Chameleon: an Architecture for Advanced End-to-End Services

Carlos Alberto Kamienski and Djamel Sadok

Computer Science Center, Federal University of Pernambuco
Caixa Postal 7851, Cidade Universitária, Recife/PE, 50732-970, Brazil
Tel: +55 81 3271-8430, Fax: +55-81 3271-8438
E-mail: {cak, jamel}@cin.ufpe.br

Abstract

Current Internet is not able to deploy advanced end-to-end services to support new applications, like interactive multimedia. The best effort service, supported by the Internet, does not offer Quality of Service (QoS) performance guarantees to these applications. Although technology for providing QoS-based services inside domains (networks) is available, the challenge is how domains should interconnect to each other in order to deploy such services in an end-to-end fashion, i.e., crossing domain borders. This paper presents the Chameleon Architecture, which enables the delivery of advanced end-to-end services without being affected by the number of domains involved and the underlying QoS technology that they use. A hierarchical model for service negotiation is proposed, which is compared, via a simulation study, with the traditional bilateral (cascade) model, most commonly used today for relationships between domains.

Key-works: quality of service (QoS), service negotiation, resource provisioning.

1. Introduction

The Internet has become a reality for many people around the world. However, the current best effort service is not suitable for supporting new advanced applications, such as interactive multimedia, that will only be provided with the introduction of Quality of Service (QoS). In order to achieve this, QoS needs to be constant along the entire path between source and destination, i.e. end-to-end, across several domains (networks) with different administrations and technical characteristics.

There are basically two approaches for introducing QoS in the Internet [17]. IntServ reserves resources for every flow in every router along the path using the RSVP protocol. There are serious concerns about the scalability of IntServ, because it generates too many signaling messages and needs to keep a lot of state information. DiffServ deals with flow aggregation in order to be scalable and has been used in some experimental projects to deploy QoS. Some people also state that if one has an over-provisioned network it is not necessary to implement complex mechanisms because QoS is obtained automatically. Our opinion is that no matter if a particular network uses IntServ, DiffServ, over-provisioning or other mechanism. Alone they are not enough for offering end-to-end guarantees.

The idea of deploying advanced services has increasingly been receiving more attention in the last years [1][2][14]. This paper presents the Chameleon architecture, which aims to contribute with the present discussion by placing the required functions into three different planes: service plane, operation plane and monitoring plane. In the service plane, attention is given to the hierarchical model of service negotiation, which offers efficiency and scalability. This model showed significant results when submitted to a performance evaluation.

The rest of the paper has the following structure. Section 2 presents an overview of advanced services in the Internet. The Chameleon architecture is presented in Section 3 while Section 4 emphasizes services negotiation models. In Section 5 results of a simulation study are shown. Section 6 presents related work and finally Section 7 draws some conclusions and presents future work.

2. Advanced Services in the Internet

The traditional best effort service applies similar treatment to all users and applications. On the other hand, an advanced service may be seen as one that offers a differentiated treatment for its traffic (probably at a higher cost).

Services may be classified by their geographic scope. An intra-domain service is to be used only within the boundaries of a domain, which is a network, e.g., an user network, an ISP or an autonomous system. An end-to-end service may be used when data sources and destinations are located in distinct domains, possibly with several other domains along the path between them.

Transport services make up the necessary infrastructure for deploying end-to-end services, which are implemented and negotiated by domains. End-user services are those that are meaningful to end users and are related to their network requirements. Providers that sell services to end-users must map end-user services into transport services in order to be able to participate in the deployment of an end-to-end service.

A Well-Known Service (WKS) [7] is a transport service with a clear and unambiguous definition of the performance guarantees a provider offers to or wants to receive from another provider when an agreement is being negotiated. It must have the same behavior in all domains where it is implemented, so that end-to-end services may be deployed to the end-users (although in principle each domain is free to implement a given service using a different technology). Domains must use an identifier (WKSID) for specifying the desired service during the negotiation.

A SLA (Service Level Agreement) is a description of an agreement between a provider and a client, which may be an end user or even another provider. The SLA contains, apart from some business clauses, a technical specification of the service called SLS (Service Level Specification). According to [5][10][12][16], a SLS includes information such as: geographic scope, data flow identification, traffic profile (rate, burst), excess traffic treatment (drop, shape, etc.), performance guarantees and service schedule.

3. The Chameleon Architecture

The Chameleon Architecture, which aims to provide advanced end-to-end services in the Internet is depicted in Figure 1. It is divided in three logical planes, in order to provide flexibility to service definition and negotiation, efficient implementation and control of proper operation of contracted services. The planes are service plane, operation plane and monitoring plane. This organization provides users with a homogeneous and integrated view of the network, although de deployment of an end-to-end service may need the cooperation of many networks, possibly implementing different QoS technologies.

