Internet Engineering Task Force Z. Zhang, C. Sanchez, B. Salkewicz, E. Crawley
Internet-Draft Bay Networks, Avici Systems
Redback Networks, Gigapacket Networks
draft-zhang-qos-ospf-01 September 1997

Quality of Service Extensions to OSPF or Quality Of Service Path First Routing (QOSPF)

Status Of This Memo

Abstract

This document describes a series of extensions for OSPF[1] and MOSPF[2] that can be used to provide Quality of Service (QoS) routing in conjunction with a resource reservation protocol such as RSVP[4] or other mechanisms that can notify routing of the QoS needs of a data flow. Advertisements indicating the resources available and the resources used are advertised to the OSPF routing domain and paths are computed based on topology information, link resource information, and the resource requirements of a particular data flow.

1.0 Introduction

QoS signalling protocols such as RSVP allow the instantiation of network state to provide a specific service level to a data flow. RSVP is specifically not a routing protocol but it does have interfaces to routing in order to determine the forwarding of its own state messages. Existing routing protocols are usually concerned only with topology information and not network resources such as bandwidth, thus they all have their limitations in providing integrated services. The following figure is a simple illustration:

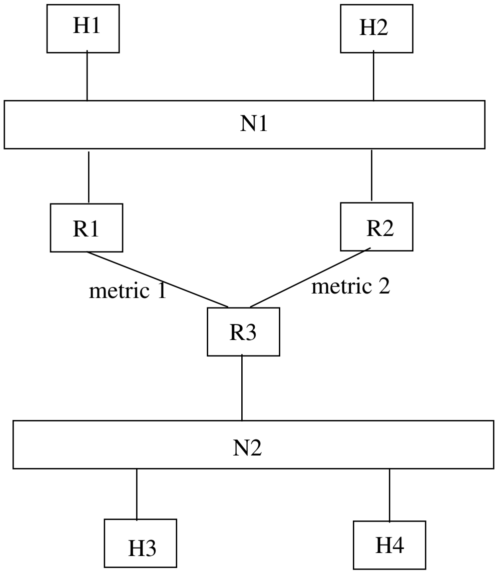


FIGURE 1. Example Topology

Suppose host H1 is sending data to host H3 at rate R. The routing protocol in use gives the shortest path as defined by the metrics, H1-->R1-->R3-->H3. However, even if R1 does not have adequate resources on its interface to R3 to handle the flow at the rate R, the route H1-->R2-->R3-->H3 that does have adequate resources available, is not used because the routing protocol always uses the shortest path.

One solution is to let the routing protocols consider network resource information as well as topology information when they calculate routes. With the OSPF protocol, complete topology information is used to calculate routes; in QOSPF, network resource information is added and used to calculate "QoS routes" that can provide the resources needed for the flow even though the route may not be strictly the shortest path.

2.0 Protocol Overview

2.1 Network Resource Information

In QOSPF, routers advertise network resource information as well as topology information. A route for a data flow[1] is calculated based on topology, network resource information, and QoS requirements (e.g. the TSpec of the RSVP

[1] In this document, a flow is identified by (source address, destination address) instead of (destination address/protocol/port, source address/port). This means that all flows for a given (source address, destination address) pair will follow the same route. Allowing multiple flows from the same (source address, destination address) pair to use different routes is a topic for further study.

PATH message) for the flow[2].

The network resource information includes available link resources on a router as well as existing link resource reservations on the router. The resource information is advertised in Link Resource Advertisements (RES-LSAs) and Resource Reservation Advertisements (RRAs). Another type of advertisement, Deterministic Area Border Router Advertisements (DABRA), are needed for inter-area multicast QOSPF.

There are a lot of ways to represent network resource information. In this document, we use Token Bucket parameters, as in the Controlled-Load Service model[5]. It is expected that resource advertisements that are related to other service models could be added over time.

The number of RRAs can easily get huge as the number of reserved flows and network size grow, presenting a scaling issue. A solution to this problem is addressed by Explicit Routing, discussed in Section 6.0.

2.2 Route Pinning

Topology and network resource information not only make it possible to calculate a shortest route that satisfies the required QoS for a flow, but also makes Route Pinning very easy to achieve. Route pinning means that an existing route with a reservation will not be replaced by a better route unless the existing one is no longer usable because of a topology change directly related to the existing route.

2.3 Data-driven (Source, Destination) Route Computation

MOSPF uses data-driven (source, destination) routing. In other words, a route is computed when the first packet for a (source, group) pair is received. This is in contrast to unicast OSPF that pre-computes routes based on destination only.

In QOSPF, routing for QoS flows is based on (source, destination)[3], and routing computations are triggered by external events regardless of whether the flow is unicast or multicast. The initial trigger for QoS routing computation comes from a resource reservation protocol such as an RSVP PATH message.

There are two reasons for (source, destination) routing in unicast QOSPF:

- Resource reservations and RRAs are generally based on (source, destination);
- When (source, destination) routing is used, flows with the same destination but different sources can follow different paths when necessary.

Note the (source, destination) routing used in unicast QOSPF does not mean that the distribution tree must be rooted at the source. It only means that

[2] While the TSpec is used now, it is essentially an estimate of the needs of a flow. Refining this estimate with the RSpec on a second pass would provide a better QoS metric. This is considered an optimization of the current design

[3] Best effort unicast routing is still based on destination only.

the routing table lookup is based upon (source, destination) rather than just the destination.

3.0 Resource Advertisements

Available and reserved network resources are advertised via Link Resource Advertisements (RES-LSAs) and Resource Reservation Advertisements (RRAs), respectively.

3.1 Link Resource Advertisement (RES-LSA)

A RES-LSA is very similar to a Router-LSA. The purpose of the RES-LSA is to advertise the link resources available for each router in the network. When calculating QoS routes, RES-LSAs are used instead of Router-LSAs.

Each QOSPF router originates a RES-LSA for each area, listing the largest amount of available resources for reservation on each of the router's interfaces in the area, along with the link's delay metric. This metric is roughly analogous to the standard OSPF cost metric, but is independent of the standard TOS metric to better characterize the static delay properties of a link.

A new instance of RES-LSA is originated whenever a new Router-LSA instance is originated for the area, or whenever the available bandwidth resource or delay changes (significantly) for a link in the area.

An algorithm may be used so that a new RES-LSA is originated only when the available bandwidth resource changes significantly. For example, a router may choose to originate a new RES-LSA only when the change of available bandwidth on a link exceeds a certain amount or certain percentage of total/ remaining bandwidth on the link. However, this can cause routers to have incorrect resource information of the router and the calculated routes may lead to reservation failures. Therefore, if a reservation attempt fails on a router, it should immediately advertise its correct resource information.

Like Router-LSAs, RES-LSAs are flooded throughout a single area.

The format of RES-LSAs is shown in Figure 2.

