Quality of Experience based provisioning for Service Providers

René Serral-Gracià, Eva Marin-Tordera, Marcelo Yannuzzi, Xavi Masip-Bruin, Sergi Sánchez
Advanced Network Architectures Lab, Technical University of Catalunya (UPC), Spain
{rserral, eva, yannuzzi, xmasip, sergio}@ac.upc.edu

Abstract—**Nowadays Network Operators (NOs) are providing service guaranties to customers by largely overprovisioning their networks, but as the Internet keeps growing, this approach is unveiling important issues in terms of cost and management of the network. To optimize the resource usage, and to reduce the existing overprovisioning, in this paper we present a Cross-layer Autonomic Network Management System that permits Service Providers to perform cost-effective network resource reservation with their NOs. In our novel approach, we use the end-user satisfaction level as the metric used to perform the resource provisioning. We show that our system is capable of achieving a high reduction in operational costs for the Service Providers, while keeping proper bounds in the end-user satisfaction for the offered services.**

I. Introduction

Over the years, Network Operators (NOs) have relied in overprovisioning in order to provide network services to their customers. Given the current growth in bandwidth demands, which are duplicated every two years, it has been argued that overprovisioning is not sustainable at this rate both in terms of operational and power consumption costs. Additionally, the apparition of Virtual Network Operators (VNOs), which do not own any network infrastructure but lease the required network resources from NOs, provide a cheaper and competitive alternative to network access. As a consequence, NOs need to deploy novel techniques to avoid being relegated to mere network carriers with few or no control over the contents traversing the network. Furthermore, VNO require more tunable traffic provisioning methods to reduce the operational costs, and to offer more competitive resource provisioning to the Service Providers (SPs).

At the same time, SPs hold large data centers and clusters to offer different set of services to final users, from on-line gaming to multimedia streaming solutions. Normally these SPs do not own any carrier infrastructure, and are adopting the overprovisioning imposed by the NOs, and by extension by VNOs, as the only option to offer some guaranties in their services. The imposition of this model caused some time ago the limited adoption of Quality of Service (QoS) as a feasible solution to provide service guaranties, and to provide an efficient method for resource management, since overprovisioning was easier to implement.

Opposed to this, the Internet is becoming an interactive and multimedia oriented infrastructure, encouraging novel quality assessment techniques to appear, further outdating QoS. In particular, Quality of Experience (QoE) is gathering the attention from the research community. Jointly with this spreading of multimedia content, plus the increase in CAPEX and energy consumption caused by network overprovisioning, it is becoming apparent that a new model for network management is required for NOs, for VNOs, and for SPs.

In this regard, current Network Management Systems (NMSs) propose very specific solutions which only consider single layer information, e.g., there is no coordination between applications and the network management. To overcome this limitation, in this paper we propose a novel Cross-layer Management solution which allows the SP to dynamically adjust the network resource reservation with the NOs (or VNOs) depending on the end-user perceived QoE of the offered service. Our approach proposes to monitor the QoE at the end-users applications, using in-band signaling with the data center to infer the network resource requirements, and thus updating the bandwidth reservation with the NO at the network layer accordingly.

Our solution permits the SP to reduce the operational costs with two clear advantages compared with other solutions present in the literature. First using QoE as an assessment method allows our system to monitor more precisely the end-user's perceived quality of the services than using QoS, and second, since the QoE is computed by the end-nodes, our infrastructure is inherently distributed, providing a very efficient mechanism to the SP when processing the end-user's data, opposed to other solutions which suffer from scalability [1], [2] and accuracy [3], [4] issues, caused normally by their centralized nature.

In the evaluations we validate our solution from two different perspectives, first we study the periods with unsatisfied end-users when using our system, compared to the case of having fixed bandwidth allocation. Second, we compare the reduction in operational costs in the SP premises also against the case of fixed bandwidth. We then conclude that even with lower operational costs, the final user satisfaction is increased.