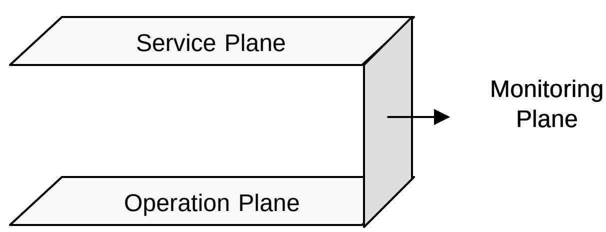


Figure 1 – The Chameleon Architecture

3.1. Service Plane

The Service Plane has a fundamental role in the Chameleon architecture, providing an abstract interface for service negotiation, so that all networks offer a similar external behavior.

An abstract interface is provided through the combination of well known services, standard SLSs an a service negotiation model.

The entity which implements the functions of the Service Plane in each domain is the Service Broker (SB). It is responsible for, e.g., the traffic prediction, service negotiation, resource provisioning control, admission control and definition of policies for collecting and passing feedback information by the Monitoring Plane. The SB may be seen as an extension of the Bandwidth Broker (BB) [8], proposed for DiffServ, but with two basic important differences. Unlike the BB, the SB is not tied to any QoS technology, and it is able to negotiate services based on other QoS parameters (delay, jitter), and not only capacity[1].

The SB is a logical entity and its actual implementation may be distributed over several software and hardware components working together, in order to get flexibility and robustness. It has a repository where all information about purchased/sold services are maintained, as well as up-to-date information about domain resources and policies used to map services to the adopted QoS technology. From the point of view of SLS enforcing and policing, border routers are basic components which the SB have to deal with (according to the proposed architecture, routers belong to the Operation Plane).

Traffic Prediction

The SB continuously receives service activation requests and traffic information collected by border routers. Sample measurements gathered at regularly spaced time intervals during a window of duration T_{meas} are made. Border routers compute the mean, m, and the variance, s^2, of the samples and send them to the SB, which uses this information as a basis for predicting future traffic. An example of used predictors [3][4][15] includes the Gaussian predictor being considered in this paper. This is due to the fact that when the number of individual flows gets large the aggregate arrival rate tends to have a Gaussian distribution under Central Limit Theorem [3].

The estimation of the capacity to be negotiated is calculated by $\hat{C} = m + as$, where a is a multiplier that controls the extent to which the predictor accommodates variability in the samples. In a Gaussian approximation to the negotiated capacity it is expected that the capacity $\hat{C}$ is exceeded with probability $1 - G(a)$, where G is cumulative distribution of the standard normal distribution.

The SB uses estimated capacity $\hat{C}$ to negotiate services with another domain. One very important point is the predictor's ability in producing good estimations. Two problems may arise, namely, overestimation and underestimation. Underestimation should be completely avoided, since it may cause contract violation as some of the QoS parameters are not fulfilled. On the other hand, overestimation should be minimized, in order to avoid the provisioning of too much resources that will not be used at all.

3.2. Operation Plane

Each domain, through an internal set of policies, maps negotiated services to some mechanism used to resource provisioning and equipment configuration. Domains may implement QoS schemes like DiffServ or IntServ, as well as use any other technology (even over-provisioning), as long as they are able to meet the SLSs. The Chameleon architecture aims to separate service offerings from adopted technology decisions, i.e., the operation plane is encapsulated into each domain and its implementation may be changed, at least in principle, without adversely impacting service offerings.

[1] Here "capacity" is being used instead of the more ambiguous term "bandwidth".

3.3. Monitoring Plane

This plane is orthogonal to the other two planes. It collects information from the operation plane and feeds it into the service plane. It is responsible for continuously measuring QoS parameters for each service, sending the results to interested parties and possibly taking correcting actions. This plane may be used within domains, or between domains, in order to police their SLSs. It also permits end users to receive information about their services performance, so that a set of actions may be taken for improving or degrading the quality[2].

4. Service Negotiation

The negotiation for end-to-end service subscription has two phases, the end-user service negotiation and the transport service negotiation. End-user service negotiation does not produce any kind of resource provisioning, just the intention of purchasing/selling this service between users and providers. The user starts the negotiation requesting a service (e.g., a telephony service). The service provider maps it to a transport service and aggregates these requests for further (re)negotiation with access providers. Access providers receive service subscription requests from several end-user service providers (manually or automated by a simple protocol), and perform transport service negotiation. If the access provider is the end-user service provider, negotiation of end-user services is clearly simplified.