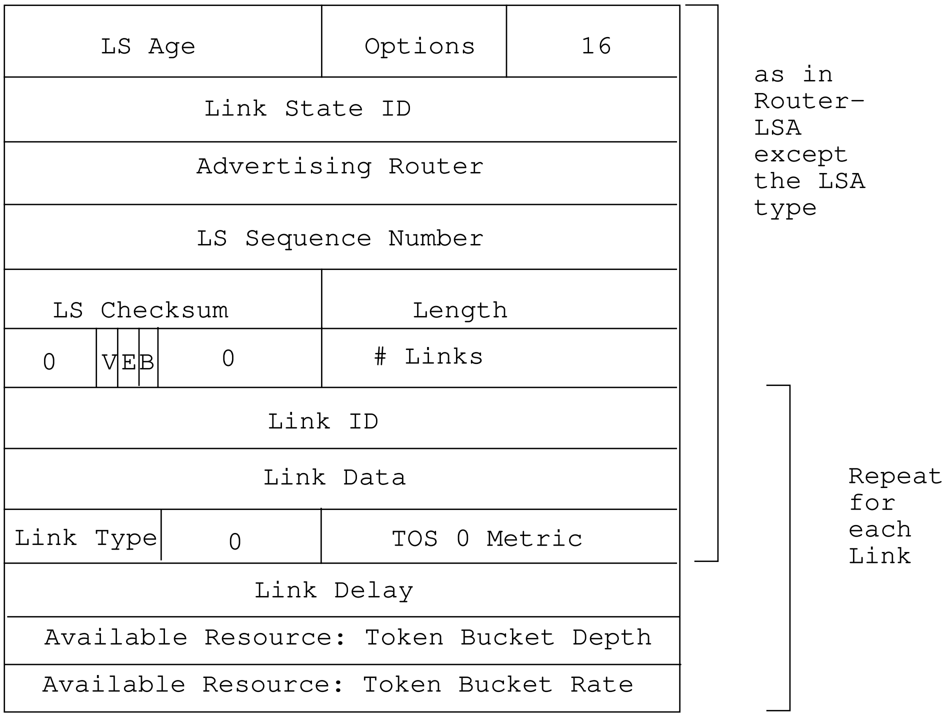


FIGURE 2. Resource LSA

The RES-LSA header is the same as all other LSA headers.

The V-bit, E-bit, B-bit, #Links, Link type, Link ID and Link data are the same as in a Router LSA.

The available link resource is represented by token bucket parameters, in IEEE single precision floating point format, as in the Controlled-Load Service model[5].

The link delay is a static delay metric for the link, in units of milliseconds.

The RES-LSA could be combined with regular Router-LSA because the delay and resource information could be encoded as special TOS metrics in Router-LSAs. However this would cause Router-LSAs to be updated much more frequently and may have some impact to some current OSPF implementations. Therefore, we choose to use a separate advertisement.

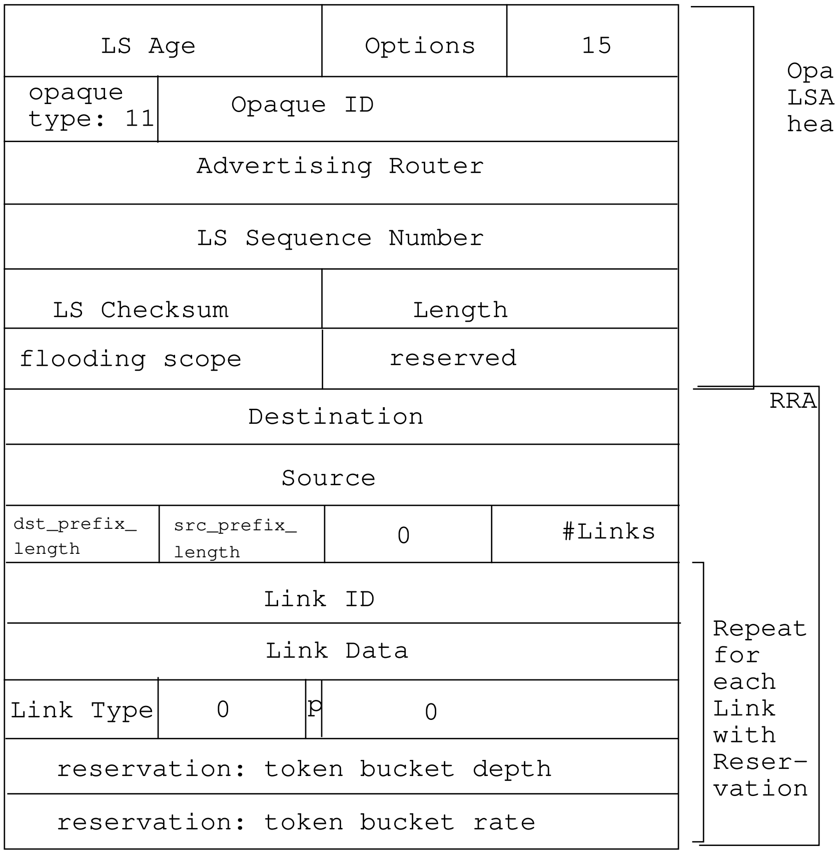
3.2 Resource Reservation Advertisement (RRA)

A Resource Reservation Advertisement describes a router's reservations for a particular flow (source, destination) on its interfaces within an area. The purpose of the RRA is to indicate the resources used by a flow such that other routers are aware of the resources used by the flow when they calculate or recalculate the tree for the flow. A new RRA is originated whenever one or more of the router's reservations change in the area.

Like RES-LSAs, RRAs are flooded throughout a single area[4].

The format of RRAs is shown in Figure 3.

FIGURE 3. Resource Reservation Advertisement with its Opaque LSA header



RRAs are encapsulated in Opaque LSAs with type = 11. The Opaque ID is chosen by the advertising router and the flooding scope is "area-local"[5].

The Destination and Source are the IP address of the destination and source of the data flow, respectively, and the dst_prefix_length and src_prefix_length correspond to the length of the network mask of the destination and source respectively. Usually they are just 0xffffffff.

The "#Links" is the number of links included in the RRA. For each link, the Link type, Link ID and Link data are identical to the values used in the Router LSA.

[4] Assuming Explicit Routing is not used. See Section 6.2.

[5] Assuming Explicit Routing is not used. See Section 6.2.

The P-bit in the 8-bit options field following the “Link Type” is a pin-flag used for route-pinning discussed in Section 5.0.

The reserved bandwidth resource is represented by token bucket parameters, in IEEE single precision floating point format, as in the Controlled-Load Service model[5].

The reservation information comes from a resource reservation protocol, such as RSVP or some other mechanism for reserving resources on the node. Whenever a reservation is made or canceled, QOSPF will originate a new instance of the RRA for the flow. RSVP SE style reservations can cause multiple RRAs to be originated depending on the number of PATH state that is matched, and a RSVP WF style reservation will cause a RRA with a wildcard source (0) to be originated.

4.0 QOSPF Route Calculation

Input to the QOSPF Dijkstra calculation includes the source and destination address and the QoS requirements for the flow, which are currently the token bucket parameters from the RSVP PATH message but could also come from other triggers.

The QOSPF Dijkstra calculation for an area is performed by processing the area's RES-LSAs, Network-LSAs, RRAs, and Group-Membership-LSAs. The latter is only used for the multicast case.