The rest of the paper is structured as follows, in Section II we discuss the related work regarding Cross-layer Autonomic Network Management Systems (ANMS) and existing QoE assessment frameworks. We continue in Section III with the main contribution of this work, that is, the cross-layer man-

This work was partially funded by Spanish Ministry of Science and Innovation under contract TEC2009-07041 and TEC2008-02552-E, and the Catalan Government under contract 2009 SGR1508.

agement system proposed. Next we focus on the simulation driven evaluation of the proposed solution, highlighting as a use case the performance of our system in a video-streaming scenario. And finally, in Section V we conclude and outline the open lines in our research.

II. Related Work

ANMS is an emerging research topic which, still nowadays, has many open issues [5], NOs and SPs require autonomic methods to reduce the budget invested in infrastructure management and to optimize the contingency recovery times. In general, such systems are focused on well-known management issues, e.g., Resource Management [6], QoS/SLA Assessment [1], etc. In [5] Samaan et. al have a thorough review of the state of the art in terms of ANMS.

Even with the considerable efforts of the community in ANMS, most solutions are very specific [5], only considering part of the problem, without analyzing a broader perspective. Opposed to this, our approach is designed to work jointly with other ANMS solutions, and to offer an efficient solution for dynamic resource reservation. Hence, we consider cross-layer information by gathering end-user's application information, without any human intervention, about their satisfaction, and then, performing the required resource reservation and release in the NO (i.e., IP layer and below).

The above task, classically is carried through QoS mechanisms, which require very complex networking infrastructures to manage the data acquisition, and the later processing in a centralized Resource Manager, incurring in noticeable scalability issues. To overcome these limitations, our solution transports the computational burden of quality assessment to the end-users applications, who report the system status using in-band signaling.

Since we use the end-users applications as quality monitors, their location is optimal to measure their perceived QoE. QoE stands for the subjective end-user perception of a given service. This *subjectiveness* can be made objective by the use of well-known techniques such as MOS [7], combined with the E-Model [8] for voice transmission, or lately, techniques such as [9], [10] for video flows. It is not the goal of this paper to discuss how the QoE is computed, we use it as a method to receive quality feedback, and to build an efficient network resource reservation and release management system.

III. Cross-Layer Management

In this section we focus on the description of the main contribution of this paper. In particular, we detail the ANMS proposed to perform on-line resource management from the SP towards the NO (vertical signaling) in order to guarantee that the end-users are properly perceiving the offered service (horizontal signaling). As we detail in Fig. 1 the system is composed by the following building blocks:

- *Service:* besides delivering the actual service to end-users, this block is also in charge of acquiring the end-users perceived QoE (horizontal signaling).
- *Decision process:* it analyzes the perceived QoE reported by the service and decides whether to ask for more resources or not (vertical signaling), as detailed later in Alg. 1.
- *Resource Manager:* this block is in charge of the resource management in the data center itself. Since we only focus on the interaction with the NO, in our work we assume that the SP already has mechanisms to monitor and control the data center, along with mechanisms to compute the resources needed by the system to offer the service, specifically in terms of bandwidth. Within the Resource Manager we identify:
  - *Network Resource Manager (NRM):* to guarantee independence of the specific service, the Network Resource Manager will use the results obtained from the Decision process, and canonical network requirements from the resource manager itself, in order to compute the current service status and new requirements, generating specific requests that will be sent to the NO to perform the reservation or release of resources, giving autonomic capabilities to on-line reservation of the necessary network resources.
  - *Billing and accounting:* depending on the business model of the Service Provider, the final resource reservation or release can be affected.

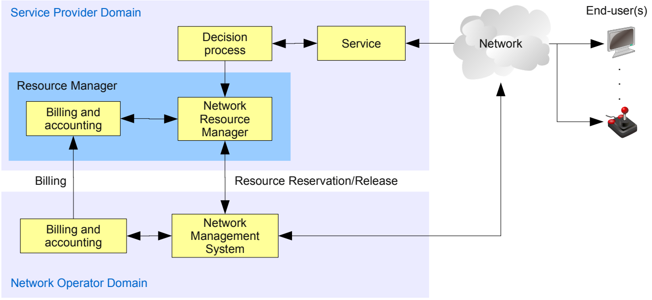

Fig. 1. Cross-layer management building blocks

Regarding the Network Operator domain the blocks interacting with the Service Provider are the Billing and accounting, which is responsible to charge the SP depending on the required resources, and the Network Management System, which enforces the on-line configuration of network devices to comply with the SP requirements.

