QoS Routing Extensions to OSPF

Jani Lakkakorpi
jani.lakkakorpi@nokia.com

Abstract

This paper briefly describes the extensions suggested to the OSPF [1] protocol to support QoS routes. These extensions are proposed in RFC 2676 [2]. This RFC presents some algorithms to compute QoS routes, and the necessary modifications to OSPF to support this function (the information needed, its format, how it is distributed, and how the QoS path selection process uses it). In addition, description from a reference implementation [3] along with some performance information is included.

1 Introduction

The most common routing protocols very rarely use routing metrics such as reliability, delay or bandwidth to calculate the optimal routes. They usually use only metrics such as the number of hops between the origination and the destination. This means that these protocols are not able to provide routes according to the loss, delay, and delay variation requirements of the served applications [4].

Link State routing protocols can support a variety of different routing metrics without altering the time to undertake the computation across the topology. Theoretically, it is possible to use several different metrics within the same topology. Routing decisions can be undertaken for example on the basis of various Quality of Service (QoS) attributes. Such attributes could be at least delay, throughput and reliability [5].

2 Open Shortest Path First (OSPF)

Open Shortest Path First (OSPF) is a widely used IP interior routing protocol that has been an Internet standard for awhile [3]. OSPF belongs to the family of Link State routing protocols (versus Distance Vector protocols). OSPFv2 is documented in RFC 1583, which is now replaced with RFC 2178. The basic operation of OSPF is briefly described in the following section. These same rules apply (in some extent) to other Link State protocols, too.

2.1 Basic Operation

OSPF is a protocol with a wide range of configurable metrics. This type of routing protocol requires each router to maintain at least a partial map of the network. When a new router joins a network, it has to learn the identity of all its neighboring routers. When this is done, each router will construct a message containing the identities and costs of the links attached to that particular router. These messages are called Link State Advertisements (LSA). Whenever a link changes its state, an LSA is flooded throughout the network. All the routers will notice the change, and recompute the routes accordingly.

Each router will save the most recent LSA from every router in the network; now each router knows the current network topology and is able to compute the shortest routes. Shortest paths to all other nodes in the network are computed with Dijkstra's algorithm [5]. The basic operation of the algorithm is illustrated in Figures 1 to 8. Table 1 is the resulting forwarding table for router A.

