Transit-Rate に基づくフロー管理を用いた スケーラブルな QoS 保証フレームワーク

山下 高生

日本電信電話株式会社

東京都武蔵野市緑町 3-9-11

E-mail: yamasita@slab.ntt.co.jp

本論文では、アプリケーションの利用するフローに対して QoS を保証するためのスケーラブルなフレームワークを提案する。本方法は、QoS 保証に必要な各フローへの受付制御, ポリシング, およびシェーピングの役割を階層的に分散化することでスケーラビリティーを改善する。本階層化は、各自律システム (AS) への、入力側 AS と出力側 AS の組み合わせによって行う。このことにより、トラフィックについても同様に階層化したグルーピングを行うことができ、各 AS で識別・管理しなければならないフローの数を削減し、トラフィックを制御するための待ち行列の数を削減する。また、本フレームワークでの経路変更時における、予約資源のマイグレーション方法についても提案を行う。

キーワード: QoS, 帯域保証サービス, リアルタイム・アプリケーション, 経路制御

A scalable framework for providing guaranteed QoS using transit-rate-based flow management

Takao Yamashita

Nippon Telegraph and Telephone Corporation

9-11, Midori-Cho 3-Chome Musashino-Shi, Tokyo 180 Japan

E-mail: yamasita@slab.ntt.co.jp

This paper proposes a scalable framework to provide guaranteed QoS to each flow originated by each application. Our framework hierarchically distributes roles of admission control, policing and shaping functions that are essential for guaranteed QoS. Our method classifies packets into groups that are distinguished by a pair of an incoming and outgoing autonomous systems (ASs). Each group contains multiple flows originated by applications in multiple ASs. The aggregation of flows by the pair decreases the number of states managed in routers especially in a backbone of the Internet. In addition, the aggregation of flows distributes a role of signaling. This enables migration of reserved resource when a routing entry is updated. To reduce affection of routing updates to real-time applications, we also propose a method to migrate reserved resource.

Keywords: QoS, guaranteed service, real-time application, routing

1 Introduction

The more the number of users in the Internet, the greater the variety of applications are used. These includes real-time applications that handles time sensitive data such as motion picture and audio. These applications are called a real-time application. Real-time applications are classified into two types: soft and hard real-time. Applications allowing soft real-time are tolerant of infrequent violation of a deadline. These applications gracefully change their quality so that users don't clearly notice deterioration of communication quality. On the contrary, hard real-time applications, such as broadcasting of extremely high quality continuous media and mission critical application requires packets to be transmitted within a deadline with a high probability. In order to transmit a packet in real-time, various researches and development have been carried out.

These methods are divided into two types: *reservation* and *policy*. The former includes resource reservation protocol (RSVP), stream protocol version 2 (ST-II). The latter includes random early detection (RED), differentiated services (DiffServ). The former executes signaling to reserve network resources. Each router reserves bandwidth of its link for the flow. It tests if characteristics of the flow meets QoS parameters previously declared by an application. If they meets these parameters, a packet is forwarded and output. Output packets are shaped so that its characteristics never violates the declared QoS parameters against a next router. The policy type does not have any states for outside of a domain. It treats packets based on states of a router and/or groups of packets. For example, RED keeps watch the number of packets in a queue and drop packets so that forwarding delay does not extremely long. For another example, a framework of DiffServ classifies packets and a domain has quality policies for each class, such as bandwidth, priority and so on.

The reservation type can provide guaranteed QoS for each flow generated by an application. However, it requires management of a number of states to distinguish many flows, so it is difficult to make it scalable. On the other hand, the policy type is advantageous in terms of scalability and communication quality will be much improved in some condition. However, it cannot provide guaranteed QoS for each application's flow in any condition.

This paper proposes a new framework to provide guaranteed QoS to each flow. Our framework hierarchically distributes roles of admission control, policing and shaping functions. In this framework, trusted autonomous systems (ASs) connect with each other and provides guaranteed QoS by cooperating, while each AS does not need to trust users in it. Our method classifies packets into groups that are distinguished by a pair of an incoming and outgoing links, or interfaces, in a router of an AS. Each group contains multiple flows originated by applications in multiple ASs. The aggregation of flows by the pair decreases the number of states managed in routers especially in a backbone of the Internet. In addition, the aggregation of flows distributes a role of signaling. This enables migration of reserved resource when a routing entry is updated. We propose a method to migrate reserved resource so that real-time applications are not affected by a burst of deadline violation.

