Chinese Journal of Electronics
Vol.10, No.1, Jan. 2001

# PRM: A Resource Management Framework for Policy-driven QoS Control in Enhanced Internets*

FU Xiaoming and ZHANG Yaoxue

*(Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China)*

Abstract — **The Internet has evolved from a traditional best-effort delivery data network into an enhanced Internet that can provide a certain Quality of Service(QoS) mechanisms for applications. However, these QoS mechanisms usually do not take policy factors into account such as priority and time-of-day. The paper presents a Resource Management Framework for Policy-driven QoS control (PRM) in enhanced internets, which consists of four components: Domain Policy Controller(DPC) resides an administrator domain, Border Resource Manager(BRM) resides between adjacent domains, Interior Resource Managers(IRM) in each router between two BRMs within the same domain, and Endsystem Resource Manager(ERM) in charge of end-node/host router resources. We first introduce challenges and demands facing the enhanced internets, then give a formalized definition of policy and describe the functions and interaction of components of PRM. Specifically we take QoS-pricing policy used in a Diff-Serv environment as an example to illustrate the operational phases of PRM.**

Key words — **Quality of service (QoS), Resource reservation protocol (RSVP), Policy, Resource management, QoS pricing.**

## I. Introduction

The emerging end-to-end Quality-of-Service (QoS) requirements of the Internet applications and the distributed aspect of resources call for a new methodology to attain effective and automated resource management. To meet the increasing QoS requirements of distributed multimedia applications, IETF presents an "Integrated Service (IntServ)" model[2], which can provide each flow a certain QoS, e.g., Guaranteed Service (GS)[17] and Controlled-Load Service (CLS)[19]. However, due to its high overhead of storage and maintenance of status information and computation for each flow, as well as requiring configuration for complicated signaling protocols, e.g., Resource Reservation Protocol (RSVP)[3], admission control, packet classifying and scheduling functions, its scalability becomes a big problem. To overcome these shortcomings, researchers are presenting a simpler, more scalable model called "Differentiated Service (DiffServ)"[1], where packets are marked as a small number of aggregate traffic class. In DiffServ model, to support aggregate QoS for different class services, it only needs providing data packets of different service type with different Per Hop Behavior (PHB), such as Assured Forwarding (AF)[9], Expedited Forwarding (EF)[10]. These networks are named "enhanced internets", or QoS-capable networks. From the point of view of users, they often wishes to obtain a certain degree of service even during period of network congestion, and if necessary, dynamically adjust desired degree of service during congestion.

To attain the goal of QoS control in such an enhanced internet, a critical problem is to present a suitable resource management model which can allocate resources and dynamically manage these resources accordingly. However, most resource management schemes employ a capacity-based approach, i.e., allocating resources according to capacity demands of flows, not discriminating between different flows. Once the network resources are exhausted, new requests can not be admitted and only be able to obtain service until other existing flows release their consumed resources. Thus users can always be not serviced according to some special requirements from themselves or network administrators. In fact, QoS mechanisms can be completely realized only when combined with some set of policies. For instance, reserving resources merely according to all users' capacity requests will lead to the resource utilization becoming low, since most users are inclined to apply for best services, which will result in a low resource utilization and lead to a way of best-effort again. Moreover, the complexity caused by

*Manuscript Received Jan. 2000; Accepted Aug. 2000. This work is supported by the National Natural Science Foundation of China.

many factors, e.g., security, time-of-day and QoS pricing, appeals for policy control for network and system resources. In Ref.[23] Gai, *et al.* provide a framework for QoS traffic management but it is mostly concerned with operation in a single administrative domain, i.e., inside a Policy Domain.

The objective of this paper is to present a policy-driven resource management framework that is capable of providing QoS control for enhanced internets that may consist of a single or multiple administrative domain(s). After an analysis of requirements of resource management for enhanced internets (Section II), we give a formalized definition of policy and propose Resource Management Framework for Policy-driven QoS Control (PRM) in Section III. Section IV takes the example of QoS-pricing policy usage to illustrate the operational phases of PRM. Section V concludes this paper and introduces our future work.

