NEW FRONTIERS IN TELECOMMUNICATIONS

2003 IEEE
International Conference on Communications

Volume 1

ICC 2003

Anchorage, Alaska
USA

11-15 May 2003
Anchorage Alaska, USA

ICC
GLOBECOM

IEEE COMMUNICATIONS SOCIETY

IEEE

2003 IEEE International Conference on Communications

Copyright © 2003 by the Institute of Electrical and Electronics Engineers, Inc.
All rights reserved.

Copyright and Reprint Permission

Abstracting is permitted with credit to the source. Libraries are permitted to photocopy beyond the limit of U.S. copyright law, for private use of patrons, those articles in this volume that carry a code at the bottom of the first page, provided that the per-copy fee indicated in the code is paid through the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923.

Other copying, reprint, or reproduction requests should be addressed to:
IEEE Copyrights Manager, IEEE Service Center, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331.

IEEE Catalog Number 03CH37441 (softbound)
03CH37441C (CD-ROM)

ISBN 0-7803-7802-4 (softbound)
0-7803-7803-2 (CD-ROM)

Library of Congress 81-649547

Additional copies of this publication are available from

IEEE Operations Center
P.O. Box 1331
445 Hoes Lane
Piscataway, NJ 08855-1331 USA

1-800-678-IEEE
1-732-981-1393
1-732-981-9667 (FAX)
email: customer.services@ieee.org

A Hierarchical Multipath Approach to QoS Routing: Performance and Cost Evaluation

Scott Seongwook Lee, Shirshanka Das, Giovanni Pau, and Mario Gerla
Computer Science Department,
University of California, Los Angeles, CA 90095-1596, U.S.A.
{sslee, shanky, gpau, gerla}@cs.ucla.edu

***Abstract*- Efforts to provide connection oriented service over the inherently best-effort Internet started almost right after its birth. Today, there exist a multitude of solutions that have been proposed but have never been implemented due to their impracticability. We propose a practical solution for fast, low cost, scalable, and yet accurate QoS routing. We propose to use hierarchical approaches to make the scheme practical and cost-effective. At the same time, we increase network utilization and decrease inaccuracy of stale information by the use of multiple paths. An extensive simulation of the various permutations of schemes over a large set of topologies and traffic conditions validate the proposed schemes and prove conclusively that hierarchical schemes with multiple path capabilities not only result in significantly lower overhead, but also give high levels of QoS performance. This paper presents the architecture of our schemes, the multiple path computation algorithm, and the simulation results validating our claims.**

I. Introduction

Substantial research contributions to provisioning QoS support in the Internet have been made by the research community in the last two decades developing features which span the entire framework ranging from the application layer down to the physical layer. However, few networks with QoS support capability have been deployed. In this paper, we address practical and specific issues involved in deploying QoS routing strategies.

The very first issue is *scalability* of QoS routing solutions in very large networks. Networks operated by most well known organizations easily comprise of several hundreds of nodes. In recent networks with link state approaches, it has been proposed to achieve scalability via hierarchical network composition. For both scalability and security reasons, networks consist of a hierarchy of subnetworks called domains (or subnetworks) which do not reveal details of their internal structure to the outside. This hierarchical property becomes a significant technical motivation to adopt QoS routing schemes proposed and developed so far. Thus, we propose hierarchical QoS routing schemes with the premise of *fast QoS provisioning* and *efficient network resource use*.

We also propose to use multiple QoS paths in each domain. The well-known benefits of provisioning multiple paths are *reliable QoS support* and *more evenly balanced network load*. In order to achieve these goals, we revise the scheme presented in [1] such that each subnetwork provisions multiple QoS paths. This QoS path provisioning is exercised transparently to the outside of each subnetwork. The mechanism is also accompanied with *QoS constraints quantization* for path precomputation which leads to fast QoS provisioning. Based on this fundamental mechanism development, we compose several different variants which are adherent to the hierarchical nature of network systems. These different schemes are primarily to show the fundamental benefits of hierarchical routing and also QoS-enabled routing. The detailed mechanisms are discussed in the following sections.

