A Resource Management Framework for Enhanced Internets*

Xiaoming Fu, Yaoxue Zhang
Department of Computer Science and Technology, Tsinghua University, China
E-mail: {fxm, zyx}@sun475.cs.tsinghua.edu.cn

Abstract

The emerging quality of service(QoS) mechanisms such as IntServ[2] and DiffServ[1] usually do not take policy factors into account. This paper proposes a framework for resource management in enhanced internets. This framework has the advantage of providing policy control and QoS support for meeting the demands of scalable applications. It consists of four components: domain policy controller(DPC), border resource manager(BRM), interior resource managers(IRM), and end-system resource manager(ERM). Principles of these components are discussed. We describe the methodology by the operation and interaction among those components. An example of QoS pricing policy in a DiffServ environment is presented and a prototype system is under development.

Key words: QoS, Policy, Resource Management, Pricing

1. Introduction

To meet the increasing Quality-of-Service(QoS) requirements of Internet applications, IETF presents an "Integrated Service(IntServ)" model[2], which can provide each flow a certain QoS, e.g., Guaranteed Service(GS) and Controlled-Load Service(CLS). Recently, a simpler, more scalable model called "Differentiated Service(DiffServ)"[1] is proposed. DiffServ model only needs providing packets of different type with different Per-Hop-Behaviors(PHBs). These networks are named "enhanced internets", or QoS-capable networks.

A critical problem of obtain QoS control in such enhanced internets is to present a suitable resource management model which can allocate and dynamically manage resources accordingly. However, most 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 exhaust, new requests cannot be admitted. In fact, QoS mechanisms can be completely actualized only when combined with some set of policies. For instance, reserving resources merely according to all users' capacity requests will lead to a low utilization, since most users are inclined to apply for best services, which will result in a way of best-effort again. Moreover, the complexity caused by many factors, e.g., security, time-of-day and QoS pricing, appeals for policy control for resources.

The objective of this paper is to present a resource management framework capable of providing QoS and policy support for enhanced internets. After an analysis of requirements of resource management for enhanced internets(section 2), The principles and operational phases are described (section 3), illustrated by an example(section 4). Section 5 concludes this paper and outlines our future work.

2. Requirements of resource management for enhanced internets

For a network that can only provide best-effort service, it is enough to employ a single FIFO queueing mechanism for resource management. However, it is insufficient for enhanced internets with diverse QoS objects. Specifically, an enhanced internet must satisfy some key requirements:

- Scalable QoS management. It should support the needs of administrative domains which have different traffic control policies.
- Flexible QoS support. Different applications may have diverse QoS requirements, e.g., low transmission delay vs. a low loss possibility.
- Optimizing resource utilization. Resource should be allocated and dynamically managed reasonably to accommodate more users/applications.

3. The framework

3.1 Policy

An important aspect of our proposed framework is policy support. In this paper, policy is defined as "the set of desired rules for behaviors of distributed applications, system and network resources within a specific

* Partly supported by the National Natural Science Foundation of China(under grant F020303).

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. A policy rule consists of policy condition(s) and policy action(s). We only consider the aspects related to QoS control while we use "policy" in this paper.

"Policy" can be formalized 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>

where, <EndSystemCondition> denotes 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. More details are not discussed here due to lack of space.

The advantages of employing the concept of policy for a resource management framework for enhanced internets are as follows:

- The Internet is a large internetwork which combines multiple autonomy administrator domains. Each has its own management policy and agreements on traffic delivery and accounting with adjacent domains.
- Policy is an active concept and capable of providing flexible QoS support and can be used for distributed resource management. For example, a QoS pricing policy may initiate, change or even tear down some 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.

3.2 Principle of the framework

Our proposed framework is embodied through the functionality of four components: Domain Policy Controller(DPC), Border Resource Manager(BRM), Interior Resource Manager(IRM), and End-system Resource Manager(ERM). Its principle can be shown as Figure 1.