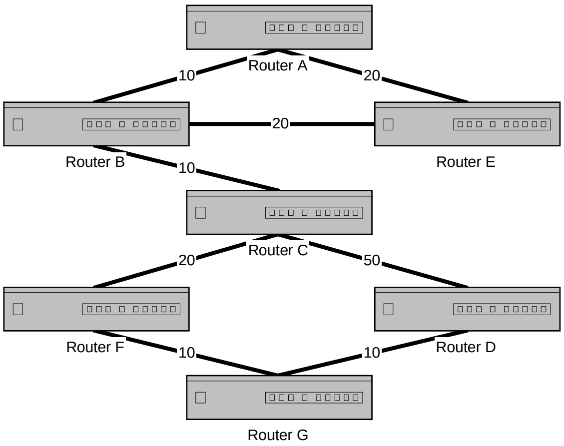

Figure 1. Example network

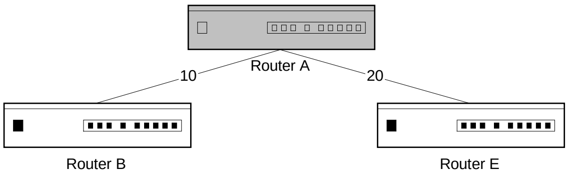

Figure 2. Start from router A and its LSAs

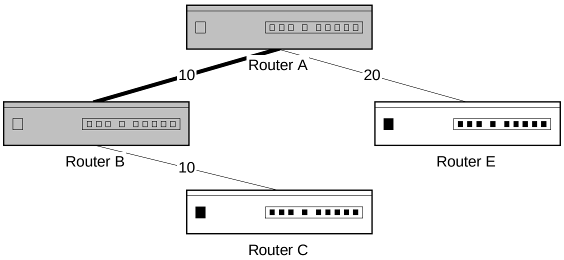

Figure 3. Select router B, and add its LSAs

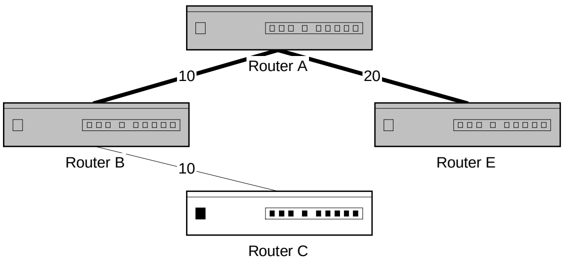

Figure 4. Select router E, no better path to router C

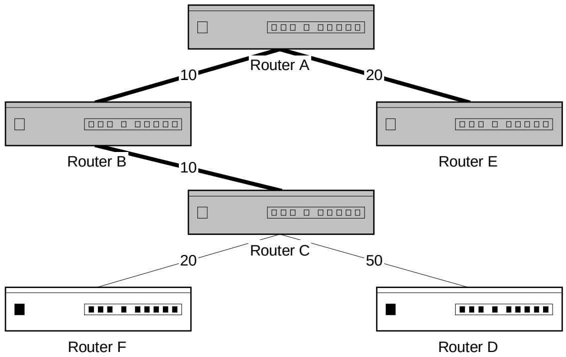

Figure 5. Select router C, and add its LSAs

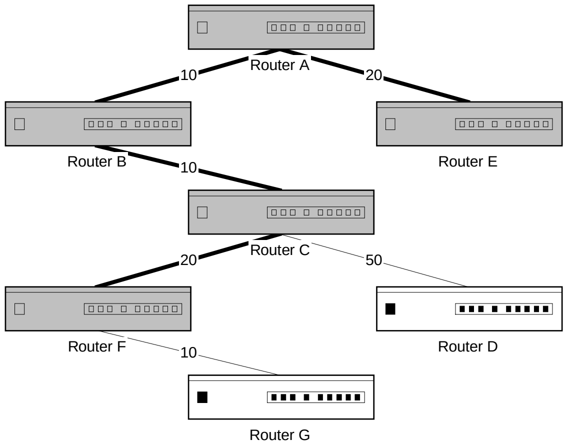

Figure 6. Select router F, and add its LSAs

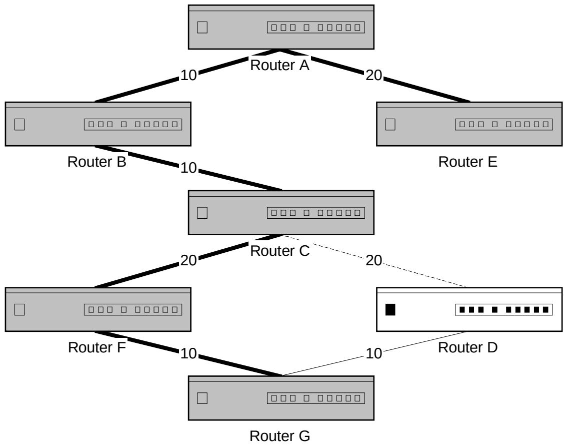

Figure 7. Select router G, and add its LSAs. Better path to router D found.

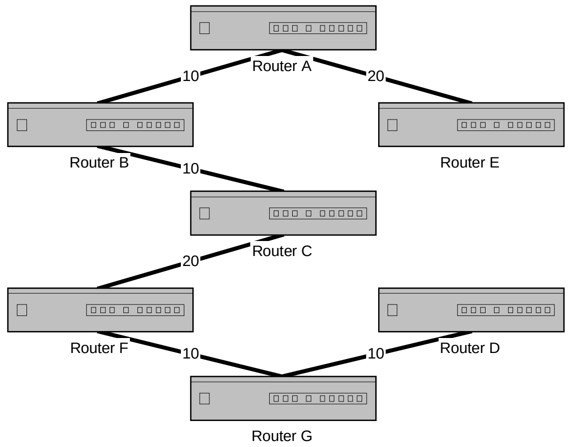

Figure 8. Select router D

Table 1: Forwarding table for router A

Destination	Link	Metric
A	Local	0
B	1	10
C	1	20
D	1	60
E	2	20
F	1	40
G	1	50

In steady state (no links or routers going out of service), the only OSPF routing traffic is periodic Hello packets between neighboring OSPF routers and the occasional refreshes of the link-state database. Hello packets are usually sent every ten seconds, and failure to receive Hello messages from a neighbor means that there are problems either in the connecting link or in the neighboring router. Every 30 minutes, an OSPF router refloods its contribution for the link-state database, just in case those pieces have been lost from or corrupted in some other routers database [1].

3 QoS Extensions to OSPF

IP has support for five different Types of Service (TOS) for datagram delivery. These types are normal service, minimize monetary cost, maximize reliability, maximize throughput, and minimize delay. An application decides which TOS will be applied to its datagrams by setting the TOS field in IP headers. Now it is possible for the routers to route IP packets to a common destination via different routes according to their TOS [1].