This paper is organized as follows. Section II discusses related works which are the foundation and groundwork of our approach. Section III explains the details of our proposed schemes and associated components. The proposed schemes are evaluated via simulation experiments and their properties are investigated in Section IV.

II. Related Works

Most QoS routing schemes are based on "source routing" approaches. That is, traffic sources compute feasible paths when QoS requests arrive. In turn, the computed paths are fixed using path pinning-down facilities such as MPLS [2]. However, there is another approach: distributed hop-by-hop QoS routing [3], [4], [5], [6]. But, distributed routing is known to be prone to loop problems. Thus, most QoS routing approaches follow source routing schemes, and the schemes that we are presenting also fall into the same category. In order to deploy a source routing, one must implement a link state type of routing algorithm (e.g., OSPF). With link state, every router in the network can maintain a local copy of the entire network topology and resource utilization information. This prevents the scheme from being scalable since the size of network information increases sharply as the network size grows. Thus, large networks are usually composed hierarchically of several domains (a.k.a. areas in OSPF networks), and hierarchical source routing is considered as the most promising scalable QoS routing approach [4], [7], [8]. Hierarchical approaches have been already used in ATM networks with the PNNI routing protocol [9].

In hierarchical QoS routing, network topology and resource information about a specific domain are summarized before being exchanged with other domains. This process is called topology aggregation [10], [11], [12], [13], [14]. With the summarized (i.e., aggregated) network information, traffic sources run QoS routing algorithms to compute a source route. The computed path is then explored with proper signaling to reserve network resources, possibly in combination with a crankback feature [15], [16]. Topology aggregation summarizes and compresses topology information. It is used when it is important to know how a network (a routing domain or OSPF area or PNNI peer group) can be traversed without having to know all the topology details. After the aggregation step, the same topology may be represented by a mesh, star, or spanning-tree network [15]. The challenge is to calculate the link metrics in such a way that the traversing characteristics remain the same or similar in the aggregated topology [13], [14]. When aggregated, some of the detailed network information can be inevitably lost due to the information abstraction.

0-7803-7802-4/03/$17.00 © 2003 IEEE

When a network uses a single additive metric (e.g., link weight or delay), the aggregated link metrics can be computed using a shortest path algorithm. Likewise, if a network uses available bandwidth as a link metric, then the aggregated link metrics can be computed using a widest path algorithm [3], [17]. However, if multiple link metrics such as available bandwidth and delay are used simultaneously, the traversing characteristics can no longer be represented using simple scalar numbers. The reason for this is that there usually exist multiple paths through the network, each of which has a lower delay or a higher bandwidth than the others. Consequently, there is a complete set of paths which are efficient in either one of the metrics [13], [14]. The set of efficient paths is also called efficient frontier. The efficient frontier can also be viewed as a function that maps available bandwidth to delay. In this two-dimensional case, the problem of aggregating a topology becomes finding the efficient frontier among all border routers.

Traffic sources compute QoS paths with the aggregated network information. This network information may not represent the network resource map with sufficient accuracy due to the discrepancy of true network states and aggregated information. Thus, after computing QoS paths, the sources issue signaling packets traversing the computed paths to reserve necessary network resources or to ensure resource availability. This additional signaling is performed with either RSVP-TE [18] or CR-LDP [2]. When the signaling packet encounters the problem of unavailable resources, it must retrace its steps from the point of blockage back to its original entry point into the domain [10]. This crankback continues until either the network decides that no route can be found, or certain protocol timers expire. Clearly, lowering the frequency of crankbacks is a significant benefit of having higher fidelity aggregation schemes since it results in faster setup times and less load on the network control [10].

The QoS information aggregation has to include the costs of all possible transits through the subnetwork. However, the size of such a representation can grow quite large as the number of border nodes increases. Thus, the performance benefits of having a high fidelity aggregation representation need to be again properly traded off against the practical cost imposed by the use of large representations. That is, QoS information abstraction may relax the burdens of large and detailed information representations with certain information losses. Also, the QoS-enabled network architecture is expected to provision QoS within preset time constraints. Thus, the time introduced by additional signaling must be carefully optimized in order to comply with the constraints. Obviously, this requires that any discrepancy between network information updates and the true network status be avoided for undesirable crankbacks not to occur.

