AQUILA: Adaptive Resource Control for QoS Using an IP-Based Layered Architecture

Thomas Engel, Hermann Granzer, Bert F. Koch, and Martin Winter, Siemens AG
Petros Sampatakos and Iakovos S. Venieris, National Technical University of Athens
Heinrich Hussmann, Dresden University of Technology
Fabio Ricciato and Stefano Salsano, CoRiTeL — Consorzio di Ricerca sulle Telecomunicazioni

ABSTRACT

Support for quality of service is an essential component of the next-generation Internet. The European research project AQUILA is committed to defining a DiffServ-based architecture for delivering on-demand QoS to requesting applications. Focal characteristics of the proposed solution are backward compatibility to the existing Internet and scalability to very large networks. To achieve such goals, AQUILA implements an overlaid distributed control layer, the Resource Control Layer, implementing a novel mechanism for dynamic control of intradomain resources, the Dynamic Resource Pool. On the interdomain aspects, the AQUILA architecture extends the BGRP framework for the aggregation of interdomain reservations to overcome scalability issues. This article describes the general AQUILA architecture, with a special focus on the DRP and BGRP mechanisms.

INTRODUCTION

Today's Internet is powerful and flexible, but, due to its unpredictable behavior, unsuitable for many of the telecommunication applications used in daily life. The next wave of innovations for an Internet-based global network will be mainly powered by the inclusion of new users willing to get reliable service for audio or video communication or for transactions such as stock trading and interactive games. The convergence of such applications toward the Internet requires the support of quality of service (QoS) [1].

The first step toward QoS is the ability to differentiate between different kinds of traffic. The well-known differentiated services (DiffServ) paradigm establishes a general framework for providing traffic handling differentiation within a common IP platform. Within such a framework, the challenge is to dynamically provide QoS guarantees to applications in an easy-to-use way for the human user. Moreover, as the Internet is essentially an enormous collection of various networks run by different organizations, QoS must be provided end-to-end through a heterogeneous infrastructure.

The AQUILA project [2] is targeted to define and implement an architecture for dynamic provision of QoS that is scalable to the global Internet and easily accessible by users. Any architecture delivering on-demand QoS to single users on the scale of the global Internet must be carefully designed to be flexible and *scalable*. The AQUILA key to scalability is the application of a resource reservation paradigm based on hierarchical structures, *resource trees*. This model is found at the intradomain level, with the Dynamic Resource Pool (DRP) mechanism, as well as at the interdomain level with BGRP+ quiet grafting. The AQUILA architecture has been implemented in a working prototype and is being tested in a field trial. This article reports on key concepts of the AQUILA approach.

The remainder of this article is organized as follows. We describe the AQUILA architecture, focusing on intradomain aspects. We cover the interdomain aspects and describe the BGRP+ model. Finally, we report some quantitative results applied to the resource tree model, focusing on interdomain application.

THE AQUILA ARCHITECTURE

QOS DIFFERENTIATION: NETWORK SERVICES AND TRAFFIC CLASSES

In order to provide QoS differentiation at the service level, AQUILA defines a limited set of *network services*. Each network service is meant to support a class of applications with substantially similar requirements and characteristics (Table 1).

0163-6804/03/$17.00 © 2003 IEEE

Network service	Targeted application	QoS target	Traffic class
Premium constant bit rate	Real-time applications with low bit rate variability (e.g., VoIP calls, VoIP trunks)	Very low delay, very low loss	1
Premium variable bit rate	Streaming real-time applications with high bit rate variability (e.g., videoconferencing, high-quality video distribution)	Very low delay, very low loss	2
Premium multimedia	Elastic long-lived applications, with TCP or TCP-like bit rate adaptation (e.g., file transfer, cached video download, low-quality video distribution)	Guaranteed throughput	3
Premium mission critical	Transaction-oriented data applications (e.g., finance transactions, database access, online games)	Low delay, low loss	4