## II. Requirements of Resource Management for Enhanced Internets

For a network that can only provide a simple best-effort service, it is enough to employ a First-in-first-out single queueing mechanism to attain the goal of resource management. However, it is far insufficient for an enhanced Internet with diverse QoS objects, e.g., GS defined in IntServ requires a stringent delay bound (quantitatively) while AF group of DiffServ assigns one of three levels of drop precedence (qualitatively). Specifically, an enhanced internet must satisfy some key requirements:

- Scalable resource management. It should support the needs of administrative domains which have different traffic classifying and treatment policies.
- Flexible QoS support. Different applications may have diverse QoS requirements. For example, some applications desire a low transmission delay, while for some applications, a rather low loss possibility is more desirable.
- Optimizing resource utilization. Resource should be allocated and dynamically managed reasonably to drive competing users/applications into a more efficient way of utilizing resources and satisfy more users/applications (achieve a high throughput).

Policies that define the desired behavior of resources have been recognized as a concept to support this complex management task by specifying means that enable to enforce this behavior. Since the agreements between policies of adjacent domains are to be coordinated separately in accordance with business strategies, they are beyond the scope of this paper and we emphasize the task of resource management in a specific domain, assuming traffic agreements are set beforehand (manually) by domain administrators. In the next section we will give a formalized definition of policy and propose a Resource Management Framework for Policy-driven QoS Control (PRM).

## III Resource Management Framework for Policy-driven QoS Control (PRM)

### 1. Policy

In this paper, we define policy as "the set of desired rules for behaviors of distributed applications, system and network resources within a specific administrative domain". Security, time-of-day, QoS mechanism and preempt priority treatment are examples of policies, but we only consider the aspects related to QoS control while we refer to policy in this paper. This is a scalable definition which may be used in DiffServ and IntServ models. As the definition implicates that policy is composed of policy rules, each one of the rules can be divided into policy condition and policy action. We can formalize "policy" using Backus-Naur Form (BNF) notation as follows:

⟨Policy⟩::=⟨PolicyRule⟩|⟨Policy⟩
⟨PolicyRule⟩::=IF ⟨PolicyCondition⟩ THEN ⟨PolicyAction⟩
⟨PolicyCondition⟩::=⟨EndSystemCondition⟩|
  ⟨BorderRouterCondition⟩|
  ⟨InteriorRouterCondition⟩
⟨PolicyAction⟩::=⟨IntServAction⟩|⟨DiffServAction⟩
⟨IntServAction⟩::=[⟨Set⟩|⟨TrafficShape⟩]
  ⟨FlowQoSAllocation⟩⟨FwdAction⟩
⟨DiffServAction⟩::=⟨PoliceAction⟩|[⟨Allow⟩|⟨Forbid⟩]
⟨TrafficConditionerAction⟩
⟨DSResourceAllocation⟩ ⟨FwdAction⟩
⟨TrafficConditionerAction⟩::=⟨Shape⟩⟨Mark⟩⟨Drop⟩
  ⟨Meter⟩⟨Police⟩
⟨DSResourceAllocation⟩::= [⟨Set⟩|⟨Monitor⟩]
  ⟨DSAggregateBandwidthAllocation⟩|
  ⟨DSAggregateBufferAllocation⟩
⟨FwdAction⟩::=⟨Drop⟩|⟨Delay⟩|⟨Forward⟩|⟨Remark⟩

where, ⟨EndSystemCondition⟩ represents policy conditions employed to/by applications, hosts, or users, which are responsible for enforcing policy actions on applications directly. ⟨FlowQoSAllocation⟩ is to reserve QoS resources for flows such as bandwidth and delay. ⟨TrafficConditionerAction⟩ is to configure parameters and functions regarding traffic conditioner.

The issue of "pricing for QoS" has been investigated in QoS research community since the early 1990s[6,16]. By "pricing for QoS" we means each user/application should pay for achieving QoS by way of consuming system and network resources. However, this work has been largely limited to optimal pricing computation and auction method, leaving the gap between QoS pricing and a viable resource management scheme unpadded. Without losing generalization, we consider a QoS-pricing policy in this context. Meanwhile, since RSVP is a suitable signaling protocol for resource management and has been widely used in industry, we take it as the fundamental of our resource management framework.