Transport service negotiation results in resource provisioning in every domain involved in an end-to-end communication. As this process is independent from the end-user service negotiation, it gives providers a great deal of flexibility in choosing the best way of transport service negotiation.

Resource provisioning may be done through advanced or immediate reservations. Advanced reservations are based on utilization statistics and must guarantee that most service activation requests will be accepted. On demand reservations are necessary in order to adapt available resources to instantaneous needs of users/applications. This paper only considers advanced reservations.

4.1. Bilateral Negotiation

Bilateral negotiation is a more traditional model, based upon current relationship models between domains, and it has been proposed for Bandwidth Brokers [8]. In this model, a domain negotiates with a neighboring domain, which in turn negotiates with a next domain, “rippling” through until the destination domain. This negotiation may be done on demand (immediate reservations) or triggered by a pre-defined event, like a time interval, which permits the use of advanced reservations. The negotiation proceeds by SLS exchanges between domains, through the use of some protocol, like Internet2/Qbone SIBBS [13].

4.2. Hierarchical Negotiation

This model introduces the new concept of Service Exchange (SE), which is a central entity that coordinates service definition and negotiation among a group of participant domains. It is similar to the Clearing House model used in the telecom area and some new bandwidth trading services (called Bandwidth Exchanges). The SE performs negotiations on behalf of his participant domains, unlike the bilateral model, where each domain needs to have individual agreements with several other neighboring domains.

---

[2] If the SLA contains some clause about cost reduction for lower quality.

In order to achieve its goal, a SE needs some information, such as: available services (WKSs); topology of its area and inter-domain links characteristics. Furthermore, each SB periodically sends service purchase and sales information (Figure 2a). Using this information, the SE performs periodically negotiation “rounds”, which result in service permission (total or partial) or refusal. Service permission implies that resources are granted all the way from source to destination.

There are two types of domains in this model: service buyers and sellers. Some domains may play both roles. Seller domains send information to the SE about their external links and internal paths. For each service, the SLS specifies its identification (WKSID), and some QoS parameters, like bandwidth, delay, jitter and packet loss. For the buying of services, important information include service identification, scope, traffic specification and QoS parameters.

The hierarchical negotiation process starts with domains making predictions based on past service utilization and sending purchase and sale information requests to the SE. Then, the SE computes which requests will be permitted (accepted) and the reservations that must be done in every participant domain. Then it informs domains about their respective negotiation results. This process is repeated at every negotiation round. Additionally, domains may negotiate details of services with other neighboring domains such as given particular QoS mechanisms (e.g., RSVP signaling) or packet marking strategies for identifying flows.

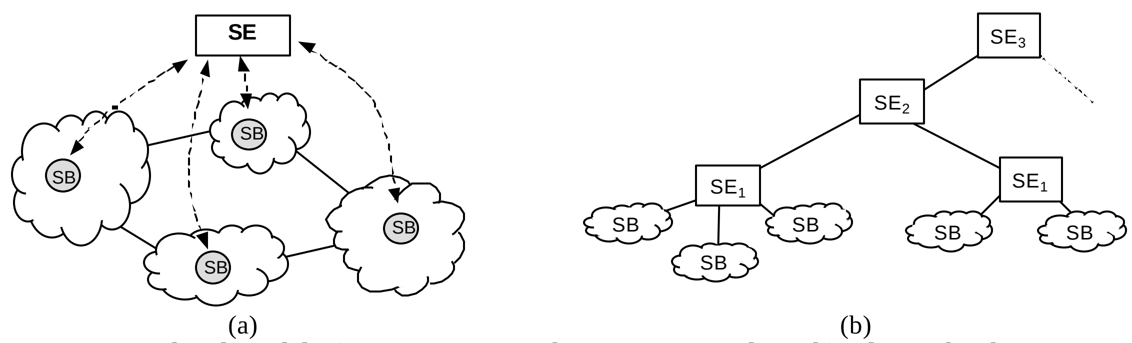


Figure 2 – Hierarchical model; a) Communication between SBs and SE; b) relationship between SEs

This model provides scalability and efficiency to service negotiation. A serious criticism to the bilateral model is the possible generation of too many signaling messages [6]. A SE always receives service purchase requests for aggregate traffic. Within an SE area, SBs aggregate traffic by service and destination, make predictions and send them to the SE. SEs also have a hierarchical organization. Each SE aggregates the requests where the destination is outside its area and sends them to a higher level SE, and so on (Figure 2b). SEs at a same level are not allowed to communicate peer-to-peer. In this paper only one level of SE is being considered, that is, one SE and SBs from participant domains.