■ **Table 1.** *The AQUILA network services and corresponding traffic classes.*

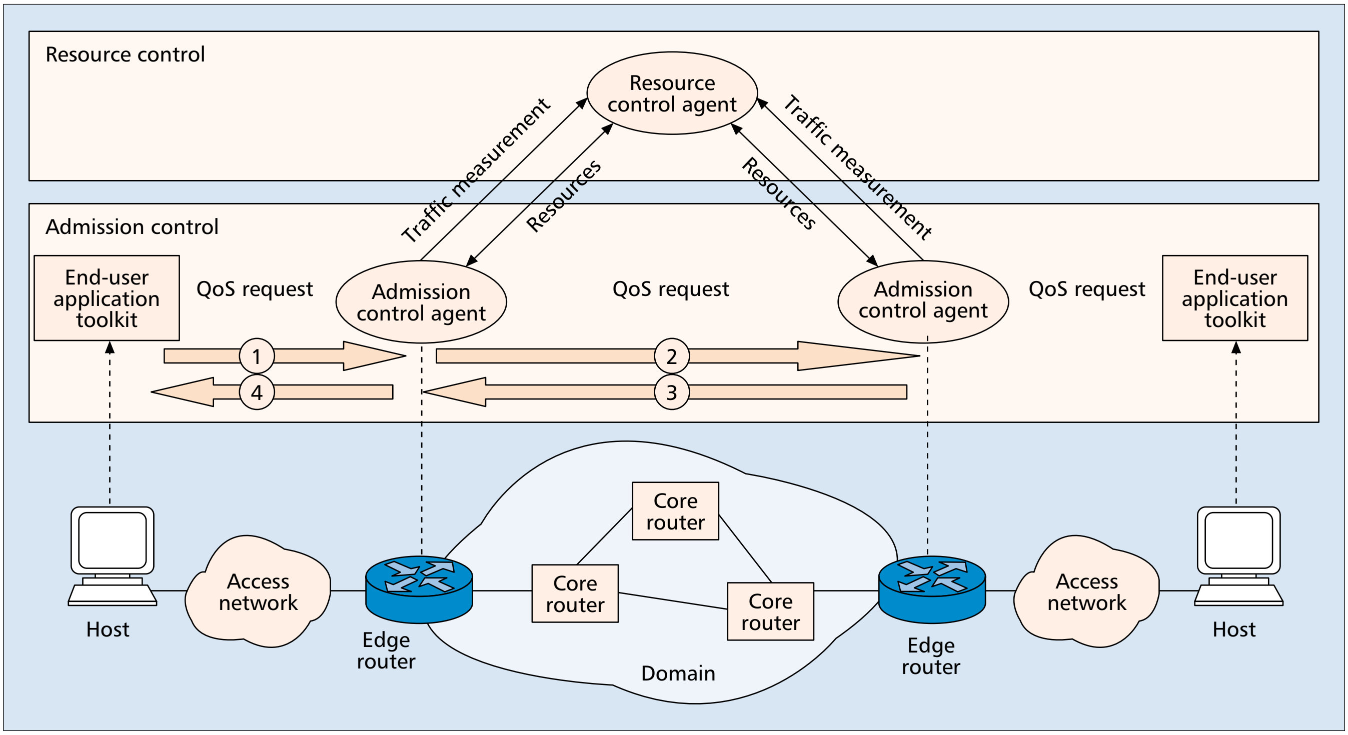
Differentiation at the service level is obtained through differentiation at the packet handling level. In fact, in the spirit of the DiffServ paradigm several scheduling and/or active queuing algorithms available in commercial routers are used to differentiate packet delay and loss on a per-class basis, and at the same time provide separation for traffic aggregates that should not compete on the same network resources (e.g., controlled flows from best-effort traffic, UDP from TCP traffic, greedy TCP connections from short-lived TCP connections). Also, any traffic regulation mechanisms (admission control at flow level, traffic conditioning at packet level) must be tailored to the specific characteristic of the type of traffic.

In designing the traffic handling mechanisms AQUILA has taken into account all of that. AQUILA defines a set of four *traffic classes* beyond the standard best effort, which map at the network level the network services offered to the applications (Table 1). For each traffic class AQUILA defines the scheduling/queuing mechanisms local to the router interfaces, the packet-level conditioning functions at the network edge (marking, policing, shaping), and the call-level admission control criteria. Moreover, all the admission control and resource management mechanisms within the network are applied on a per-traffic-class basis. Details of the implementation of traffic classes can be found in [3].

THE RESOURCE CONTROL LAYER

The mechanisms for dynamically delivering QoS are implemented in AQUILA in a distributed fashion. Several interacting logical elements collectively constitute the Resource Control Layer (RCL), which can be seen as a distributed overlay control network on top of the DiffServ domain (Fig. 1). The RCL mainly has three tasks, assigned to different logical entities:

- To offer an interface to the QoS infrastructure to legacy applications. This is covered by the *End-User Application Toolkit* (EAT) located in the end-user host. From the network point of view the EAT acts as



■ **Figure 1.** *AQUILA intradomain architecture. The RCL agents are associated with different network entities (EAT → host, ACA → edge router, RCA → domain).*

the RCL front-end, while from the user point of view it provides a QoS portal.

- To control the traffic accessing the network by performing policy control and admission control. This is the task of the *Admission Control Agent* (ACA) located at the network edge. Each edge router of the DiffServ domain is controlled by a single ACA.
- To monitor, control, and distribute the *network resources*. This task is assigned to the *Resource Control Agent* (RCA). The RCA interacts with the ACAs through the DRP mechanism explained below.

The AQUILA architecture distinguishes between *resource control* and *admission control*. The former refers to the dynamic assignment of per-class bandwidth limits to the edge routers — the admission limits introduced below. The latter refers to the handling of per-flow reservation requests done at the edge routers and based on such limits. Both processes run at different timescales and are implemented by independent entities (ACAs, RCAs). This separation and the autonomous operation of these entities are the AQUILA key to scalability and reliability. With this architecture QoS can be dynamically handled within a large AQUILA domain.

