Quality-of-Service Routing

Antti Pietiläinen
Nokia Research Center
P.O. Box 422, FIN-00045 NOKIA GROUP

Abstract

Quality-of-service (QoS) routing is a natural consequence of emerging QoS services. The meaning of QoS routing and differences compared to today's routing are explained. Several individual aspects that closely relate to QoS routing are reviewed. Routing strategies and examples of new routing algorithms are considered. Finally, scaling and link layer considerations are shortly discussed.

Introduction

QoS-based services offer controlled delay, delay jitter, loss ratio, or bandwidth. Depending on the client, different combinations of the above properties are requested. The goal of QoS routing is to provide routing algorithms that are capable of identifying such paths so as to satisfy the maximum possible number of flows with QoS requirements [1]. Today's routing algorithms are mainly based on path length. Thus, for supporting QoS services new QoS routing algorithms must be developed.

Compared to today's routing algorithms there are some significant properties that have to be added and some properties that have to be changed: First, to support traffic using integrated-services class of services, multiple paths between node pairs will have to be calculated. Second, today's routing will shift traffic from one path to another as soon as a "better" path is found. The traffic will be shifted even if the existing path can meet the service requirements of the existing traffic. Frequently changing routes can increase the variation in the delay and jitter experienced by the end users. Third, today's optimal path routing algorithms do not support alternate routing. If the best existing path cannot admit a new flow, the associated traffic cannot be forwarded even if an adequate alternate path exists. [3]

In this article we first describe how QoS affects the routing information in routers. Then we review basic routing strategies and algorithms with respect to QoS. In the same part we address also multicast and reservation protocol. In the third part we consider scaling to a large network. Link layer aspects are discussed in part four.

1. Basic Requirements

As mentioned before, constraints such as delay and bandwidth can be used as link metrics in addition to cost. Thus, the network state could be portrayed as in Fig. 1 [3].

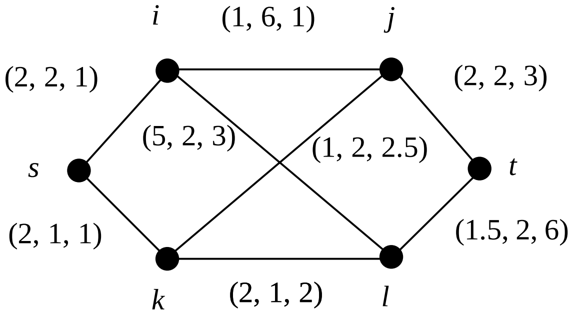


Figure 1: Network state. Link state = (bandwidth, delay, cost), after [3]

Here the link state is a triple consisting of residual (or unused) bandwidth, delay, and cost. The node delay and node bandwidth are included in the link parameters. The global state viewed in node *s* is presented in Table 1. Next hop, see Table 1, determines where to a packet is routed when it is transmitted from node *s*. As can be seen, the next hop that is chosen depends both on the destination and on the QoS constraint that is used to calculate the route.

Table 1: Global state in distance vectors at node *s*.

Destination	*i*	*j*	*k*	*l*	*t*
Bandwidth	2	1.5	2	2	1.5
Next hop	*i*	*k*	*k*	*i*	*i*
Destination	*i*	*j*	*k*	*l*	*t*
Delay	2	3	1	2	4
Next hop	*i*	*k*	*k*	*k*	*k*
Destination	*i*	*j*	*k*	*l*	*t*
Cost	1	2	1	3	5
Next hop	*i*	*i*	*k*	*k*	*i*

As mentioned in the introduction, a drawback today is that routing will shift traffic from one path to another as soon as a "better" path is found. A change in the route can repeat many times during a single flow and cause untolerable jitter. In a small network the changing of routes between nodes is a small problem as compared to large networks consisting of many autonomous systems (AS's). It is suggested [2] that in interdomain routing the QoS information should be relatively static, determined from the engineered topology and capacity of an AS rather than ephemeral fluctuations in traffic load through the AS. Ideally, the QoS supported in a transit AS should be allowed to vary significantly only under exceptional circumstances, such as failures or focused overload.

The third basic property required of QoS routing is the need of calculating alternate paths. A greedy multi-path routing algorithm has been proposed [4]. Each node computes multiple alternate paths to a given destination. The available capacity process, e.g., available bandwidth, available buffer, etc., is derived from the link state updates received at each node and the forwarding rule is based on some function of the maximum available capacity process applied to the set of possible nodes corresponding to the next hop on an alternate path for a particular source destination pair. A consequence of this path selection rule is the pooling of network resources when the load is increased.

