EDC setups, a cluster of SON Appliances *locally* shapes the flow of client requests to enforce a *global* maximum access rate defined by a service host. In this paper, we identify the architectural limitations present in off-the-shelf appliances in order to introduce a SON Appliance architecture fit for *multi-service* traffic shaping. Subsequently, we proposed and validated via simulation an algorithm for *multipoint-to-multipoint* service traffic shaping in two-tier EDCs which solves the multipoint-to-multipoint service traffic shaping problem while pushing the system to its maximum capacity.

In this body of work, we focused on studying the performance of our approach in data centers deployed in a single site, where the network that connects the preprocessing and service tiers has very small latency. Consequently, further study is required for geographically distributed data centers, as the network may now introduce a higher latency. Moreover, the design and implementation of a practical version of the proposed approach on a real world testbed which would allow to properly measure the impact of the algorithm in an actual production environment. Finally, by design choice, our algorithm calculates appliance weights using queue sizes as its main metric. Nonetheless, in the future, it could be useful in to investigate approaches for assigning weights to appliances based on user history.

References

[1] R. D. Callaway, A. Rodriguez, M. Devetsikiotis, and G. Cuomo, "Challenges in service-oriented networking," in *IEEE Globecom*, San Francisco, CA, USA, November 2006.

[2] K. Boloor, B. Callaway, M. Dias de Amorim, A. Rodriguez, and Y. Viniotis, "Meeting Service Traffic Requirements in SOA," in *IEEE Workshop on Enabling the Future Service-Oriented Internet*, New Orleans, LA, USA, November 2008.

[3] Y. Jarma, K. Boloor, M. Dias de Amorim, Y. Viniotis, and R. Callaway, "Dynamic Service Contract Enforcement in Service-Oriented Networks," *IEEE Transactions on Services Computing*, preprint, 4 august 2011, doi:10.1109/TSC.2011.45.

[4] S. Zilora and S. Ketha, "Think inside the box! optimizing web services performance today," *IEEE Communications Magazine*, vol. 46, no. 3, pp. 112 – 117, March 2008.

[5] M. Head, M. Govindaraju, R. Engelen, and W. Zhang, "Benchmarking XML processors for applications in grid web services," in *ACM/IEEE Conference on Supercomputing*, Tampa, FL, USA, November 2006.

[6] IBM, "IBM SOA Appliances: Redefining the Boundaries of Middleware," *IBM White Papers*, 2007. [Online]. Available: http://www-01.ibm.com/software/integration/datapower/library/white_papers.html

[7] G. Cuomo, "IBM SOA "on the edge"," in *ACM Sigmod*, Baltimore, MD, USA, June 2005.

[8] A. Parekh and R. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: the single-node case," *IEEE/ACM Transactions on Networking*, vol. 1, no. 3, pp. 344 – 357, June 1993.

[9] A. Elwalid and D. Mitra, "Traffic shaping at a network node: theory, optimum design, admission control," in *IEEE Infocom*, Kobe, Japan, March 1997.

[10] B. Raghavan, K. Vishwanath, S. Ramabhadran, K. Yocum, and A. Snoeren, "Cloud Control with Distributed Rate Limiting," in *ACM Sigcomm*, Kyoto, Japan, August 2007.

[11] A. Varga, "The OMNeT++ Discrete Event Simulation System," in *European Simulation Multiconference*, Prague, Czech Republic, June 2001.

[12] M. Zukerman, T. Neame, and R. Addie, "Internet traffic modeling and future technology implications," in *IEEE Infocom*, San Francisco, CA, USA, January 2003.