

Hierarchical Path QoS on a QoS-based Multicast Protocol SRSVP

TAKAAKI SEKIGUCHI[†], KENJI FUJIKAWA^{††}, YASUO OKABE^{††}
and KAZUO IWAMA[†]

[†]Department of Communications and Computer Engineering

^{††}Department of Intelligence Science and Technology,
Graduate School of Informatics, Kyoto University

Abstract In this paper, we argue a method to collect information of each existing multicast flow on hierarchical networks. SRSVP, a QoS-based multicast routing protocol, is designed as it collects flow-specific information, called PQ, by putting it into signaling messages, so that the derived QoS path becomes more efficient. HQLIP, an underlying QoS-based unicast routing protocol, handles a network as a hierarchical structure for scalable QoS-based routing. We have designed and implemented an algorithm to compute PQ (hierarchical PQ) corresponding to aggregated link information on hierarchical networks for SRSVP to compute better QoS paths. We have attempted to make the algorithm more efficient by examining behaviors of routers.

1. Introduction

IP multicasting is designed to enable the delivery of packets to a set of hosts that have been configured as members of a multicast group¹⁾. Various protocols for IP multicast routing such as PIM-SM²⁾ have been developed. But these existing protocols are based on the best-effort service, so QoS guarantees are not considered. On the next-generation Internet, it is necessary to accomplish some services, for example, multi-site video conferences and broadcasting over the whole of the Internet. Therefore IP multicast routing with QoS guarantees on a large-scale network is required.

Traditional routing protocols such as OSPF³⁾ distribute single arbitrary metric, while QoS-based routing protocols distribute additional routing metrics such as transmission delay and available bandwidth. If any of these metrics change frequently, routing updates may become more frequent and they consume more network resources. That is, there exists a scalability issue in QoS-based routing on a large-scale network. One of techniques for the issue is to aggregate local information by handling a network as a hierarchical structure⁴⁾ thereby avoid flooding messages over the whole network.

For scalable QoS-based multicasting, a QoS-based multicast routing protocol, called SRSVP⁵⁾, and a QoS-based unicast routing

protocol, called HQLIP⁶⁾, have been proposed by Real Internet Consortium (RIC, <http://www.real-internet.org/>). SRSVP uses a mechanism to collect flow-specific information, called PQC (Path QoS Collection)⁷⁾, to compute better QoS routes in order to let receivers join multicast distribution trees. HQLIP handles a network as a hierarchical structure so that it archives a scalable QoS-based routing.

In this paper, we argue an algorithm to collect PQ on hierarchical networks, so that SRSVP compute better QoS routes moreover on hierarchical networks handled by HQLIP.

2. A Framework for QoS Multicasting Routing

2.1 PQC

PQC is a mechanism to collect flow-specific information for QoS-based routing.

In any QoS-based multicast routing model, it is important how routers collect flow-specific information. That is, how much information routers have about existing multicast trees affects routing heuristics very much.

For example, in the PNNI signaling protocol, QoS routes are determined without collecting flow-specific information. For QoS-based multicast routing with such mechanisms, it is impossible to compute efficient routes reflecting current multicast trees. Because of a lack of information about resources consumed by mul-

unicast flows, it may appear that there exists no routes accommodating the requested QoS, and route determinations may fail.

On the other hand, QOSPF⁸⁾ attempts to collect all information. It advertises flow-specific information on links by messages called RRA. Routers are notified all states of multicast flows by them and they will compute efficient routes^{9),10)}. But the number of RRA messages can easily become large as the number of flows increase. There exists a scalability issue on a large-scale network, so it will be unrealizable.

PQC is a mechanism as it collects flow-specific information on links, called PQ (Path QoS), by putting it into Path messages of signaling protocols. Using this information, each router updates link-state information flooded by QoS-based routing protocols and computes better QoS routes. PQ includes transmission delay and available bandwidth for a flow. For example, assuming that there exists a flow consuming 3Mbps bandwidth on a link and QoS-based routing protocol advertises the link has 6Mbps available bandwidth, then PQ indicates that the link has 9Mbps available bandwidth.

The following figures illustrate examples of PQC. Figure 1 illustrates a network and its link states. The numbers beside links indicate available unidirectional bandwidths on each link. For simplicity, bandwidths on links are considered to be the same value as the opposite direction, originally they are different in directions.