2. Routing strategies, algorithms and other aspects to consider

There are three routing strategies: source routing, distributed routing and hierarchical routing [3]. They are classified according to how the state information is maintained and how the search of feasible paths is carried out.

2.1 Source routing

In source routing, each node maintains the complete global state, including the network topology and state information of every link. Based on the global state, a feasible path is locally computed at the source node. A control message is then sent out along the selected path to inform the intermediate nodes of their precedent and successive nodes. A link-state protocol is used to update the global state at every node. Source routing achieves its simplicity by transforming a distributed problem to a more centralized one. It avoids dealing with distributed computing problems. It guarantees loop-free routes. Source routing has several problems. First, the global state maintained at every node has to be updated frequently enough to cope with the dynamics of network parameters such as bandwidth and delay. This makes the communication overhead excessively high for large-scale networks. Second, the link-state algorithm can provide only approximate global state due to the overhead concern and nonneglible propagation delay of state messages. As a consequence, QoS routing may fail to find an existing feasible path due to the imprecision in the global state used [5]. The problems increase when the size of the network increases so source routing can not be scaled to large networks.

2.2 Distributed routing

In distributed routing, the path is computed by a distributed computation. Therefore, the routing response time can be made shorter, and the algorithm is more scalable. Searching multiple paths in parallel for a feasible one is made possible which increases the chance of success. Control messages are exchanged among the nodes, and the state information kept at each node is collectively used for the path search. Most distributed routing algorithms need a distance-vector protocol (or a link-state protocol) to maintain a global state in the form of distance vectors, see Table 1, at every node. Based on the distance vectors, the routing is done on a hop-by-hop basis.

Although the distributed algorithms which do not need global state information are more scalable than source routing algorithms, they tend to send more messages. This reduces the payload traffic's share of the bandwidth.

2.3 Hierarchical routing

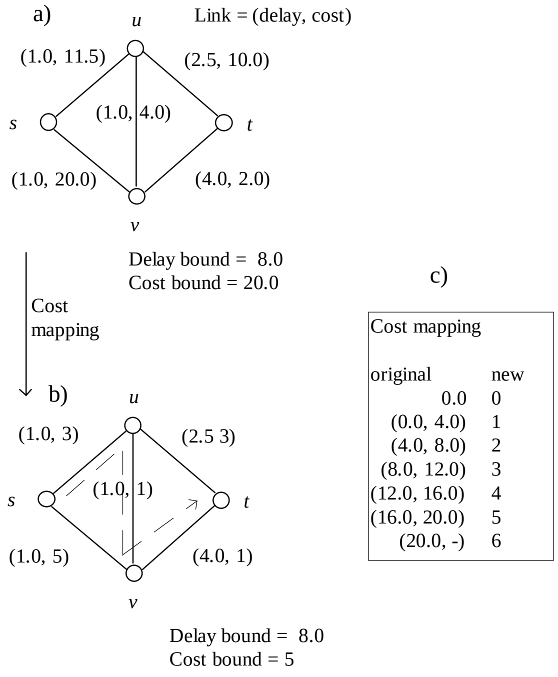
In hierarchical routing, nodes are clustered into groups recursively, creating a multilevel hierarchy. Each physical node maintains an aggregate global state. This state contains detailed state information about the nodes in the same group and aggregate state information about the other groups. Source routing is used to find a feasible path on which some nodes are logical nodes representing groups. A control message is then sent along this path to establish the connection. When the border node of a group represented by a logical node receives the message, it uses the source routing to expand the path through the group. A network where hierarchy is built in the above-mentioned way poses several advantages. First, hierarchical routing scales well because each node only maintains a partial global state where groups of nodes are aggregated into logical nodes. Second, since source routing algorithms can be used directly at each hierarchical level to find feasible paths based on the aggregate state, hierarchical routing retains many advantages of source routing. Third, it has also some advantages of distributed routing because the routing computation is shared by many nodes.

However, because the network state is aggregate, additional imprecision is introduced which has a significant negative impact on QoS routing [6].

2.4 Algorithms

New routing algorithms emerge all the time. We present a few examples.

