Design of a Multi-layer Bandwidth Broker Architecture

George A. Politis, Petros Sampatakos, Dr. Iakovos S. Venieris

National Technical University of Athens
Greece

Abstract. Internet is widely known for lacking any kind of mechanism for the provisioning of Quality of Service guarantees. The Internet community concentrates its efforts on the Bandwidth Broker architecture towards this problem. This paper presents a design model of a multi-layer Bandwidth Broker architecture that introduces a Resource Control Layer, which is divided into two sub-layers. The upper one is responsible for the overall network administration, while the lower one performs per-flow policy-based admission control. The design models, the mechanisms, and algorithms adopted in this architecture will be delineated.

Keywords: Bandwidth Broker, Quality of Service, Integrated Services, Differentiated Services, Resource Control Point, Resource Control Agent, Application Middleware.

1. Introduction

Internet is the technology that has become part of our every-day life over the past years and gains significant momentum day by day. Although it started as an experiment [1,2], nowadays, it is a serious business and it aims to be the integrated infrastructure that will concentrate most or even all of the services, existing and future ones. However, the protocols and mechanisms of the current Internet technology seem to be insufficient for delivering the traffic of the arising and demanding multimedia applications with the appropriate Quality of Service (QoS) characteristics, and thus enhanced mechanisms have to be deployed to provide a QoS-enabled Internet infrastructure.

Despite the notion that many have adopted, QoS is not solved merely by increasing the capacity of the links, since there are always merging points in the network that inevitably lead to congestion situations. But first, it should be clarified what this term means; a working definition states: "*IP QoS enables a network to deliver a traffic flow end to end with the guaranteed maximum delay and guaranteed rate required by the user process, within agreed error boundaries*" [3]. In order to bring QoS into the network, three components have to be deployed: traffic handling, signalling, and provisioning and configuration [3,4,5]. The first refers to the classification of data packets into separate flows, the scheduling and the buffer management algorithms performed on each flow at the network devices. The second component allows the end-user to signal specific flow requirements and enables the end-to-end co-ordination of QoS between the network nodes. Finally, the third component decides which network device performs which specific traffic handling mechanism based on the policies of the network operator. Moreover, it refers to monitoring, measurement and traffic engineering mechanisms needed for evaluating the QoS guarantees, fixing overloaded links, measuring the characteristics of traffic.

There are several initiatives from the Internet community to resolve this problem, in principle the *Integrated Services* (IntServ) [6,7] and the *Differentiated Services* (DiffServ) [8,9] approach. The first approach, which uses explicit resource reservations, is considered as

S. Rao and K.I. Sletta (Eds.): INTERWORKING 2000, LNCS 1938, pp. 316-325, 2000.

rather difficult to scale up to a worldwide network, while the second one does not yet provide all required mechanisms for end-to-end QoS provisioning. Although they are two independent models where the DiffServ model was introduced as a rather simple and easily deployable model that came to replace the IntServ model and overcome the scalability issues that follow it, finally it is realised that they are not competitive but rather complementary [4,10,11].

The concept of the Bandwidth Broker (BB) that has been introduced from the early stages of the DiffServ model [9] is responsible for performing policy-based admission control, managing network resources, configuring specific network nodes, among others. Nowadays, the Internet community directs its efforts towards the specification and standardisation of the mechanisms of the BB, as well as the development of a prototype [12,13,14]. This paper presents the architecture and design decisions of a multi-layer BB, which is currently under development [15].

This paper is organised as follows: Section 2 gives a short presentation of existing approaches for QoS provisioning over IP. Section 3 discusses the basic concepts of the proposed architecture and specially the Resource Control Layer. Section 4 describes the design model of the Resource Control Point and its mechanisms. Finally, the work to be done in the future is delineated in Section 5, while a summary of the main topics of this paper are given in Section 6.

2. Existing Approaches

Both telecommunications industry and research community have spent a lot of effort on investigating and developing new technologies that could provide QoS over IP-based networks, during the last years. The first attempts focused on providing an automatic optimisation of IP traffic over switched-based networks, such as ATM (e.g. MPOA, IP switching). However, the disadvantage of those approaches is that the application software does not have an interface which can control the specific capabilities of the underlying network.