In order to effectively deal with the twin issues of hierarchical network composition and fast and accurate QoS provisioning, we propose a new scheme stemming from the fundamental premises of *statistical QoS guarantees* in conjunction with the benefits of *multiple QoS provisioning*.

III. Mechanism Overview

In this section, we present the elements for hierarchically provisioning QoS paths: constraints quantization, local path computation / path summary exchange, call admission with path summary information, and multiple QoS path computation. These elements are mainly to overcome the limits of the previously discussed methods and to provide practicality and effectiveness in hierarchical QoS routing.

A. Constraints Quantization

As mentioned, calculating the efficient frontier with multiple QoS metrics may not be feasible due to its limited number of manipulated metrics and broad range of possible outcomes. To simplify matters, we can predefine QoS constraints for the demands of certain QoS applications. We call this process "constraints quantization" and this enables the network to effectively deal with multiple QoS metrics and even precompute QoS paths for the quantized constraints. This also complies with a DiffServ architecture in which traffic classes are defined a priori and QoS services are granted per each class in an aggregated fashion. This constraints quantization limits the outcome boundaries and results in fewer path searches. Moreover, this allows paths per quantized constraints to be precomputed and pinned down between border routers. Thus, additional path verification with control messages can be avoided and fast QoS provisioning can be achieved.

B. Local Path Computation and Path Summary Exchange

As the first step toward hierarchical QoS routing, we are aiming to deploy our architecture in an autonomous system. The main reason is that all the subdomains in a single AS are controlled by a common administration policy. Thus, the same quantized constraints are applied to all subdomains.

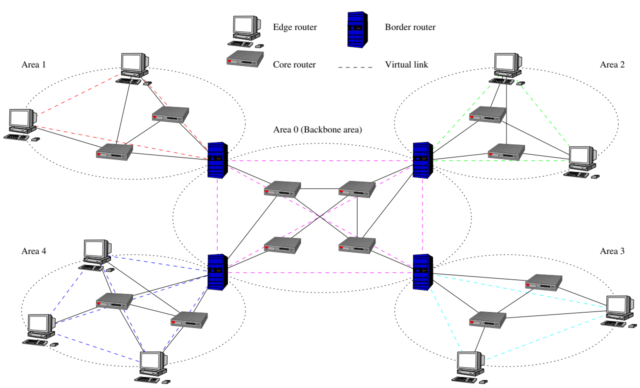

Fig. 1. Example of local path computation for virtual connections between border routers.

First, we assume that an AS consists of several subdomains (i.e., "areas" in our case) and each of them is connected to a backbone area. Fig. 1 shows an example. Each area runs its own routing protocol not revealing its topology information to the outside world. This topological subdivision includes different types of routers, namely:

- Edge router

A router that is directly connected to end hosts and that receives connection requests is an edge router. Edge routers are treated as traffic sources and destinations, and they make routing decisions according to the mechanisms we are presenting here.

- Area internal router (a.k.a. core router)

A router with interfaces to networks in one area only is considered to be an area internal router. Internal routers flood each area with complete routing and QoS information about changes that occur within the area.

- Area border router

A router with one interface to the backbone network and one or more interfaces to additional areas is considered to be a border router. Each border router connects one or more areas to the backbone. Using the backbone, border routers distribute routing and QoS information "summaries" for all areas throughout the AS.

