

A Dimensioning and Deployment Tool for on Demand Policy-based Resource Management System

Kamel Haddadou¹, Samir Ghamri-Doudane¹, Yacine Ghamri-Doudane², and Nazim Agoulmine²

¹ LIP6, Pierre & Marie Curie University, 104 avenue du Président Kennedy 75015 Paris – France

{Kamel.Haddadou,Samir.Ghamri-Doudane}@lip6.fr

² LRSM , ENSIIE & University of Evry joint research group, 18 allée Jean Rostand, 91025 Evry Cedex – France

ghamri@ensiie.fr, Nazim.Agoulmine@iup.univ-evry.fr

Abstract. Today the Policy-Based Management (PBM) approach is recognized as an efficient solution to simplify the complex task of managing and controlling networks. To this end, the IETF has introduced a reference framework to build PBM systems. However, this framework only addresses the provisioning of relatively long validity period services based on pre-defined SLAs (Service Level Agreements). Furthermore, very little work addresses the scalability properties of the instantiation of this framework in a real network. In our previous work, we presented an extension of the IETF PBM framework in order to support dynamic provisioning of short term services (end-system signaling) as well as an instantiation scheme that is scalable (distributed provisioning of edge routers). This instantiation scheme is based on the distribution of the provisioning process while keeping centralized only the parts that involve critical resources, namely the Bandwidth Brokerage. In this paper, we propose an extensive analytical study of this extended architecture. The results of this work are intended to be used as a guideline to help network operators to design a scalable PBM system in order to offer to their customers' services with QoS assurance in an on-demand basis.

1 Introduction

A major challenge in emerging multi-service, QoS-capable telecommunication networks is the deployment of high-quality multimedia applications. Both of network operators and end users are willing to offer and use multimedia communications with a large range of QoS-guarantees. To achieve this aim, an efficient control and management of network resources are submitted to be the key issues in the telecommunications world. We argue that a combination of QoS signaling and Policy-Based Management (PBM) [1] is required to enable proper multimedia sessions.

PBM aims to facilitate the management activity as it allows network administrators to define high-level objectives of network management schemes based on a set of policies. The latter is a set of pre-defined rules controlling network resources. Rules established by the network administrator, include actions to be triggered when a set of conditions is fulfilled. PBM approach allows in its turn the translation of these high-level rules to a set of low-level device-compliant configuration commands [2].

In a previous work [3], we proposed a novel architecture that integrates QoS signaling with dynamic per-session QoS provisioning. Thus, our proposed architecture was twofold. First, we proposed to transfer parts of the network management and control mechanisms to the user's terminal. This latter is rendered responsible, for each of its sessions, to generate QoS requests towards the PBM system and to wait upon the reception of request acceptance by this one. Then we proposed to distribute the resource provisioning process while keeping centralized only the parts that involve critical operations, namely the Bandwidth Brokerage. The performance improvement in terms of scalability of such architecture was demonstrated through practical experiments. In this paper, we propose an extensive analytical study of this extended architecture. The results of our current work are intended to be used as a guideline to help network operators to design and dimension their scalable on-demand policy-based resource allocation system. So, this later will allow them to offer to their customers services with QoS assurance in an on-demand basis.

As far as we know, the only contribution to the specification of an analytical model for PBM systems is the one proposed in [4]. This analytical study aimed at demonstrating that a dynamic PBM architecture could be scalable according to the size of the administrative domain. This work is without doubt very interesting however the hypotheses of the authors are not realistic. Indeed, their analysis did not take into account numerous practical constraints related to the implementation of the PBM architecture, and it also ignores the fact that the critical resources cannot be distributed. This means that when wishing to dimension a PBM system for real networks, the analytical model proposed in [4] cannot be used. Conversely, we design in this paper an analytical model that depends closely on functional constraints and we compare it to real experiments showing its accuracy to model a real system.

The rest of this paper is organized as follows: Section 2 introduces our previously proposed architecture. Section 3 develops the analytical model and analysis of our architecture. The dimensioning formulas for on-demand policy-based resource allocation systems are presented in Section 4. This is followed by the description of the deployment strategy to be used by network operators. Finally, Section 6 concludes the paper and presents some future works.

2 Scalable on-demand Policy-based Resource Allocation

It appears nowadays that management systems following the PBM architecture are neither responding to operators' scalability issues nor to customers needs. In fact, customers are willing to dynamically request network resources depending on their instantaneous needs and without having to contract a SLA for a long period of time. This tendency is confirmed by the forthcoming 3G IP Multimedia Subsystem (IMS)/Multimedia Domain (MMD) [5] network architecture. Indeed, this architecture is evolving toward including per-session resource management and control using policies. However, from the operator perspective, the integration of dynamic resource allocation to the existing IETF's PBM architecture is not feasible at a large scale.