TOS-based routing has recently been dropped from the OSPF specification due to lack of deployment. However, to assure backward compatibility, routers are still able to advertise TOS specific metrics in their LSAs. This gives an opportunity to test QoS routing as an extension of the standard OSPF TOS features.

The QoS routing extensions to OSPF are based on two main ideas: first, the LSAs and the topology database have to include network resource information such as available bandwidth, and second, the route computation algorithm has to take this information into account.

3.1 QoS - Optional Capabilities

OSPF Hello Packets, Database Description packets and all LSAs include OSPF options field, which enables OSPF routers to support optional capabilities, and to advertise their capability level to other routers [2]. With the help of this mechanism, routers of different capability levels can communicate within an OSPF routing domain. The OSPF standard [1] used to specify the following bits in the options octet:

Table 2: OSPF options octet

*	*	DC	EA	N/P	MC	E	T

The least significant bit, T-bit, that was used for indicating TOS routing capability has now been removed [2]. However, as it was earlier mentioned, this information can be included in the options field for backward compatibility.

It is proposed in [2] that the T-bit should be used as an indicator of router's QoS capability and renamed as the Q-bit. In Hello packets this bit indicates whether the router is capable of supporting QoS routing or not. When this bit is set in a router or summary links LSA, it indicates that the packet contains QoS fields. In a network LSA this bit indicates whether the network described in the advertisement is QoS capable or not. This approach should be implemented quite carefully so that any old style (RFC 1583 based) implementations would not be confused.

3.2 Encoding Resources as Extended TOS

Since QoS extensions to OSPF should ideally be compatible with existing OSPFv2 routers, extensions in packet formats should be defined so that they are understood, ignored, or gracefully misinterpreted by OSPFv2 routers [2]. Encoding of QoS metrics in the TOS field makes it possible to mimic this new feature as extended TOS capability. These definitions should be either disregarded or considered unspecified by OSPFv2 routers.

Through the use of the fifth bit in TOS fields, there are 32 different combinations available for QoS resources. Since the TOS field was originally defined as being four bits long, this definition should not conflict with existing values. Because some implementations might not take all bits in the TOS field into account, the values of bandwidth and delay (which are specified as of today - link reliability and jitter may be defined later) are mapped onto 'maximize throughput' and 'minimize delay' if the most significant bit is discarded. Tables 3 and 4 represent TOS and QoS in OSPF according to [2].

Table 3: OSPF encoding of RFC 1349 TOS values

Decimal representation	Binary representation	Meaning
0	0000	Normal service
2	0001	Minimize monetary cost
4	0010	Maximize reliability
6	0011	
8	0100	Maximize throughput
10	0101	
12	0110	
14	0111	
16	1000	Minimize delay
18	1001	
20	1010	
22	1011	
24	1100	
26	1101	
28	1110	
30	1111	

Table 4: OSPF encoding of RFC 2676 QoS values

Decimal representation	Binary representation	Meaning
32	10000	
34	10001	
36	10010	
38	10011	
40	10100	Bandwidth
42	10101	
44	10110	
46	10111	
48	11000	Delay
50	11001	
52	11010	
54	11011	
56	11100	
58	11101	
60	11110	
62	11111	

3.3 Encoding Bandwidth

Since the actual metric field in OSPF packets only provides 16 bits to encode the bandwidth, linear representation is not feasible. The solution described in RFC 2676 [2] is exponential encoding using appropriately chosen implicit base value and a certain number of bits for encoding the mantissa and the exponent.

Given a base of eight, the three most significant bits should be reserved for the exponent part and the remaining 13 for the mantissa. This allows a simple comparison for these two numbers encoded in a form, which is useful during implementation.

An example may clarify things a bit: let us assume a link with bandwidth of 8 Gbits/s = 1024^3 Bytes/s. Its encoding would consist of an exponent value of six since $1024^3 = 4096 * 8^6$, which has a granularity of 8^6 (approx. 260 kBytes/s). The associated binary representation would thus be 1100 1000 0000 0000 or 53248_8. The bandwidth cost of this link, when idle, is the two's complement of the above binary representation, i.e., 0011 0111 1111 1111 which corresponds to a decimal value of $(2^{16} - 1) - 53248 = 12287$.