A different approach, coming from the Internet Engineering Task Force (IETF), is the IntServ architecture, which provides a starting point to establish the necessary infrastructure for advanced multimedia services on top of the IP protocol suite. Integrated Services architectures have been defined using protocols which are being implemented for IP routers (e.g. RSVP [16]). The basic concept of the IntServ model is the enhancement of the existing IP router with tasks traditionally executed in switch-based networks and thus giving Internet a connection-oriented character. Hence, operations like policing, shaping, admission control and QoS management must be provided by all of the RSVP routers on a per IP flow basis. However, in a large scale network with millions of connected users, the number of IP sessions handled by a core RSVP router can be very large. Therefore, the execution of the above functions for every active IP flow in a core IP router leads to pure performance and to a non-scalable network architecture. Furthermore, many important issues remain unsolved, in particular appropriate charging and admission control mechanisms in order to make an Integrated Services architecture economically viable.

The above considerations have forced the Internet community to define a new model for QoS provisioning over IP networks. The new model, known as DiffServ model defines a set of traffic classes each of which is designed to serve applications with similar QoS demands. A traffic class describes the Per Hop Behaviour (PHB) that the packets of this class should receive in each network node. The per hop behaviour determines the priority, the maximum delay in the transmission queues, the link-sharing bandwidth and the probability of a packet to be dropped. The DiffServ model ensures high scalability by separating the operations performed in the borders of the network from those accomplished in the core network. Border routers perform more complex tasks such as traffic policing and shaping, marking and prioritised routing. Marking is the process of classifying IP packets belonging to a specific IP flow and assigning them to the appropriate traffic class. All of the above operations are performed on per flow basis as in the IntServ model.

The small number of active IP flows handled by a border router does not cause the scalability problems that exist in the IntServ architecture. On the other hand, core routers carry out only one simple task that is prioritised routing. DiffServ core routers do not keep any information for the established IP flows. On the contrary, they simply serve packets according to the traffic class that the Ingress border router has assigned to. Hence, each DiffServ core router has to know only the number of traffic classes and the corresponding per hop behaviour of each class. However, in the DiffServ model the functions that would allow end users to request network resources in a dynamic manner are missing. In other words, the *signalling* interface between users and border routers is still not defined.

The framework, which takes advantage of both models and harmonises their different qualities, introduces a logical entity, the so-called Bandwidth Broker (BB) [9,12,13,14]. The main components of this architecture, as well as their functionality are depicted in Figure 1. It can be seen that this model attacks the problem of QoS provisioning in the three aspects mentioned above i.e. signalling, traffic handling and management.

In this new model, Internet is assumed to be separated into various Administrative Domains or Internet Service Providers (ISPs), where each core network is based on the DiffServ model forwarding the aggregate IP traffic based on the DiffServ Code Points (DSCPs) of the traffic flows. Nevertheless, this traffic should be policed, shaped and marked on a per-flow basis at the ingress points of a DiffServ domain, and this is performed by the Edge Devices (EDs). Moreover, since adjacent ISPs have contracts between them, the so-called Service Level Agreements (SLAs), that specify, among others, the traffic characteristics that one domain injects to the other, the egress points of a domain has the responsibility to shape the aggregate traffic sent downstream to the neighbouring domain, in order not to violate the contracts. This is the task of the Border Router (BR). Therefore, the routers are configured appropriately with traffic conditioning mechanisms, as well as scheduling and buffer management modules that specify the PHB of the router according to the network services offered by each domain.

Furthermore, the user may signal her/his QoS requirements to the BB of the domain via a mechanism that could be based on the IntServ/RSVP model or another mechanism such as COPS [17], CORBA [18] etc. The BB, which is responsible for monitoring and controlling the available bandwidth within the DiffServ domain, identifies the path of the new flow and checks whether there are enough available resources in the DiffServ routers that belong to this path. If the path goes beyond the set of DiffServ routers controlled by a specific BB, then the request is forwarded to the appropriate adjacent BB until the BB, which handles the destination ED is reached. Finally, if all of the DiffServ routers, which are involved in the data path of the user's request, have enough available resources, the request is accepted.