The key difference between the QOSPF Dijkstra and the normal OSPF/MOSPF Dijkstra is that a router’s RES-LSA rather than Router-LSA is used to discover its neighbors, and links will be ignored if they do not have sufficient resources (resource available plus already reserved) for the flow.

To calculate the best or lowest-delay path, the delay metric in RES-LSAs is used in the same way OSPF uses the TOS zero cost metric of Router LSAs.

4.1 Multicast QOSPF

4.1.1 Intra-area Multicast QOSPF

Like in normal MOSPF, the intra-area QoS SPF tree is forward-linked. This means that the best path is chosen based on the delay metrics from the source to the target.

4.1.2 Inter-Area Multicast QOSPF

In MOSPF, for a (source, group) pair, a tree has to be calculated for each area and then the trees are combined into a global tree. When calculating a tree for an area, if the source is in another area, the root of the tree is set to all the ABRs that support MOSPF and have valid Summary LSAs containing the source.

As shown in Figure 4, suppose the source is in area 0.0.0.0. When R5 and R6 calculate their trees for area 0.0.0.1, they will root the trees at R2, R3, and R4.

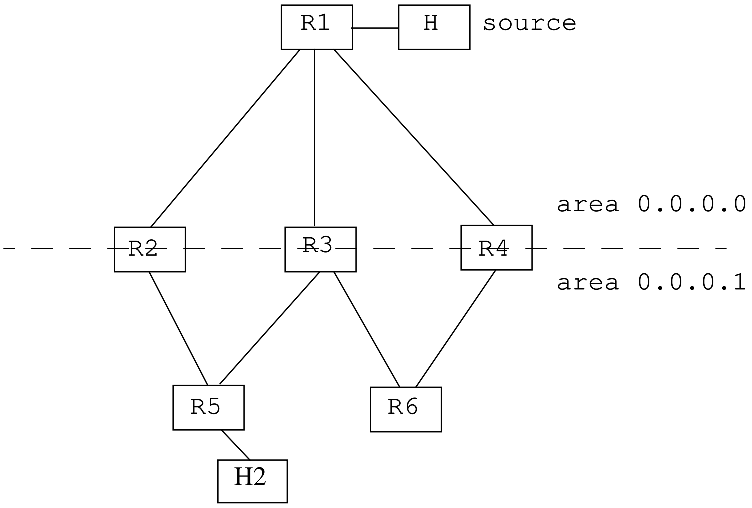
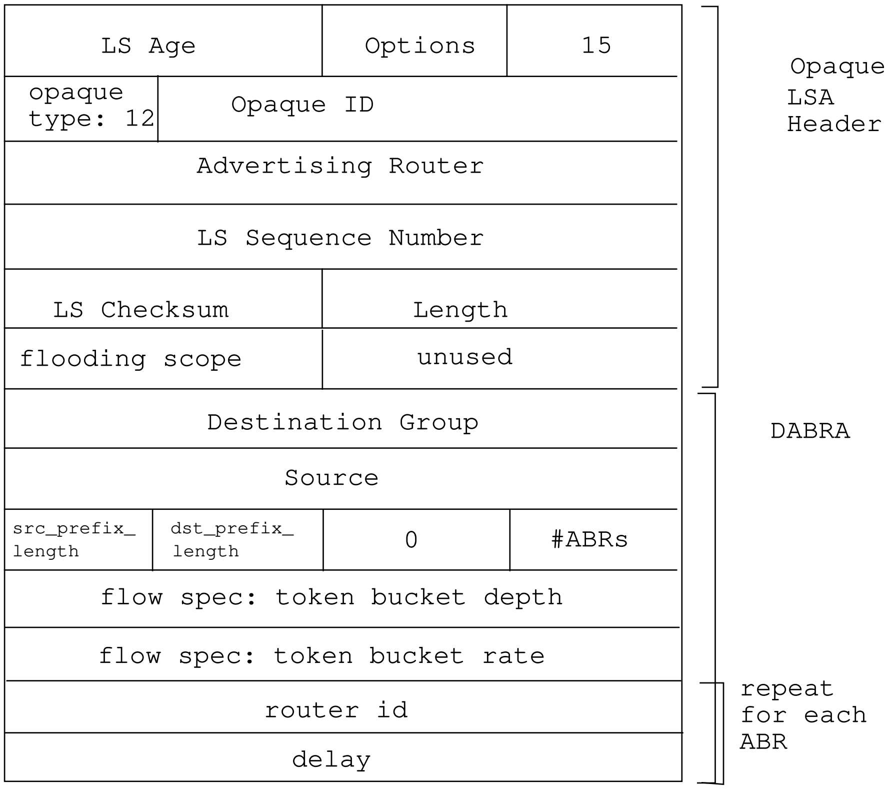


FIGURE 4. A Tree

In QOSPF, links without adequate resources for a data flow are not considered. So, in Figure 4, suppose the link R1->R3 does not have enough bandwidth, then R3 will not be on the multicast tree for area 0.0.0.0 so it will not get the packets. Now when R5 and R6 calculate trees for area 0.0.0.1, they should root the trees only at R2 and R4.

For this reason, after R2 and R4 finishes calculation for area 0.0.0.0, they should notify routers in area 0.0.0.1 how to root the tree via Deterministic ABR-Advertisements (DABRA).


<table>
<tr><td colspan="2">LS Age</td><td>Options</td><td>15</td></tr>
<tr><td>opaque type: 12</td><td colspan="3">Opaque ID</td></tr>
<tr><td colspan="4">Advertising Router</td></tr>
<tr><td colspan="4">LS Sequence Number</td></tr>
<tr><td colspan="2">LS Checksum</td><td colspan="2">Length</td></tr>
<tr><td colspan="2">flooding scope</td><td colspan="2">unused</td></tr>
<tr><td colspan="4">Destination Group</td></tr>
<tr><td colspan="4">Source</td></tr>
<tr><td>src_prefix_ length</td><td>dst_prefix_ length</td><td>0</td><td>#ABRs</td></tr>
<tr><td colspan="4">flow spec: token bucket depth</td></tr>
<tr><td colspan="4">flow spec: token bucket rate</td></tr>
<tr><td colspan="4">router id</td></tr>
<tr><td colspan="4">delay</td></tr>
</table>

Opaque LSA Header
DABRA
repeat for each ABR

FIGURE 5. DABRA with its Opaque LSA header

Each ABR on the QoS tree for the "source area" of a flow originates a DABRA, listing all the ABRs on the tree, and floods it throughout all "downstream areas". If the source of the flow is in one of the router's directly attached areas, then the area is the "source area" and all other areas are "downstream" areas; otherwise (the source is in an area not directly attached to the router), the backbone area is the "source area" and all non-backbone areas are "downstream areas".

4.1.3 Inter-AS Multicast QOSPF

Similar to the inter-area case, there should be a notification about how to root the tree. The details are not explored in this document.

4.1.4 Detailed Multicast QOSPF Dijkstra Calculation

The following procedure is a modification to section 12.2 in the Multicast Extensions to OSPF, RFC 1584. It tries to build a multicast distribution tree that satisfies the bandwidth resource requirement first, then probably a partial best effort tree to cover the rest of routers and networks.