In order to overcome these limitations, we proposed in a previous work [3] a novel solution for on-demand policy-based resource allocation in IP networks. This solution aims to distribute the decision making operations among several distributed PDPs. Therefore, the PBM architecture has been decomposed into a set of functional com-

ponents. The idea of this decomposition is to identify which components represent critical sections in the decision-making process. Once this phase achieved, the solution consists on proposing a new instantiation model where non-critical components are distributed according to none functional requirements (such as performance objectives, network size, etc.). Hence, the impact of critical operations on the overall management system performances is minimized. To maintain the consistency of the decision-making process, critical operations are kept centralized. These operations are identified as those operations that need to access to critical resources (shared information, common databases, etc.) in the system.

The critical operations in our framework were identified as those related to the bandwidth brokerage. All other operations related to decision making appeared as replicable. Based on these statements verified in our previous work [3], we propose to keep centralized the Bandwidth Brokerage while distributing all other functional components. Fig. 1 presents in details our framework (Fig. 1(b)) and its instantiation (Fig. 1(c)), and shows its differences with the IETF's PBM framework (Fig. 1(a)).

As our objective was to demonstrate its scalability features, we both realized a complete implementation of our proposed framework. The practical experiments highlighted the scalability property of our approach. These experiments also permitted to identify the effect of each component of the framework on the overall performance of the management system. A detailed description of our proposed architecture and its practical performance evaluation can be found in [3].

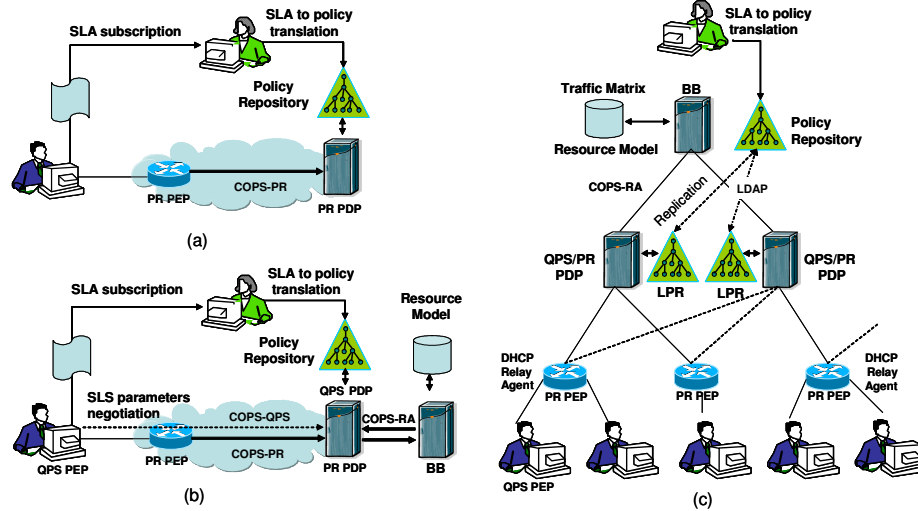


Fig. 1. (a) IETF PBM architecture, (b) Proposed on-demand policy-based resource allocation architecture, (c) Scalable instantiation of the proposed architecture.

3 Analytical Analysis.

In order to understand the behavior of our architecture, we have developed an analytical model. Our main goal in designing this model is to understand the problem analytically and to try to provide network operators with a dimensioning tool to help them to design their system efficiently. First, we will describe the components and characteristics of the analytical model, then we will apply this model to practical experiment results and compare them, in order to validate our model.

3.1 Analytical Model

Based on the instantiation schema presented in Figure 1(c), we deduce that all the customer requests follow the same path: (1) PDP processing, (2) Local Policy Repository (LPR) access (LDAP), (3) bandwidth brokerage, and if there is enough resources (positive response from the Bandwidth Broker): (4) configuration of the appropriate edge router, (5) service completion, and (6) resource release (access to the BB and to the adequate edge router). According to this we specify the queue network model shown in Figure 2.

The first queue (P), which represents local processing at a PDP, is composed of C parallel independent servers, each of which corresponds to one connection controlled by a PDP. The aim is to study the scalability of the framework, i.e. to compute the maximum value of C before reaching system instability (i.e. here we are interested in the limits of the system). Each server serves the requests at a certain rate. We have chosen an exponential¹ distribution which is fairly representative of the real world. As we are interested in the number of active connections in the system, the number of PDPs has no effect on the model design. Each PDP handles a part of these connections and C represents the total value.