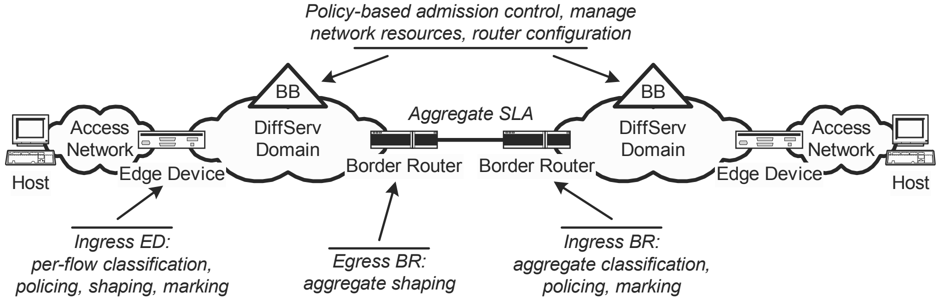

Fig. 1. Bandwidth Broker Principle Architecture

Moreover, all involved BBs update their resources database in order to reflect the new established flow. After the successful establishment of the flow, user's packets belonging to the specific IP flow are policed, shaped, and classified (according to the traffic profile sent initially by the user in the reservation message) as mentioned above.

3. Resource Control Point (RCP)

This section presents the main principles, components and mechanisms of the multi-layer BB architecture. Moreover, special attention is given to the structure and functionality of a specific logical entity, the Resource Control Point (RCP).

3.1 Architectural Principles

This architecture is highly related to the general BB architectural framework described in the previous section. In order to comprehend the decisions made for the design of this specific architecture, it would be helpful to give some fundamental assumptions. First, it is assumed that the data-plane consists of DiffServ-aware routers, while there is no intention in developing any new technology in this field. Therefore, this plane is used as it is and the focus is on the design of an overlay Resource Control Layer (RCL) that manages the resources of the underlying DiffServ data-plane. Second, this architecture is limited to the single-domain case, thus no inter-domain mechanisms are discussed; although most of the ideas presented apply to both cases.

Last but not least, there is a compromise that has to be accepted between provisioning hard QoS guarantees or simplifying the design by sacrifycing a percentage of the network resource utilisation. The resulting architecture is rather simplified, but of course, such a decision can only be taken by also considering the network services to be offered. Therefore, the target network services for this model address to applications that exhibit a “light” DiffServ behaviour:

ñ Delay and jitter sensitivity with small IP packet lenghts e.g. Voice over IP.
ñ Delay sensitivity (looser than the above) and high-bandwidth requirements e.g. videoconferencing.
ñ Packet loss sensitivity, security requirements and low duration sessions e.g. SAP.
ñ No need for guarantees i.e. best-effort.

3.2 Resource Control Layer (RCL)

The Resource Control Layer (RCL) is separated into three logical entities that have been assigned distinct tasks. First, the *Resource Control Point* (RCP) is responsible for managing and distributing the network resources to the corresponding elements. The initial values of the network resources come from the network administrator during the start-up configuration. Second, the Resource Control Agents (RCAs) are assigned the task of performing policy-based admission control so that each reservation request is accepted after ensuring that the customer has administrative rights and there are sufficient resources in the network. In order to perform admission control, the RCAs are assigned by the RCP an amount of resources for which they are responsible. Moreover, each RCA controls an Edge Device or a Border Router, configures the traffic conditioning parameters, allocates the bandwidth resources received by the RCP and handles the users reservation requests. Third, the Application Middleware (AMW) provides an interface to the end-user applications that enables the end-user to signal her/his requirements to the QoS infrastructure. Figure 2 depicts the logical components of the architecture, their associations, as well as the relation to the underlying data-plane.