In the following sections we detail the specific behavior of the QoE assessment within the Service block, the Decision Process, and the Network Resource Manager, excluding the rest of the blocks since they have been studied previously in the literature in work such as [2], [11], [6], [12], [13], and are out of the scope of this paper.

A. Previous considerations

Before discussing the different building blocks in the system, in this section we outline the different considerations and assumptions made in the design of our system.

In general, the possible causes of service degradation can be reduced to three different categories, *i)* insufficient resources in the SP, *ii)* insufficient contracted resources in the network, and *iii)* insufficient resources in the end-user premises. As it can be noted, as a first step we focus on the realistic case of service degradation, leaving failure scenarios as future work, since currently, the failure recovery is carried out entirely within the NOs premises.

Of these three causes, we focus on the case of insufficient resources contracted with the NO. In particular, our interest resides on the network resource management (cross-layer vertical signaling) between the SP and the NO, and the QoE assessment (horizontal signaling with the end-user application).

In order to ease the comprehension of the full design, we make the following reasonable assumptions aligned with the above causes of service degradation:

1) The SP has enough resources, and internal management facilities to cope with the load caused by all the clients, case *i* above is not met.
2) The network provider has enough resources to satisfy the load demanded by the Service Provider, i.e., the network is not the bottleneck if the resources defined by the SP were properly reserved, case *ii* above is not met.
3) The end-user has enough hardware and network resources to use the requested service, i.e., the case *iii* stated above does not apply.

B. The service QoE assessment

Nowadays data centers offer a wide variety of services to the end-users, e.g., live multimedia content, movies, E-mail, on-line games, and so on, each one having specific requirements in terms of performance.

On the one hand, services such as live multimedia streaming require a reliable network (with bounded packet losses and delivery delay). The same applies for movies or stored multimedia content broadcasting.

On the other hand there are other services, such as E-mail, on-line gambling sites or turn based games, where the constraints are more related with the end-user wait time and response delay than the actual network performance in terms of bandwidth; that is, end-users only need a fluid interaction with the system.

Finally, other kind of services, such as on-line gaming, require non negligible bandwidth but most importantly a very small delay in data delivery since its real-time nature derives in very tight network constraints, i.e., end-users require very low delay, with moderate bandwidth usage in order to beat their opponents.

Considering the current usage of the Internet, all the services have a potentially high number of end-users, which makes very hard to monitor and analyze in real-time the quality from a centralized location. Despite of this, all such services follow in one way or another the client/server paradigm, which implies that the end-user needs a specific client-side application, namely, a web browser with JavaScript support, or a native application installed to interact with the service, e.g., in the case of E-mail, the application can monitor end-user wait times since the click on an E-mail until it is available for the user; or in the case of a video streaming application, it can compute the MOS through the analysis of buffer underflows and delivery delays with the embedded timestamps in the video frames [14]. It is thus clear that, these applications offer a vantage point to analyze the delivered service, and since they are running at the end-user premises are good candidates to perform the monitoring. On top of that, using the end-nodes as measurement entities inherently distributes the load of the QoE computation.

As a consequence, our approach proposes to upgrade such client-side applications in order to perform the QoE assessment. Using QoE as a trigger provides a more accurate method for resource reservation than simply using QoS, given that QoE can measure metrics such as response time, or perceived video quality, which is not feasible with approaches such as QoS.

It is out of the scope of this paper to propose specific mechanisms to assess the QoE, but for the sake of completitude, we discuss the specific case of video streaming later in Section IV, and provide some details on how to compute the QoE in that particular environment.

For each type of service the SP defines a series of constraints, which are embedded within the client-side application, and determine how the QoE is computed. The system then defines a quality threshold $\phi \in [0,1]$ that determines the lower acceptable bound for the service quality, the higher ϕ the better is the experienced quality. This limit is used to manage the network resources in the service provider as we show in the next section. A consideration that we leave as part of our future work, is to propose a business model which regardless of ϕ, in some situations the cost of increasing the resources towards the NO is not feasible because it does not derive in increased revenue.