The second queue (L) characterizes the LPR access. This queue is also composed of C parallel independent servers, since an LDAP server is able to handle the requests in parallel. For the same reasons as for the PDP service time modeling, the service time of the LPR is modeled as an exponential distribution. The number of LPRs has no effect on the model design, either.

The third part of the model (B) represents the bandwidth broker. It is a single server queue, as the BB processes the requests sequentially, and it should be noted that there is only one BB in the system. Its service law is approximated by an exponential distribution.

The fourth part (the N parallel queues) symbolizes the set of configurable edge routers in the operator network. We assume that these routers are targeted equitably by customer requests (probability set to $1/N$). Each edge router is modeled as a single server queue and its service time is approximated by an exponential distribution. For high values, the edge router number (N) has no significant impact on the performance of our system (this assertion will be argued later).

The next queue (S) characterizes the effective service time, i.e. the time necessary for a user to complete the desired service. Finally, the last queues (W and F) represent the waiting time between two consecutive requests from the same customer (initiated using the same QPS connection). The Queue F is visited in the case of a blocked call (Failure: insufficient resources to satisfy the client request), while the queue W is visited after a successful call.

After service completion, the system should release the reserved resources, which represents another access to the BB and the corresponding edge router. Consequently, the outflow of the edge router queues is connected to both the service and waiting queues, with the same probability: $1/2$. Also, depending on the available resources, the Bandwidth broker may accept or reject the client request. Thus, the outflow of the

¹ The reason for choosing the exponential distribution as the service time is that all our experiments have shown that the mean and the standard deviation values are close enough for both PDP and LPR. Indeed, the characteristic of this distribution is that its standard deviation is equal to its mean.

BB queue is divided into two different paths with a blocking proportion: ' α '. This latter depends on the call blocking probability (p), and is calculated as follows:

First, let us redefine the relevant parameters:

p : The call blocking probability.

α : The blocking proportion at the outflow of the Bandwidth Broker queue in the proposed model.

$T_{\langle I \rangle}$: The throughput of the queue $\langle I \rangle$, with $I \in \{P, L, B, R, S, W, F\}$. Note that, in the steady state and for each queue in the network, the input and output throughputs are the same. T_R corresponds to the aggregated throughput of the N Parallel queues (R).

Then, based on the queue network flows in the steady state, we extract the following equations:

$$\begin{aligned} T_B &= T_L + T_S \\ T_R &= (1 - \alpha) \times T_B \\ T_S &= \frac{T_R}{2} \end{aligned}$$

Also, based on the definition of the call blocking probability, it is obvious that:

$$T_S = (1 - p) \times T_P \quad \text{with: } T_P = T_L$$

Finally, from the previous equations, the following formula is extracted:

$$\alpha = \frac{p}{2 - p} \quad (1)$$

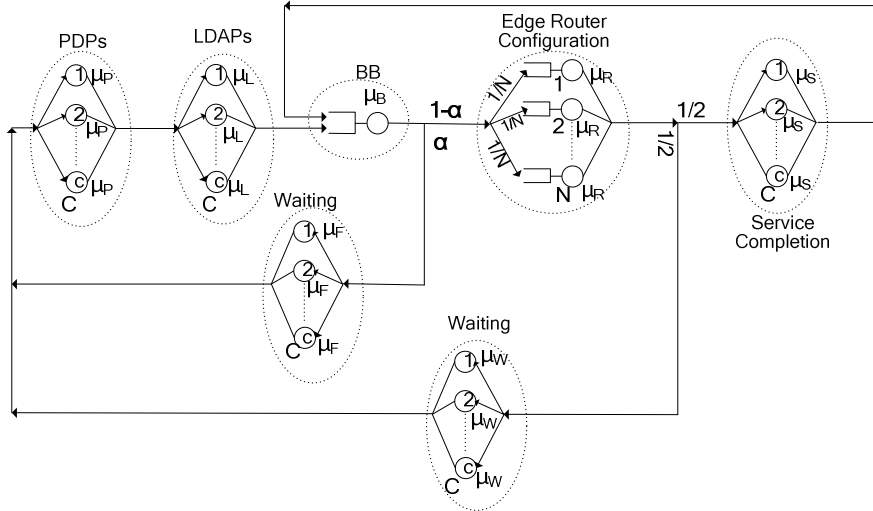


Fig. 2. Analytical model of the PBM system.