Figure 1 shows a representation of the AQUILA architecture. The EAT enables the user application to state its QoS request to the network (i.e., to specify the required network service) and advertise the relevant traffic parameters. This request is sent from the host to the network, that is, from the EAT to the ingress ACA (arrow 1). For some services, specifically those associated with a point-to-point scope, admission control is performed independently at the ingress and egress network edges. In this case it is up to the ingress ACA to contact the egress ACA responsible for the egress edge router on behalf of the requesting EAT (arrows 2 and 3). Finally, it communicates to the EAT the final admission response (arrow 4). With this procedure QoS can be dynamically delivered edge to edge through an AQUILA domain. More detailed information about the EAT can be found in [2].

An IP-QoS architecture must be designed to provide QoS guarantees beyond QoS differentiation. Delivering QoS guarantees requires controlling the amount of traffic entering the network. In AQUILA this task is accomplished by the distributed Resource Control Layer.

QOS GUARANTEES: OVERVIEW OF TRAFFIC CONTROL MECHANISMS

An IP QoS architecture must be designed to provide QoS *guarantees* beyond QoS *differentiation*. Delivering QoS guarantees requires controlling the amount of traffic entering the network. In AQUILA this task is accomplished by the distributed RCL. Several mechanisms act complementary within the RCL at different timescales: provisioning, DRP and admission control. We now provide a brief global view of such mechanisms and their mutual relationships; the interested reader is referred to [3] for the algorithmic details.

The admission control function is fully distributed: each ACA acts autonomously and admits or rejects the QoS request based on the declared parameters and locally stored state information about resource availability in the network. This is expressed in terms of a set of admission limits maintained locally at the ACA. The admission limits represent the maximum amount of inbound/outbound traffic for each traffic class allowed to access the edge router. The rationale behind the concept of admission limits is that congestion in the network core can be avoided by a coarse restriction on the per-class traffic at the network edge. The enforcement of such limits during the admission control phase is meant to prevent congestion or QoS degradation within the core network. Additionally, the ACA can apply any policy constraint configured by the provider.

The global network resources are handled on a per-class basis. Based on a global expectation of the traffic matrix for each class and routing information, an initial *provisioning* algorithm identifies the expected amount of bandwidth used by each class on each link. These values are used for setting the router interface parameters (e.g., Weighted Fair Queuing, WFQ, weights) and to provide the initial values for the set of admission limits introduced above. Based on feedback from global network measurements, occasional reprovisioning can be enforced to track long-term changes in the average traffic distribution.

The assignment of admission limits done during the (re)provisioning phase represents a static per-class bandwidth assignment to the edge routers. The assignment algorithm is based on the expected spatial distribution of input traffic, and must take into account the network topology as well as any relevant routing information. In order to track and react to fluctuations and deviations of the actual spatial pattern of offered traffic from the expected one, the concept of Dynamic Resource Pools (DRPs) is introduced by AQUILA.

THE DYNAMIC RESOURCE POOL MECHANISM

For each class, the DRP algorithm is committed to dynamically adjust the admission limits according to the current request arrival intensity at each edge router. The DRP mechanism is fully distributed and hierarchically organized, and runs between the ACAs and the single or multiple RCA(s). The DRP mechanism relies on the concept of *resource pools*, which is a key novelty introduced by AQUILA. To explain the concept, consider the simple resource pool shown on the left of Fig. 2. Three edge routers are connected to a common core router (CR4) and share the bandwidth of the common bottleneck link behind CR4. The bandwidth values given in the picture represent the maximum amount of traffic allowed on each link for a generic traffic class j: preserving such a limit on any link will avoid QoS degradation occurring for that class. Each edge router n has assigned a maximum budget for traffic class j inbound traffic, the admission limit denoted $l_j(n)$, so their sum does not exceed the bottleneck link bandwidth, $\Sigma_{n=1,2,3}\, l_j(n) \leq$ 18 Mb/s.

Let's assume that ER1 is exhausting its budget due to high intensity of QoS requests for class j, while at the same time ER3 is using only a small portion of its budget due to lack of requests. In this case it is desirable to dynamically shift a portion of the resource budget from ER3 to ER1, that is, to decrease $l_j(3)$ and