Two new states are added to each vertex: the delay from the source to the vertex, and the resource flag indicating if there is enough bandwidth resource from the source to the vertex.

1. Initialize the algorithm's data structures as in RFC 1584. Set the initial delay to infinity and resource flag to FALSE.

2. Initialize the candidate list as in RFC 1584, with the following differences:

   A. In intra-area case, when a Network vertex is put into the candidate list, set the resource flag to TRUE and set the delay to 0.

   B. In intra-area case, when a Router vertex is put into the candidate list, If its RES-LSA exists and is valid, set the resource flag to TRUE, and set the delay to 0.

   C. In inter-area cases, if the DABRA(s) for this flow exist and is/are valid, and the RES-LSA for an area border router that is both in one of the DABRAs and in the calculating area exists and is valid, set the resource flag of the vertex for the border router to TRUE, and set the delay to the delay value from the DABRA.

3. If the candidate list is empty, the algorithm terminates.

   Same as RFC 1584.

4. Move the closest candidate vertex to the shortest-path tree.

   If there are vertices with TRUE resource flags, the one with least delay is chosen. The same tie-breaker as in RFC 1584 applies.

   Otherwise, the one with least regular OSPF cost is chosen, and the same tie-breaker as in RFC 1584 applies.

5. Examine Vertex V's neighbors for possible inclusion in the candidate list. If V is a router vertex with a TRUE resource flag, consider the links in its RES-LSA. Otherwise, consider the links in its Router-LSA or Network-LSA.

   Each link (say L) describes a connection to a neighboring vertex (say W) or a stub network. Skip links connecting to stub networks.

   If W is already on the SPF tree, or if W's LSA does not contain a link back to vertex V (if vertex W is a router vertex use vertex W's Router LSA to make this determination as it is irrelevant whether or not there is reservable bandwidth in the reverse direction), or if W's LSA has LS age of MaxAge, or if W is not multicast capable (indicated by the MC-bit in W’s Router LSA or RES-LSA’s options field), skip the link.

   For each remaining link, perform the following:

   a. Calculate the cost between the source and vertex W (forward or backward), which is the sum of the cost between the source and V and the cost between V and W. Let it be C. Same as in RFC 1584.

If all the following conditions are met:

- o V has a TRUE resource flag
- o if V is a router vertex, the resource on the link satisfies the requirement (the sum of available resource and existing reservation for the flow is equal to or greater than the requirement)
- o if W is a router vertex, the RES-LSA for W exists and is valid

the delay from the source to W is also calculated as the sum of the delay from the source to V and the delay of the link from V to W. Let this sum be delay D.

The delay of link L is 0 if V is a network vertex, otherwise it’s the delay metric from vertex V's RES-LSA. It is always in the forward direction.

b. If vertex W is not yet on the candidate list then install W on the candidate list and modify its parameters as described in RFC 1584. If the delay D is calculated in step A, record it in W’s delay state and set W’s resource flag to TRUE (step 5d).

c. Otherwise W is already on the candidate list and there are four possibilities:

- o W has a TRUE resource flag and D is NOT calculated in step 5a - W is already reachable via a path that has enough resource and this new path does not have enough resource - go to next link.

- o W has a FALSE resource flag and D is calculated - the old path does not have enough resource but the new one has enough so it should be used - modify W as in RFC 1584, set W’s resource flag to TRUE and record the delay D (step 5d).

- o W has a FALSE resource flag and D is not calculated - we are now building a best effort (partial) tree - process as in RFC 1584 - go to next link if the new path has higher cost, or modify W’s parameters (step 5d) if the new path should be used because of either lower cost or a tie-breaker.

- o W has a TRUE resource flag and D is calculated - process as in RFC 1584 but use delay instead of regular OSPF cost - go to next link if the new path has higher delay, or modify W’s parameters (step 5d) if the new path should be used because of either lower delay or a tie-breaker.

d. Same as in RFC 1584, plus recording the delay value D and setting the resource flag to TRUE when necessary.

6. go to step 3.

After the tree for area A is built, the calculating router determines if area A is used to determine the upstream node in the same way as described by RFC 1584. If the router is an ABR and area A is the “source area” for the flow, a DABRA is also originated to advertise all area border routers that are on the tree and have a TRUE resource flag. It is flooded to all "downstream areas"[6].

4.2 Unicast QOSPF

In terms of adding to and moving from the candidate list, unicast QOSPF Dijkstra is very similar to multicast QOSPF so the Dijkstra details are not discussed here.

4.2.1 Unicast QOSPF Dijkstra is needed in only one area

If the calculating router has multiple areas, then the best effort route to the destination has to be found first to identify the area that needs to run the Dijkstra:

1. If the route is an intra-area route, then the area that the route belongs to needs to run the Dijkstra to find a QoS route to the destination network.
2. If the route is an inter-area route, then backbone area needs to run the Dijkstra to find a QoS route to one of the ABRs that advertises the best effort route.
3. Suppose the route is an external route. If the ASBR used by the external route is within one of the router's directly attached areas, then that area needs to run the Dijkstra to find out a QoS route to the ASBR; otherwise, backbone area needs to run the Dijkstra to find out a QoS route to one of the ABRs that advertise the ASBR.

Unlike best-effort Dijkstra, a complete tree for the area is not needed. Once the shortest path to the destination network or the ABR or the ASBR is found, the Dijkstra terminates.

4.2.2 Inter-area and Inter-AS Unicast QOSPF

In the case that the destination is not in a directly attached area, things are more complicated because OSPF areas hide detailed topology and network resource information. Using the topology in Figure 4 again; when R1 calculate a QoS route for (H, H2), it finds a QoS route to ABR R2 that has a shortest best-effort route the destination, but R2 can not find a QoS route to the destination. R3 has a QoS route to the destination but the QoS route from R1 to R3 was not calculated.
One way to solve the problem is let R2 send a "summary" to area 0.0.0.0 indicating that it does not have a QoS route for the particular flow, so R1 will try to find a QoS route to R3. A router should send the summary to each area that it sends the Type 3 Summary LSAs for the destination network. However this may not be good idea because there would be a large number of such summaries.

4.3 QOSPF Dijkstra Recalculation

[6] Assuming Explicit Routing is not used. See Section 6.3.

Recalculation occurs upon one or more of the following situations:

- New instances of conventional OSPF/MOSPF LSAs, namely Network-LSAs, Summary-LSAs, AS External LSAs and Group-Membership-LSAs in multicast case - some or all QOSPF routes need to be recalculated (see MOSPF protocol spec for details in multicast case).
- New instances of RRAs, and DABRAs in multicast case. Only the QOSPF routes related to the RRAs and DABRAs need to be recalculated.
- New instances of RES-LSAs - All QOSPF routes need to be recalculated.

5.0 QOSPF Route Pinning

Route Pinning means that once reservations on a route from a source to a destination have been made, the route will not be replaced with a better route, unless the original one is no longer usable. Therefore, a pinned path may not continue to be the shortest path. Control over route pinning can be from a number of sources, such as configuration, flags from a signaling protocol or other administrative controls.