As in the topology aggregation approach, edge/border routers are involved in gathering and processing QoS information to summarize their corresponding areas. Within each area, the conventional QoS information exchanges are carried out via the OSPF flooding mechanism. This area-bounded flooding carries the internal QoS information to all edge/border routers in the area. Then, the edge and the border routers perform path computations similar to the one required to find the efficient frontier. However, in our case, we do not need to find all possible paths between the edge/border routers. This is because we predefine sets of quantized QoS constraints and QoS services are granted for each set. Thus, if Q sets of QoS constraints are defined, at most Q paths are computed for each pair of edge/border routers (i.e., edge-to-edge, edge-to-border, and border-to-border). This path computation can be easily carried out by the QoS routing algorithm presented in [1]. Every area performs these path computations independently with its own link state database. The computed paths are pinned down in their corresponding areas, and they are refreshed when new link state information arrives. These paths can be viewed as "virtual links" between edge/border routers within areas as depicted in Fig. 1.

After paths are computed for each quantized constraints set within each area, they are, in turn, summarized and the path summaries are exported to other areas. This path summary describes the QoS-constrained reachability between edge/border router pairs. For example, an area consists of 4 edge/border routers and 4 sets of quantized QoS constraints. Then, the QoS constraints can be quantized in the format of Table I.

TABLE I

QOS CONSTRAINTS QUANTIZATION EXAMPLE.

QoS index	Bandwidth	Delay	Jitter
1	b1	d1	j1
2	b2	d2	j2
3	b3	d3	j3
4	b4	d4	j4

Then, the path summary can be generated as in Table II. The table is an example of path summary created by node 1. It shows that node 1 can reach node 2 satisfying all the four QoS constraint sets. Likewise, it also reaches node 3 and 4 satisfying constraint sets 2, 3, and 1, 3, 4 respectively. The detailed path information is locally kept by node 1. Also, each path is recorded with its QoS status (i.e., available bandwidth, end-to-end delay, etc.) for further QoS path computations with these path summaries. Basically, this looks similar to the link state table entry created by node 1. The link state entry describes the reachability to adjacent neighbors with link metrics. On the other hand, the path summary shows the QoS-constrained reachability to other edge/border routers.

TABLE II

PATH SUMMARY TABLE OF NODE 1.

Neighbor node	QoS indexes
2	1, 2, 3, 4
3	2, 3
4	1, 3, 4

This simplified path summary is exchanged between all edge/border routers. That is, edge routers in an area distribute their path summaries to other edge and border routers of the same area. However, border routers of an area send out their path summaries not only to edge and other border routers in the area but also to edge/border routers in other areas they also belong to. In the path summary exchange process, instead of the OSPF flooding mechanism, a predefined source routing scheme is used. That is, in the previous example, if node 1 is sending out its path summary to node 2, 3, and 4, node 1 may compute the shortest paths to those nodes. Then, the shortest paths are used to carry the path summary by forming direct "virtual links" between them. This is to avoid unnecessary control overhead in areas. Since core routers are not involved in the path decision process and do not need to collect the path summaries, single shortest paths between edge/border routers minimize the control overhead. This path summary exchange between edge/border routers can be viewed as a restricted flooding mechanism. Each edge/border router sends out its path summary to other edge/border routers of the same area. After collecting path summaries from other edge/border routers, the routers do not rebroadcast them as normal flooding does. This is because other edge/border routers in the same area get the same path summary information directly from the information source. However, border routers need to relay the path summary information of an area to other areas to which the router belongs.

Consequently, all the edge/border routers in the AS maintain three different databases: link state database, locally precomputed path database, and the entire path summary table. Obviously, the path summary table resembles the link state table, but the detailed information about the physical paths built in other areas is not included.

C. Call Admission Control with Path Summary Information

Since edge routers maintain the path summary tables, they can exercise a call admission control with the path summary information. The edge routers determine whether or not a new call is accepted by examining preestablished QoS paths. That is, when a new call comes in and requests a "quantized" QoS service, the edge router searches for a path in the path summary. Since the path summary table contains the abstracted path information to all other edge routers in the AS, the edge router can determine whether or not to accept the new call by performing similar QoS path computation. This computation takes into account preestablished path information (i.e., path summary) instead of link state information. It can be also viewed as QoS path computation with the "virtual links" as the conventional QoS routing does it with physical link states. Most importantly, after accepting the new call, the edge router does not need to issue any path setup messages since the QoS-constrained path is already established from entry to exit edge router across several areas. The source edge router simply stamps the path label corresponding to the quantized QoS constraints in the packet headers, and the traffic follows the precomputed paths across the predefined border routers. The call set up does not require any additional computation or path checking. Thus, it complies with our premise of fast QoS provisioning.