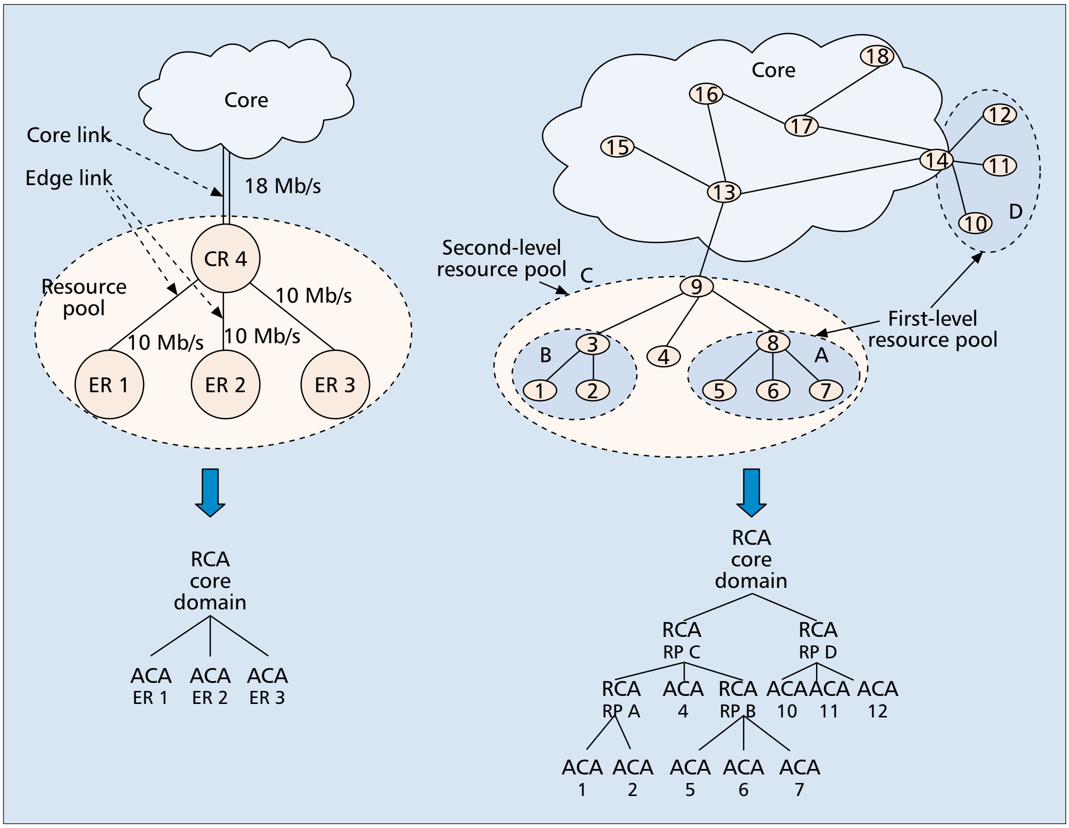


Figure 2. *Examples of a simple resource pool (left) and resource pool hierarchy (right), and the associated resource trees participating in the DRP algorithm.*

The application of the Resource Pool concept is straightforward in case a set of edge routers is connected in a star configuration to a common core router. This approach can be hierarchically extended to build Resource Pools whose elements are not edge routers but other Resource Pools.

increase $l_j(1)$ so that the bottleneck limit is never exceeded. If we consider the per-class bandwidth on bottleneck links as the relevant network resource, we can say that ER1 and ER3 exchanged a portion of their resources. The concept of resource pools arises from the general consideration that not any pair of edge routers can exchange resources with each other, just those sending traffic to the same bottleneck link. The individuation of resource pools is done during the initial provisioning phase, and is related to the presence of bottleneck links, which in turn depend on the topology and routing configuration.

Resource exchange within a resource pool is accomplished through the *Dynamic Resource Pool* algorithm. The DRP runs between the ACAs responsible for the edge routers (the resource pool leaves) and the RCA, which is the logical entity responsible for the core router, the resource pool root (Fig. 2-right). The DRP is based on feedback from the admission control process. In fact, each ACA continuously compares the current level of per-class bandwidth reservation to the assigned budget (i.e., the admission limits). Whenever this value decreases below a lower watermark, the ACA returns an amount of unused resources (the *release block*) to the pool root, that is, it diminishes the local value of the limits and communicates the differential to the RCA. The released resources constitute a resource cushion available to the RCA for successive reassignment to other requesting ACAs. In fact, whenever the level of budget utilization exceeds a higher watermark, the ACA requests a block of more resources (the *request block*) from the pool root (i.e., it asks the RCA to increase its admission limit). The RCA can accept or reject the request based on the current size of the spare cushion. Such a request/release procedure for dynamic adaptation of the admission limits was implemented with the Common Object Request Broker Architecture (CORBA). The DRP mechanism is very simple to implement; any transaction is initiated by the ACAs, and there is no need for the RCA to maintain knowledge of the ACA state.

By tuning the algorithm parameters (namely the lower/higher watermarks and the request/release block size) it is possible to trade off between global utilization efficiency on one side, and DRP signaling overhead on the other. AQUILA implements algorithms for automatic adaptation of such parameters (see [4] for details).

The application of the resource pool concept is straightforward if a set of edge routers is connected in a star configuration to a common core router,[1] as in the above example. This approach can be hierarchically extended to build resource pools whose elements are not edge routers but other resource pools. This is depicted in Fig. 2-right, where pool C is composed of pool A, pool B, and router ER4, since all these elements share the potential bottleneck link between nodes 9–13. Hierarchical resource pools can easily be identified in those network areas that are intrinsically hierarchically structured. In such cases, it is possible to implement the resource management function in a distributed fashion:

---

[1] Dual homing is often used today to improve resilience. With dual homing an edge router is connected to two core routers for redundancy. In this case the edge router may belong to two separate resource pools, and its input traffic will split accordingly.