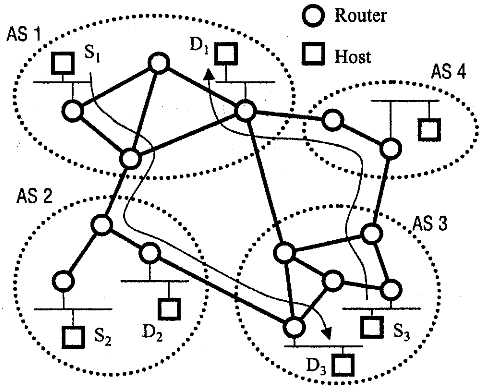

Figure 1: Three types of ASs in a scalable framework to provide QoS.

This paper is organized as follows. Section 2 describes a network model of the Internet that we assume in this paper. In Sec. 3, we propose a scalable framework to provide guaranteed QoS. Section 4 concludes the paper by stating remaining problems.

2 Network model in the Internet

A router used in the Internet forwards packets to one or more neighboring routers. This neighboring router is called next-hop router. Next-hop routers for an unicast and multicast packet is determined by a destination address and a pair of a destination and source addresses, respectively. A router manages a combination of a destination address and next-hop routers in a routing table for unicast communication. A routing table is constructed by manual configuration, called static route, and/or routing protocols such as BGP4 [1], OSPF [2], RIP [3] and so on.

Routing protocols are classified into two types. One is an interior gateway protocol (IGP) and the other is an exterior gateway protocol (EGP). IGPs are used inside an AS. EGPs calculate routing entries for inter-AS communication. Examples of IGPs are RIP and OSPF. An example of an EGP is BGP4.

When we represent the direction from every router to its next-hop routers for a destination address by a directed edge, it should be a directed acyclic graph with only one node whose out-degree is zero, because packets cannot

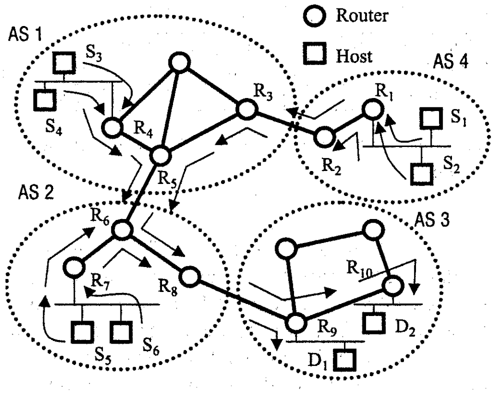

Figure 2: Hierarchical role distribution.

be transmitted to the destination if it includes any loops. BGP4 prevents loops by transmitting information about a path toward a destination AS, called path attribute. Some methods prevent even transient loops [4].

3 Transit-rate based framework

3.1 Approach

A key of our framework is to manage aggregated flows by a pair of an incoming and outgoing ASs. A router in an AS manages traffic of aggregated flows as *transit-rate* by a pair of an incoming and outgoing interfaces. This reflects AS-level transit-rate. Flows transmitted from an end to the other end are hierarchically aggregated by a chain of this transit-rate. The aggregation of flows distributes a role of signaling. This enables migration of reserved resource when a routing entry is updated.

3.2 AS types

A network of an AS usually has a hierarchical structure. It can simply be divided into a backbone network and the others. In our framework, there are three types of ASs for a pair of a source and destination: source AS, transit AS, destination AS. A source AS includes hosts sending packets to a destination. A source AS forwards a packet generated by a user's host to a neighboring AS via its own backbone network. A transit AS receives packets from a neighboring AS and forwards them to another via only its backbone network. A destination AS includes hosts that receive these packets. Figure 1 represents three types of ASs. For communication from S_3 to D_1, $AS1$, $AS2$, and $AS3$ are a source, transit, and destination ASs, respectively. For communication from S_1 to D_2, $AS4$ and $AS3$ are a source and destination ASs, respectively. $AS1$ and $AS2$ are transit ASs. The thicker solid lines stand for backbone networks of ASs.

3.3 Reservation requests

To guarantee QoS for each flow, a end-to-end reservation of network resources is necessary. Our framework divides a reservation request into two levels: inter-AS and intra-AS requests. An inter-AS request is a request to reserve network resources from a border router in a source AS to that in a destination AS. This request is initiated by routers in a backbone network. An intra-AS request is a request to reserve network resources from a source host to a border router in a source AS and those from a border router in a destination AS to a destination host. Because border routers in an AS have to cooperate to migrate reserved resources when a routing entry is updated, information of aggregated reservation requests is shared among border routers. For example, it is achieved by various reliable multicast protocols [5].