Because Resource Reservation Advertisements describe existing reservations, the route pinning algorithm can be accomplished with a simple modification to the QOSPF Dijkstra algorithm:

When the Dijkstra is run for a flow, if the links with existing reservations for the flow are preferred the original path is automatically preserved when possible. This will occur even if a new and better path is available.

Sometimes it is desirable that only part of a QoS distribution tree is pinned because it is possible to have some receivers that desire pinning and some that do not. This can also be easily achieved if RSVP or some dynamic mechanism can signal the desire for route pinning.

Suppose a router/host sends a RESV message to its previous hop router A, and it indicates in the RESV message that it wants the path to be pinned. Router A makes the reservation and notifies QOSPF that the path should be pinned. When A originates an RRA for the flow, it sets the P-bit (pin-flag) in the reservation for the link. When the route is recalculated, instead of preferring all links with reservations, only those links with “pinned” reservation are preferred, hence only part of the route is pinned.

Before the support from a signal protocol is available, a router simply sets the p-bit in its RRAs to indicate that route pinning should be used if it is configured so.

5.1 Route Pinning Dijkstra Modification

Their are two changes that are made to the QOSPF Dijkstra algorithm to implement route pinning.

5.1.1 Adding vertices to the candidate list

When adding a vertex to the candidate list, if its parent has a reservation for the flow on the link that leads to the vertex, and the reservation has the P-bit set in the RRA, the vertex is marked as "reserved"; or, if its parent is a network vertex marked as "reserved", it is also marked as "reserved".

If a neighbor W, of a vertex V that is just moved to the SPF tree, is already on the candidate list but not marked as "reserved", and it would not be updated in the normal Q/MOSPF Dijkstra, it still is updated if there is a reservation with the P-bit set for the flow on the link from V to W.

5.1.2 Moving a vertex from the candidate list to the SPF tree

Of those vertices with TRUE resource flags, a vertex marked as "reserved" is chosen with the smallest delay, even if there is an un-reserved vertex with a smaller delay. Vertices that are un-reserved are only moved to the SPF tree when there are no more "reserved" vertices on the candidate list.

In summary, vertices are moved from the candidate list to the SPF tree in the order of three preference groups: vertices with the "reserved" marks; vertices with the TRUE resource flags; and finally the rest best-effort ones.

6.0 Explicit Routing OSPF (EROSPF)

QOSPF needs both available resource information and existing resource reservation information in addition to the normal topology and membership information. When the size of a routing domain or the number of QoS data flows increases, there is a scaling problem because it takes a lot of bandwidth, memory and CPU power to flood, store and process the resource reservation information even though many of the routers may not be interested in the information.

To alleviate this scaling problem, Explicit Routing (ER) can be used: for a flow (source, destination) only the source router(s) (see Section 6.1.1 and Section 6.2) calculate a route, and then the forwarding information is distributed to the downstream routers along the path.

Because other routers do not need to perform the Dijkstra calculation, they are saved from this possible CPU-intensive computation. In the QOSPF case, the resource reservation information only needs to be kept on the source routers, thus saving bandwidth, memory, and CPU cycles. EROSPF is also applicable to standard MOSPF to reduce the computation needs of the transit routers.

6.1 Multicast Explicit Routing

The following discussion is in terms of a single area. In the multi-area case, each area maintains a forwarding table, and a global forwarding table comes from the merge of all the areas' forwarding tables.

6.1.1 Source Router Determination

The source router for a flow in an area is determined by one of the following conditions:

- the source of a flow is on a directly connected network within the area.
- the router is an ABR and the source is not in the area.

In other words, explicit routes are only calculated by the source router and the border routers that the flow travels through. It is very possible to have multiple source routers for a (source, destination) pair. In this case, each source router will calculate the tree separately, and then distribute forwarding information (i.e., its subtree) to the downstream routers on its subtree.

6.1.2 Explicit Routing Advertisements (ERAs)

The forwarding information for a (source, destination) pair is contained in an Explicit Routing Advertisement (ERA), which is passed in an Opaque LSA along the subtree described by the ERA. The passing scope is determined by information contained in the ERA.

There are two kinds of ERAs. One is an Installation-ERA, used to distribute forwarding information and the other is a Flushing-ERA, used to flush obsolete forwarding information.

6.1.2.1 Format of Installation-ERA

The Format of Installation-ERAs is shown in Figure 6:

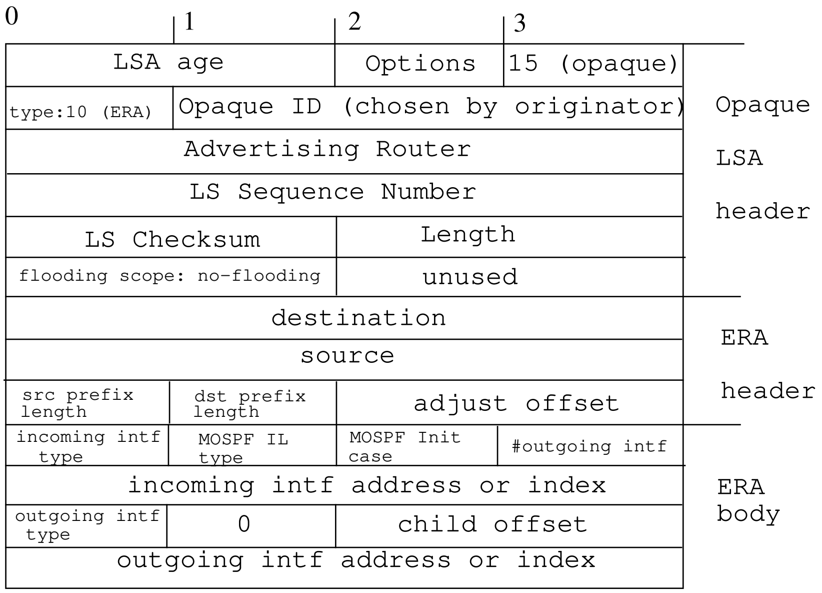


FIGURE 6. Format of Installation-ERA

ERAs are carried in Opaque LSAs with the Opaque type 10. The Opaque ID is chosen by its originator. The flooding scope is no-flooding, meaning that the receiver should not flood it out. However, when the receiver parses the ERA, it will build new ERA(s) off the received one and send out new ones with the same Opaque LSA header and ERA header (see Section 6.1.6).

Each ERA describes routers on a route tree. For each router, its incoming interface and a list of outgoing interfaces are listed. The interface type is the same as in OSPF Router LSAs. The interface is represented as one of the following:

- for a numbered interface, it is the ip address of the upstream (for incoming interface) or downstream (for outgoing interface) neighbor.
- for an unnumbered point-to-point interface, it is the interface index.

The offset fields (adjust offset and child offset) are used to encode the subtree into the ERA body, as explained in Section 6.1.3 and Section 6.1.6.

6.1.2.2 Flushing-ERA

A Flushing-ERA is used to flush a previously advertised Installation-ERA when the route changes (see Section 6.1.8). The flushing-ERA uses the MaxAge instance of the previously advertised ERA with an empty ERA body.