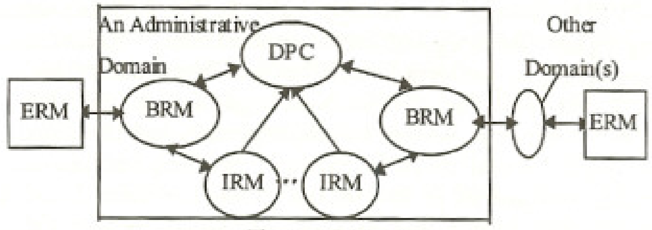


Figure 1 Components

DPC: DomainPolicy Controller ERM: Endsyst Resource Manager

Domain Policy Controller. DPC is a logical entity through which policy information of the domain is maintained and policy decisions are made. It is further subdivided into four modules: *PolicyMgmt* performs indexing, setting, updating, starting/closing policy rules; *PolicyDecidePnt* module performs policy retrieval and reasoning to acquire decisions of policy actions, which are passed to BRM to be enforced; *PolicyStorage* stores policy data, and *StatusTable* 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 as Figure 2.

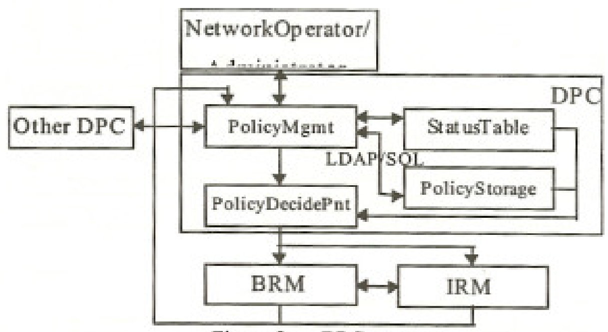


Figure 2 DPC structure

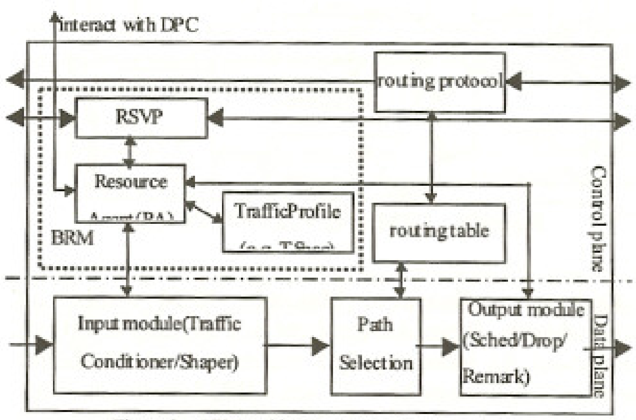


Figure 3 BRM and its location in a border router

Border Resource Manager. BRM resides in the border of a domain. A BRM consists of 3 modules: RSVP[3] module to manage individual(eg.IntServ) or aggregate flows(eg.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 causes resource exhausted so that unable to accommodate, BRM should also report this to DPC, trying to resolve the conflict via policy. BRM is shown as Figure 3.

Interior Resource Manager. IRM resides in each router between two BRMs within the same domain. IRM ignores RSVP and TrafficProfile modules since it should only be responsible for monitoring local resources. Again, when it encounters failure, IRM will report this to DPC of its 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 Figure 4 for the structure of IRM.