D. *Multiple QoS Path Provisioning*

Another notable property of our schemes is that networks provision multiple QoS paths. There are many reasons for using multiple paths: reliable QoS support, load balancing, cost-effective QoS provisioning as presented in [1], [19]. When multiple QoS paths are provisioned, they can be maintained seamlessly with the described constraints quantization and path summary process. Instead of searching for a single path for each quantized constraints set, we can preestablish multiple QoS paths for the described reasons. The path summary process still represents the same reachability as if there is only one path. However, the QoS status of the represented path summary will be the combination of computed multiple paths. That is, represented available bandwidth will be the minimum available bandwidth of all the paths, and delay will be chosen from their maximum value. Thus, the path summary information still keeps the actual path setup hidden and represents virtual links with the reachability and QoS status information.

When multiple paths become available, they can be used mainly in two different ways: call-by-call allocation onto each path and packet-by-packet allocation over them. Usually, the call-based allocation can be exercised in a round-robin fashion or simply by picking one of multiple paths randomly. The packet-based allocation scatters packets over multiple paths. That is, load balancing effect is made with either flow or packet granularity. This different usage also affects packet jitter, and these issues will be evaluated in the experiments section.

Multiple path provisioning is orthogonal to hierarchical network composition in our case since each flat subdomain provisions multiple paths independently. Thus, we can directly apply the multiple QoS path computation algorithm presented in [1].

With the described building components, in the following section, we introduce QoS routing variants which use hierarchically provisioned multiple QoS paths. Also, the conventional flat-space QoS routing is used and compared to show both fundamental benefits of hierarchical QoS routing and performance effects of multiple QoS path provisioning.

IV. Simulation Experiments

In this section, we investigate the benefits of hierarchical QoS routing when combined with multiple path QoS provisioning by simulating a hierarchical network in the SENSE [20] simulator developed at the NRL in UCLA. The topologies that we examine are of the generic form shown in Fig. 1. Thus, it consists of a backbone area connected to a number of stub areas forming a two-level hierarchy. The bandwidth on the links on the backbone is 45 Mbps and the delay on them is 10 μs. In the stub areas, the bandwidth on the links is 10 Mbps and the delay on them is 1 μs. For purposes of simulation, we consider traffic to belong to one of four QoS constraint classes. These classes correspond to:

- IP Telephony: bandwidth = 64 Kbps, delay = 100 ms
- Radio over the Internet: bandwidth = 192 Kbps, delay = 100 ms
- Video-on-Demand: bandwidth = 1.2 Mbps, delay = 50 ms
- CBR VPN traffic: bandwidth = 1.5 Mbps, delay = 10 ms

We also settled upon the distribution of these classes as 30, 35, 25, and 10 percent of the total number of connections generated.

A. *Parameters varied*

We vary the topology, the QoS routing schemes, the flooding interval of the link state advertisements and the degree of traffic congestion in the network. Each topology can be characterized by the number of stub areas and the number of source routers in each area. Thus, the parameters that we vary in the topology are the number of area routers and the number of "traffic originating/receiving" routers in each area. Topologies with larger number of areas are expected to show lower routing overhead when hierarchical schemes are used. Topologies with larger number of sources injects more connections into the network, thus testing the scalability of the schemes. The routing schemes that we evaluate are the following:

- Flat Single: Conventional single QoS path routing with flat topology.
- Hier Single: Single path hierarchical QoS routing.
- Hier Rand: Multiple QoS paths over hierarchical topologies of QoS areas with random selection of one path out of the multiple paths.
- Hier Scat: Multiple QoS paths over hierarchical topologies of QoS areas with packet scattering over the multiple paths.
- Flat Scat: Multiple QoS paths over flat topologies with packet scattering over the multiple paths.