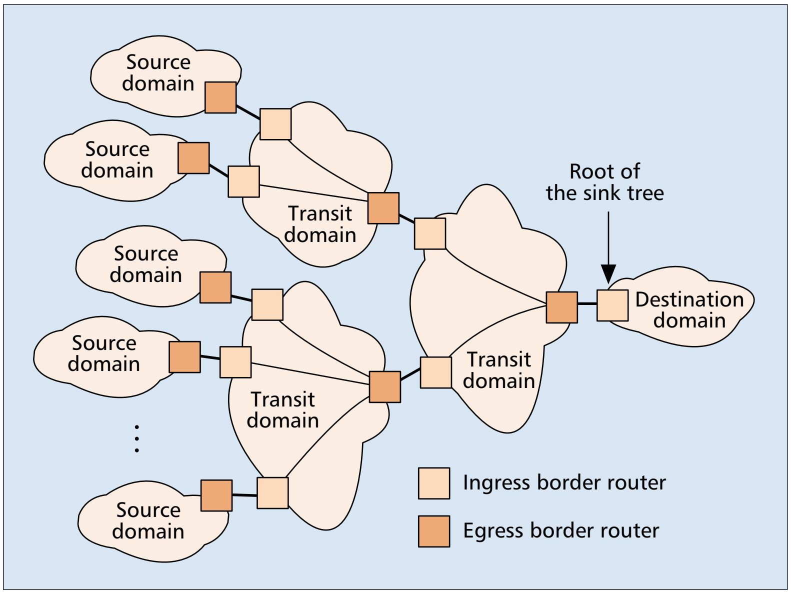


Figure 3. *A BGP sink tree.*

rather than a single centralized RCA for the whole network, it is possible to activate an RCA for each resource pool, and to hierarchically apply the request/release DRP transactions described above between children RCAs and a parent RCA. In other words, a hierarchical resource pool can be mapped into a *resource tree* where each node is an RCA and the leaves are the ACAs, as shown on the right in Fig. 2. The ACAs map the QoS requirements into bandwidth demands on a per-class basis; then the resource tree is used to dynamically distribute the bandwidth toward the root (i.e., the core network). The state information maintained at each node in the tree is very simple. In fact, each element *n* has to remember the amount of bandwidth currently assigned to it by its parent (this is the bandwidth reserved from *n* all the way up to root of the tree) and the portion reserved by its children, but it does not need to know how the bandwidth further splits at lower levels. Thus, reservation states are aggregated from the leaves toward the root. Moreover, the maintenance of a resource cushion at each node reduces the number of signaling messages along the resource tree.

This hierarchically aggregated control scheme, characteristic of DRP in the context of intradomain resource management, is reused by AQUILA for interdomain resource management.

INTERDOMAIN RESERVATIONS

In considering interdomain QoS provisioning it is assumed that all or a subset of the network services defined by AQUILA are supported by all domains. Such services are called *Globally Well-Known Services* (GWKS) [5] in AQUILA. The existence of a common set of GWKS supported by all domains is a necessary condition for delivering predictable QoS through a multidomain path.

The interdomain QoS architecture developed in AQUILA originates from the Border Gateway Reservation Protocol (BGRP) framework proposed in [6]. BGRP provides a mechanism for the aggregation of resource reservations spanning multiple domains. The reservations are negotiated between BGRP agents, which are deployed at each Border Gateway Protocol (BGP)-capable border router of each DiffServ domain. BGRP exploits the property of the BGP routing protocol [7] that builds sink trees while tracing a route toward a particular AS, as depicted in Fig. 3. With BGRP the reservations are aggregated along the BGP sink trees.

BGRP mainly uses three messages: PROBE, GRAFT, and REFRESH. The source domain initiates a PROBE message to establish a new reservation toward the destination domain. It can be issued by a BGRP agent of the source domain for a user reservation request directed outside the domain. The PROBE message contains information about requested network service and amount of required bandwidth for the new flow. The PROBE message is forwarded hop by hop between BGRP agents, alternately on ingress and egress border routers, until it reaches the destination domain (i.e., the root of the sink tree). On its way toward the root domain, it records path information about the traversed domains and border routers. When the PROBE message reaches its destination, a GRAFT message is generated containing an identification of the destination domain (sink tree id). This message travels back to the source along the recorded path. Upon reception of the GRAFT message each BGRP agent has to reserve the requested bandwidth on the local downstream hop (an ingress border router is responsible for a whole domain, while an egress border router is responsible for the local interdomain link). The BGRP agent aggregates the reservation with the existing ones pertaining to the same sink tree. Finally, REFRESH messages are exchanged regularly in order to preserve the established reservation state through the interdomain path. REFRESH messages are also used to reduce the amount of reserved resources when the source domain releases reservations. Figure 4 depicts the exchange of BGRP messages over the interdomain resource control architecture.

With this scheme, reservation states are aggregated on a per-destination basis; in fact, the generic intermediate BGRP agent will not maintain a single reservation state per each active reservation, but rather an aggregated state associated with each sink tree. This dramatically reduces the amount of *state information* stored in the network. However, the aggregation of reservations is just the first step toward scalability. In order to limit the signaling load and processing power required in the BGRP agents, it is also necessary to reduce the number of *signaling messages*. A hint for a possible strategy was already given in [6], the so-called *quiet grafting* mechanism. The idea is to have an "early" response to reservation messages (PROBE) so that they do not always have to travel all the way along the sink tree up to the root. AQUILA extended the original BGRP proposal by fully specifying the quiet grafting mechanism and the related procedures. The resulting proposed architecture is called BGRP Plus (or BGRP+), which is synthetically described in the following.