We define QoS-pricing policy as follows:

⟨QoSPricingPolicy⟩::=⟨QoSPricingPolicy⟩|
  IF [⟨QoSClass=class *i*⟩⟨userBid⟩
  =Thredshold *i*⟩]
THEN [⟨ResrcAllocate *i*⟩|⟨DecLowPreferResrc⟩]

which means if a user/application of QoS class $i$'s preference is not below a certain thredshold for class $i$, then either allocate system and network resource to it or decrease some reources consumption of low preference user/application(s).

The advantages of employing a policy-driven resource management framework for QoS control are as follows:

• The Internet is a large internetwork combining multiple autonomy systems(ASs), each of which is somewhat an administrator domain and has its own manage policy, routing protocol and agreements about traffic delivery and accounting with adjacent domains. A distributed policy-based resource management is a good solution to attain QoS control.

• Policies are an active concept and capable of providing flexible QoS support. For example, a QoS-pricing policy may initiate, change or even tear down some characteristics of ongoing services.

• Policy needs participation from applications, systems and networks. This makes up an integral way for a specific domain to manage its resources effectively. We assume applications have their some differentiated price preferences (or bids) under different degrees of congestion. Through a reasonably designed scheme, resources will be utilized in a more efficient way.

## 2. Components of PRM

In order to implement QoS control, policies must be able to act on their targets, i.e. resources, hence they have to be transformed into functional components in network elements. Our proposed Resource Management Framework for Policy-driven QoS control (PRM) consists of the following functional components: Domain Policy Controller (DPC), Border Resource Manager (BRM), Interior Resource Manager (IRM), and End-system Resource Manager (ERM). The PRM structure is shown in Fig.1.

(1) Domain policy controller

Domain policy controller (DPC) is a logical entity through which policy information of the domain is changed and policy decisions are made. It is further subdivided into four modules: the PolicyMgmt module performs indexing, setting, updating, starting/closing policy rules; the PolicyDecidePnt module performs policy retrieval and reasoning to acquire decisions of policy actions, which are passed to BRM to be enforced; the PolicyStorage module is the place where policy data are stored, and the StatusTable module reposits information about resource usage. The latter two modules can be accessed by another two modules via standard LDAP or SQL operations. The logical structure of a DPC is shown in Fig.2.

(2) Border resource manager

Border Resource Manager (BRM) resides in the border of a domain. It is connected inward to some Interior resource manager(s) and outward to an adjacent BRM or an Endsystem Resource Manager. A BRM consists of three modules: RSVP module to manage individual (IntServ) or aggregate flows (DiffServ) reservations, Resource Agent (RA) to perform TrafficProfile (TSpec for IntServ, TCS and SLS for DiffServ) management, monitoring resource usage in border routers, reporting and checking for policy action to DPC. When a new request comes and causes resource conflict, i.e., resource exhausted so that unable to admit the new request, it should also report this to DPC, trying to resolve the conflict by policy. BRM and its location in a border router are shown in Fig.3.

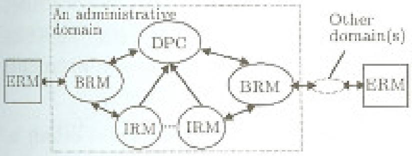



Fig. 1. Components of PRM. DPC: Domain policy controller; ERM: Endsyst resource manager; IRM: Interior resource manager; BRM: Border resource manager