We are aware that classically the QoE is measured in terms of MOS, which ranges from $[1,5]$, even if in our approach we use $\phi \in [0,1]$ the latter can easily be mapped to MOS by using the R-Factor described in the E-Model in [8].

C. The decision process

As we discussed above, the decision process is one of the key parts of our infrastructure. In particular, it is in charge of deciding whether the resources used are well suited for the current load depending on the observed QoE by the end-users. In our proposal, the decision process solely uses the perceived QoE as a metric to decide whether to reserve or release resources, since at this point there is no knowledge about the current resource reservation or about bandwidth requirements. The specific resource management is relayed to the NRM, which allows a clean design with a generic NRM, while keeping a per service decision process specifically developed to assess its QoE.

In a context with thousands of concurrent users, obtaining the perceived QoE of a single end-user is of no practical interest from the SP point of view. On the contrary, having many end-users with degraded quality is relevant, as it might alter the users perception of the service, specially when the users are geographically close[1]. Moreover, given the existing policies between SPs and NOs, it might occur that the number of updates in the reservation is limited by the NO due to infrastructural or configuration restrictions. To overcome these limitations, we propose the following assessment stategy.

1) Assessment strategy:: The decision process periodically queries the Service to obtain the perceived QoE for each end-user, then the decision process keeps track of the end-users' perceived quality in two different ways, depending whether the QoE degradation is localized (i.e., the issues occur within the same regional zone, or within the same network prefix), or general (i.e., there are insufficient reserved resources, normally bandwidth, in the whole system to cope with the current end-user demands). In both cases, the SP and NO can only improve the service when end-users and the SP are tied to the same NO. To ease the discussion during the rest of the paper we assume that the localized degradation refers to network prefixes without loss of generality.

With the two critera defined above, the next step is to decide when a service is considered degraded in order to request more resources. To this end, we define two different quality thresholds, namely ρ and σ, as follows:

Definition 1: Let $\rho \in [0,1]$ be the threshold from where a service is considered generally degraded and more bandwidth resources are required for the whole SP.

More specifically, ρ is the lower permitted bound of unsatisfied end-users over the total using the service. Hence, the goal of the system is that at any instant of time the ratio of unsatisfied users υ is lower than the threshold ρ. Such υ can be computed using Eq. 1:

$$\upsilon = \frac{\sum_{i=0}^{n-1} \mathcal{U}_i}{n} \tag{1}$$

where n is the amount of end-users in the system, and $\mathcal{U}_i$ represents whether the user is satisfied, $\mathcal{U}_i$ is computed as detailed in Eq. 2:

$$\mathcal{U}_i = \begin{cases} 1 & , \, if \; q_i \leq \phi \qquad \text{(User unsatisfied)} \\ 0 & , \, otherwise \qquad \text{(User satisfied)} \end{cases} \tag{2}$$

where q_i is the quality experienced by the i^{th} end-user, and ϕ has been previously defined in Section III-B as the lower acceptable per end-user quality limit.

Definition 2: Let $\sigma \in [0,1]$ be the threshold from where a service is considered degraded within a network prefix, and more resources are required towards that specific network.

That is, the amount of unsatisfied end-users over those using the service from the same network prefix. Let's define $\mathcal{P} = \{P_1, \ldots, P_{|\mathcal{P}|}\}$ as the set of prefixes using the service, and $k \in [1, |\mathcal{P}|]$ an specific prefix. Then, the ratio of unsatisfied end-users ω_k at a given instant of time is computed analogously to υ, but only considering the end-users belonging to that prefix.