This proposed model is a BCMP model and can be solved as a product form solution [8]. As we are interested in the scalability performance of such a system, we have studied the system in the worst case, by assuming that each customer disconnecting from the system is immediately replaced by a customer entering the system. Thus, the resulting model is a closed queue network, where inputs and outputs are removed.

3.2 Test-bed results

In order to validate our analytical model, we compare its results to those obtained using practical experiments. To that end, an integrated test-bed containing all our architecture entities has been implemented. The BB, the PEP, the PDP, and the signaling protocol have been implemented using Java 1.5. The policy repository and its replicas (LPRs) are instances of the LDAP repository, in which the management information is modeled using CIM. The management information replication and updates are handled automatically by LDAP [9]. In addition to PBM components, specific Linux-based conditioning software has been used to enforce the QoS decisions (traffic classification and packet marking) in the Linux-based edge routers.

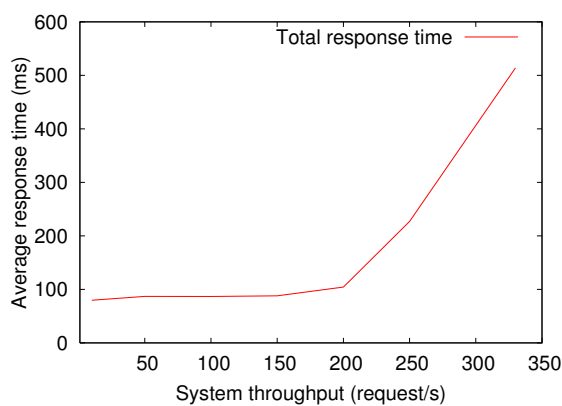


Fig. 3. Average response time as a function of system throughput.

As shown in Figure 3, the obtained practical results demonstrated that the overall-system delay is always below the ITU-T recommended signaling-delay limit [6] and that the system throughput is higher than the 200 req/s as recommended by the ITU-T [7]. For a detailed performance evaluation of our architecture the reader can refer to [3].

From the practical experiments using our test-bed, we have determined the median value of the PDP's mean service time, which is set to 7ms. Similarly, the LPR mean service time is set to 40ms and the edge router configuration mean time is fixed to 35ms. For the BB, the mean time necessary to process a request is equal to 2,25ms (μ_B). Finally, the obtained value for the call blocking probability (p) is equal to 4.6%. This value is extracted from the practical evaluation during the stable state.

3.3 Instantiation and Validation of the Model

In addition to the values obtained from experimental results and depicted above, we chosen the following values to validate our model: the number (N) of edge routers is set to 58 (as assumed in the test-bed). For the BB, the mean time necessary to process a request is equal to 2,25ms (μ_B). The service execution time and the waiting times are assumed to follow an exponential distribution with a mean equal to 2 minutes (μ_S), 5 minutes (μ_F) and 15 minutes (μ_B) respectively. The ' α ' parameter is computed using the call blocking probability value (p) and is equal to: 2.35%.

Based on the analytical model, which is a BCMP model, it is possible to compute the different scalability and performance parameters of the proposed framework. The

parameters which are essential for our study are (1) the average response time, which is the delay time to be served by the PDP, the LPR, the bandwidth broker and, if the call is accepted, the appropriate PR PEP queues respectively, and (2) the effective system throughput calculated as the number of requests correctly served per second (T_P). These parameters are calculated for different values of C , the number of customers (connections) in the system. Therefore, this value is increased until the system reaches an unstable state.

The analytical results are represented by the curves in Figure 4(a-b) and we see that the instability point corresponds to a value of $C = 230,000$ concurrent connections. The average response time remains low enough (less than 150 ms) when the connection pool is under 220,000. Beyond this number, the average time increases very quickly and reaches values of several seconds. At the same time, effective throughput increases linearly until it reaches its maximum value which coincides with the instability point, and then it remains steady. From this, we can conclude that the maximum throughput of our system is about 227 effective requests per second and the value obtained is in concordance with the result obtained in the test-bed.