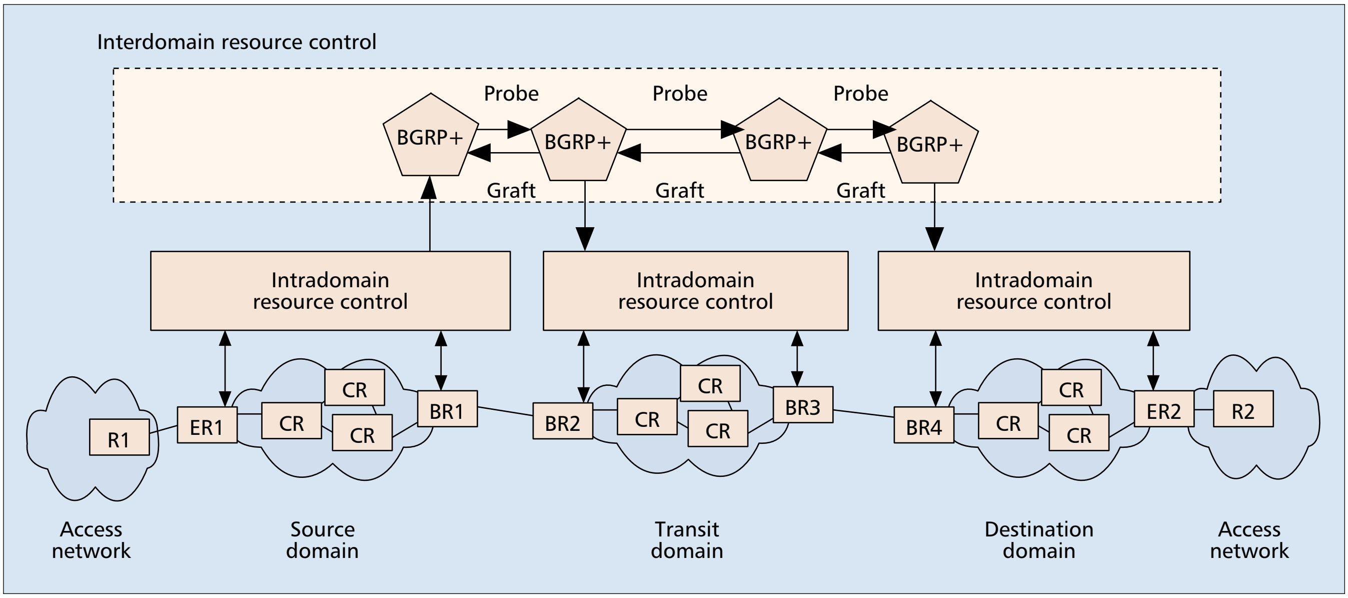


Figure 4. *A reference network configuration for interdomain QoS.*

The objective of quiet grafting is to achieve a reduction in the number of PROBE and GRAFT messages. Quiet grafting enables an intermediate BGRP+ agent to answer positively a PROBE message without the need to further forward it along the sink tree. To provide such early response, the intermediate BGRP+ agent must have pre-reserved a resource cushion for this sink tree so that it can guarantee resource availability to new reservations up to the destination domain without interacting with the downstream agents. Therefore, downstream forwarding of PROBE messages along the sink tree can be avoided as long as the resource cushion for that destination is not exhausted. In principle, several strategies are possible to dynamically build up such a resource cushion at each BGRP+ agent. The mechanism used in the AQUILA implementation is the so-called *delayed release* mechanism. Resources are normally released by the source BGRP agent by sending REFRESH messages with a lower bandwidth value, and intermediate BGRP agents should immediately release the unused resources by propagating REFRESH messages downstream toward the root. Instead a BGRP+ agent will delay the release of unused resources (i.e., the forwarding of REFRESH messages) in order to build up a temporary resource cushion to accommodate successive new reservations toward the same destination.

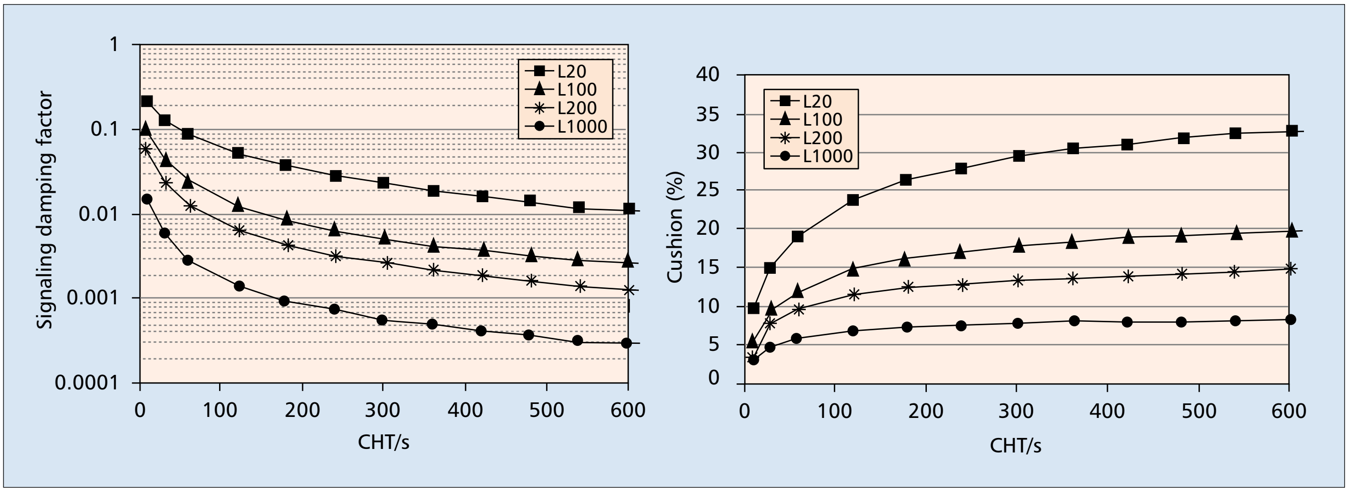
Further details about the BGRP+ mechanism can be found in [4, 5].