Algorithm 1 *decisionProcess*

```
Input: $s$ {$s$: End-user service status}
Output: $trigger$

$trigger \leftarrow \varnothing$
$updateStats(\mathcal{S}, s)$ {Update the system status with $s$}
if $\upsilon > \rho$ then
    $trigger \leftarrow \langle all, \upsilon \rangle$ {Set trigger for a global resource query}
else
    for all $k \in \mathcal{S}_{\mathcal{P}}$ do
        if $\omega_k > \sigma$ then
            $trigger \leftarrow trigger \cup \langle k, \omega_k \rangle$ {Set trigger for a prefix resource query}
        end if
    end for
end if
```

This information is stored in a system wide status descriptor, namely $\mathcal{S}$, which holds the overall status for the different end-users. In particular, it holds $\mathcal{S}_{\mathcal{P}} = \{\omega_1, \ldots, \omega_k\}$ for all the prefixes in the system, containing the ratio of unsatisfied end-users for each prefix.

In Alg. 1 we detail the full set of operations performed by the decision process. The algorithm first updates the overall status $\mathcal{S}$ of the data center with the new set of values received by the Service module, namely the QoE of the users. After this, the system computes the υ following Eq. 1, deciding if there is general service degradation, and triggers a general resource request. In case that the user satisfaction is below the threshold ρ, the algorithm follows with the assessment of ω_k which is computed for each prefix $k \in \mathcal{P}$, in this case the system schedules as many requests as prefixes under severe service degradation. As it can be observed, the decision process does not consider the case of resource release, which is deferred to the NRM.

It is important to notice that the final decision to request for resources will be taken by the NRM, because together with QoE it also considers factors such as the feasibility of the reservation, the resource release when the number of end-users decreases, or the cost of such changes.

In the rest of the paper we focus on the case with generic service degradation, because localized degradations are a particular case of the generic ones.

D. Network Resource Manager

The Network Resource Manager (NRM) is in charge of interpreting the values obtained by the decision process, reserve or release the required resources, and to receive feedback about the reservation status from the NO.

Initially one might think that relying in resource usage (e.g., used bandwidth) is enough to provide a proper estimate of the end-user's perception. However, the advantage of measuring QoE over bandwidth usage is that, bandwidth, does not consider events such as lost frames, or end-user wait times for the service, which are critical from the end-user point of

[1] If some users are not satisfied with a service they are bound to inform their relatives, which will affect the perceived reliability on the service, and lower the number of users of the service.

view. Additionally, having per end-user QoE information can be used to internally improve the data center behavior, e.g., increasing the priority of the processes serving the unsatisfied end-users or, as we discuss in this work, to manage the network resources.

Opposed to the decision process, the NRM considers two different aspects regarding the system quality, first the overall resource usage, and second the overall end-user satisfaction. In the case of resource usage the system proactively sends a request for more resources towards the NO before the network reaches an overload situation. To reduce the number of updates, the bandwidth will be requested in blocks of fixed size δ, which has to comply with the policies imposed by the NO.

In the second case, whenever the end-user satisfaction drops below the specified boundaries ρ or σ, NRM will ask for more resources in case the system is close to overloading, or will tighten the network constraints with the NO in terms of delay, loss, and jitter; which are the classical tunable values by the NO.

As a consequence, the NRM will merge in a single request operation the information obtained from the used network resources (by the Resource Manager) and the satisfaction ratio (from the decision process) with the following criteria:

- If the curent system load L is close to the reserved resources $\mathcal{R}$, that is, $L \geq \Delta\mathcal{R}$, where $\Delta \in [0,1]$ is the threshold to avoid network overloading.
  Then the system reserves k bandwidth blocks complying with $k\delta \geq (1+(1-\Delta))\mathcal{R}$.
- If the service requires tighter network constraints, i.e., there are still resources but the ratio of unsatisfied end-users is higher than ρ or σ, the SP must increase the priority of the flows.
  Such priority increase is usually mapped by the NO to different classes of service. Generally classified into four different categories, namely, *Gold*, *Silver*, *Bronze* and *Best effort*.

Once the requested resources are computed, the system must compute the cost of reserving these resources, and depending on the available budget (or on the reservation policies) the request can be accepted or not. Another cause for denying the resource reservation request involve the policies in the NO, for example, NOs might limit the number of updates per time period.