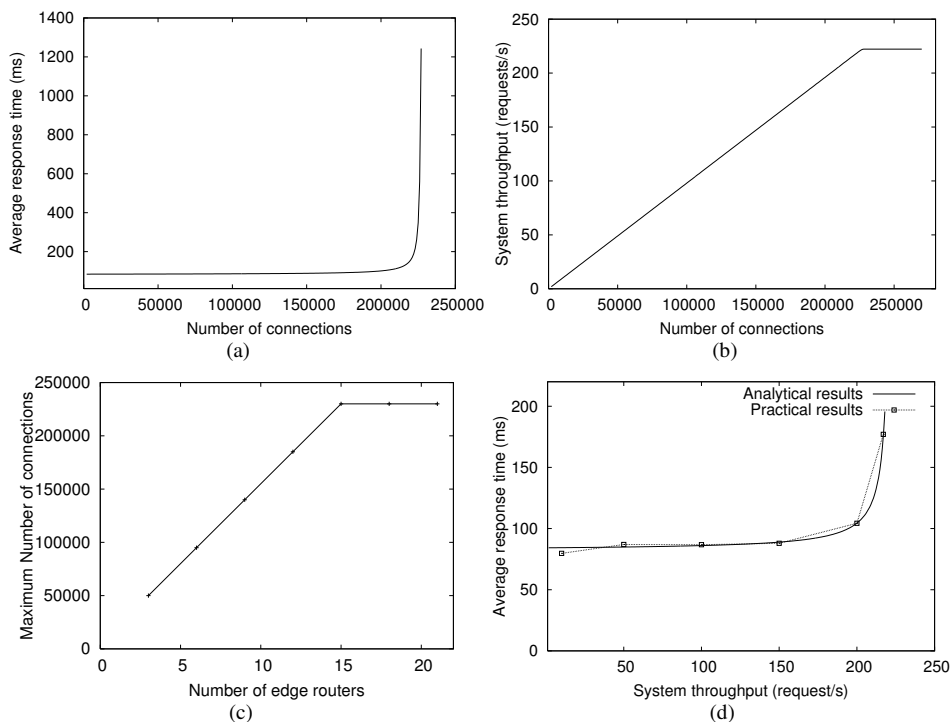


Fig. 4. Analytical evaluation:

(a) Average response time to resource allocation requests, and (b) system throughput as a function of the number of connections.

(c) Maximum number of connections shown as a function of the number of edge routers.

(d) Average response time to resource allocation requests as a function of the system throughput: practical vs. analytical results.

In the last experiments, we fixed the number N of edge routers, PR PEPs, in the system to 58. In some cases, this value may have a perceptible influence on the scalability performances of the architecture. So, to deal with this assertion, we vary the N

value in the range [3 ... 21] and compute the maximum connection number (stability threshold) for each of them. The obtained results are shown in Figure 4(c). When the N value is less than 16, it has a direct impact on the scalability of the system. However, beyond this point, the two parameters become completely independent. This can easily be explained by the fact that the provisioning operations of each edge router are in fact critical sections. Hence, for a low number of edge routers, the time between the arrivals of two consecutive provisioning decisions within the same edge router is very low. For a high number of edge routers, which is the case for most of the 1st and 2nd tier ISPs, the bottleneck of the system is clearly the BB. All the scalability and performance parameters are strongly dependant on this latter.

Furthermore, in order to validate the proposed analytical model and all the corresponding results, we have compared these results with the test-bed results presented in section 3.2. We can see that the results are very similar (Figure 4(d)). This Figure presents the average response time to requests plotted as a function of the effective system throughput (the number of accepted requests per second). In the case of the analytical results, the curve is simply inferred by combining curves (a) and (b) of Figure 4, in which we only consider the stable state of the system (i.e. throughput values below 227 req/s). We see that the performance values obtained in each case are very close. Therefore, we can confirm that the proposed analytical model is representative of the performance of the entire system and can therefore be used as a dimensioning tool to calculate the capacity of the system to handle allocation requests while ensuring the ITU-T maximum response time boundary of 6s and the stability of the system.

4 Analytical Results and Dimensioning Formulas

Based on the analytical model presented above, we target here to extract relevant dimensioning formulas, which can act as a simple guideline for the instantiation of the PBM system.. Indeed, the main objective of our study is to provide these relevant dimensioning tools, since we have already identified the behavior of each component as well as the bottleneck of the proposed architecture. More precisely, in the subsequent study, we target the following scalability parameters: (1) T^{Max} : the maximum effective throughput of the system and (2) C^{Max} : the maximum capacity of the system (i.e. maximum number of active customers before reaching instability). Indeed, these two parameters will be the basis of our system design and dimensioning tool.