Source routing algorithm:
Chen and Nahrstedt proposed a heuristic[1] algorithm for the NP-complete[2] multi-path-constrained routing problem [7]. If all metrics except one take bounded integer values, the multi-path-constrained routing problem is solvable in polynomial time. Consider delay-cost-constrained routing. The idea is to map the cost (or delay) of every link from an unbounded real number to a bounded integer. This reduces the original NP-complete problem to a simpler problem solvable in polynomial time. Let C be the cost requirement and x a small integer. The algorithm first maps the cost of every link to an integer bounded by $x+1$. Real numbers in $[0,C]$ are mapped into integers in $[0,x]$, real numbers in $(C,\infty]$ are mapped to $x+1$, and the cost bound C is mapped to x, see fig 2. The new problem with the link cost bounded by $x+1$ can be solved in polynomial time by an extended Dijkstra's algorithm or an extended Bellman-Ford algorithm. It was proved that a feasible path of the new problem must also be a feasible path of the original problem. The performance of the algorithm is tunable by choosing the value of x: a larger x results in higher probability of finding a feasible path and a higher overhead.



Cost mapping

original	new
0.0	0
(0.0, 4.0)	1
(4.0, 8.0)	2
(8.0, 12.0)	3
(12.0, 16.0)	4
(16.0, 20.0)	5
(20.0, -)	6

Figure 2: Chen-Nahrstedt heuristic: (*C* = 20.0 and *x* = 5): a) The original problem is to find a path from *s* to *t* such that the delay is bounded by 8.0 and the cost is bounded by 20.0; b) the costs of the links are mapped to integers in [1...6]. For link (*s*, *u*) the cost 11.5 is mapped to 3. The new problem is find a path from *s* to *t* such that the delay is bounded by 8.0 and the cost is bounded by 5. A feasible path is *s* → *u* → *v* → *t*, which, as expected is also a feasible path for the original problem. c) The cost-mapping table.

Distributed routing schemes:
Shin-Chou Algorithm [8]
Shin and Chou proposed a distributed routing scheme for establishing delay-constrained connections. No global state need be maintained at any node. The algorithm floods from the source toward the destination. Each message accumulates the total delay of the path it has traversed so far. When a routing message is received by an intermediate node, the message is forwarded only when one of the following two conditions is satisfied:

- It is the first such message received by the node.
- It carries a better accumulated delay than the previously received messages

If either condition is true, the message will be forwarded along the forwarded links whose delay plus the message's accumulated delay does not exceed the end-to-end delay requirement. Once a message reaches a

---

[1] Trial and error based
[2] A problem is NP-complete when there is at least one instance of the problem which cannot be solved in polynomial time.

destination, it finds a delay-constrained path which is the one that it has traversed. It was shown that when certain scheduling policies [9] are used and the routing messages are set to the appropriate priority, there will be at most one message sent along every link.

Chen-Nahrstedt Algorithm
Selective Probing [10]. Chen and Nahrstedt proposed a distributed routing framework based on selective probing. After a connection request arrives, probes are flooded selectively along those paths which satisfy the QoS and optimization requirements. Every node maintains its local state, based on which the routing and optimization decisions are made collectively in the process of probing. As in the Shin-Chou algorithm, each probe arriving at the destination detects a feasible path.

Algorithms were derived form the framework to route connections with a variety of QoS constraints on bandwidth, delay, delay jitter, cost, and their combinations. Several techniques were developed to overcome the high communication overhead of the Shin-Chou algorithm. First, probes are only allowed to be forwarded to a subset of outgoing links selected based on topological distance to the destination. Second, iterative probing is used to further reduce the overhead. At first iteration, the probes are sent only along the shortest paths. If the first iteration fails, probes are allowed to be sent along paths with increasing lengths in the following iterations. Simulation shows that with two iterations the Chen-Nahrstedt algorithm achieves substantial overhead reduction.

An example of a *hierarchical* routing algorithm is Private network network interface (PNNI), see Ref. [11]. Integrated PNNI [12] has been designed from the start to take advantage of the QoS Routing capabilities that are available in PNNI and integrate them with routing for layer 3 [2]. This would provide an integrated layer 2 and layer 3 routing protocol for networks that include PNNI in the ATM core. The I-PNNI specification has been under development in the ATM Forum and, at this time, has not yet incorporated QoS routing mechanisms for layer 3.

2.5 Relation to RSVP

It is important to understand the difference between QoS-based routing and resource reservation [2]. While resource reservation protocols such as RSVP [13] provide a method for requesting and reserving network resources, they do not provide a mechanism for determining a network path that has adequate resources to accommodate the requested QoS. Conversely, QoS-based routing allows the determination of a path that has a good chance of accommodating the requested QoS, but it does not include a mechanism to reserve the required resources.