The next step after deciding the allocation policies is to determine a rsource management mechanism to infer the required resources with the goal of avoiding the maximum number unsatisfied end-users as possible. To this end we propose the following mechanism:

1) Resource allocation/release policy:: Since the required amount of resources in most services are determined by the number of end-users, we use this information to determine the future requirements in terms of resources. The goal of the allocation policy is to minimize the ratio of unsatisfied end-users. As a consequence, the algorithm must quickly react to any potential overloading of the resources, while it can have a looser behavior for the resource releasing. To this end, we propose two different mechanisms, one suited for increasing the resource reservation, and the other to cope with the release of the reserved resources. This behavior is modeled in our system with the expression presented in Eq. 3.

$$\begin{aligned} \hat{L}_{t+1}^{I} &= \Theta_1 s_t + \Theta_2 s_{t-1} \\ \hat{L}_{t+1}^{D} &= \Phi_1 s_t + \cdots + \Phi_4 s_{t-3} \end{aligned} \tag{3}$$

where s is the amount of required resources at a given instant. In our case it refers to the last two and four samples respectively, and $\sum_{i=1}^{2} \Theta_i = 1$ and $\sum_{j=1}^{4} \Phi_j = 1$ the weights of each sample. Then $\mathcal{R} = (1+(1-\Delta))\hat{L}_{t+1}^{I}$ if $\hat{L}_{t+1}^{I} \geq \Delta\mathcal{R}$, hence the system asks for more resources as discussed previously. Otherwise, in the case that $\hat{L}_{t+1}^{D} \leq \Delta\mathcal{R}$ then $\mathcal{R} = \mathcal{R} - \frac{\hat{L}_{t+1}^{D}\mathcal{R}}{2}$ if $\frac{\hat{L}_{t+1}^{D}\mathcal{R}}{2} > \delta$, which produces a stepwise smooth decrease of the allocated resources.

IV. System Validation

In this section we validate our system from two different perspectives. First, we detect the periods with end-user unsatisfaction when using our system, and second, we analyze the reduction in the operational costs observed by the SP, in both cases we compare our solution against the fixed bandwidth allocation currently offered by NO.

A. Simulations

As a proof of concept, we validate our system by simulating a video streaming service provider with clients requesting a video in real-time, specifically the videos have a Constant Bit Rate requirement of 900Kbps[2], the users enter and leave the system randomly watching videos with a duration from one minute to one hour. The client applications compute the video quality by using the technique developed in [14], i.e., computing the delivery delays and the packet losses directly from the video frames and thus assessing the perceived video quality from a single measurement point. This information is sent to the SP, which computes the ratio of satisfaction ϕ and ω_k and decides whether to change the resource reservation. In order to simplify the exposition we only detail in the results global quality failures (ϕ). However, extending the simulations to per prefix failures leads to similar results.

The simulations compare our approach with the fixed bandwidth allocation present today in SPs. In the set up environment we simulated various flash-crowd events with the following criteria: each simulation lasts around 45 minutes, each flash-crowd event has a duration range spanning from 5 to 10 minutes, e.g., when a specially appealing video is submitted, and there are 5 flash-crowd events per simulation, the rest of the time the average number of users is around one order of magnitude smaller than during the flash-crowd (i.e., the number of users in the system ranges from 1000 to 16000), each calm period has a duration of approximately 90 seconds. With these conditions we run five different simulations changing the amount of flash-crowd events that require more

[2] The classical High Quality bit rate value found in sites such as Youtube

resources than the allocated by the system in the case of fixed bandwidth allocation, more specifically the amount of flash-crowds requiring more resources than the initially allocated ranges from 1 to all 5. To gather the results we monitor the allocated bandwidth through reservation requests, the monthly price of the provisioning as detailed in [12] and the periods with service degradation. In particular, we monitor the ratio of service degradation in the system. To have more realistic values we assume that each request towards the NO has a set up time of 5 seconds as discussed in [15], that is, the time between a request is issued and its set up in the NO finishes.

B. Results

We performed the simulations in order to compare the performance of our system with the use of the classical fixed bandwidth reservation offered by the NOs. Our findings are summarized in Table I, where we show both the case of using our Autonomic Network Management System (ANMS) or using fixed bandwidth (FB). As it can be observed, our system outperforms the fixed bandwidth allocation in all the tests for the different amount and intensity of flash-crowds. In the table the first column labels the flash-crowds from the trivial case of not having flash-crowds, to the case of having a highly congested site with all 5 flash-crowds overloading the system during long time periods.