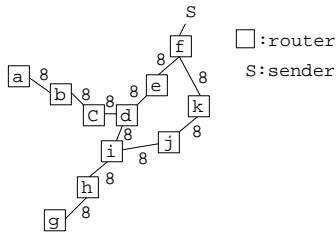


Fig. 1 A network

First, when a receiver $R1$ requests 5Mbps to S that is a sender of a multicast flow, the Resv message for the resource reservation is routed along the path $a - b - c - d - e - f$ computed by a QoS-based routing protocol. Next, another receiver $R2$ requests 4Mbps to S . The Resv message is sent to S . The path will be $g - h - i - j - k - f$ because other paths have insufficient bandwidths and cannot grant the request. Then link states and the multicast dis-

tribution tree become like Figure 2(a).

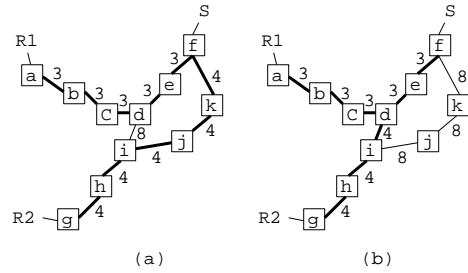


Fig. 2 (a) Reservations without PQC / (b) Reservations with PQC

PQC works as follows. When the Path message is sent from S , each router investigates more precise QoS information on links for the flow, puts it into the Path message, and forwards the message downstream to the receivers. By PQC, the routers g , h , and i can know the links among d , e , and f are available for the flow, that is, these links have 8Mbps available bandwidths. Then the Resv message will be routed along the path $g - h - i - d - e - f$ and the link states and the multicast tree becomes like Figure 2(b). Reservations with PQC consume network resources more efficiently than those without PQC.

2.2 SRSVP

SRSVP is a QoS-based multicast routing protocol that combines a resource reservation mechanism like RSVP¹¹⁾ with a multicast routing scheme like PIM-SM.

Traditional multicast routing protocols such as DVMRP^{1),12)} and MOSPF¹³⁾ are based on broadcasts for receiver discovery, so they have a scalability issue. SRSVP employs the concept of a rendezvous point (RVP) like PIM-SM to solve the issue. In SRSVP, multicast packets are transferred from a sender to a rendezvous point by unicasting and to receivers by multicasting. That is, reservations between a sender and a rendezvous point and those between a rendezvous point and receivers are established independently.

The mechanism of resource reservations and multicast routing are as follows. A receiver sends a Resv message, called Resv0. Each router forwards it along the best-effort path to a sender. A sender sends a Path message in response to the Resv0 message. Each router forwards it with PQ along the reverse path of the Resv0 message. The receiver sends a second Resv message, called Resv1. Using PQ, each

router calculates a path from the sender that accommodates to the requested QoS, forwards the message along the path. The sender sends a Path message in response to the Resv1 message. Each router establishes a resource reservation for the flow and forwards the message with PQ along the reverse path of the Resv1 message.

2.3 HQLIP

HQLIP is a QoS-based unicast routing protocol that is an extension of OSPF. HQLIP has multiple layers to be available over the Internet to aggregate QoS information and transfers that QoS state is getting worse promptly. HQLIP is designed to require no crank back processing like PNNI.

HQLIP defines each physical interface on routers as a level 0 area. Areas at levels less than $i + 1$ connected through routers make a level $i + 1$ area. The level $i + 1$ area is called a *parent area* and each of the areas at levels less than $i + 1$ is called a *sub-area* of the parent area. A border of an area exists on routers. If an interface on a router belongs to an area, it is said that the router belongs to the area. A router connecting multiple areas is called a *border router* among those areas. Each area has a center to aggregate QoS information.

Link information flooded by HQLIP has two types: internal link information and external link information. Information flooded in a level i area about how the sub-areas (at levels j and k ($j, k < i$)) connected with each other is called internal link information of level (j, k) . Information flooded in a level i area about how the sub-area (at level k ($k < i$)) connected with an area (at level j) adjacent to the level i area is called external link information of level (j, k) .

The following figures illustrates how link information is generated by HQLIP. In this paper, we use the notation $(x \leftarrow y, bw=u, dly=v)$ as link information. It indicates that available bandwidth is u and that transmission delay is v on a link from an area y to an area x .

Each router creates internal link information of level $(0, 0)$ for each interface in direction that packets are outgoing and floods it in the parent area of that interface (Figure 3).