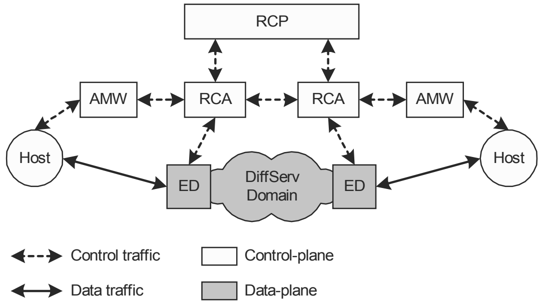

Fig. 2: Resource Control Layer Architecture

Each RCA is associated to a single ED and handles the reservation requests that come from the hosts attached to the ED. The RCA performs local admission control checking whether the local "area" is capable to handle the new traffic flow. In order to make this decision independent of any central entity (e.g. the RCP), the RCP should allocate to the RCA a resource share that would represent the resources of the nearby network. Therefore, the load from the signalling processing of a reservation request is distributed to the RCAs, while the RCP redistributes the resources among the RCAs whenever one or more of them runs out of them.

Depending on the nature of the requested network service and the degree of its guarantees, the RCL may allocate resources either at the ingress RCA or the egress one or both. In case that both RCAs make the reservation, then the ingress RCA should be able to locate the egress one and forward the reservation request. The mechanism of RCA identification is the task of another entity, not shown in the model, which is responsible for the mapping of a host IP address to the corresponding address of the RCA.

Although the admission control is restricted to the edges of the network (ingress, egress or both), this model promises to provide QoS guarantees. The key is to apply an efficient mechanism to the RCP, so that the resources distributed to the RCAs reflects the QoS traffic that can be handled by the core network without violating the requested QoS guarantees.

3.3 Hierarchical Structure of the RCP

The functionality of an RCP includes the start-up configuration of the network, the distribution of the resources to the RCAs and the reconfiguration of the available resources according to the variations of the traffic load. In order to reduce the interactions between the RCAs and the RCP and at the same time, provide an efficient resource management entity, the RCP is structured in a hierarchy of RCPs, as illustrated in Figure 3.

Each RCP has the responsibility of its "children" RCPs. Initially, following a top-down approach, the available bandwidth, which is primarily determined by the capacity of the backbone network, is distributed from each RCP to their children according to the initial amounts defined by the network administrator. In addition, after the start up process of the network and the initialisation of the resources, the RCPs are the managers of the resources that have been assigned to them. In other words, the distribution of the resources is rather a dynamic procedure than a static configuration. Thus, in case that the resources that have been assigned to an RCP are insufficient, the RCP will request more from his "parent" RCP, in order to take advantage of any unused resources. The request can be propagated to the upper levels if necessary. In this way the redistribution of the resources takes place as closest to the requester of the resources as possible and the process load is reduced.