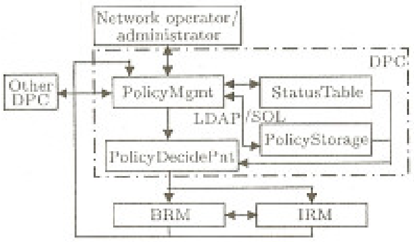



Fig. 2. DPC structure

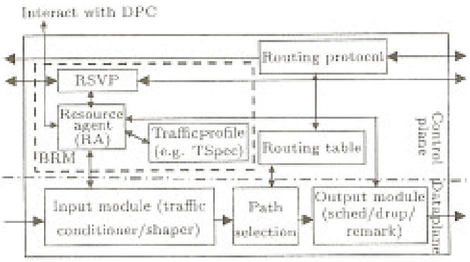



Fig. 3. BRM and its location in a border router

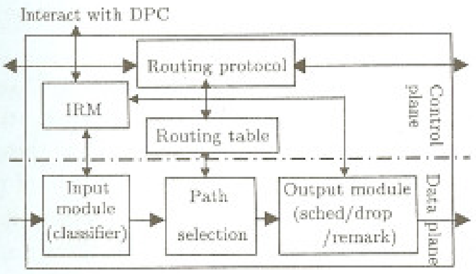



Fig. 4. IRM and its location in an interior router

(3) Interior resource manager

Interior resource managers (IRM) resides in each router between two BRMs within the same domain. An IRM is somewhat like the RA module in BRM and it ignores the signaling module and TrafficProfile module since it should only be responsible for monitoring the local interior router resources. Also, when it encounters network failure, IRM will report this to DPC in the same domain and may enforce some action, depending on the feedback DPC decision. This is in accordance with the direction of "keep the core simple and put the complexity in the border" of an effective QoS network design. See Fig.4 for the structure of IRM and its location in an interior router.

(4) Endsystem resource manager

Endsystem resource manager (ERM) resides on each end node (or host). It provides the front end of users to inform them about the incoming QoS requests, hence it should interact with the nearby DPC via a BRM to decide if the network can acknowledge or reject the request. Fig.5 shows ERM and its location in a PRM end-node/system. ERM provides policy-driven both resource allocation and adapting represents for applications. Various events like failures, queue sizes exceed a thredshold etc. from modules in data plane interacts with applications and other resource managers. The data plane includes representatives (DataREPRs) which provide data manipulation functions such as transcoding, mixing, etc. The node also has a CPU scheduler which provides computational QoS to representatives. The flow classifier is responsible for delivering incoming data flows for processing.

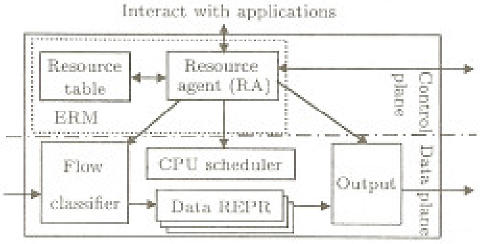


Fig. 5. ERM in an end-node/system

## IV. PRM Operations and an Example

### 1. Overall operations under PRM

**Two phases of PRM operations** In brief, the operations of PRM can be distinguished into two phases: a QoS negotiation phase and a data transmission phase.

In the QoS negotiation phase a sender application would specify its traffic characteristics via a RSVP Path message. The Path message is forward hop-by-hop toward the destination, i.e., following a path like sender→ERM→{BRM→{IRM}→BRM}→ERM→receiver, where { } suggests the route may include one or more tandem units. After receiving this message, the receiver would specify QoS requirements (e.g., IntServ parameters such as bandwidth, delay) following the reverse path toward the sender. ERMs and BRMs use these parameters for capacity test and QoS computation which finally results either in resource reservation or in rejection of the reservation attempt if the QoS cannot be met due to a lack of local resources. Note that IRMs are not involved in this procedure and only forward the RSVP signaling messages to their next hop, since we put this task to the BRMs in each domain. If these tests succeed and the QoS can be met, required resource capacities will be reserved.

After the negotiation phase has been successfully completed, in the data transmission phase, each unit of traffic allocation (eg., flows in IntServ and traffic classes in DiffServ) is monitored with respect to the traffic profile setup bilaterally between the IRM or BRM itself and its adjacent resource management component. This is achieved by (re)marking or discarding out-of-profile traffic. Moreover the resources used to process the user data are scheduled with respect to the reserved resources (also called "QoS enforcement") in the data transmission phase.

**The role of DPCs and their interaction with BRM and IRMs** DPCs play an important role in PRM and they are responsible for the following functionality:

- Collecting periodical resource usage information and accidental resource failure information from BRM and IRM;
- Dealing with policy rule configuration and query command initiated by network operators or administrators;
- Making (or simply recording) agreements between itself and adjacent DPCs.
- Maintaining the consistence between BRMs and IRM's resource configuration procedure.