For an example of Fig. 2, an inter-AS request between $AS3$ and $AS4$ initiated by R_2. This request is propagated to R_9. This request message reserve network resource along the path from R_2 to R_9 via R_3, R_5, R_6, and R_8. Intra-AS requests for $AS3$ and $AS4$ reserves resources from S_2 to R_2 and from R_9 to ${}_2$, respectively.

3.4 Hierarchical role distribution

Our framework provides guaranteed QoS by distributing roles of admission control, policing, and shaping of traffic. The distribution of these roles decreases the number of managed reservation states and queue states for policing and shaping. A router in a backbone network of a source AS identifies each flow and executes admission control, policing, and shaping against each flow originated in it. A backbone router of a source AS has several states. It manages transit-rate from inside of the AS to neighboring ASs. A backbone routers of a transit AS also manages transit-rate from every neighboring AS to another. These routers executes policing and shaping against grouped flows. These routers have queues for every pair of an incoming and outgoing interfaces. They police and shape traffic for every pair according to an amount of reserved resources. As described above, flows are hierarchically grouped along a chain of transit-rate managed by every AS. In our framework, multiple ASs provides guaranteed QoS by cooperating. When traffic that does not meet reserved resources, transit traffic including such traffic are dropped.

This hierarchical management of flows enables each AS being able to modify an amount of reserved resources. Each AS manages an amount of reserved resources to transit traffic to each destination AS that is a routing domain. Here, a routing domain means an area in a network for which a route is common among hosts in an area. Hence, each AS can migrates reserved resource on current routes into new routes.

For example, R_1 identifies flows of S_1 and S_2 and polices them. Then they are forwarded and merged with flows of S_3 and S_4 in R_6. The merged flows are identified as only one grouped flows forwarded from R_5 to R_8.

3.5 QoS policy of AS

In our framework, QoS policy provided to users are described by *transit-rate policy*. This represents policies traffic forwarded between neighboring ASs. These are

1. maximum transit-rate accepted when being requested by a neighboring AS and
2. ratio between neighboring ASs .

1 means the allowable rate forwarded from a neighboring AS to another when it determine the increase of a rate by itself or a neighboring AS requests. 2 means the ratio at which packets destined for a destination AS are distributed to multiple neighboring ASs when there exists multiple next-hops for a destination. We call transit-rate actually admitted *committed transit-rate (CTR).*

3.6 Comparing granularity of traffic identification

In RSVP framework, applications initiate a request to obtain network resources when they want to receive a real-time stream. Routers on a path from a sender to receivers manages *path state* and *reservation state* for every flows sent by applications even in a backbone core network. This is necessary because RSVP aims at service with strict quality and violation of flow's actual QoS parameter against previously declared QoS parameters must not affect any other flows. Therefore, this framework needs a number of queues to schedule packets in classified traffic.

The scalable core (SCORE) framework [6] accomplishes a small number of states by aggregating flows at an ingress router to a multiple protocol label switching (MPLS) domain. In the SCORE framework, aggregated flows with the same label are treated in fair behavior of a router. However, the larger the MPLS domain, the more the number of identified aggregated flows.

Guaranteeing QoS requires cost and causes decrease of packet forwarding performance of routers especially in a backbone network. Therefore, it is especially necessary to reduce the number of queues for packet scheduling. In our framework, each AS has to manages queues to transit. Aggregated flows are distinguished by a pair of an incoming and outgoing ASs. This means that a router needs to manage a queue for every pair of an incoming and outgoing links, or interfaces. In the current Internet, the number of physical and logical interfaces of each router might not exceed several ten. Therefore our framework should be scalable in terms of the number of queues. On the contrary, our framework has to reservation states for every ASs. The SCORE framework needs to manage reservation states whose number is equal to the number of labels. In terms of the number of reservation states, the SCORE framework is advantageous. Clarification of this difference of the number of reservation states is our future work. Summary of the managed reservation states and queues is shown in Table 1.

	reservation state	queue
Our framework	every AS	every transit pair
RSVP	every destination	every destination
SCORE	every labels	every labels

Table 1: Summary of managed reservation state and queues.