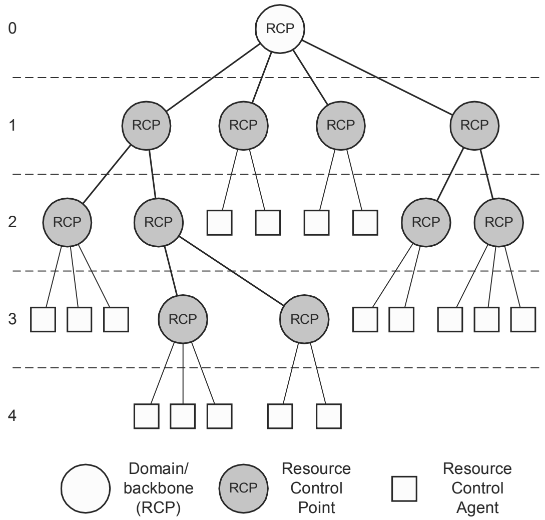

Fig. 3: Hierarchy of the RCPs

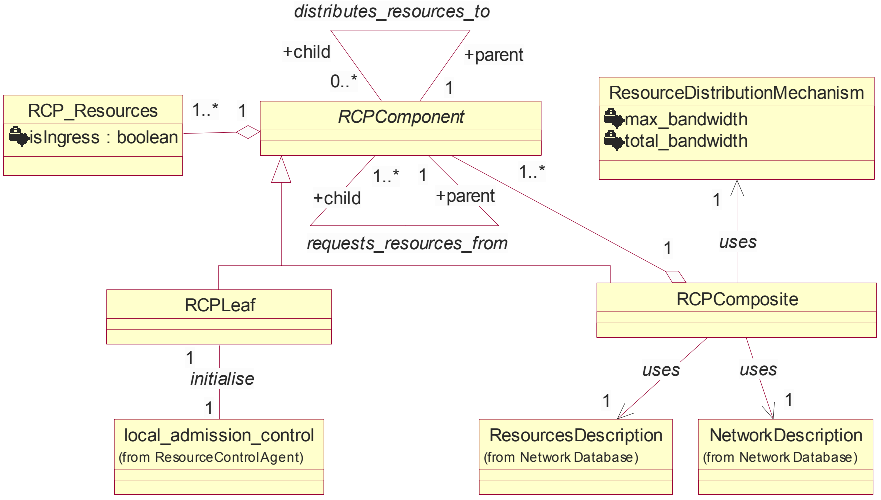

Fig. 4: Resource Control Point Coarse Design Model

The structure of the RCPs should reflect the structure of the underlying network. Therefore, it is required to take into account a number of variables in order to manage to make the best possible mapping. The more accurate this mapping the more efficient the hierarchy will become. Knowledge about network topology and routing as well as information about the expected SLAs of the customers should be helpful in solving this problem.

There are some basic principles that can be used as guidelines for the formation of the hierarchy. Firstly, the RCP should represent a set of physical links that are topologically related. They can represent the links of a sub-area or sub-network, e.g. the network of a university laboratory. When two or more sub-areas are connected to the same router then a new RCP could be formed including the two RCPs that represented those sub-areas. As it is obvious, in a network that uses a fully meshed (or nearly fully meshed) topology the concept of the hierarchy could not be applied. Furthermore, it is not allowed to an RCP to include a link that is already member of another RCP of the same level; the level of an RCP should be taken into account because the parent RCP will always include links that are already members of its children RCPs. Also, the routing information could be an additional input for the case. The sub-areas of the same level of hierarchy should not be directly linked. The local traffic for each sub-area should not use links that are members of another RCP, otherwise this will result in a leak of resources. The above guidelines are not intending to produce an optimal mapping between the network topology and the hierarchical structure of the RCPs, but provide a relatively easy guide for the network administrator.

4. Design of the RCP

4.1 Coarse Design Model

The coarse design model of the RCP is depicted on figure 4 using the Unified Modelling Language (UML) notation [19].

In order to depict the tree structure of the resource control points the composite pattern is used [20], where the following classes are defined: *RCPComponent*, *RCPComposite* and *RCPLeaf*.

The tree of the resource control points is created using the information retrieved by the network database concerning the *NetworkDescription* and *ResourcesDescription*. This information is used to create the resource control points and assign initial resources to each of them. The network database is managed by the network administrator.

Each RCP has a set of objects *RCPResources* that represent the resources that are assigned to each traffic class provided by the network. Each RCP has 5 values that are related to the resources that it possess. The *max_bandwidth* represents the maximum amount of bandwidth that could ever be assigned to the RCP. This value is restricted principally by the capacity of the physical link. The *total_bandwidth* value is the amount of bandwidth that is actually assigned to the RCP for distribution among its children RCPs, initially is the same as the initial amount of resources. The *spent_bandwidth* value apparently depicts the amount of bandwidth that is already distributed to the child RCPs or in case of an *RCPLeaf* the amount of bandwidth spent for the reservation requests of the corresponding RCA. Obviously the subtraction *total_bandwidth – spent_bandwidth* represents the available bandwidth of each RCP.

4.2 Resource (Re-) Distribution

A static approach of the resource distribution mechanism will eventually result in bottlenecks and in unfair and inefficient management of the network resources. Thus, an adaptive mechanism should be defined that will adjust the distribution of resources accordingly to the demand. For this purpose the watermarks mechanism is introduced. Two watermarks are defined, one low and one high watermark for each RCP and RCA.