The efficiency from the hierarchical model is due to the knowledge the SE has of its area allowing it to perform negotiations considering several criteria, e.g., inter-domain QoS Routing (QoSR). Here, a SE can determine the best routes among domains satisfying the requested services. When SBs receive this information, they feed it to respective interior (IGP) and exterior (EGP) routing protocols. Consequently, protocols and routing tables need to be extended in order to consider more information in route decisions, beside the destination domain (service and possibly source domain). However, QoSR is just an additional feature of the hierarchical model and it does not need to be implemented by every SE.

During the service negotiation process, SEs aims to achieve the following goals:

- Service purchase: most service purchase requests should be accepted.

- Precise reservations: once reservation are done, services should not offer lower QoS to their users caused by lack of resources.
- Service sale: most available services should be effectively sold.
- Efficient implementation: implementation has to be efficient, because SEs may experience a high processing load.
- Fairness: negotiation should give equal opportunities for all sellers and buyers.

In this paper, the focus is mainly on the two first goals.

5. Evaluation

This section presents some results of a performance evaluation based on simulation that has being carried out on the Chameleon architecture using ns-2 [9] network simulator.

5.1. Simulation Model

Simulations have been done using the RNP2 (Brazilian Research Network) [11] topology, showed in Figure 3, where each state PoP was considered a domain (27 domains). Link capacity varies from 20 Mbps between domains 1 and 2 (link 1-2) to 1 Mbps for domains with lower traffic demands. A domain's internal capacity may be 155 Mbps, 34 Mbps or 2 Mbps. Internally, each domain was configured with a delay of 10 ms. Link delays were configured according the their physical length to 10, 15, 20, 30 and 40 ms. For instance, link 1-2 and link 1-13 were configured with 10 ms and 40 ms respectively.

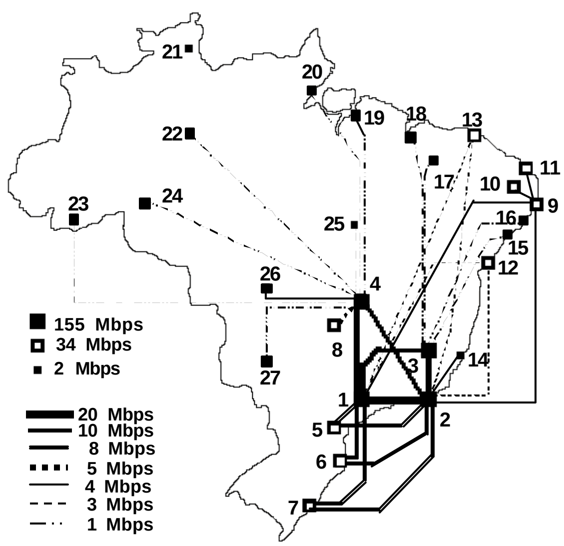


Figure 3 – Simulation topology

Service activation requests refer to a end-user voice service. The voice call arrival rate in each domain i (i = 1, 2, ..., 27) is modeled as a Poisson process of intensity l_i calls per second and call duration is exponentially distributed with a mean of $1/m = 120$ s. Traffic load arriving at each domain is defined as $r_i = l_i / m$ .Voice sources are modeled as an on-off Markov process, which alternates between "on" and "off " periods and is it also exponentially distributed with average duration of 1.004 s and 1.587 s, respectively. Each source generates CBR traffic at 80 Kbps[3] when "on" and 0 Kbps when "off ". To support this end-user service,

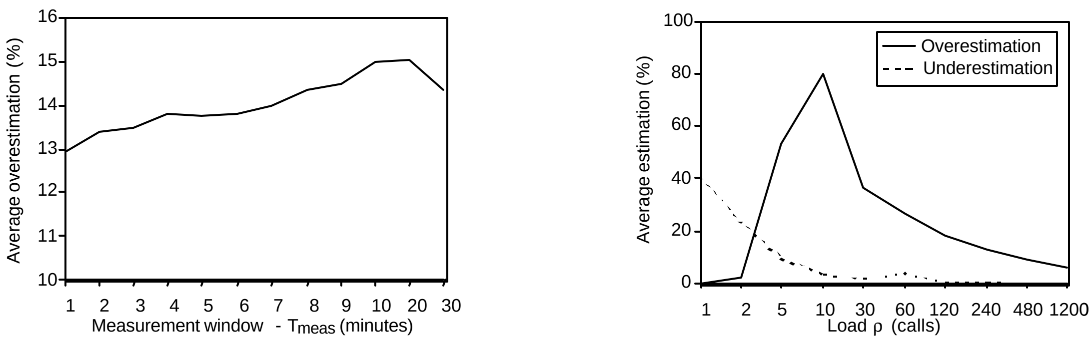
[3] Using a 64 Kbps PCM coder with a 20 ms frame and also considering IP (20 bytes), UDP (8 bytes) and RTP (12 bytes) headers, it results in 80 Kbps.

a simple transport WKS was defined, with just one parameter (besides capacity):it accepts a maximum delay of 150 ms.