If we only had 1600 Mbits/s of available bandwidth on the link, the encoding of this bandwidth would be $6400 * 8^5$, which corresponds to a granularity of 8^5 (approx. 30 kBytes/s), and has a binary representation of 1011 1001 0000 0000 or decimal value of 47360. The advertised cost of the link with this load level, is then 0100 0110 1111 1111 or $(2^{16} - 1) - 47360 = 18175$. The cost function behaves as it should - the less bandwidth is available on a link, the higher the cost. In addition, the goal of better granularity for links with narrow bandwidth is also achieved. However, it should be noted that the figures given in the previous examples match exactly the resolution of the proposed encoding, which will not always be the case in real life. The standard practice is to round the available bandwidth values to the closest numbers. Because we are interested in the cost value, we choose to round costs up and thus bandwidth down.

3.4 Encoding Delay

Delay is encoded in microseconds using the same exponential method as described for bandwidth except that the base is defined to be four instead of eight. Thus the maximum delay that can be expressed is $(2^{13} - 1) * 4^7$ (approx. 134 seconds).

4 Reference Implementation

The QoS routing implementation of Apostopoulos, Guérin and Kamat [3] is based on the RFC 2676 "QoS Routing Mechanisms and OSPF Extensions" [2]. In this implementation of OSPF based QoS routing, link available bandwidth is the only metric extension implemented. In addition, QoS routes are computed only within a single OSPF area.

The QoS routing extensions were added to the OSPF implementation that is available in most Unix systems as a part of the Gate Daemon (GateD) program. GateD is a public domain program that provides a platform for implementing routing protocols on hosts running the Unix operating system. The GateD environment offers a variety of services useful for implementing a routing protocol. These services include:

- support for creation and management of timers
- memory management
- a simple scheduling mechanism
- interfaces for manipulating the host's routing table and accessing the network
- route management (route prioritization and route exchange between protocols)

RFC 2676 [2] includes some possible variations for QoS routing extensions that include on-demand computation and pre-computation of QoS routes as well as both explicit and hop-by-hop routing modes. The reference implementation in [3] performs path pre-computation and uses hop-by-hop routing mode. Path pre-computation was chosen because it can reduce processing costs. Hop-by-hop routing mode was preferred because it can be easily combined with RSVP, the signaling protocol that is going to be used to request QoS guarantees.

The reference implementation computes QoS paths with the widest-shortest path selection criterion, which is thoroughly described in [2]. A modified Bellman-Ford algorithm is used to pre-compute paths from a router to all other routers in the network. This algorithm computes paths of all possible bandwidth values for each destination, and builds a QoS routing table, which is kept apart from the basic OSPF routing table. This QoS routing table can be considered as a matrix, where a row corresponds to a destination, and column i corresponds to paths that are no longer than i hops away, and have the largest amount of bandwidth among all such paths to the destination. Thus a single matrix entry contains the next hop(s) and the available bandwidth on such paths. The information in this routing table is used to identify all paths that can satisfy the bandwidth requirements of a request. This is done by comparing the requested bandwidth to the available bandwidth in successive columns in the flow's destination row. The search is over when an entry with an available bandwidth larger than the requested one is found. At this point the corresponding next hop is returned and the request is forwarded to that address. If there are several possibilities for next hop, one of them is randomly chosen.

5 Conclusions

In RFC 2676 [2], the used metric of hop-count is extended with available link bandwidth and link propagation delay. The route selection is more emphasised on satisfying the bandwidth requirement, since the primary purpose of the delay metric is to identify high latency links and to avoid them. Additional experimental results on the implementation of QoS extensions to OSPF on the GateD platform are described in a [3]. These results show that the QoS extensions do not excessively increase the bandwidth requirements for route updates, or the cost of implementation.

6 References

[1] John T. Moy: OSPF: Anatomy of an Internet Routing Protocol, 3rd printing, September 1998, ISBN 0-201-63472-4.

[2] G. Apostopoulos, R. Guérin, S. Kamat, A. Orda, T. Przygienda, D. Williams: QoS Routing Mechanisms and OSPF Extensions, RFC 2676 (Experimental), August 1999.

[3] G. Apostopoulos, R. Guérin, S. Kamat: Implementation and Performance Measurements of QoS Routing Extensions to OSPF, Proceedings of INFOCOM'99, New York, NY, March 21-25, 1999.

[4] Marko Halme: Feasibility of IP Transport in Universal Terrestrial Radio Access Network, Master's Thesis, August 1999.

[5] Geoff Huston: ISP Survival Guide: Strategies for Running a Competitive ISP

[6] G. Apostopoulos, R. Guérin, S. Kamat, A. Orda, S. Tripathi: Intradomain QoS Routing in IP Networks: A Feasibility and Cost/Benefit Analysis, IEEE Network September/October 1999.

[7] W. Winer: Additional OSPF Extensions for Traffic Engineering and QoS Routing, Internet Draft, February 1999.