Consequently, QoS-based routing is usually used in conjunction with some form of resource reservation or resource allocation mechanism [2]. Simple forms of QoS-based routing have been used in the past for Type of Service (TOS) routing [14]. In the case of OSPF, a different shortest-path tree can be computed for each of the 8 TOS values in the IP header [15]. Such mechanisms can be used to select specially provisioned paths but do not completely assure that resources are not overbooked along the path. As long as strict resource management and control are not needed, mechanisms such as TOS-based routing are useful for separating whole classes of traffic over multiple routes. Such mechanisms might work well with the emerging Differential Services efforts [16].

Combining a resource reservation protocol with QoS-based routing allows fine control over the route and resources at the cost of additional state and setup time. For example, a protocol such as RSVP may be used to trigger QoS-based routing calculations to meet the needs of a specific flow.

2.6 Multicast

The goals of QoS-based multicast routing are as follows [2]:

- Scalability to large groups with dynamic membership
- Robustness in the presence of topological changes
- Support for receiver-initiated, heterogeneous reservations
- Support for shared reservation styles, and
- Support for "global" admission control, i.e., administrative control of resource consumption by the multicast flow.

A receiver-oriented multicast routing model seems to have some advantage over multicast source routing. Under this model:

1. Sender traffic advertisements are multicast over a best-effort tree which can be different from the QoS-accommodating tree for sender data.
2. Receiver discovery overheads are minimized by utilizing a scalable scheme (e.g., PIM[1], CBT[2]), to multicast sender traffic characterization.

---

[1] Protocol Independent Multicast

[2] Core Based Trees

3. Each receiver-side router independently computes a QoS-accommodating path from the source, based on the receiver reservation. This path can be computed based on unicast routing information only, or with additional multicast flow-specific state information. In any case, multicast path computation is broken up into multiple, concurrent unicast path computations.
4. Routers processing unicast reserve messages from receivers aggregate resource reservations from multiple receivers.

3. Scaling to large networks

When the networks grow larger, the computational load required for routing increases. Source routing algorithms, for example, are not very scalable. Therefore, aggregation is carried out so that nodes in another autonomous system are combined to form a single logical node. Normally, the inaccuracy of the network state knowledge in hierarchical routing does not cause problems. However, as mentioned before, the imprecision has a significant negative impact on QoS routing.

The approach suggested in [2] is not to compute paths based on residual or instantaneous values of AS metrics (which can be dynamic), but utilize only the QoS capabilities engineered for aggregate transit flows. The engineering may be based on the knowledge of traffic to be expected from each neighboring ASs and the corresponding QoS needs. This information may be obtained based on contracts agreed upon prior to the provisioning of services. The AS metric then corresponds to the QoS capabilities of the "virtual path" engineered through the AS (for transit traffic) and a different metric may be used for different neighbors. This is illustrated in Fig. 3.

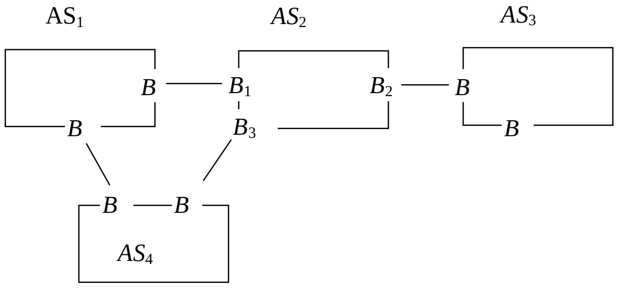


Figure 3: A large network that consists of autonomous systems (AS). *B* = border router, after [2]

Here, border router B_1 may utilize an AS metric specific for AS_1 when computing path metrics to be advertised to AS_1. This metric is based on the resources engineered in AS_2 for transit traffic from AS_1. Similarly, B_3 may utilize a different metric when computing path metrics to be advertised to AS_4. Now, it is assumed that as long as traffic flow into AS_2 from AS_1 or AS_4 does not exceed the engineered values, these path metrics would hold. Excess traffic due to transient fluctuations, however, may be handled as best effort or marked with a discard bit.

4. Link-layer considerations

To determine whether the QoS requirements of a flow can be accommodated on a link, a router must be able to determine the QoS available on the link. It is still an open issue as to how the QoS availability is determined for broadcast multiple access links (e.g., Ethernet). A related problem is the reservation of resources over such links. Solutions to these problems are just emerging [17].