6.1.3 Creating Installation-ERAs

After a source router finishes a route calculation, it builds an ERA to encode the subtree that has the router itself as the root. The subtree is traversed in "preorder". In the example in Figure 7 (numbers are interface addresses or indices), the source router A will build an ERA listing routers in the order of A,B,D,E,C.

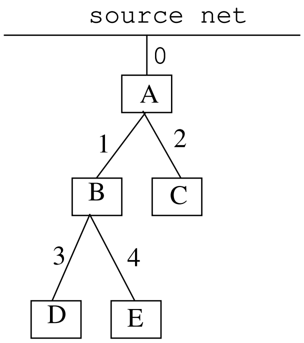
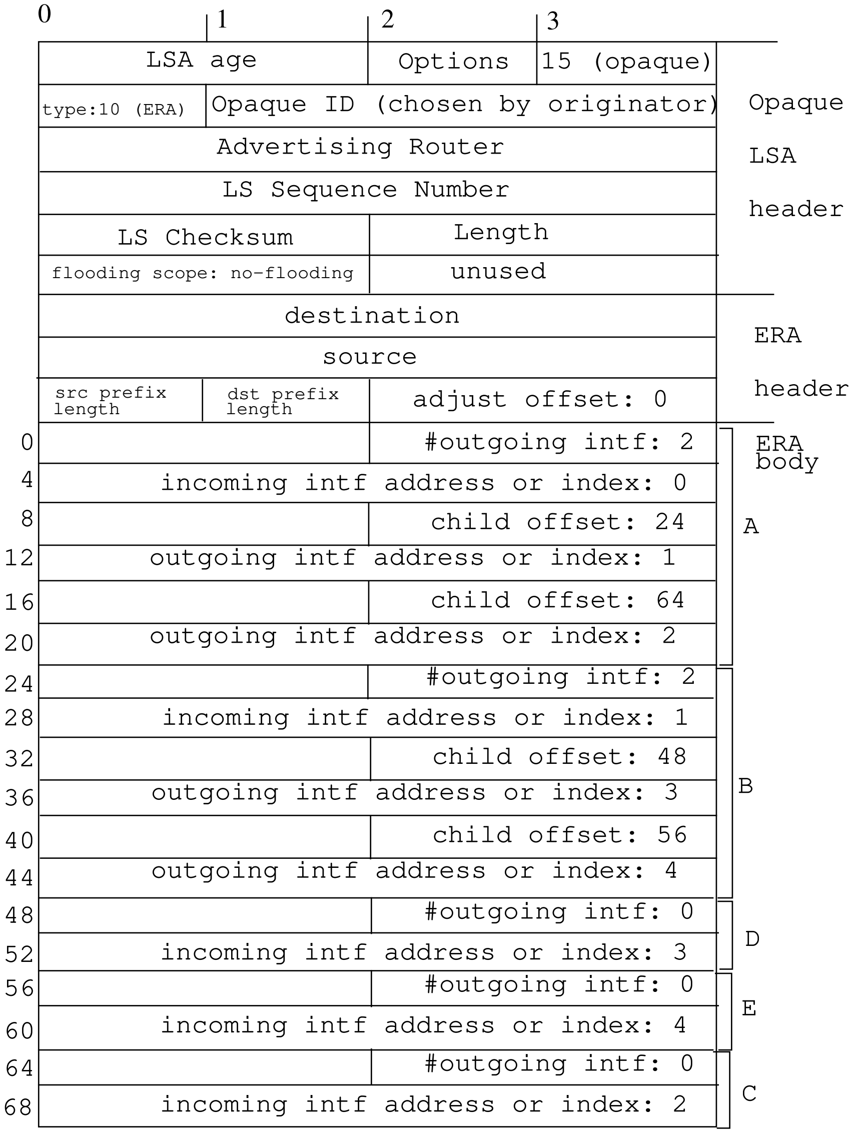


FIGURE 7. An example

The "adjust offset" is set to 0 by the source router. Except for the first router placed into the ERA, when a router is added to the ERA, the "child offset" of the parent's outgoing interface leading to the router is set to the offset of the router in the ERA body. *Note that all offsets are relative to the ERA body.* After building the whole ERA, the source router builds one ERA for each subtree under itself and unicasts the ERA to the root of the subtree, which is the first router listed in the ERA. For example, router A will build an ERA for the subtree rooted at B and unicast it to B, and build an ERA for the subtree rooted at C and unicasts it to C. This building process is pretty simple and is described in Section 6.1.6. However, the source router only stores the ERA for the whole tree and not newly built ERAs. The ERA for the subtree rooted at A is shown in Figure 8.



```
0                    1                    2                    3
+--------------------------------------+--------------+----------------+
|                LSA age               |   Options    |  15 (opaque)   |
+------------------+-------------------+--------------+----------------+  Opaque
| type:10 (ERA)    |     Opaque ID (chosen by originator)              |
+------------------+---------------------------------------------------+  LSA
|                      Advertising Router                              |
+----------------------------------------------------------------------+  header
|                      LS Sequence Number                              |
+--------------------------------------+-------------------------------+
|             LS Checksum              |            Length             |
+--------------------------------------+-------------------------------+
|     flooding scope: no-flooding      |            unused             |
+--------------------------------------+-------------------------------+
|                          destination                                 |  ERA
+----------------------------------------------------------------------+
|                            source                                    |  header
+------------------+-------------------+-------------------------------+
| src prefix       | dst prefix        |       adjust offset: 0        |
| length           | length            |                               |
+------------------+-------------------+-------------------------------+
0  |                                      |      #outgoing intf: 2        |  ERA
4  |          incoming intf address or index: 0                           |  body
8  |                                      |       child offset: 24        |
12 |          outgoing intf address or index: 1                           |  A
16 |                                      |       child offset: 64        |
20 |          outgoing intf address or index: 2                           |
24 |                                      |      #outgoing intf: 2        |
28 |          incoming intf address or index: 1                           |
32 |                                      |       child offset: 48        |
36 |          outgoing intf address or index: 3                           |  B
40 |                                      |       child offset: 56        |
44 |          outgoing intf address or index: 4                           |
48 |                                      |      #outgoing intf: 0        |  D
52 |          incoming intf address or index: 3                           |
56 |                                      |      #outgoing intf: 0        |  E
60 |          incoming intf address or index: 4                           |
64 |                                      |      #outgoing intf: 0        |  C
68 |          incoming intf address or index: 2                           |
```

FIGURE 8. The ERA for the subtree in Figure 7

6.1.4 Using Multiple ERAs for Long Routes

The structure and processing of the ERA allows the router computing the route to encode as much of the route as can fit in a packet. The source router can send an ERA to a downstream router that is not an immediate neighbor providing the subtree that continues from the downstream router. It is not likely that this facility would be used often in many networks.

6.1.5 Transmitting, acknowledging, and storing of ERAs:

A source router stores in its database ERAs (together with their Opaque LSA header) for trees with itself as the root. An ERA built for an immediate downstream neighbor is unicast to the incoming interface of the first router in the ERA (the first router in the ERA is always the receiver), encapsulated in an Opaque LSA.