The results show that the amount of time the service is degraded for all the users is independent of the intensity of the flash-crowds in our system, while it is strongly related, as expected, with the congestion level in the case of having fixed bandwidth, as we show in the column labeled as $p(\upsilon > \rho)$ in the table, where it can be noted that the periods with service degradation are always below 10% for ANMS, raising up to $\sim$ 50% for the fixed bandwidth. The only case where both results are comparable is when having flash-crowds within the service limits, where our solution performs slightly worse because it tries to adapt to the current load, and causing brief periods of service degradation on the beginning of the flash-crowds due to the 5 second lag caused by the NO before committing the new resource reservation. Nevertheless, with this adaptability our proposal achieves a great reduction both in average used bandwidth and operational cost (see columns labeled as Average Bandwidth and Monthly Cost in the table). An aspect worth noticing is that even with flash-crowds within limits, that is, flash-crowds not reaching the reserved fixed bandwidth, there are cases that cause service degradation, the reason of this behavior is that the resource requirements are close to the reserved (e.g., link load higher than 80%) the network buffers and the jitter of the video flows derive in mild service degradation, which in some cases cause the υ to raise above the threshold ρ.

Aligned with the above discussion, it can be noted that the maximum interval with service degradation in the case of having fixed bandwidth is generally as long as a whole flash-crowd period (around 300 seconds), and it is always longer than the one caused by our system, which is always bounded by the adaptation to the load demand derived from Eq. 3. Another aspect to consider is the amount of time the system is underprovisioned, i.e., periods with severe service degradation, which is in all the cases below 3% for our solution, while it raises up to $\sim$47% when using fixed bandwidth.

Regarding the used bandwidth and the operational cost, the last two pair of columns show that our solution, even with constant flash-crowd events, is able to outperform the fixed bandwidth while still reducing the service degradation periods.

The last analysis we perform in this work refers to the demanded bandwidth versus the offered one. In Fig. 2 we detail the ratio of demanded versus offered bandwidth. In the figure, the value of 1 represents exact balance between offer and demand, values below one imply overprovisioning, while values above one refer to underprovisioned network. As it can be noted, our solution is always below 1, which is that the resources are overprovisioned by δ as discussed in the previous section. However, at the edges of the flash-crowd, there is severe underprovisioning caused by the adaptive algorithm as we pointed out previously.

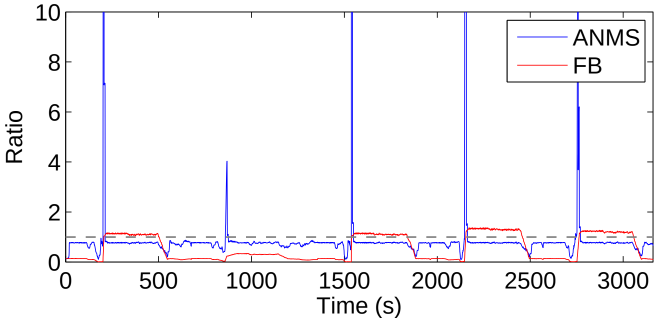

Fig. 2. Demanded versus reserved bandwidth for FC. 4 out-limit

V. Conclusions and Future Work

In this paper we presented a reactive approach for Service Providers to efficiently manage traffic provisioning with the Network Operator. The core of the proposed solution uses the end-user perceived QoE of the offered service in order to trigger the resource reservation with the final goal of minimizing the service degradation periods. The motivation behind the proposal is the increased complexity in terms of cost for Network Operators to provide overprovisioned networks, and the need of Service Providers of more efficient and accurate means to provision the required network services and, while delivering a reliable service, reduce the operational costs of the network.

To validate the solution we performed a series of simulations with a different set of flash-crowd and high load scenarios that demonstrate that our Cross-Layer Autonomic Network Management System can, at the same time, reduce the required network resources and its operational costs, while offering a better service than the classical fixed bandwidth reservation.