Traffic load generated in and received by each domain is proportional to their output and input link capacity. Domains constantly generate calls targeted to other domains. System load, given by the number of simultaneous active services, has been varied between 1000 and 10000. For each call its destination domain is generated according to the load share.

5.2. Gaussian Predictor

Simulations in this section evaluate the Gaussian Predictor, related to the extent it can deals with changes in aggregate throughput, avoiding underestimation and minimizing overestimation. For this study, it was considered a predictor of a single domain. Simulation time was 1 hour, similarly to the following sections.



(a) (b)

Figure 4 – Estimation effectiveness; a) overestimation related to the measurement window Tmeas; b) overestimation and underestimation with different loads (ρ);

The effectiveness of the predictor for generating estimates close to the real future traffic is evaluated. Figure 4a presents the capacity overestimation for a load r = 240 calls varying the measurement window T_{meas} from 1 up to 30 minutes. The lowest overestimation value (about 13 %) is obtained for T_{meas} = 1 minute, just because it can follow short time-scale traffic fluctuations. For T_{meas} = 20 it presents its highest overestimation value (almost 15 %) and after that up to T_{meas} = 30 it falls down. Thus, it is not strictly necessary to use short prediction measurement windows and consequently intervals between negotiation round may be more spaced. However, voice traffic typically varies during a day and too long measurement windows (more than 1 hour) should be avoided [4]. For all simulated values of T_{meas}, underestimation was under 1 %.

Figure 4b shows an underestimation with T_{meas} = 1 for load r varying from 1 to 1200 calls. For r under 10 calls, estimates are incorrect. Overestimation had a peak of 80% for r = 10 and the underestimation was 40 % for r = 1. Below r = 100 calls, overestimation was higher than 20 %, because for lower loads the aggregate arrival rate does not have a normal distribution. In general, the higher the load, the lower the over and underestimation. In the following sections, a measurement window T_{meas} = 1 minute was used.

5.3. Negotiation models

Bilateral negotiation model is quite simple. SBs exchange messages (SLSs) containing basically the requested capacity, maximum accepted delay and accumulated delay. The actual amount of resources granted to a request is the lowest (perhaps null) available amount along the way, if the maximum delay is not exceeded.

To the hierarchical model, a SB sends requests to the SE, which accomplishes a negotiation round. Negotiation consists in analyzing during every service request whether there are resources in the end-to-end path, checking also for the maximum delay accepted. The SE uses an algorithm based on Dijkstra´s shortest path algorithm with some changes. Two schemes were defined, depending on the main priority: 1) lowest delay: chosen path always has the lowest delay; 2) highest capacity: chosen path has the higher capacity. Computational complexity is a concern here, but it does not invalidate the use of such schemes, since there are efficient polynomial algorithms for QoSR when capacity and another QoS parameters (like delay) are considered [17]. Each service request involves a search for an available path for the tuple *<source domain; destination domain; service>*.

The first simulations of this section compare these two hierarchical model schemes with the bilateral model. Figure 5 compares granted with requested resources for load r varying from 1000 to 10000 calls. Under these conditions, both hierarchical schemes present similar performance, granting 100 % for r below 2000 calls and more than 98 % for r up to 3000 calls. For the simulated scenario, r = 3000 calls is the maximum acceptable bound, so that the SE is able to refuse less than 2% of requested resources.