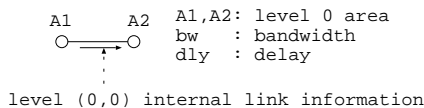


Fig. 3 (0,0) internal link information

Each router on a border of a level i area ($i \geq 1$) creates external link information of level $(0, i)$ for each interface that does not belong to the level i area and floods it in the parent area of the interface (Figure 4).

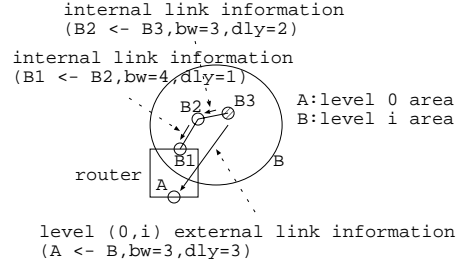


Fig. 4 (0,i) external link information

Each center of level j areas ($j \geq 1$) calculates link QoS information in direction that packets are outgoing using internal link information flooded in that area. By using that information and external link information, the router calculates QoS information to the center of an adjacent area (at level i) and floods it in the parent area as link information of level (j, i) (Figure 5). If the parent area is also the parent of the level i area, this link information is treated as internal link information. Otherwise it is treated as external link information.

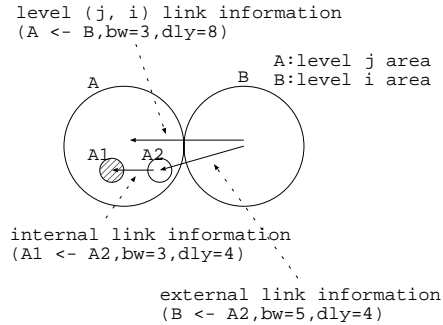


Fig. 5 (j,i) link information

3. Hierarchical PQ

3.1 Hierarchical PQ

To compute better QoS routes with PQ on hierarchical networks, SRSVP requires PQ corresponding to link information among areas besides original PQ. We call it *hierarchical PQ*.

3.2 Computation of Hierarchical PQ

Computation of hierarchical PQ is classified the following five cases.

When a Path message is going out of an area

B and B does not include the sender, the border router of B calculates a hierarchical PQ from an area A to B using the following information, where A is a maximum level area of the previous-hop areas of B .

Case 1 If B and A are level 0 areas, the border router calculates $PQ(B \leftarrow A)$ with link information from A to B , flow-specific information on the link, and resources consumed by the Path message itself.

Case 2 If B is a level 0 area and A is an area at level greater than 0, the border router calculates $PQ(B \leftarrow A)$ with external and internal link information from A to B and PQ in the Path message received from the previous-hop router.

Case 3 If B is an area at level greater than 0, the border router calculates $PQ(B \leftarrow A)$ with external link information from A to a border sub-area of B , internal link information among sub-areas of B , and PQ in the Path message received from the previous-hop router.

When a Path message is going out of an area A (at level greater than 0) and A includes the sender, the border router of A computes information called FAQ (First Aggregated QoS) using the following information.

Case 4 In case a Path message is going out of an area A (at level greater than 0) that includes a sender, the border router computes $FAQ(A \leftarrow S)$ with internal link information among sub-areas in A , PQ in the Path message received from the previous-hop router, and FAQ from the sender to a sub-area of A that includes the sender.

Unlike PQ, there is no link information corresponding to FAQ. FAQ is not used at the QoS route determination, but used to check whether the determined route really accommodates the requested QoS or not.

Finally, when the Path message reaches a receiver, the following computation is hold.

Case 5 When a Path message reaches a receiver R , it computes the same PQ as Case 3, considering it is going out of an area that include R and that border is not on R . This computation is started from the lowest level area and is repeated while the previous-hop area exists.

The following figures illustrate examples of Case 2, Case 3, and Case 4. Figure 6 illustrates a network. In the figure, S indicates a sender, R indicates a receiver, T indicates a router, and other characters indicate areas. Areas with

hatched lines indicate that they include center of the parent area, but centers those are not used in the following explanations are not indicated. Here, a Path message is passing the routers in order of $T1 \rightarrow T2 \rightarrow T3 \rightarrow T4 \rightarrow T5$.