3.7 Determination of committed transit-rate

A reservation request for an inter-AS is originated by a source AS. A reservation request message consists of a prefix for a destination AS and a variation or absolute value of a rate. When a source AS wants a an amount of a rate and multiple next-hop routers exist, it is divided according to a ratio described in its QoS policies described in 3.5. Then every request is sent to each next-hop.

When a transit AS receives requests from neighboring ASs, it then aggregates requests for the same destination into one request and sends it to neighboring ASs.

A backbone router in a destination AS returns acknowledgment messages to previous-hop routers. They are propagated for a source AS that originates a reservation request. Each router manages a rate for a destination AS that can be forwarded by it.

Initial state The initial value of a committed transit-rate is zero. All the entry of a routing table is initially disabled for real-time traffic.

Request message processing Let $r_k^{(i)}(j)$ be a rate, or bandwidth, for destination j requested to k by i. Let $N_j^{(i)}$ and $P_j^{(i)}$ be sets of next-hops and previous-hops for destination j, respectively. Let $t_i^{(j)}$ be a ratio of a rate at which traffic is forwarded to next-hop router i by j. The rate requested to neighboring router l for destination j is

$$\sum_{k\in P_j^{(i)}} r_i^{(k)}(j)\frac{t_l^{(i)}}{\sum_{m\in N_j^{(i)}} t_m^{(i)}}. \qquad (1)$$

Calculation of CTR By using Eq. 1, the CTR for transit from i to j at k is

$$\sum_l r_k^{(i)}(l) \frac{t_j^{(k)}}{\sum_{m \in N_l^{(k)}} t_m^{(k)}}, \tag{2}$$

where l is a destination AS.

3.8 Packet forwarding

Packet forwarding process Routers forward packets based on a routing table. When multiple next-hop routers for a destination exist, packets are forwarded to multiple next-hop routers according to QoS policies described in 3.5. Each next-hop router is used at a proportion of rates.

When a packet flows into a router, it determines a pair of an incoming and outgoing interface according to the following table. When multiple next-hop routers exist, packets are distributed to pairs of an incoming and outgoing interfaces.

Policing and shaping process for transit Routers of our framework manages queues for transit traffic. When a neighboring router has more than two interfaces, Traffic shaped by more than one queue of an upstream router joins one queue. It causes narrower inter-packet gap than determined by the CTR. A policing function for a transit queue has to allow a burst of traffic caused by merging multiple shaped traffic. Rate based packet scheduling methods [7][8] calculates a rate of flow using inter-packet gap. When using virtual clock [8] and maximum packet length is L_{max}, VC_i, which is a variable called *virtual clock*, assigned to flow i whose average rate is R_i can exceed a real time by L_{max}/R_i. Because a router cannot distinguish traffic merged in a previous router, VC_i has to allow $n \cdot L_{max}/R_i$ of excess from a real time, where n is the number of queues through which traffic joins one queue for transit traffic.

3.9 Transit-rate migration in a route change

When a router changes a nexthop router for a destination AS, real-time traffic should be transmitted within its deadline even just after the change. In addition, network resources reserved for traffic forwarded to a current next-hop should be deleted due to effective use of resources. Accordingly, before changing a nexthop router, network resources along the path where traffic increases should previously be allocated. After changing a next-hop router, resources along the path where traffic decreases should be deleted.

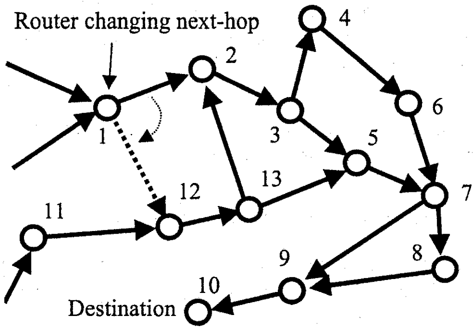

Figure 3: Approach to view divergence control of replicated data.

Theorem 1 *Assume node i changes a next-hop router from node j to k for destination d. We consider two sub-DAGs with only one node whose in-degree is zero. The sub-DAGs are subgraphs of the DAG representing packet forwarding route for a destination. One consists of nodes that are ancestors of node j. The other consists of nodes that are ancestors of node k and are not ancestors of node j. We call a node via which all the routes from j to d and from k to d modification end point. Downstream nodes of a modification end point does not need to change any committed transit-rate.*

Proof: Traffic forwarded by node i does not change both before and after changing a next-hop. Traffic forwarded to m by p is equal to Eq. 1, where p is a modification end point. Eq. 1 is not dependent on distribution of traffic transmitted from previous hops but is determined by only the total traffic destined for a destination. Hence, Ancestors of a modification end point does not need to change any committed transit-rate. □