We vary the link state advertisement interval from 5 seconds to 30 seconds. Here, we present the results for the two extremes (i.e., 5 and 30 seconds). High link refresh rates are expected to lead to more accurate CAC decisions at the cost of greater control overhead and vice versa for low refresh rates. Similarly, we tested the topologies with varying amounts of congestion and we present two scenarios: one corresponding to a lightly loaded network and the other corresponding to a highly loaded network. In a lightly loaded network, packet scattering schemes are expected to do better than random path selection schemes, while in heavily loaded networks, both are expected to give an equal degree of load balancing in the network.

B. *Parameters evaluated*

We evaluate the various QoS schemes on the following criteria:

1. Routing overhead: We simply calculate the number of QoS scheme-related control packets generated during the simulation

and call it the routing overhead. Hierarchical schemes are expected to give lower overhead since they perform area-bounded link state flooding and area-crossing path summary exchanges.
2. QoS performance: This takes into account the QoS performance seen by the connections and the network during the simulation. For each quantized QoS constraint class, we calculate the average QoS service received by the connections belonging to that class in their lifetime. This consists of the following parameters:

- Throughput satisfaction: The ratio of the received and the requested throughput.
- Delay satisfaction: The ratio of the received and the requested delay.
- Jitter: The average jitter experienced by the connections.
- Load balancing index: This gives us the network wide degree of load balancing. The details of how we arrive at the number to represent the load balancing degree is described in [1]. The lower the index, the better the degree of load balancing.

Given the distribution of the various classes, we gave them those weights while calculating the average throughput satisfaction, delay satisfaction, and jitter experienced by all connections irrespective of class. Throughput and delay statistics are expected to be similar for all the schemes, while jitter is expected to be larger for packet scattering schemes due to out-of-order arrival of packets. Load balancing on the other hand is expected to be better for multiple path schemes and even better for packet scattering schemes due to a finer (i.e., packet level) granularity.

Thus, for each topology, flooding interval and congestion degree, we get the *routing overhead, throughput satisfaction, delay satisfaction, jitter and load balancing index*. We present results for both lightly and highly loaded cases and with low and high rates of flooding. Since delay satisfaction was found to closely follow throughput satisfaction, we just present throughput satisfaction graphs here. The x-axis consists of the various topologies depicted as follows: (A, B) refers to a generic topology with A sources/area and B stub areas. The y-axis consists of the various parameters as experienced by the various QoS schemes. We first look at the statistics for low refresh rate and lightly loaded network and then we present the graphs which differ significantly for the other cases.

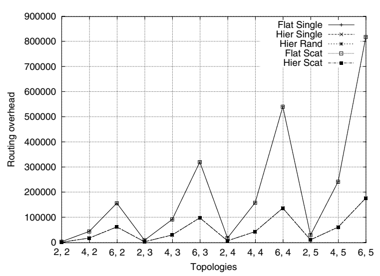

Fig. 2. Lightly loaded and low refresh rate: routing overhead.

Fig. 2 shows that hierarchical schemes show much lower routing overhead compared to the flat QoS schemes. All the hierarchical schemes are clustered near the low curve while the flat schemes are clustered near the top curve. Since the only parameter that influences the routing overhead is the refresh rate and the kind of scheme used, the graphs are the same for all low refresh cases.

Fig. 3 shows that all the schemes perform almost similarly when it comes to providing QoS satisfaction, since all of these schemes are capable of supporting QoS. We can still see, however, that multiple path schemes perform better than single path schemes in providing the service they promised. This is due to the benefits of load balancing in the network.

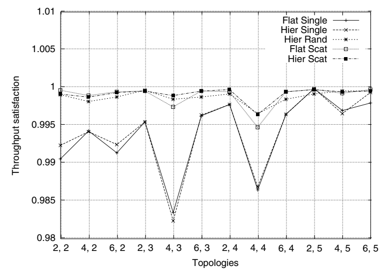

Fig. 3. Lightly loaded and low refresh rate: throughput satisfaction.