First, let us define and organize all the flow equations of our model (the queue network flows):

$$\left\{ \begin{array}{l} T_L = T_P \\ T_B = T_L + T_S \\ T_R = (1 - \alpha) \times T_B \\ T_S = \frac{T_R}{2} \\ T_W = \frac{T_R}{2} \\ T_F = \alpha \times T_B \\ T_P = T_W + T_F \end{array} \right. \Rightarrow \left\{ \begin{array}{l} T_L = T_P = \frac{1 + \alpha}{1 - \alpha} \times T_W \\ T_B = \frac{2}{1 - \alpha} \times T_W \\ T_R = 2 \times T_W \\ T_S = T_W \\ T_F = \frac{2\alpha}{1 - \alpha} \times T_W \end{array} \right. \quad (2)$$

Previously, we have shown that the bottleneck of system is the bandwidth broker or the PR PEPs, depending on the number N of edge routers. Consequently, the maximum throughput of the system, which is designated by: ($T^{Max} = T_P^{Max}$), is dependant on these components. Since the BB and the edge routers are modeled as single server queues, we have:

$$\text{If the BB is the bottleneck: } T_B^{Max} = \frac{1}{\mu_B}$$

$$\text{If not: } T_R^{Max} = \frac{N}{\mu_R}$$

$$\text{Also: } T_R = (1-\alpha) \times T_B$$

Thus, based on these definitions, we can extract the following condition on our system:

$$\text{If } \left(N > (1-\alpha) \times \frac{\mu_R}{\mu_B} \right) \text{ the BB is the bottleneck of the system. } \quad (3)$$

Furthermore, based on the flow equations in 2, the maximum throughput of the system is defined by:

$$T^{Max} = T_P^{Max} = \begin{cases} \frac{1+\alpha}{2} \times T_B^{Max} & \text{if } N > (1-\alpha) \times \frac{\mu_R}{\mu_B} \\ \frac{1+\alpha}{2 \times (1-\alpha)} \times T_R^{Max} & \text{if not.} \end{cases}$$

$$\Rightarrow T^{Max} = \begin{cases} \frac{1}{(2-p) \times \mu_B} & \text{if } N > \left(\frac{2 \times (1-p)}{2-p} \right) \times \frac{\mu_R}{\mu_B} \\ \frac{N}{2 \times (1-p) \times \mu_R} & \text{if not.} \end{cases} \quad (4)$$

$$\Rightarrow T^{Max} = \text{Min} \left(\frac{1}{(2-p) \times \mu_B}, \frac{N}{2 \times (1-p) \times \mu_R} \right) \quad (5)$$

When applied to our test-bed, these formulas (equations 3 to 5) show the following result: If ($N > 15$), the bottleneck of the system is the BB and, in this case, the maximum effective throughput is $T^{Max} = 227$ req/s.

Also, the number of active connections (customers) in the system can be defined as:

$$C = \sum \overline{N_{\langle I \rangle}} \quad I \in \{P, L, B, R, S, W, F\}$$

With: $\overline{N_{\langle I \rangle}}$: The mean number of customers in the queue $\langle I \rangle$.

However, if we assume that most of the customers are in the service and waiting queues during the steady state, the number of customers can be approximated by:

$$C \approx \overline{N_S} + \overline{N_W} + \overline{N_F}$$

$$\text{With: } \begin{cases} \overline{N_S} = T_S \times \mu_S \\ \overline{N_W} = T_W \times \mu_W \\ \overline{N_F} = T_F \times \mu_F \end{cases} \Rightarrow \begin{cases} \overline{N_S} = \mu_S \times T_W \\ \overline{N_W} = \mu_W \times T_W \\ \overline{N_F} = \frac{2\alpha}{1-\alpha} \times \mu_F \times T_W \end{cases}$$

$$\text{Then: } C \approx \left(\mu_S + \mu_W + \frac{2\alpha}{1-\alpha} \times \mu_F \right) \times T_W$$

Consequently, the maximum capacity of the system is defined as follows:

$$\begin{aligned}
T_W^{Max} &= \begin{cases} \frac{1-\alpha}{2 \times \mu_B} & \text{if } N > (1-\alpha) \times \frac{\mu_R}{\mu_B} \\ \frac{N}{2 \times \mu_R} & \text{if not.} \end{cases} \\
\Rightarrow C^{Max} &\approx \begin{cases} \frac{(1-\alpha) \times (\mu_S + \mu_W) + 2\alpha \times \mu_F}{2 \times \mu_B} & \text{if } N > (1-\alpha) \times \frac{\mu_R}{\mu_B} \\ \frac{((1-\alpha) \times (\mu_S + \mu_W) + 2\alpha \times \mu_F) \times N}{2 \times (1-\alpha) \times \mu_R} & \text{if not.} \end{cases} \\
\Rightarrow C^{Max} &\approx \begin{cases} \frac{(1-p) \times (\mu_S + \mu_W) + p \times \mu_F}{(2-p) \times \mu_B} & \text{if } N > \left(\frac{2 \times (1-p)}{2-p} \right) \times \frac{\mu_R}{\mu_B} \\ \frac{((1-p) \times (\mu_S + \mu_W) + p \times \mu_F) \times N}{2 \times (1-p) \times \mu_R} & \text{if not.} \end{cases} \quad (6)
\end{aligned}$$