PERFORMANCE ANALYSIS OF BGRP+

The scalability of BGRP+ depends on two factors: the number of states and the number of signaling messages that have to be processed by each BGRP+ agent. As seen above, sink tree aggregation reduces the number of states to the order of the number of possible destination domains (estimated around 10^4 in today's Internet), while the quiet grafting mechanism reduces the number of signaling messages.

To evaluate the performance of quiet grafting, we simulated a sink tree with a total of 12 domains: 10 source domains (the leaves), one common transit domain, and the destination domain (the root). The sink tree was fed with a reservation process modeling voice over IP calls: a Poisson traffic model was used for the call process (i.e., exponentially distributed interarrival and holding times), and each call required a single bandwidth unit. During each simulation we measured the ratio of the number of signaling messages forwarded and received by each BGRP agent (i.e., the signaling reduction factor). Such a factor can be decreased by enlarging the resource cushion available to the agent, that is, the amount of bandwidth reserved downstream toward the root but not currently allocated to any upstream child of the sink tree. On the other hand, since the resource cushion represents a volume of *unused* resources, such enlargement means lower utilization efficiency. Again, a trade-off exists between *scalability* and *efficiency*, analogous to what was found with DRP. In the AQUILA implementation the control parameter regulating the cushion size is the *cushion holding time* (CHT), which determines the delay before a certain amount of unused resources is released. We ran several simulations by varying the offered input load and the CHT parameter. The results are shown in Fig. 5. The left graph shows the measured signaling reduction factors vs. CHT for different input loads. The right graph reports the corresponding mean cushion size as a percentage of the total allocated bandwidth.

In the experiments with sink trees of a depth of three domains and 10 source domains originating voice traffic, large load shifts were applied in order to stress the resource management. The results showed that the border routers of the source domains forwarded about 0.9 percent of incoming requests, while the border router of the common transit domain passed 2.3 percent



■ **Figure 5.** *Measured signaling reduction factors (log scale, left) and resource cushions (right) vs. cushion holding times (CHT) for different loads (L20–L1000).*

of the received requests, corresponding to about 0.02 percent of the original requests. The bandwidth cushions were 16 percent at the source domains and less than 6 percent at the transit domain, with 23 percent total. Globally, the number of signaling messages was reduced by two orders of magnitude. Moreover, from Fig. 5 it can be seen that the performance even improves with increasing traffic load. In fact, when high loads are applied, relatively smaller cushion sizes are sufficient to achieve higher signaling reduction factor. Further quantitative results can be found in [4].

CONCLUSIONS AND OUTLOOK

We have given a technical description of an approach enabling end-to-end QoS over large IP-based networks. All the AQUILA architecture can be realized using standard equipment and commercial routers, so the proposed approach is highly backward compatible to the currently existing Internet infrastructure.

The proposed architecture has been implemented completely as a working prototype, which is being used in practical tests in a field trial in the last quarter of 2002. Currently, the QoS features are available to a number of legacy applications (including, e.g., NetMeeting and Real Player) as well as to specially created multimedia Web services.

From a technical point of view, a satisfactory status of QoS over IP has been reached by the efforts of AQUILA and other related projects. An important next step of research will be to investigate in detail the economics of QoS. Currently, nobody can forecast whether the Internet actually will be revolutionized by QoS support. But the AQUILA approach provides a solid technical foundation for these investigations as well as for future realizations.

ACKNOWLEDGMENTS

The AQUILA project is partially funded as IST-1999-10077 by the European Commission within the IST Research Programme. It started January 2000 and finishes December 2002.

The authors wish to express their gratitude to the European Commission's Project Officer Dr. Paulo T. De Sousa, who coached the IST Premium IP Projects, as well as all other members of the AQUILA consortium for valuable discussions. These members are: arvato systems, Germany; Elisa Communications, Finland; Salzburg Research, Austria; Q-Systems, Greece; T-Systems Nova, Germany; Telekom Austria, Austria; Polish Telecom and Warsaw University of Technology, Poland.