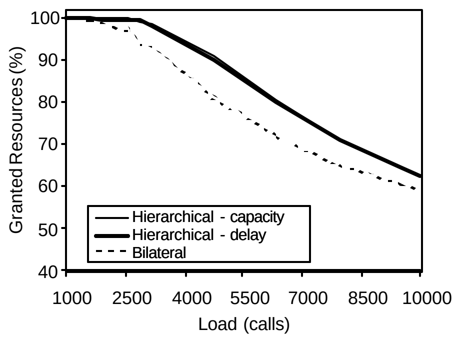
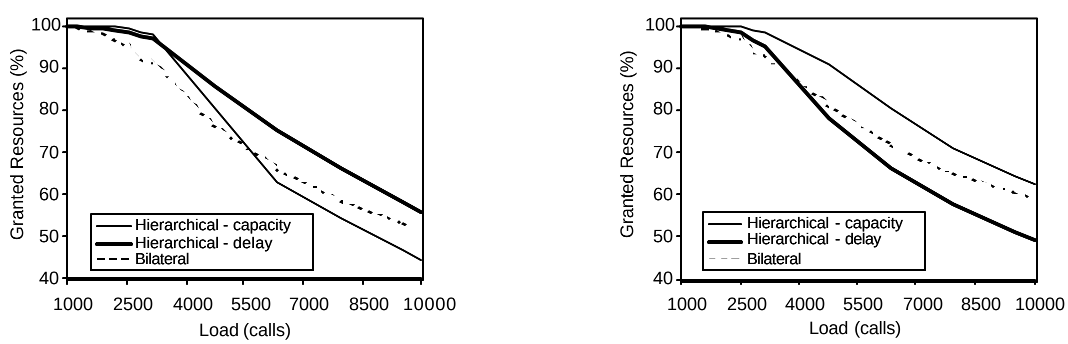


Figure 5 – Granted resources

The bilateral model presented a lower performance for every value of r higher than 1500. This is because the hierarchical model performs a type of QoSR and it is able to find alternate paths for negotiating services and granting resources. The gain of the hierarchical model varies from 0 % to 10 %, with 5 % in average for the higher capacity scheme. For r = 3000 the observed gain is 5.5 %.



(a) (b)

Figure 6 – Granted resources with changes in link 1-2 configuration; a) capacity changed to 5 Mbps; b) delay changed to 50 ms.

In order to compare the two schemes of the hierarchical model, we made some changes in link 1-2, and results may be observed in Figure 6. First, its capacity was changed to 5 Mbps from the original value of 20 Mbps (Figure 6a). The highest capacity scheme presented a considerably lower performance than the lowest delay scheme (up to 12 %). From r = 6000 the highest capacity scheme presents lower performance than the bilateral model, with almost 8 % for r = 10000. This behavior occurs because the bilateral model always looks for paths with lower capacity, even though this path involves more domains and hops. This means that in order to take care of a particular service request, frequently it uses up resources that otherwise would be granted to other requests. This result is in line with the well known preference for routing based on shortest paths (considering the number of hops), which uses up less resources.

For the same reason, the lower delay scheme has a worst performance than the higher capacity scheme when the delay of link 1-2 is changed to 50 ms (Figure 6b). The lower delay scheme chooses alternate paths (like 1-4-2 and 2-3-1), which have a lower delay now. Thus, a larger amount of resources is used up, to deal with the same number of service requests. The outcome is that the ratio of granted resources falls down. An interesting aspect is that performance is not changed for r up to 3000, in both cases of Figure 6 The reason is straightforward: involved links do not represent a bottleneck with this load.

5.4. Admission control

In this section the call admission control (CAC) mechanism is evaluated, comparing generated, requested, granted and admitted calls. Figure 7 shows the results. The number of requested calls (based on predicted estimates) sent by all SBs to the SE is 19 %, 13 % and 12% higher than the system load (generated calls) for r = 1000, 3000 and 10000, respectively. These results are compatible with those of section 5.2.

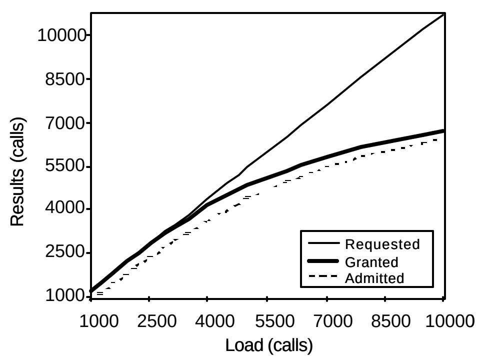


Figure 7 – Generated (load), requested, granted and admitted calls

CAC for the WKS is done by checking whether there are sufficient resources (capacity) to admit a service activation request (call). The rate used for each call is 32 Kbps. First, the SB checks for availability of internal resources and then it also checks the link to the next domain. If there are enough resources, the available resource count is updated and the call is admitted. Otherwise, it is blocked. When a call is finished, a tear-down message is sent the CAC mechanism and the available resource count is increased.

Figure 7 shows that 97 % of generated calls were admitted for r = 3000, which represents definitely the best load for the simulated scenario. For r = 10000, 67 % of generated calls are admitted, although it represents 96 % of granted calls. Admitted calls represent 91 % of generated calls for r = 1000. This is because under lower loads the average rate of each call differs considerably of 32 Kbps. Even for the best case, SBs are not able to admit every call, because of the bursty on-off traffic nature, where calls are generated when no resources are available.