A router also stores in its database ERAs received from its parents, but not those ERAs built for its downstream neighbors.

The acknowledgment and retransmission mechanism is the same as that used for conventional LSAs. Since the transmission and acknowledgment of OSPF LSAs are between adjacent neighbors while sometimes ERAs and DABRAs need to be sent to non-adjacent routers, a special pair of update/ack packets are needed for ERAs for DABRAs. See Section 7.2

6.1.6 Processing of Installation-ERAs:

The first listed router in a received ERA is always the receiver itself.

Upon ERA receipt, the forwarding entry for a (source, destination) pair is installed (or updated) and associated with the ERA.

If there is a previous instance of the Installation-ERA, to each immediate downstream neighbor listed in the previous instance of the ERA but not in the new ERA, send a Flushing-ERA with the same header as that of the previous instance.

For each immediate downstream neighbor listed in the received ERA, a new ERA is constructed from the received ERA and sent to the incoming interface of the first listed router in the newly constructed ERA. The Opaque LSA header and the ERA header remain the same, however. The new ERA's "adjust offset" is set to the "child offset" associated with the outgoing interface in the received ERA that leads to the neighbor. The child offsets are not changed in the new ERA. The subtree for the neighbor is then copied into the new ERA. The subtree is in the following range of the RECEIVED ERA BODY:

```
  [child offset - old adjust offset, next child offset - old adjust offset]
```

If there is no "next child", then the remaining portion of the ERA body is copied. Notice that the encoding work done by the source, and the offset fields make the downstream routers' job a matter of copying and shifting.

In the example in Figure 7, A will build two ERAs from the ERA for itself, one for B and the other for C. The two ERAs are illustrated in Figure 9.

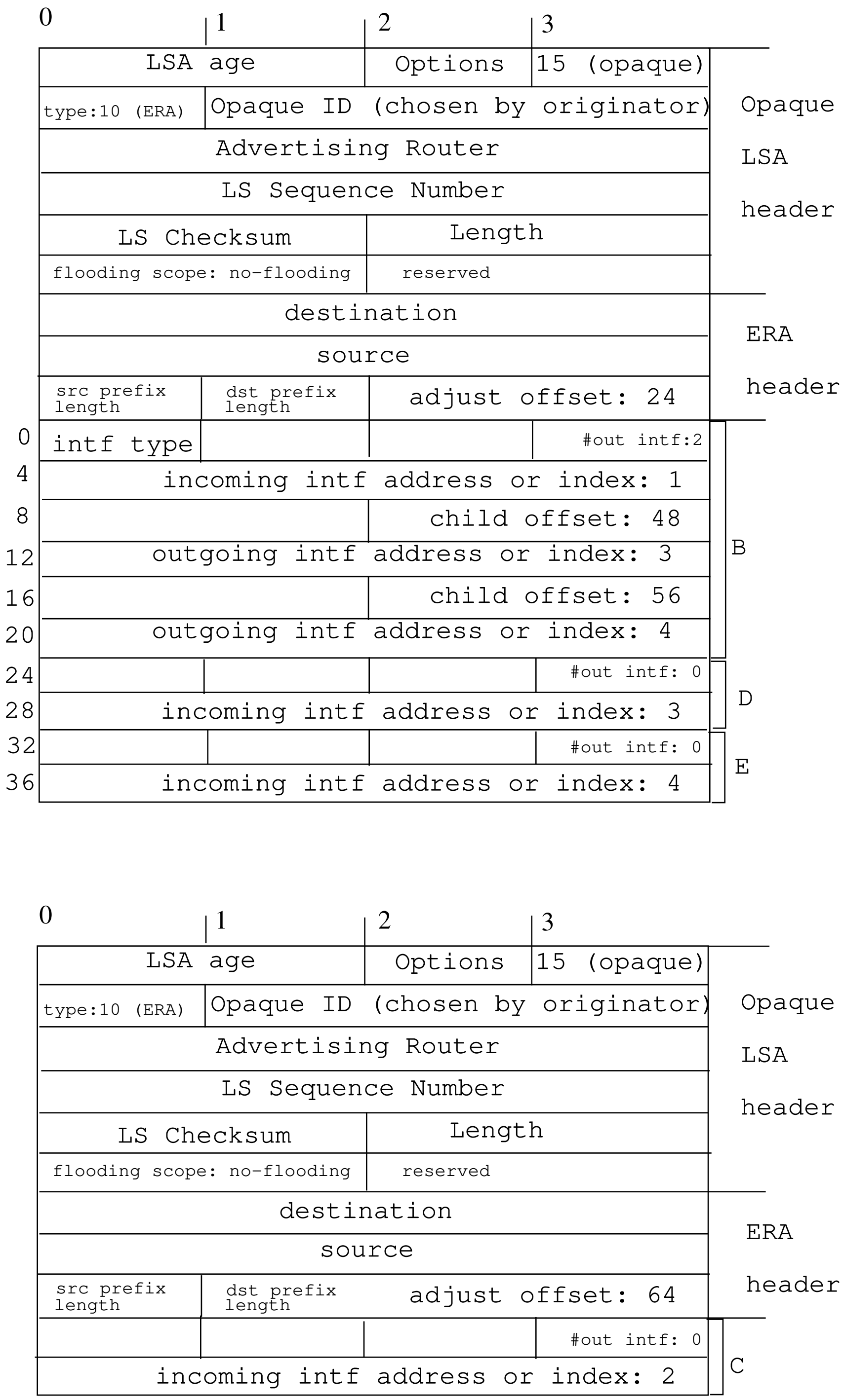


FIGURE 9. ERAs built by A for B and C

6.1.7 Processing of Flushing-ERAs

Upon receipt of a Flushing-ERA, the corresponding Installation-ERA is found and a MaxAge Flushing-ERA is constructed and sent out with same header as the existing ERA for each immediate downstream neighbor in the Installation-ERA. If a forwarding entry exists for the corresponding Installation-ERA, the forwarding entry’s incoming interface is set to NULL (so that no packets for the (source, group) will be accepted on the interface) if there are no other Installation-ERAs for the (s, g). If other Installation-ERAs exist, a new forwarding entry is constructed for the (source, group) pair. If there is no forwarding entry for the (source, group), forwarding entry with a NULL incoming interface is installed to prevent forwarding of any received packet for the (source, group) pair.

6.1.8 Route Change

For all routers, if the upstream neighbor or interface of the first router in an ERA goes down, a MaxAge Flushing-ERA is immediately sent to each immediate downstream neighbor to flush the ERA. This does not need to wait until the source finishes recalculation.

When there is a topology change, the source routers recalculate the tree, and send updated ERAs along their subtrees. New ERAs are carried in the Opaque LSAs with the same Opaque ID as in the old ones, but with a larger sequence number.

For all routers, if a previous downstream neighbor is no longer listed in a newer ERA, a Flushing-ERA with the same header of the previous instance of the new ERA is sent to the neighbor to flush its corresponding Installation-ERA.

6.2 Unicast Explicit Routing

While Multicast ER makes sense even if QOSPF is not used, Unicast Explicit Routing is needed only for QoS routing.