In order to validate this approximation, we initiated a comparison with the results obtained using the analytical model previously presented. The curve (a), in Figure 5, shows the impact of the mean waiting time on the maximum system capacity, while the curve (b) focuses on the impact of the mean service time of the BB. In all cases, the results, obtained using the approximation formula (equation 6), are very close to the exact values (the analytical model values).

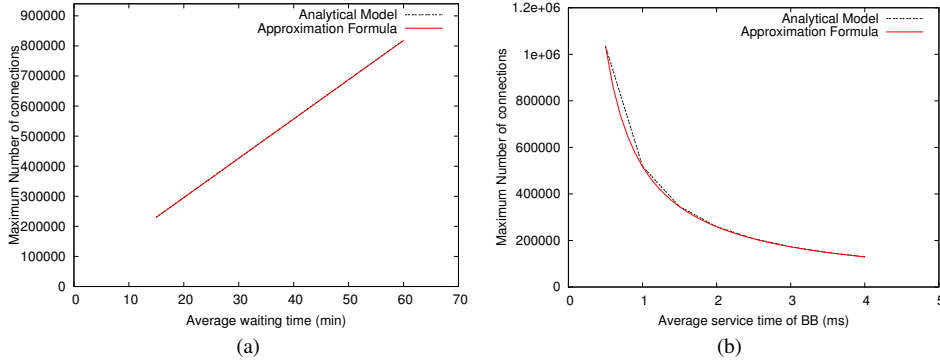


Fig. 5. Maximum capacity of the system (number of connections) shown as a function of (a) the average waiting time, (b) the average service time of the bandwidth broker: Analytical Model vs. Approximation Formula.

5 Deploying a Scalable PBM System

From the proposed analytical model and the previous dimensioning formulas, it is possible for an operator, once he has characterized the behavior of their customers in terms of average service duration (μ_S) and average waiting time between two consecutive requests for the same customer (μ_W, μ_F), to calculate the maximum number of active connections that will be supported by the PBM system. However, in order to instantiate this architecture in a real network, the operator also needs to determine the number of PDPs (respectively LPRs) to deploy physically in the network. The number of PDPs depends on the capacity of each instance to process a certain number of active connections simultaneously. Of course, this delay depends on the frequency of

resource allocation requests issued by the QPS PEP and the hardware architecture of the device that is supporting the PDP software. We have conducted a number of experiments to highlight the evolution of the processing performance of the Management Center (PDP+LPR) on a particular platform according to the number of active customers (i.e. The number of customers currently connected to the PDP).

The test-bed includes an instantiation of the PDP and the LPR on two different computers running on a Linux-based OS. Each of them has the following characteristics: RAM=512Mo, CPU=P4-2489Mhz, Cache Memory=512Kb. Each PDP is connected to 58 edge routers. Each connected customer generates a resource allocation request every minute (this is deliberately high but in the reality it should be lower).

The results of these experiments are illustrated in Figure 6, which shows how the processing performance of the Management Centre evolves with the increase in the number of active customers. From this figure we see that the PDP is able to manage efficiently up to 7,000 active customers without any real loss of performance. With this load, the system doesn't use the memory swap and the RAM usage increases linearly with the number of connected customers. In addition, the mean CPU usage ratio of the PDP is around 12%. When the customer pool is below the 7000 limit, the Management Centre service time is always below 100ms with a mean of about 45ms (this is approximately the same value used in the analytical study: $(\mu_p + \mu_l)$).