Next, we evaluate the effectiveness of the provisioning performed by the SE for all domains and inter-domain links. To this end, the local CAC mechanism described before was adapted to check the end-to-end availability of resources, that is, in every domain from source to destination. Thus, we may compare the number of calls admitted by local and end-to-end CAC mechanisms. This scenario is depicted in Figure 8. For a load r = 3000, 3.5 % more calls are admitted by the local CAC mechanism compared to the end-to-end CAC mechanism. This suggests that possibly the network may face some problems to meet the QoS parameters agreed in the SLS. A more accurate way to analyze the reasons for this behavior is by simulating real traffic in the network, which was not done in this study because of practical limitations.

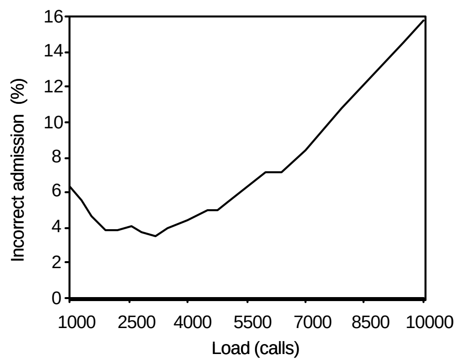
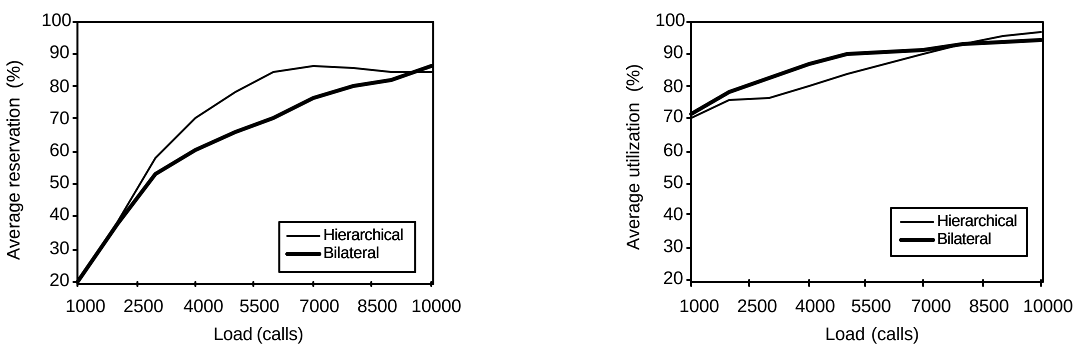


Figure 8 – Incorrectly admitted calls

From r = 3000 this difference increases according to the load up to 15.7 % for r = 10000. When call load gets higher, beyond network capacity, its performance is uncertain. With 15.7 % of calls being admitted incorrectly, probably most calls will have a performance below the expectations.

5.5. Resource provisioning and utilization

Our next simulations evaluate the amount of reserved resources compared to the available resources and the amount or used resources compared to the reserved resources, for inter-domain links.



(a) (b)

Figure 9 – Resource provisioning and utilization; a) link reserved resources (% out of total resources); b) link used resources (% out of reserved resources).

Figure 8a shows that the hierarchical model can reserve more resources than the bilateral model. It reserves 9.4 % more resources for r = 3000 with an average of 8.9 % (considering all simulated loads). On the other hand, bilateral model uses 3 % more resources than hierarchical model, proportionally to the reserved resources (Figure 8b). Utilization refers to resources used up by admitted calls. Our conclusion is that in the hierarchical model the over-provisioning is slightly higher for most loads. It is 2.2 % higher for r = 3000, 5.5 % higher for r = 5000, and 4 % lower for r = 10000.

The relation between resource provisioning and utilization within domains is similar to inter-domain links. The hierarchical model generates reservations 1.1 % higher in average and the bilateral model uses 3.8 % more resources (proportional to reserved resources). For both negotiation models the percentage of reserved resources was under 24 %. This is partially because local traffic was not simulated and also because domains are well provisioned internally.

6. Related work

Currently, the Bandwidth Broker (BB) [8] represents the most thorough concept for resource negotiation and provisioning for DiffServ networks. In [15] it is suggested a two-tier model, intra and inter-domain, for managing resources in a DiffServ network. The Internet2 QBone project has come up with its BB model and a protocol for inter-domain signaling between BBs [13].