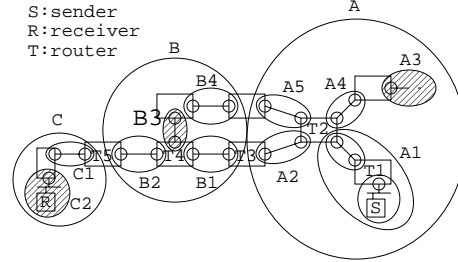


Fig. 6 A hierarchical network

Figure 7 illustrates that the router $T5$ computes $PQ(C1a \leftarrow B)$ (Case 2). The PQ is computed with the following information.

- external link information ($B2a \leftarrow B3$, $bw=4$, $dly=2$)
- internal link information ($B2b \leftarrow B2a$, $bw=1$, $dly=1$)
- $PQ(B2b \leftarrow B2a$, $bw=3$, $dly=3$)

Internal link information ($B2b \leftarrow B2a$) is replaced to $PQ(B2b \leftarrow B2a)$ because it is more precise link information for the flow. Then the PQ is computed with the following information.

- external link information ($B2a \leftarrow B3$, $bw=4$, $dly=2$)
- $PQ(B2b \leftarrow B2a$, $bw=3$, $dly=3$)

The bandwidth of PQ becomes the minimum value and the delay becomes the sum of values above. Then the PQ becomes $PQ(C1a \leftarrow B)$, $bw=3$, $dly=5$.

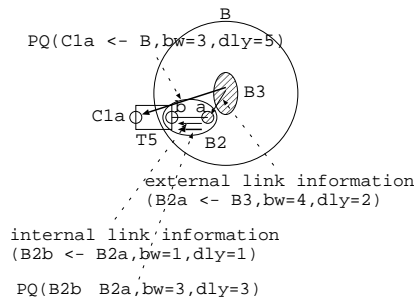


Fig. 7 Computation of $PQ(C1a \leftarrow B)$ (Case 2)

Figure 8 illustrates that the router $T5$ computes $PQ(B \leftarrow A)$ (Case 3). Assuming that link information ($B \leftarrow A$) has been computed with external link information ($B1 \leftarrow A$) and internal link information ($B3 \leftarrow B1$), The PQ is com-

puted with the following information.

- external link information ($B1 \leftarrow A$, $bw=1$, $dly=2$)
- internal link information ($B3 \leftarrow B1$, $bw=3$, $dly=1$)
- $PQ(B1 \leftarrow A$, $bw=2$, $dly=2$)

External link information ($B1 \leftarrow A$) is replaced to $PQ(B1 \leftarrow A)$. As a result, The PQ becomes $PQ(B \leftarrow A$, $bw=2$, $dly=3$).

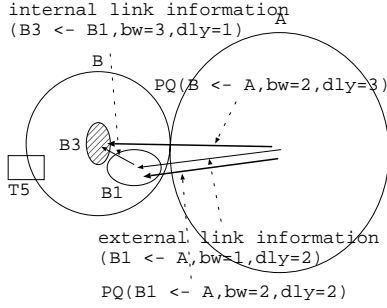


Fig. 8 Computation of $PQ(B \leftarrow A)$ (Case 3)

Figure 9 illustrates that the router $T3$ computes $FAQ(A \leftarrow S)$ (Case 4). The FAQ is computed with the following information.

- internal link information ($A3 \leftarrow A4$, $bw=1$, $dly=2$)
- internal link information ($A4 \leftarrow A1$, $bw=3$, $dly=2$)
- $FAQ(A1 \leftarrow S$, $bw=2$, $dly=1$)

The FAQ becomes $FAQ(A \leftarrow S$, $bw=2$, $dly=5$).

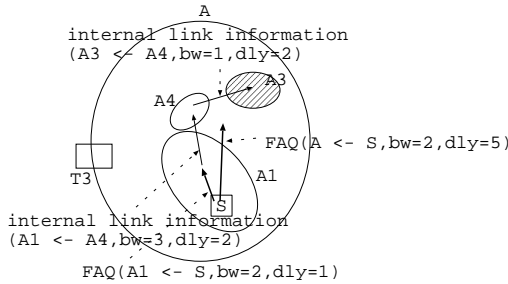


Fig. 9 Computation of $FAQ(A \leftarrow S)$ (Case 4)