Similar problems arise when a router is connected to a large non-broadcast multiple access network, such as ATM. In this case, if the destination of a flow is outside the ATM network, the router may have multiple egress choices. Furthermore, the QoS availability on the ATM paths to each egress point may be different. The issues then are,

- how does a router determine all the egress choices across the ATM network?
- how does it determine what QoS is available over the path to each egress point?, and
- what QoS value does the router advertise for the ATM link.

Typically, IP routing over ATM (e.g., NHRP[1]) allows the selection of a single egress point in the ATM network, and the procedure does not incorporate any knowledge of the QoS required over the path. An approach like I-PNNI [12] would be helpful here, although it introduces some complexity.

5. Conclusions

Currently, research on QoS routing is very intensive which can be seen from the fact that large portion of the references in this article are from the last year. It appears that QoS routing can be relatively easily implemented in a small network. In a larger network the

---

[1] Next Hop Routing Protocol

load caused by routing traffic and computation increases sharply and makes QoS routing more difficult. On the other hand, if the logical appearance of the network is simplified by aggregation of several nodes into one logical node, we end up with problems that are due to the imprecision of the network state. Therefore, more “circuit switched” routing may be feasible in the core network. Failing in QoS routing can lead to significant increase of unused high-quality capacity. Therefore, QoS routing will be an important research area.

References

[1] Guérin, R. A., Orda, A., and Williams, D.:QoS Routing Mechanisms and OSPF Extensions, Proceedings of 2nd Global Internet Miniconference (joint with Globecom'97), Phoenix, AZ, November 1997, p. 1903.

[2] Crawley, E., Nair, R., Rajagopalan, B., and Sandick, H.: A Framework for QoS-based Routing in the Internet, Request for Comments: 2386, August 1998, Internet Engineering Task Force.

[3] Chen, S. and Nahrstedt, K.: An Overview of Quality of Service Routing for Next-generation High-Speed Networks: Problems and solutions”, IEEE Network, Nov./Dec. 1998, p. 64-79 and references therein.

[4] Mithal, S.: Bounds on End-to-End Performance via Greedy, Multi-Path Routing in Integrated Services Networks, INFOCOM ’98, p. 19-26.

[5] Shaikh, A., Rexford, J., and Shin, K.: Dynamics of quality-of-service routing with inaccurate link-state information, U. of MI Tech. rep. CSE-TR-350-97, Nov. 97.

[6] R. Guerin and A. Orda: QoS-based Routing in Networks with Inaccurate Information: Theory and Algorithms, IEEE Infocom ’97, Japan, Apr. 1977.

[7] Chen, S. and Nahrstedt, K.: On Finding Multi-Constrained Paths, IEEE ICC ’98, June 1998.

[8] Shin, K. G. and Chou, C.-C.: A Distributed Route-Selection Scheme for Establishing Real-Time Channel, 6th IFIP Int’l. Conf. High Perf. Networking, Sept. 1995, pp. 319-329.

[9] Kandlur, D. D., Shin, K. G., and Ferrari, D.: Real-Time Communication in Multi-Hop Networks, Proc. 11th Conf. Dist. Comp. Syst., 1991, pp. 300-307.

[10] Chen, S. and Nahrstedt, K.: Distributed Quality-of-Service Routing in High-Speed Networks Based on Selective Probing, Tech. rep. Univ. of IL at Urbana-Champaign, Dept. Comp. Sci., 1998.

[11] ATM Forum PNNI subworking group: Private network network Interface Spec. v.1.0 (PNNI 1.0), afpnni-0055.00, March 1996.

[12] ATM Forum Technical Committee: Integrated PNNI (I-PNNI) v1.0 Spec. af-96-0987r1, Sept. 1996.

[13] Braden, R., Zhang, L., Berson, S., Herzog, S., and S. Jamin: Resource ReSerVation Protocol (RSVP) -- Version 1, Functional Spec", RFC 2205, September 1997.

[14] Ma, Q.: Quality-of-Service Routing in Integrated Services Networks, PhD thesis, Computer Science Department, Carnegie Mellon Univ. 1998.

[15] Postel, J.: Internet Protocol, STD 5, RFC 791, Sept. 1981.

[16] Black, D., Blake, S., Carlson, M., Davies, E., Wang, Z., and W. Weiss: An Architecture for Differentiated Services, work in progress.

[17] Ghanwani, A., Pace, J. W., Srinivasan, V., Smith, A., and Seaman, M.: A Framework for Providing Integrated Services over Shared and Switched IEEE 802 LAN Technologies, Work in progress.