REFERENCES

[1] X. Xiao, "Internet QoS: A Big Picture," *IEEE Network*, Mar. 1999.
[2] AQUILA home page: http://www.ist-aquila.org/
[3] S. Salsano, Ed., "Specification of Traffic Handling for the Second Trial," Del. IST-1999-10077-WP1.3-COR-1302-PU-O/b1, AQUILA project. work in progress; http://www.ist-aquila.org/
[4] T. Engel *et al.*, "Resource Management in QoS Enabled IP Networks with the AQUILA RCL," ICTSM10, Monterey, CA, Oct. 2002.
[5] S. Salsano, Ed., "Inter-domain QoS Signaling: the BGRP Plus Architecture," Internet draft, draft-salsano-bgrpp-arch-00.txt, May 2002, work in progress; http://www.ist-aquila.org
[6] P. Pan, E. Hahne, and H. Schulzrinne, "BGRP: Sink-Tree-Based Aggregation for Inter-domain Reservations," KICS 2000.
[7] Y. Rekhter, "A Border Gateway Protocol 4 (BGP-4)," RFC 1771, Mar. 1995.

ADDITIONAL READING

[1] P. Traina , "BGP-4 Protocol Analysis," RFC 1774, Mar. 1995.
[2] E. Nikolouzou, Ed., "BGRPP: Performance Evaluation of the Proposed Quiet Grafting Mechanisms," Internet fraft, draft-nikolouzou-bgrpp-sim-00.txt, July 2002, work in progress; http://www.ist-aquila.org/

BIOGRAPHIES

THOMAS ENGEL received a Dipl. degree in mathematics from Technical University of Darmstadt in 1987. He is currently a principal research scientist at the Networks and Multimedia Communications department of Siemens Corporate Technology, Munich, Germany. Areas of interest are scalable QoS support in IP networks, voice over IP, self-organizing resource management, traffic engineering, and network planning.

HERMANN GRANZER received his physics diploma at Friedrich-Alexander-Universität Erlangen-Nürnberg in 1993. He got his Ph.D. at Freie Universität Berlin in 1996. Starting at Siemens in 1997, he became group leader of the Rapid Prototyping

Group in Information and Communications Networks R&D. Besides coordinating internal prototype projects, he is technical coordinator of the IST project AQUILA.

HEINRICH HUSSMANN holds a diploma in computer science from Munich University of Technology and a doctoral degree from the University of Passau. For several years he was a systems engineer and team leader for advanced development at Siemens. Since 1997 he is full professor of computer science at Dresden University of Technology. He is the author of over 60 publications, including three internationally published books.

BERT F. KOCH (berthold.koch@siemens.com) received his diploma in computer science from Technical University of Berlin, Germany, in 1978. He joined Siemens Corporate R&D in Munich and was software development manager responsible for CAD for microelectronics, operating systems, and administration software for ATM systems. In Siemens' Information and Communications Group he has been project manager of many European ESPRIT and ACTS Projects. Presently he is project manager of the IST Project AQUILA.

FABIO RICCIATO received his Laurea degree in 1999 from the University of Rome "La Sapienza," where he is currently working on his Ph.D. Since 1999 he has collaborated with CoRiTeL. He participates in the AQUILA project funded by the EU. His current research areas include QoS and traffic engineering in IP networks, MPLS, IP over optics, and network optimization.

STEFANO SALSANO received his Laurea degree in 1994 from the University of Rome "Tor Vergata" and his Ph.D. in 1998 from the University of Rome "La Sapienza". Since November 2000 he has been an assistant professor at the University of Rome "Tor Vergata." From 1997 to 2000 he was with CoRiTeL, a research institute on telecommunications, coordinating research activities in the IP-related area. His current research interests include QoS and traffic engineering in IP networks, IP telephony, MPLS, and IP over optics.

PETROS SAMPATAKOS received his Dipl. degree in electrical engineering and computer science from National Technical University of Athens (NTUA) in 1999, where he is currently working on his Ph.D. Since 1999 he has collaborated with the telecommunications laboratory of NTUA. He has participated in several European and national research projects (ELISA, AQUILA). Areas of interest include QoS over IP, resource management, network management, and object-oriented software technologies. He has published a number of papers related to these fields.

IAKOVOS S. VENIERIS received a Dipl.-Ing. degree from the University of Patras in 1988 and a Ph.D. from NTUA in 1990, all in electrical and computer engineering. He is currently an associate professor at NTUA. His research interests include broadband communications, the Internet, mobile and intelligent networks, service creation and control, distributed processing, agents, and performance evaluation. He has over 100 publications including two internationally published books.

MARTIN WINTER studied physics at the University of Stuttgart and received his Master's degree in 1986. In 1986 he joined Siemens in Munich, working on software development for the product EWSD until 1994. Since 1995 he has worked on several internal and international projects in the Rapid Prototyping Department of Siemens Information and Communication Networks. Major areas of interest are IP QoS and modern software technologies.

From a technical point of view, a satisfactory status of QoS over IP has been reached by the efforts of AQUILA and other related projects. An important next step of research will be to investigate in detail the economics of QoS.

Copyright notice