A router is a source router for a unicast flow (source, destination) when one of the following conditions exists:

- The source is on one of the router’s directly connected networks in the area that needs the Dijkstra, or,
- The source is not in the area, and the router is an ABR.

The multicast ERA is also used for unicast, but in the unicast case, the “MOSPF IL Type”, “MOSPF Init Case”, and incoming interface are not used, and the number of outgoing interface is always 1.

6.3 Changes of behavior of QOSPF if Explicit Routing is used

Explicit Routing is introduced to address QOSPF’s scaling problem[7], but QOSPF does not logically depend on Explicit Routing. The discussions in Section 3.0 and Section 4.0 have been assuming that no ER is used. When ER is used, the following behaviors of QOSPF are changed:

[7] It can also greatly reduce MOSPF’s calculation burden.

6.3.1 Flooding scope of RRAs

RRAs are no longer flooded throughout an area. Instead, a RRA is sent to the source router that advertised the explicit route (branch) to the originator or the RRA. If the source router is on the source network in the same area, it then uses “link-local” scope to flood the RRAs to other routers on the source network.

6.3.2 Flooding scope of DABRAs

DABRAs are no longer flooded throughout “downstream areas” of the “source area”. Instead, a DABRA is sent to all the ABRs on the route in the “source area”.

6.4 Quick Scaling Performance Analysis

The Scaling problems with QOSPF are primarily caused by RRAs, so let’s do a scaling analysis in terms of number of RRAs flooded per second, based on the following area configuration:

Number of routers in the area:	R
Average number of routers on a multicast tree:	M = abs(sqr_root(R))
Average number of flows that sources from a router:	F
Period of time during which to set up all the flows:	T = 10 seconds

For each flow, each router has to originate a RRA, so there will be (R * M * F) RRAs originated.

If explicit routing (ER) is not used, each router will get all the RRAs, so the R routers will receive (R * F * M - F) RRAs (a router does not need to receive its own RRAs), i.e, (R * F * M - F)/10 RRAs have to be transmitted per second.

If ER is used, only the source routers will receive the RRAs. Assuming those RRAs are sent to the source router following the reversed multicast path, then at most[8] (1 + 2 +,,, + (M - 2) + (M - 1)) transmissions are needed for each flow, or F * (1 + 2 +... + (M -2) + (M - 1))/10 RRAs have to be transmitted per second.

Changing the value of R, we have the following result:

Number of routers (R):	9	16	25	36	49	64
Number of RRAs per sec w/ ER:	0.3F	0.6F	1.0F	1.5F	2.1F	2.8F
Number of RRAs per sec w/o ER:	2.6F	6.3F	9.9F	21.5F	34.2F	51.1F

It is clear that QOSPF does not scale without ER but it scales well with ER.

7.0 Changes to OSPF to accommodate QOSPF/ER

Because of the new functionality and new types of LSAs, the following changes are needed to accommodate QOSPF or ER.

[8] when the multicast tree degrades to a line.

7.1 The Options field

The Options field in OSPF Hello, Database Description packet and all LSAs indicates what optional capabilities a router supports.

A new bit must be added to the Options field: the Q-bit. If set, it means the router supports QOSPF and understands RES-LSAs. The Q-bit matters only in Database Description packets and Router LSAs.

*	Q	DC	EA	N/P	MC	E	*

When a router exchanges its database with a neighbor, it only sends the neighbor those types of LSAs that the neighbor understands. If the neighbor does not set the Q-bit in its Database Description packets, the router should not include RES-LSAs in its Database Description packets and LS Update packets.

QOSPF Dijkstra should not be used if there is at least one router that does not support QOSPF in an area. This is indicated by the existence of a valid Router-LSA with the Q-bit cleared in the Options field.

However, note that if all multicast-capable routers supports QOSPF, then the QOSPF Dijkstra for multicast can still be used.

7.2 New Types of OSPF packets

OSPF requires that any LSAs be exchanged between neighbors that are supposed to become adjacent and a Link State Update/Ack packet would simply be discarded if it is from a neighbor with a state less than ExStart. However, when ER is used, the RRAs and ERAs may be sent to non-adjacent routers. The solution is to invent a new pair of update/ack packets that do not require adjacency to transmit/acknowledge RRAs and ERAs when ER is used. The same acknowledgment/retransmission scheme as those between adjacent neighbors can be used to ensure reliable transmission of RRAs and ERAs.

8.0 Security Considerations

Given that QOSPF could be triggered by RSVP, it is expected that the security mechanisms for RSVP will provide authorization and access control for QOSPF routing calculations. Additionally, the OSPF security mechanisms for authenticating neighbors and data received are very important for explicit routing since ER packets can change forwarding state in a very direct manner. Especially, since an ERA can be sent to a router on a different network, ERA packets' authentication should be per area instead of per interface.

9.0 Acknowledgments

The authors gratefully acknowledge the following people/organizations for making this protocol come together:

- Tim Trapp of Thompson International for the initial problem, constraints, as well as constructive discussions.
- E-Systems, Inc. Particularly, Hai Nguyen, Gerry Rosen, and Thomas Grill for their patience and perserverence during some of the difficult design and development phases.
- John Krawczyk, Ross Callon, Mohd Bashar, Mike Davis, Ambrose Kwong, Billy Ng and Dennis Baker for useful design comments.
- The IP group and Multimedia group at Bay Networks for lots of coding and debugging support.

10.0 Notice Regarding Intellectual Property Rights

```
Bay Networks may seek patent or other intellectual property protection for
some or all of the technologies disclosed in this document. If any standards
arising from this disclosure are or become protected by one or more patents
assigned to Bay Networks, Bay Networks intends to disclose those patents and
license them on reasonable and non-discriminatory terms. Future revisions of
this draft may contain additional information regarding specific intellectual
property protection sought or received.
```

11.0 References

1. J. Moy, *OSPF Version 2*, Request for Comments (RFC) 1583
2. J. Moy, *Multicast Extensions to OSPF*, Request for Comments(RFC) 1584, March 1994.
3. R. Coltun, *The OSPF Opaque LSA Option*, Internet Draft, draft-coltun-ospf-opaque-01.txt
4. R. Braden, L. Zhang, S. Berson, S. Herzog, S. Jamin. *Resource ReSerVation Protocol (RSVP) - Version 1 Functional Specification*, Internet Draft, draft-ietf-rsvp-spec-12.txt, May 1996.
5. J. Wroclawski, *Specification of the Controlled-Load Network Element Service*, Internet Draft, draft-ietf-intserv-ctrl-load-svc-01.txt, November, 1995.

12.0 Authors' Address

```
Zhaohui (Jeffrey) Zhang
Bay Networks, Inc.
2 Federal Street
Billerica, MA 01821
+1 508-670-8888
zzhang@baynetworks.com

Cheryl Sanchez
csanchez@baynetworks.com

Bill Salkewicz
bills@redbacknetworks.com

Eric S. Crawley
Gigapacket Networks, Inc.
25 Porter Road
Littleton, MA 01460
508-486-0665
```

esc@gigapacket.com