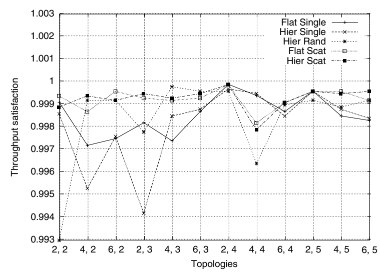

Fig. 4. Lightly loaded and high refresh rate: throughput satisfaction.

Fig. 4 shows that we now have even better QoS adherence, since the refresh rate is higher, we have more accurate information. Even here, multiple path cases give better QoS performance. The graphs for heavily loaded cases are similar and are therefore omitted here.

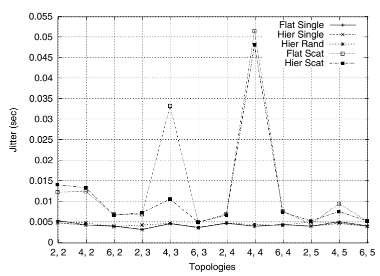

Fig. 5. Lightly loaded and low refresh rate: average jitter.

Fig. 5 shows that schemes with packet scattering give larger jitter compared to single path schemes, which was expected. Thus, a scheme like Hier Rand is a good option for jitter-sensitive traffic. The jitter graphs for all the other cases are similar and are therefore omitted.

The load balancing index is a measure of how well the traffic is spread in the network. Fig. 6 shows that single path schemes (Flat Single and Hier Single) have much worse load balancing than the multiple path schemes (Hier Rand, Hier Scat, and Flat

Scat). Hier Rand is a little worse since it uses only single path for each flow, but even with that, it substantially reduces the load balancing index by allocating traffic connections over multiple paths with the flow-level granularity.

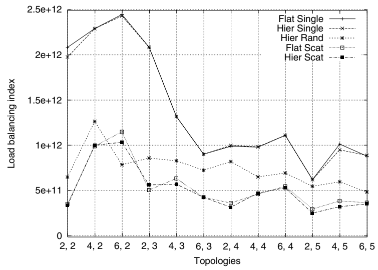

Fig. 6. Lightly loaded and low refresh rate: load balancing index.

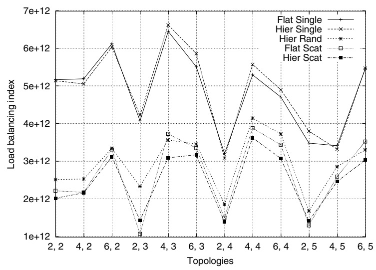

Fig. 7. Highly loaded and low refresh rate: load balancing index.

It is expected that in heavily loaded networks, when all the network is being used, Hier Rand will perform as good as Hier Scat or Flat Scat in terms of load balancing, and Fig. 7 proves that the presumption is correct.

The simulation results underline a few characteristics of the various schemes evaluated. Flat QoS schemes are accurate, and when combined with multiple paths, can lead to good QoS performance and load balancing. However, they suffer from high routing overhead, almost an order of magnitude more than corresponding hierarchical schemes. Hierarchical QoS schemes provide quite high QoS performance in spite of low refresh interval. They also lead to much lower routing overhead and when combined with multiple paths, they achieve fairly good load balancing. However, due to the nature of packet scattering, relatively high jitter is experienced, and this problem can be alleviated by selecting only one out of the multiple paths at random. Thus, a scheme like Hier Rand, while giving low jitter and tolerable load balancing, can also provide high QoS performance. For non-jitter sensitive traffic, Hier Scat is deemed to be ideal because of its adherence to hierarchical nature of network systems and fairly effective load balancing. Thus, we conclusively prove that hierarchical QoS schemes are practical due to their low cost and provide quite a high degree of adherence to QoS guarantees even with low refresh interval rates.