The TEQUILA [14] project is meant for deploying services with QoS guarantees and it has been involved with SLS definition, negotiation protocols for SLS and intra-domain and inter-domain traffic engineering schemes so that a network will be able to honor commitments assumed in SLSs. Compared to Chameleon, TEQUILA has a broader scope, but Chameleon is more concise and its hierarchical service negotiation model is innovative.

Another project, AQUILA [1], aims to define, evaluate and implement an advanced architecture for QoS in the Internet by introducing a software layer for distributed and adaptive resource control. AQUILA is focused on tool construction and trials in real networks with multimedia services.

The CADENUS [2] project proposes an integrated solution for the creation, configuration and provisioning of end-user services with QoS guarantees in IP Premium networks. It may be considered a complementary work to Chameleon.

In [3] a QoS provisioning method based on a clearing house architecture is proposed, which was the first motivation for our hierarchical model. However, it is limited to capacity negotiation, that is, it is not able to identify services and perform negotiations based on other QoS parameters.

7. Conclusion

In this paper, we presented the Chameleon architecture, which permits the deployment of end-to-end services in the Internet, no matter the number of networks that exist along the path between data source and destination and the QoS technology they adopt. It introduces the Service Broker (SB), which, among other tasks, is responsible for the service negotiation that produces the end-to-end resource provisioning. Two negotiation models are presented: the traditional bilateral model which generates signaling messages between neighboring domains, and the hierarchical model, which uses the Service Exchange (SE) for coordinating the negotiation among domains.

In order to evaluate efficiency and scalability of the hierarchical model, a simulation study was conducted. A comparative evaluation of both models showed that, for our simulated scenario, the hierarchical model could grant up to 10 % more resources for services, considering call load in its optimal level. The effectiveness of the provisioning performed as a result of the hierarchical negotiation was demonstrated by the admission of up to 97 % of generated calls. However, we also found out that under the hierarchical model over-provisioning is 2.2 % higher than the bilateral model, although it can admit more calls.

As future work, we aim to include the operation plane in our simulations, with routers, internal links and real traffic. Other possibilities are simulating the behavior of the hierarchical model when two or more services are being negotiated and the introduction and modeling of fairness in an efficient way in the negotiation performed by the SE.

References

[1] AQUILA Project, http://www-st.inf.tu-dresden.de/aquila, 2000.

[2] CADENUS Project, http://www.cadenus.org, 2000.

[3] Chuah, C. N., et. al., "QoS Provisioning Using a Clearing House Architecture", IWQoS'2000, June 2000.

[4] Duffield, N.G., Goyal, P. & Greenberg, A., "A Flexible Model for Resource Management in Virtual Private Networks", ACM SIGCOMM'99, September 1999.

[5] Goderis, D. et. al., "Service Level Specification Semantics and Parameters", Internet Draft, <draft-tequila-sls-01.txt>, June 2001.

[6] Günter, M. & Braun, T., "Evaluation of Bandwidth Broker Signaling" IEEE 7th International Conference on Network Protocols, October 1999.

[7] Kamienski, C.A. & Sadok, D., "Service Definition and Negotiation in the Chameleon Architecture", First IEEE International Symposium on Telecommunications (IST 2001), September 2001.

[8] Nichols, K., Jacobson, V. & Zhang, L., "A Two-bit Differentiated Services Architecture for the Internet", RFC 2638, June 1999.

[9] Network Simulator (version 2.1b7), http://www.isi.edu/nsnam/ns/, October 2000.

[10] Rajan, R., Celenti, E. & Dutta, S., "Service Level Specification for Inter-domain QoS Negotiation", Internet Draft, <draf-sls-somefolks-00.txt>, November 2000.

[11] RNP, "RNP2 backbone map", http://www.rnp.br/backbone/bkb-mapa.html .

[12] Salsano, S. et. al., "Definition and usage of SLSs in the AQUILA consortium", Internet Draft, < draft-salsano-aquila-sls-00.txt>, November 2000.

[13] Teitelman, B. & Chimento, P., "QBone Bandwidth Broker Architecture", Internet2 QBone Draft, June 2000, http://qbone.internet2.edu/bb/bboutline2.html.

[14] TEQUILA Project, http://www.ist-tequila.org,, 2000.

[15] Terzis, A., et. al., "A Two-Tier Resource Management Model for the Internet", Global Internet'99, December 1999.

[16] Trimintzios, P. et. al**.,** "A Management and Control Architecture for Providing IP Differentiated Services in MPLS-Based Networks", IEEE Communications Magazine, May 2001.

[17] Xiao, X. & Ni, L. M., "Internet QoS: A Big Picture", IEEE Network, March 1999