When a new reservation request has been received and the already spent bandwidth plus the bandwidth required by the new request exceed the high watermark, a request for more resources is made to the parent RCP. The amount of resources that will be redistributed to the requester is decided by the parent and depends on the amount of the available bandwidth of the parent RCP and the amount of the original request. There are two obvious solutions, the parent could distribute exactly the amount that has been requested or can give all the available bandwidth. To find the optimal solution is not an easy task and also depends on the network topology. The algorithm should minimise the communication overhead but should also be able to distribute the resources without permitting greedy components to dominate. A first approach of such an algorithm is described in pseudocode.

```
if (3 * req • 25% of the available resources)

give 3*req;

elseif (2 * req • 25% of the available resources)

give 2*req;

elseif (req • available bandwidth)

give req;

else (request_resources from parent);
```

The algorithm described could be adaptive by altering the factor that will be multiplied with the *req* value (requested bandwidth) or the percentage of the available bandwidth according to the network needs. The concept is to distribute more resources than requested in order to avoid a frequent communication with the specific child, but to save also enough resources for the other children.

The low watermark is an indication that there are unused resources which should be released in order to be used be other children. When the low watermark is crossed the child calls the *release_resources* of his parent. The amount of resources that is going to be released is determined by the high watermark, e.g. in an RCP that has 1Mbps of total bandwidth, a high watermark at 800Kbps, a low watermark at 200Kbps and the spent bandwidth is 128Kbps there should be released 200Kbps so the new total bandwidth is 800Kbps. Obviously the watermarks have to be reconfigured based upon the new total bandwidth value, so the new high watermark is (800*80%) = 640Kbps and the low watermark (800*20%) = 160Kbps.

4.3 Software and Hardware Platform

Since, the functions of the particular RCL comprises actions on various platforms of an IP network and the hosts connected to this network, it is reasonable to use a platform-independent system. Thus, the RCL is implemented as a distributed software system, where the OMG's Common Object Request Broker Architecture (CORBA) is applied. Therefore, the interfaces between distributed components are described using OMG's Interface Description Language (IDL), and that an object request broker (ORB) is applied to allow communication of the distributed components. As the system is implemented using Java (SDK 1.2.2), the IDL-to-Java-compiler as well as the ORB included in this SDK is used.

5. Future Work

The architecture presented in this paper is restricted to the case with one ISP and thus the inter-domain mechanisms between adjacent RCPs and adjacent RCAs and how they provide QoS guarantees are not investigated, yet. However, the algorithms described in the single-domain scenario should be examined and tested that the QoS guarantees they offer, at least justify the simplicity of the model.

Moreover, the RCL should be enhanced with some additional mechanisms necessary for the provisioning of hard guarantees requested by advanced multimedia applications. Such mechanisms include taking advantage of the routing information, monitoring the core DiffServ domain and developing a measurement platform that enable the RCL estimate and foresee the traffic loads, and therefore, take more effective decisions. Last but not least, a powerful technology that will be investigated in the context of this work, is the Multiprotocol Label Switching [21,22].

6. Conclusions

The overall architecture presented in this paper addresses the problem of QoS provisioning in IP networks, in a complete and consistent manner. It introduces a multi-layer Resource Control Layer that is responsible for the handling of the reservation requests, performing policy-based admission control, provisioning and configuring the network in a top-down approach, managing the network resources and dynamically redistributing them among the network elements.

This paper presents the specific design model of the Resource Control Point which is structured in a hierarchical manner in order to manage effectively the network resources and control the Resource Control Agents which are distributed at the edges of the network. Two field trials are expected to take place in the near future (for the intra- and inter-domain, correspondingly) that will provide useful information about the efficiency and the level of QoS provisioning this model can offer.

Acknowledgements

This paper is partly funded by the European research project AQUILA, Information Societies Technology (IST) programme, IST-1999-10077. The authors are currently engaged in the implementation of the concept presented in this paper.