As lines open for future work, as we already mentioned, having a clear business model can raise the interest of Service Providers and Network Operators to implement the system

	$p(\upsilon > \rho)$		Max. Duration (s)		Below BW.		Avg BW.		Monthly Cost	
	ANMS	FB	ANMS	FB	ANMS	FB	ANMS	FB	ANMS	FB
No FC.	0	0	0	0	0	0	1425	9000	11996	30000
FC. within limits	0.099	0.043	35	95	0.023	0	3953	9000	16674	30000
FC. 1 out-limit	0.085	0.136	29	303	0.023	0.088	4584	9000	19159	30000
FC. 2 out-limit	0.087	0.238	35	308	0.020	0.181	5536	9000	22116	30000
FC. 3 out-limit	0.070	0.367	42	318	0.025	0.301	6791	9000	25302	30000
FC. 4 out-limit	0.077	0.391	35	314	0.024	0.378	6727	9000	26131	30000
FC. 5 out-limit	0.057	0.491	24	321	0.019	0.471	7918	9000	29436	30000

TABLE I

Comparison in system performance with the presence of Autonomic Network Management System (NMS) and Fixed Bandwidth reservation (FB) for the different out-limit flash crowds (FC).

while maintaining a feasible economic model for both parties. More in the technical side of the work, designing a more adaptable resource provisioning and release function can improve the end-user perceived quality, and further reduce the used bandwidth.

References

[1] René Serral-Gracià, and et. al. “Coping with Distributed Monitoring of QoS-enabled Heterogeneous Networks”. *4th International Telecommunication Networking Workshop on QoS in Multiservice IP Networks*, pages 142–147, Venice, Italy, February 2008.

[2] Xavier Masip-Bruin, and et. al. ”The EuQoS System: A Solution for QoS Routing in Heterogeneous Networks”. *IEEE Commun. Mag*, 45(2):96–103, 2007.

[3] René Serral-Gracià, and et. al. ”Network performance assessment using adaptive traffic sampling”. *IFIP Networking*, LNCS 4982:252–263, Singapore, May 2008.

[4] Joel Sommers, and et. al. ”Accurate and Efficient SLA Compliance Monitoring”. In *Proceedings of ACM SIGCOMM*, pages 109–120, Kyoto, Japan, August 2007.

[5] Nancy Samaan and Ahmed Karmouch. Towards autonomic network management: an analysis of current and future research directions. *Communications Surveys & Tutorials, IEEE*, 11(3):22–36, Quarter 2009.

[6] Vlad Nae, and et. al. Efficient management of data center resources for massively multiplayer online games. *SC ’08: Proceedings of the 2008 ACM/IEEE conference on Supercomputing*, pages 1–12, 2008.

[7] ITU-T Recommendation G.113. Transmission impairments due to speech processing, 02/2001.

[8] ITU-T Recommendation G.107. The E-model, a computational model for use in transmission planning, 03/2005.

[9] Jirka Klaue, and et. al. EvalVid - A Framework for Video Transmission and Quality Evaluation. In *In Proceedings of the 13th International Conference on Modelling Techniques and Tools for Computer Performance Evaluation*, pages 255–272, 2003.

[10] Kazuhisa Yamagishi and Takanori Hayashi. Parametric packet-layer model for monitoring video quality of iptv services. In *IEEE International Conference on Communications. ICC ’08.*, pages 110–114, May 2008.

[11] Mudhakar Srivatsa, and et. al. A Policy Evaluation Tool for Multisite Resource Management. *Parallel and Distributed Systems, IEEE Transactions on*, 19(10):1352–1366, Oct. 2008.

[12] Hao Wang, and et. al. Optimal ISP subscription for Internet multihoming: algorithm design and implication analysis. 4:2360–2371 vol. 4, March 2005.

[13] Partha Sarathi Chakraborty and et. al. A platform for charging, accounting and billing in telecommunication networks and the internet. pages 633–638, 2009.

[14] René Serral-Gracià and et. al. Packet Loss based Quality of Experience of multimedia video flows. 2009.

[15] Edward Wustenhoff. Service Level Agreement in the Data Center. *Sun Professional Services - Sun BluePrints OnLine*, apr 2002.