In addition, BRMs will always forward RSVP Resv message from ERMs to DPC, which determines whether to initiate a new resource allocation or appropriate resource reconfiguration of the old applications and forces them to release the resources needed to admit the new application via policy (and if so, DPC would revise its StatusTable). Reconfigurating the old applications may be achieved by one or both of the following two means: 1) stop data transmission for existing applications; and 2) decreasing the QoS of existing applications. Therefore a setup phase for a certain application may cause other applications of the same class to stop their QoS enforcement due to lack of resources, a consequent requirement of which is for BRM to modify its TrafficProfile.

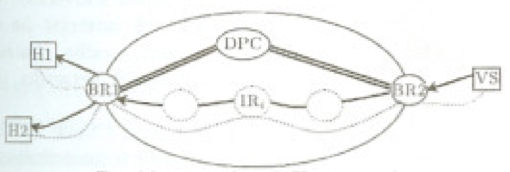
**Other issues when employing policies** When a new PolicyAction is decided to be taken, the problem of conflict detection and resolution should be resolved. However, due to the variety of mechanisms and degrees of sophistication to support this functionality, we do not specify a generic resolution and take it as a local decision for each domain. For instance, a rule priority maybe used in conflict resolution.

As described before, in PRM, network and system resources are constrained by QoS policies such as explicit priority, time-of-day, and QoS pricing. Thus these policy requirements are the driven force for the task of resource management. In a general procedure, firstly it should take the policy requirement into account, then examine related resource requirements and QoS satisfiability. When a BRM or IRM encounters failures or conflicts, it should report them to DPC, which then checks the policy repository and determine policy rules for related BRM or IRM.

**2. An example: treatment of QoS pricing policy**

In order to satisfy their QoS needs, users of applications (such as multimedia, transaction processing and database) compete for resources from servers of the information system, and since most users are inclined to be greedy, users will always request the best possible quality, specifying a large resource demand to the network, thus leading to an ineffective utilization of resources. Hence there's a need for QoS pricing policy. As for applications, some are non-adaptive and prefer to deliver their services to their users in an all-or-nothing fashion. On the other hand, adaptive applications such as layered video stream applications are apt to change their QoS (including sending rate) adaptively in different network environments. QoS pricing policy may be applied in PRM to overcome this problem. Refs.[6, 8, 11, 15, 18] proposed various approaches of QoS pricing policy but we do not put our emphases on them in this paper due to lack of space. Instead we take a specific example of adaptive 3-layered video applications (similar to Ref.[13]) to illustrate the PRM management operation under such a simple QoS pricing policy: to service the high bid first for the same class of service.

Consider a network setting shown in Fig.6 where transition network is a DiffServ network with border routers BR1, BR2, some interior routers IR1, $\cdots$ IR$i$, $\cdots$, and an administration station running DPC functions. Hosts H1 and H2 connected to BR1 are clients requesting video playback from a video server VS, which is connected to BR2. Suppose the maximal available bandwidth for the traffic class "3-LayeredVideoTransfer" between BR1 and BR2 is 2Mbits/s (predefined by the DPC and keep the same during the time interval regarding this example).




Transition network: a DiffServ network

Fig. 6. An example of PRM management. (== Policy message path; ······ RSVP signal path; —— Data path)

Suppose the 3 QoS operating points for the layered video are 1.2 Mbits/s for full quality video containing all three layers, 0.9 Mbits/s for medium quality video containing 2 layers, and 0.6 Mbits/s for low quality video with just the base layer.

Users from H1 and H2 can request three differentiated quality playback of layered video from VS, correspondingly bidding with three qualitative levels: high, middle and low. Following the definition in Section III, we define QoS pricing policy of this case conceptually as follows:

Policy example: QoS-pricing policy for 3-LayeredVideoTransfer

PocilyRule-1: IF "bid=high" THEN "rate=1.2Mbps" /*"QoSOperationPoint='playback layer 1, 2 and 3"'*/

PocilyRule-2: IF "bid=middle" THEN and "rate>=0.9Mbps"/*"QoSOperationPoint='playback at least layer 1, 2"'*/

PocilyRule-3: IF "bid=low" THEN "QoSOperationPoint='playback at least layer 1"' and "rate)=0.6Mbps" /*"QoSOperationPoint='playback at least layer 1"'*/

Our assumed scenario is as follows: H1 first requests a "bid=low" 3-layered video stream from VS; some time later, H2 starts a "bid=high" application that requires streaming of full quality video from VS. Fig.6 shows the sequence of event treatment in PRM, described as below:

(1) H1 requests a video from VS. The video server sends a RSVP Path message to H1 containing application information (including the identification of class of the layered video, and the price H1 bid for its transmission).