Appendix

AMW	Application Middleware
ATM	Asynchronous Transfer Mode
BB	Bandwidth Broker
BR	Border Router
COPS	Common Open Policy Service Protocol
CORBA	Common Object Request Broker Architecture
DiffServ	Differentiated Services
DSCP	DiffServ Code Point
ED	Edge Device
IDL	Interface Description Language
IntServ	Integrated Services

IP	Internet Protocol
ISP	Internet Service Provider
MPOA	Multiprotocol over ATM
PHB	Per-Hop Behaviour
QoS	Quality of Service
RCA	Resource Control Agent
RCL	Resource Control Layer
RCP	Resource Control Point
RSVP	Resource Reservation Protocol
SLA	Service Level Agreement
UML	Unified Modelling Language

References

1. Krol, E.; Hoffman, E.: FYI on "What is the Internet?". RFC 1462, May 1993.
2. McQuillan, J.; Walden, D.: The ARPA Network Design Decisions. Computer Networks, 1, pp. 243-289, 1977.
3. Roberts, Lawrence.: Quality IP. April 21, 1999. http://www.data.com/issue/990421/roberts.html
4. Bernet, Y.: The Complementary Roles of RSVP and Differentiated Services in the Full-Service QoS Network. IEEE Commun. Mag., Vol. 38, No. 2, Feb. 2000.
5. Cisco: Introduction: Quality of Service Overview. http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/12cgcr/qos_c/qcintro.htm
6. Clark, D.; Shenker, S.; Zhang, L.: Supporting real-time applications in an integrated services packet network: architecture and mechanism. Proc. ACM SIGCOMM, pp. 14-26, August 1992.
7. Braden, R.; Clark, D.; Shenker, S.: Integrated services in the Internet architecture: an overview. RFC 1633, 1994.
8. Black, D.; Blake, S.; Carlson, M.; Davies, E.; Wang, Z.; Weiss, W.: An Architecture for Differentiated Services. RFC 2475, 1998.
9. Nichols, K.; Jacobson, V.; Zhang, L.: A Two-bit Differentiated Services Architecture for the Internet. RFC 2638, 1999.
10. Eichler, G.; Hussmann, H.; Mamais, G.; Venieris, I.; Prehofer, C.; Salsano, S.: Implementing Integrated and Differntiated Services for the Internet with ATM Networks: A Practical Approach. IEEE Communications, January 2000.
11. Bernet, Y.; Yavatkar, R.; Ford, P.; Baker, F.; Zhang, L.: A Framework for End-to-End QoS Combining RSVP/Intserv and Differentiated Services. Internet Draft, IETF, 1998.
12. Neilson, R.; Wheeler, J.; Reichmeyer, F.; Hares, S.: A Discussion of Bandwidth Broker Requirements for Internet2 Qbone Deployment. Internet2 Qbone BB advisory Council, August 1999.
13. Report from the First Internet2 Joint Applications/ Engineering QoS Workshop. http://www.internet2.edu/qos/may98Workshop/9805-Proceedings.pdf, May 1998.
14. Terzis, A.; Ogawa, J.; Tsui, S.; Wang, L.; Zhang, L.: A Prototype Implementation of the Two-Tier Architecture for Differentiated Services. RTAS99 Vancouver, Canada, 1999.
15. Adaptive Resource Control for QoS Using an IP-based Layered Architecture, AQUILA. http://www-st.inf.tu-dresden.de/aquila/
16. Braden, R.; Zhang, L.; Berson, S.; Herzog, S.; Jamin, S.: Resource ReSerVation Protocol (RSVP) - Version 1 Functional Specification. RFC 2205, 1997.
17. Boyle, J.; Cohen, R.; Durham, D.; Herzog, S.; Rajan R.; Sastry, A.: The COPS (Common Open Policy Service) Protocol. Internet Draft, IETF, 1999.
18. CORBA/IIOP 2.3.1. Specification. http://www.omg.org/corba/cichpter.html
19. OMG Unified Modelling Language Specification, version 1.3. 1999.
20. Gamma, E.; Helm, R.; Johnson, R.; Vlissidis, J.: Design Patterns – Elements of Reusable Object-Oriented Software. Addison-Wesley, 1995.
21. Rosen, E.; Viswanathan, A.; Callon, R.: Multiprotocol Label Switching Architecture. Internet Draft, IETF, 1998.
22. Awduche, D.: MPLS and Traffic Engineering in IP Networks. IEEE Commun. Mag., Vol. 37, No. 12, Dec. 1999.