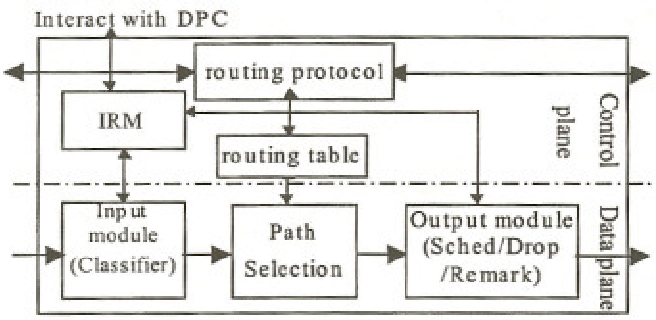


Figure 4 IRM and its location in an interior router

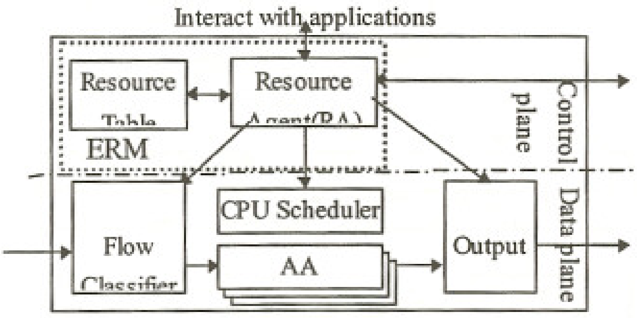


Figure 5 ERM in an end-node/system

End-system Resource Manager. ERM resides on each end node. It enables the front end of users to inform them about the incoming QoS requests, hence it should interacts with the nearby DPC via a BRM to decide if the network can acknowledge or reject the request. Figure 5 shows an ERM.

ERM allocates end-system resources and makes adaptation when necessary. The data plane includes Application Agents(AAs) which provide application data processing such as transcoding, mixing, etc. Chen[4], and Klyne[15] each provides a content negotiation for the resources with which they interact.

The node also has a CPU scheduler which provides computational QoS to AAs. The flow classifier is responsible for delivering incoming data flows for processing.

3.3 Operations

In brief, the operations 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. After receiving this message, the receiver would specify QoS requirements following the reverse path toward the sender. ERMs and BRMs use these parameters for capacity test and QoS computation which finally results in either success or rejection of the reservation attempt. Note 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 resources will be reserved.

Then, in the *data transmission* phase, each unit of traffic allocation(eg., flows in IntServ and traffic classes in DiffServ) are 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 reserved resources in the data transmission phase are scheduled.

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

4. An example: QoS pricing policy handling

The issue of "pricing for QoS" has been investigated in QoS community since the early 1990s[7][8][11]. However, this work has been largely limited to optimal pricing computation and auction method[8][12], leaving the gap between QoS pricing and a viable resource management framework unexplored. Here we take an example of adaptive 3-layered video applications (similar to[6]) to illustrate our framework under a policy under QoS pricing: to serve high bids for the same class of services.

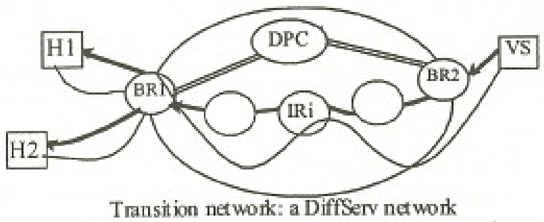
Consider a network setting shown in Figure 6—a DiffServ network with border routers BR1, BR2, interior routers IR1, ... IRi, ..., and an administration station running DPC functions. Users from H1 and H2 can request 3 differentiated quality playback of layered video from VS, correspondingly bidding with three qualitative levels: high, middle and low. QoS pricing policy for this is given conceptually as follows:

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 "rate>=0.9Mbps" /*"QoSOperationPoint='playback at least layer 1, 2'"*/

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



Transition network: a DiffServ network
═Policy message path ——RSVP signal path ——data path

Figure 6. An example exploring the scheme

Our assumed scenario is as follow: 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. Figure 6 shows the sequence of event treatment:

1. H1 requests a video from VS. VS sends a RSVP Path message to H1 containing application information (including the video stream ID and the price H1 bids).

2. H1 reserves 1.2Mbits/s over the path between H1 and VS using RSVP Resv message. It will succeed. BR2 sends a message to DPC describing the 3 operating points of the video and its bid, along with the stream ID. DPC stores this in StatusTable. The video server starts transmitting video streams to H1 and BR2 monitors them.