3.3 Behaviors in Routers

By examining the dependency of information required by all cases above, routers on the route along the Path message is passing must compute PQ and FAQ observing the following algorithm.

```
procedure hierarchical_pathqos
begin
```

```

    a := parent(D);
    while prev(a) ≠ nil and
        level(a) ≤ level(B) do
        Case3(a, prev(a));
        a := parent(a);
    endwhile;
    if C ≠ nil then
        while level(a) ≤ level(B) do
        Case4(a);
        a := parent(a);
        endwhile;
        Case2(C, B);
    else
        while prev(a) ≠ nil do
        Case5(a, prev(a));
        a := parent(a);
        endwhile;
    endif
end
```

Here, B is the maximum level area of which the Path message is going out, C is a level 0 area representing the outgoing interface of the Path message, and D is a level 0 area representing the incoming interface of the Path message. $level(x)$ returns the level of the specified area x , $parent(x)$ returns the parent area of x , and $prev(x)$ returns the previous-hop area of x . $parent(x)$ and $prev(x)$ returns nil if the router cannot compute the value. $Case2(x, y)$ computes $PQ(x \leftarrow y)$. $Case3(x, y)$ and $Case5(x, y)$ are the same as above. $Case4(x)$ computes $FAQ(x \leftarrow S)$.

3.4 Finding Previous-hop Areas

The function $prev(x)$ in the algorithm above considered to return the previous-hop area of x . But, to return the previous-hop area, the router must investigate the route along which the Path message is passing, that is, it must scan PQ in the Path message. So, each router must scan PQ multiple times every areas of which the Path message is going out.

Assuming that each router run the algorithm above and append the computed PQ at the last part of the Path message, the multiple scans are not necessary and each router can find all previous-hop areas by scanning only one time seen from the whole.

```
function prev(x:area): area
begin
    while pq ≠ nil do
        a := begging area of pq;
        b := parent(a);
        c := parent(ending area of pq);
```

```

if  $b \neq c$  and  $c = x$  then return  $a$ ;
   $pq :=$  previous element of  $pq$ ;
endwhile;
return nil;
end

```

Here, all PQ in the Path message received from the previous-hop router are stored in a list structure. At receiving the Path message, the last element of the list is set to the variable pq .

4. Concluding Remarks

In this paper, we have designed an algorithm to compute hierarchical PQ on networks handled by HQLIP for SRSVP to compute QoS routes more efficiently. We have considered what occurs when a Path message is going out of a router and have attempted to speed up the algorithm by eliminating the extra scans of PQ in the Path message.

As a current implementation of SRSVP and HQLIP, RICD by Real Internet Consortium have been developed. We have implemented a mechanism to compute hierarchical PQ based on the algorithm. A new implementation of them are now developing and we are planning to implement the mechanism on it and to evaluate the performance.

References

- 1) C. Semeria, T. Maufer: Introduction to IP Multicast Routing, <http://www.3com.com/>, 3Com Corporation (1997).
- 2) D. Estrin, et. al.: Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification, RFC 2362, Network Working Group (1997).
- 3) J. Moy: OSPF Version 2, RFC 1583, Network Working Group (1994).
- 4) ATM Forum PNNI subworking group: Private Network-Network Interface Spec. v1.0 (PNNI 1.0), afpnni-0055.00 (1996).
- 5) K. Fujikawa, I. Sheng, Y. Goto: Simple Resource ReSerVation Protocol (SRSVP), Real Internet Consortium (1999).
- 6) M. Ohta: Hierarchical QoS Link Information Protocol (HQLIP), Real Internet Consortium (1999).
- 7) Y. Goto, M. Ohta, K. Araki: Path QoS Collection for Stable Hop-by-Hop QoS Routing, inet97, proceedings (1997).
- 8) Z. Wang, C. Sanchez, B. Salkewicz, E. Crawley: Quality of Service Extensions to OSPF or Quality of Service Path First Routing

- (QOSPF) , Internet Draft (1996).
- 9) B. M. Waxman: Routing of Multipoint Connections, IEEE JSAC 6, pp. 1617-1622 (1988).
- 10) M. Imase, B. Waxman: Dynamic Steiner Tree Problem, SIAM J. Disc, Vol. 4, No. 3, pp. 369-384 (1991).
- 11) R. Braden, et. al.: Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification, RFC 2205, Network Working Group (1997).
- 12) D. Waitzman, C. Partridge, S. Deering: Distance Vector Multicast Routing Protocol, RFC 1075, Network Working Group (1988).
- 13) J. Moy: Multicast Extensions to OSPF, RFC 1584, Network Working Group (1994).