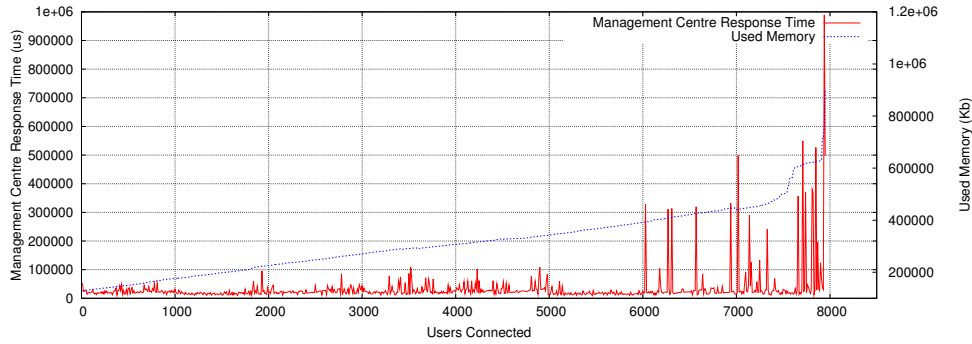


Fig. 6. PDP performance vs. PDP load.

Above this number of active customers, the experiments show long service times. This is principally due to the high usage of the physical memory and the need for the system to start swapping. At this point, the system is in an unstable situation and the Management Centre service delay increases significantly with the number of active customers. According to these results we can consider that, for this hardware architecture, the maximum number of active customers a particular PDP with this particular hardware can support efficiently is 7,000.

From these results and the analytical study, we can provide a very simple guideline for the instantiation of the PBM system. Depending on customers' behavior (average service duration and average waiting time between two consecutive requests), the analytical model gives the maximum capacity of the PBM system (i.e. maximum number of active customers). Let C^{Max} denote this maximum capacity. This capacity can be expressed in terms of the number of necessary PDP devices. According to the experiments, the number of PDPs needed should be equal to $C^{Max} / 7,000$ (if using the same hardware architecture as in the experiments). If, for example, an operator expects to have a maximum of 220,000 active customers and if he is willing to provide a Voice over IP service and let's say that in this case the average duration of a phone

session is $\mu_s = 2\text{min}$ and the average time between two consecutive calls for each customer is $\mu_w = 15\text{min}$, then the number of PDPs to instantiate will be $220,000/7000 \approx 32$ PDPs. This means that if the operator deploys one BB and 32 PDP, and provides the customers with the QPS PEP software, he will provide a scalable and stable on-demand policy-based resource allocation system until it reaches 220,000 active connections. This number does not represent the number of customers, which can be much greater, since not all the customers are active at the same time. No doubt, numerical performance can be enhanced using more advanced hardware.

7 Conclusion and Future Work

In this work, we have presented an analytical model for an on-demand policy-based resource allocation and provisioning system for stateless IP networks. Our objective behind this work is to use this analytical model to derive dimensioning formulas to be used to design such systems. As it stands, the performance of the system depends on the performance of the BB as well as the underlying PDP instantiations. Using these results, we also proposed a simple approach to calculate the number of PDPs to deploy in the system and ensure the targeted scalability feature. Both the derived dimensioning formulas and the deployment rules can be used by network operators as a design tool for their on-demand policy-based resource allocation and provisioning system.

As for future work, one point that can be considered is the design of a distributed scheme for the dynamic assignment of PDP to the customers (QPS PEP). This dynamic assignment should necessarily deal with the balancing of load among PDPs and the limitation of the load within each PDP to a desirable value (i.e. to a maximum of 7000 connections per PDP for our plate-form implementation).

References

1. Verma, D.: Policy-Based Networking—Architecture and Algorithms. New Riders Publishing, (2000)
2. Aib and R. Boutaba, "On Leveraging Policy-Based Management for Maximizing Business Profit", IEEE Transactions on Network and Service Management. Vol. 4 - n°3, pp. 25-39, (2007)
3. Haddadou, K., Ghamri Doudane, S., Ghamri Doudane, Y., Agoulmine, N.: Designing Scalable On-Demand Policy-Based Resource Allocation in IP Networks. IEEE Communications Magazine, Vol. 44 - n°3, pp. 142-149, (2006)
4. Pujolle, G., Korner, U., Perros, H.: Resource allocation in the new fixed and mobile internet generation. ACM International Journal of Network Management, (2003).
5. 3GPP TSG SA WG2. IP multimedia subsystem (IMS). Stage 2 (release 8). Technical Report SP-36 v8.1.0, 3GPP, (2007)
6. ITU-T Recommendation No. E.721: Network grade of service parameters and target values for circuit-switched services in the evolving ISDN. (1999)
7. ITU-T Recommendation No. E.500: Traffic intensity measurement principles. (1998)
8. Bolch, G., and al.: Queuing Networks and Markov Chains. Modelling and Performance Evaluation with Computer Science Applications, Wiley- InterScience, (1998)
9. Hodges, J., Morgan, R.: Lightweight Directory Access Protocol (v3): Technical Specification. RFC 3377, (2002)