3. Some time later, H2 requests a same video from VS with a high bid.

4. VS sends H2 a Path message containing the operating points of the requested video stream.

5. H2 send out an RSVP Resv message. Router BR2's QoS negotiation algorithm detects that H2's request cannot be admitted. Then 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 checks policy storage and finds that the rejected reservation has a higher bid than the existing reservation (by H1). It also determines that bandwidth for the new flow can be accommodated by shutting down layers 2 and 3 of the low bid video for H1.

7. It sends a PolicyAction message to ERM on H1 directing it to shut down layers 2 and 3 of the video received by H1. When the application on H1 receives this, it sends a RSVP Teardown message to shut down flows associated with layers 2 and 3 of the 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. At receiving the PolicySignal from H1, DPC tells BR2 to reserve resources for H2. Finally the video session is established.

5. Conclusions and related work

There is an increasing need to develop a scheme for resource management calable of supporting QoS and policy control. Our scheme has the following advantages over other schemes as far as we know:

- Distributed and autonomous policy-driven control;
- Flexible QoS support and dynamic management in either DiffServ or IntServ networks;
- Scalable applications support (eg, adaptability)

Currently, with the the QoS negotiation manager presented in [4] and QoS pricing policy given in [5], we are working toward a prototype system for the framework within our proposed testbed for QoS control[11], as well as deploying application APIs and agents for resource management used in SED-08 routers[10].

References

[1] S. Blake, D. Black, and *et al.* An Architecture for Differentiated Services. *RFC 2475*, IETF, Dec 1998.

[2] R. Braden, D. Clark, and S. Shenker. Integrated Services in the Internet Architecture: an Overview. *RFC 1633*, IETF, Jun 1994.

[3] R. Braden, L. Zhang, and *et al.* Resource Reservation Protocol(RSVP) - Version 1 Functional Specification, *RFC 2205*, IETF, Sep 1997.

[4] H. Chen, Y. Zhang, and *et al.* Multimedia Quality-of-Service(QoS) Classification Negotiation Manager. *Chinese Journal of Electronics*, 7(1), 1998, 44-48.

[5] X. Fu, Y. Zhang, and *et al.* A Market-Based Approach for Bandwidth Allocation. *Acta Electronica Sinica*, 27(9), 1999, 127-129.

[6] R. Ramanujan, A. Ahamad, and K. Turber. An Automated Resource Management Tool for Policy-Driven QoS Control within Integrated Services Intranets. In: *Proc 23rd IEEE Conf. Local Computer Networks,* Boston, Oct 1998.

[7] N. Sermret, R. Liao, A. Campbell, and A. Lazar. Market Pricing of Differentiated Internet Services. *Technical Report*, CTR-TR 503-98-37, EE Dept, Columbia Univ, 1998.

[8] S. Shenker, D. Clark, and *et al*. Pricing in Computer Networks: Reshaping the Research Agenda. *ACM Comp. Comm. Rev.*, 26(2), 1996, 19-43.

[9] Y. Zhang, Y. Zhao, and *et al*. Study on the SED-08 Router. *High Tech. Letters*, 7(10), 1997, 32-34.

[10] Y. Zhang and *et al*. A Testbed for QoS Control on Enhanced Internet. *IEEE/ACM Tran. Netw.*,to appear.

[11] X. Fu, Y. Zhang and *et al*. A Survey and Prospect of Pricing the Internet, *Journal of Computer and Modernization*, 15(2), 1999, 1-6.

[12] Q. Wang, J.M. Peha, and *et al*. Sirbu. Optimal Pricing for Integrated Services networks with Guaranteed Quality of Service. In: J. Bailey and W. McKnight(eds.), *Internet Economics,* MIT Press, Cambridge, MA,1997.

[13] Reichmeyer F, Ong L, and *et al*. A Two-Tier Resource Management Model for Differentiated Services Networks. *Work-in-progress*, IETF, Nov 1998.

[14] Nichols K, Jacobson V, and Zhang L. A Two-bit Differentiated Services Architecture for the Internet. *Work-in-progress*, IETF, Apr 1999.

[15] G. Klyne, Protocol-independent Content Negotiation Framework, *RFC 2703*, IETF, Sep 1999.