(2) H1 reserves 1.2Mbits/s over the path between H1 and VS using RSVP Resv message. It would succeed since the available resource is enough. Then a TrafficConditioner is established in BR2 and BR2 sends a message to DPC describing the 3 operating points of the video and its bid, along with the source and destination addresses of the video stream. The DPC stores this information in StatusTable. The video server starts transmitting the video stream to H1. BR2 monitors the stream using the TrafficConditioner parameters.

(3) At some later time, H2 requests a same video from the video server with a high bid.

(4) The video server sends H2 a Path message containing the operating points of the requested video stream.

(5) H2 sends a RSVP Resv message to the network. Router BR2's QoS negotiation algorithm detects that H2's request cannot be admitted. Upon that R2 denies H2's reservation and generates a PolicySignal message addressed to DPC through RSVP. The PolicySignal message carries information about the parameters of the rejected reservation request.

(6) DPC queries in its policy storage and finds that the rejected reservation has a higher bid than the existing reservation (that was made by H1). It also determines that enough bandwidth for the new flow can be accommodated by shutting down layers 2 and 3 of the low bid video stream destined for H1.

(7) It sends a PolicyAction message to the ERM on H1 directing it to shut down layers 2 and 3 of the video received by H1. When the application on H1 receives the PolicyAction, it sends a RSVP Teardown message to the network to shut down data flows associated with layers 2 and 3 of the 3-layered video. It also sends a PolicySignal message to DPC to report its new status. The available bandwidth on the link BR1-BR2 is now 0.4 Mbits/s.

(8) When receiving the PolicySignal from H1, DPC tells BR2 to reserve resources for H2. Finally the reservation process for H2 succeeds and the video session is established.

## V. Conclusions and Related Work

There is an increasing need to develop a scalable framework for QoS control that will enable interoperability among multiple provider networks (administrator domains) that must work together to achieve effective QoS control and automated resource management. This paper attempts to present an appropriate resource management framework, PRM, toward this goal. Although this paper takes QoS pricing policy in DiffServ network as an example to illustrate the operation of our proposed framework, PRM, we intend to use any policy/policy set for PRM implementation and management with DiffServ, IntServ or the mixture of them in general. PRM has the following advantages over other schemes as far as we know:

- Distributed and autonomous policy driven;
- Flexible QoS support and dynamic management in either DiffServ or IntServ networks;
- Taking account of different demands (such as adaptability) of applications.

Currently, with the QoS negotiation manager presented in Ref.[4] and QoS pricing policy proposed in Ref.[8], we are working toward a PRM prototype within our proposed testbed for QoS control[22]. The major challenges to the deployment of PRM will be the tradeoffs achieved between usability, flexibility and performance. Also we will be deploying application APIs and agents for resource management used in SED-08 routers[21] to provide scalability and tests of security and usability, and to demonstrate the advantages of the framework.

## References

[1] Blake S, Black D *et al.*, "An Architecture for differentiated services", *RFC 2475, IETF,* Dec 1998.

[2] Braden R, Clark D, Shenker S., "Integrated services in the internet architecture: an overview", *RFC 1633, IETF,* June 1994.

[3] Braden R, Zhang L *et al.*, "Resource reservation protocol (RSVP)-version 1 functional specification, *RFC 2205, IETF,* Sep. 1997.

[4] Chen H, Zhang Y, Chen K, "Multimedia quality-of-service (QoS) classification negotiation manager", *Chinese Journal of Electronics,* Vol.7, No.1, pp.44–48, 1998.

[5] Clark D, Fang W, "Explicit allocation of best-effort packet delivery service", *IEEE/ACM Trans on Networking,* Vol.6, No.4, pp.362–373, 1998.

[6] Ferrari D and Delgrossi L, "Charging for QoS", *Proc IWQoS'98,* Napa, CA, May 1998.