A router changing a next-hop router c sends modification messages to the current and new next-hop routers. These messages consists of identifiers of router c, a next-hop router, a variation of a rate for a destination, fun-out level, and joining level. A way to process two levels is as follows. The initial values of both fun-out and joining levels are 1s. When a path with fun-out level i and joining level j branches out into n path, fun-out level is changed into $i \cdot n$. A way to change a rate increased or decreased is the same as described in 3.7. When merging multiple modification message with the same next-hop router and the same destination AS, fun-out and joining levels are $\prod_i$ and $\sum_i J_i \prod_{j \neq i} F_j$, respectively.

When a router receives different multiple messages with fun-out level F_i and joining level J_i of message i, we calculate $\sum_i J_i/F_i$. If $\sum_i J_i/F_i$ of messages with a current next-hop and new next-hop is equal to 1, this router receives all the modification messages. This router notifies termination detection of modification message propagation to all the previous hops.

We describe an algorithm to change reserved rate of current and new paths using the above techniques below. Let O be a router that changes a nexthop router for a destination address.

Initial process A router sends a modification message for both a current and new next-hop routers.

Receive process of modification message

1. When a router receives a modification message, it locks routing table entry for a destination included in the message.
2. A router modifies the variation of a reserved rate in a message according to the method described in 3.7.
3. When there exists n next-hop routers, it changes fun-out level f in a message into $n \cdot F$.
4. A router sends a message to every next-hop routers.
5. Let F_i and J_i be a fun-out and joining levels in message i, where $0 \leq i \leq n$. If $\sum_i J_i/F_i$ for a current next-hop router is equal to 1 and $\sum_i J_i/F_i$ for a new next-hop router is also equal to 1, then a router determines that all modification messages have arrived.
6. A router notifies the completion of reception of all modification messages to all the previous-hop routers.

Theorem 2 *The above algorithm detects termination of modification message delivery.*

Proof: When a modification message is delivered to a current next-hop, a total amount of J_1/F_1 in a message is 1 because the initial values of F_1 and J_1 are 1s. Assume that n messages exist in a network and a total amount of J_i/F_i $(1 \leq i \leq n)$ is 1. Then a modification message k branches out m next-hops. A fun-out and joining levels of messages branched out from message k are J_k and $m \cdot F_k$, respectively. Because the total amount of J_l/F_l is J_k/F_k, the total amount of J_r/F_r $(1 \leq r \leq n+m-1)$ is 1, where r is a modification message. Hence, a modification end point can detect termination of reception of all modification messages. □

4 Conclusion

This paper has proposed a scalable framework to provide guaranteed QoS to each flow. Our framework hierarchically distributes roles of admission control, policing and shaping functions. This framework decreases the number of reservation states and queues managed by routers especially in a backbone network. In addition, we propose a method to migrate reserved resource when an entry of a routing table is updated not to reduce communication quality.

A method proposed in this paper is a framework to provide guaranteed QoS. We have not gone into details of this framework yet. We will investigate details of the whole framework. In addition, our future work also includes to investigate the way to police aggregated flows in a router of a transit AS.

Acknowledgments

We thank Dr. Haruhisa Ichikawa for his continuous support. We also thank Mr. Hirohide Mikami for his support and helpful suggestions.

References

[1] Y. Rekhter and T. Li, "A border gateway protocol 4 (bgp-4)," *RFC 1771*, March 1995.

[2] J. Moy, "Ospf version 2," *RFC 1583*, March 1994.

[3] C. Hedrick, "Routing information protocol," *RFC 1058*, June 1988.

[4] J. J. Garcia-Lunes-Aceves, "Loop-free routing using diffusing computations," *IEEE Transactions on Networking*, vol. 1, February 1993.

[5] K. P. Birman, *BUILDING SECURE AND RELIABLE NETWORK APPLICATIONS*. Manning Publications, 1996.

[6] I. Stoica and H. Zhang, "Providing guaranteed services without per flow management," in *Proc. ACM SIGCOM Conference*, 1999.

[7] A. Demers, S. Keshav, and S. Shenker, "Analysis and simulation of a fair queueing algorithm," *Journal of Internetworking Research and Experience*, pp. 3–26, October 1990.

[8] L. Zhang, *A New Architcture for Packet Switched Network Protocols*. PhD thesis, Massachusetts Institute of Technology, July 1989.