[7] Floyd S and Jacobson V, "Link sharing and resource management models for packet networks", *IEEE/ACM Trans on Networking,* Vol.3, No.4, pp.365–386, 1995.

[8] Fu X, Zhang Y, *et al.*, "A market-based approach for bandwidth allocation", *Acta Electronica Sinica,* Vol.8, No.27, pp.127–129, 1999.

[9] Heinanen J, Baker F, *et al.*, "Assured forwarding PHB group", *RFC 2597, IETF,* June 1999.

[10] Jacobson V, Nichols K and Poduri K, "An expedited forwarding PHB", *RFC 2598, IETF,* June 1999.

[11] Kelly F P., "Charging and rate control for elastic traffic", *European Transactions on Telecommunications,* Vol.8, No.1, pp.33–37, 1997.

[12] Nichols K, Jacobson V, Zhang L., A two-bit differentiated services architecture for the internet, draft-nichols-diff-svc-arch-02.txt, *Internet Draft, IETF,* Apr. 1999.

[13] Ramanujan R, Ahamad A, Turber K, "An automated resource management tool for policy-driven QoS control within integrated services Intranets", *In Proc 23$^{rd}$ IEEE Conf Local Computer Networks,* Boston, MA, Oct. 1998.

[14] Reichmeyer F, Ong L, *et al.*, "A two-tier resource management model for differentiated services networks", *Internet Draft, IETF,* Nov. 1998.

[15] Sermret N, Liao R, Campbell A, Lazar A, "Market pricing of differentiated Internet services", *Technical Report CU/CTR/TR* 503-98-37, Dept of Elec. Eng., Columbia Univ, Dec. 1998.

[16] Shenker S, Clark D, *et al.*, "Pricing in computer networks: reshaping the research agenda", *ACM Computer Communication Review,* Vol.26, No.2, 19–43, 1996.

[17] Shenker S, Partridge C and Guerin R, "Specification of guaranteed quality of service", *RFC 2212, IETF,* Sep. 1997.

[18] Wang Q, Peha J M, and Sirbu M. A., "Optimal pricing for integrated-services networks with guaranteed quality of service", In Bailey J and McKnight W (eds), *Internet Economics,* Cambridge, MA: MIT Press, 1997.

[19] Wroclawski J., Specification of the Controlled-Load Network Element Service, *RFC 2211, IETF,* Sep. 1997.

[20] Yavatkar R, Pendarakis D, Guerin R., "A framework for policy-based admission control", *Internet Draft, IETF,* Apr. 1999.

[21] Zhang Y, Zhao Y, Gai F., "Study on the SED-08 router", *High Technology Letters,* Vol.7, No.10, pp.32–34, 1997.

[22] Zhang Y, Chen H *et al.*, "The design and implementation of a QoS control mechanism based on perceptual time channel", *Journal of Japanese Information Processing Society,* Vol.40, No.9, pp.3564–3576, 1999.

[23] Gai S, Strassner J, *et al.*, "QoS policy framework architecture", *Internet Draft, IETF,* Feb. 1999.

**FU Xiaoming** was born in Jiangxi, China, on Dec. 9, 1973. He received the B.S. degree in electrical engineering and the M.S. degree in computer science from Northeastern University, China, in 1994 and 1997, respectively. In 2000, he received the Ph.D. degree in computer science from Tsinghua University, China. His current research interests lie on resource management, traffic control and Quality of Service provisioning in high-speed networks (Email: fxm@mail.cs.tsinghua.edu.cn).

**ZHANG Yaoxue** was born in Hunan, China, on Jan. 5, 1956. He received the B.S. degree in electronic communication from Xidian University, Xi'an, China, in 1982, the M.S. and Ph.D. degree from Tohoku Univeristy, Japan, in 1986 and 1989, respectively, both in computer science. He joined Tsinghua University in 1990 and has been a full professor since 1993, at the Department of Computer Science and Technology, Tsinghua University, China. He was a visiting scientist at the Laboratory of Computer Science, Massachusetts Institute of Technology, USA, in 1995 and a visiting professor in the University of Aizu, Japan, in 1998, respectively. He is currently leading a research group with mainly focus on QoS control, router design and implementation, information retrieval and networked multimedia (Email: zyx@moe.edu.cn).