V. Conclusion

We proposed a set of approaches for *practical, scalable, fast QoS provisioning* with *low routing overhead* and *high state accuracy*. By proposing the use of *multiple paths* at all levels of the hierarchy, we provide both *reliability* for connections and *load balancing* over the network. Extensive simulation results validated our premises. The hierarchical schemes (Hier Rand and Hier Scat) give extremely good QoS performance even with low refresh rates, and very low overhead. Hier Scat provides much better load balancing while Hier Rand gives much better jitter performance. Therefore, an ideal combination would be to use Hier Rand for jitter-sensitive traffic and Hier Scat for other bandwidth-delay sensitive traffic. Throughout the simulation experiments with all these variants, we showed the cost effectiveness and high performance of the proposed routing mechanisms. Also, their practical application to the hierarchical nature of network system composition was shown to be effective.

References

[1] Scott Seongwook Lee and Mario Gerla. Fault Tolerance and Load Balancing in QoS Provisioning with Multiple MPLS Paths. *Lecture Notes in Computer Science*, 2092:155–, 2001. International Workshop on Quality of Service (IWQoS).

[2] E. Rosen, A. Viswanathan, and R. Callon. Multiprotocol Label Switching Architecture. Request for Comments 3031, Internet Engineering Task Force, January 2001.

[3] S. Kamat S. K. Tripathi G. Apostolopoulos, R. Guerin. Quality of Service Based Routing: A Performance Perspective. In *SIGCOMM*, Vancouver, British Columbia, September 1998.

[4] S. Kamat S. Tripathi G. Apostolopoulos, R. Guerin. Improving QoS Routing Performance under Inaccurate Link State Information. In *ITC-16*, June 1999.

[5] Z.-L. Zhang S. Nelakuditi, R. P. Tsang. Qualitity-of-Service Routing without Global Information Exchange. In *IEEE IWQoS*, March 1999.

[6] J. Crowcroft Z. Wang. Quality-of-Service Routing for Supporting Resource Reservation. *IEEE JSAC*, September 1996.

[7] N. Fujita A. Iwata. A Hierarchical Multilayer QoS Routing System with Dynamic SLA Management. *IEEE Journal on Selected Areas in Communication*, 18(12), December 2000.

[8] Shigang Chen and Klara Nahrstedt. An Overview of Quality of Service Routing for Next-Generation High-Speed Networks: Problems and Solutions. *IEEE Networks*, 12(6):64–79, November 1998.

[9] M. Goguen D. Dykeman. PNNI Draft Specification V1.0. Technical Report af-pnni-0055.000, ATM Forum, March 1996.

[10] B. Khan-Y. Shavitt B. Awerbuch, Y. Du. Routing Through Networks with Hierarchical Topology Aggregation. In *IEEE ISCC*, Athens, Greece, 1998.

[11] Y. Shavitt B. Awerbuch. Topology Aggregation for Directed Graphs. *IEEE/ACM Transactions on Networking*, 9(1), February 2001.

[12] R. Acharya D. Ghosh. A Probabilistic Scheme for Hierarchical QoS Routing. In *IEEE International Conference on Computer Networks*, October 2001.

[13] S. Chen K.-S. Lui, K. Nahrstedt. Hierarchical QoS Routing in Delay-Bandwidth Sensitive Networks. In *IEEE LCN*, Tampa, FL, USA, 2000.

[14] S. Chen K.-S. Lui, K. Nahrstedt. Routing with Topology Aggregation in Bandwidth-Delay Sensitive Networks. Technical Report UIUCDCS-2000-2187 UILU_ENG-2000-1744, Computer Science Dept., University of Illinois at Urbana-Champaign, 2000.

[15] J. C. Copeland Y.-K. Ke. Aggregation Algorithms for Asymmetric QoS-Routing Information. In *GLOBECOM*, November 2001.

[16] J. C. Copeland Y.-K. Ke. Scalability of Routing Advertisement for QoS Routing in an IP Network with Guaranteed QoS. In *GLOBECOM*, November 2000.

[17] George Apostolopoulos, Sanjay Kamat, and Roch Guerin. Implementation and Performance Measurements of QoS Routing Extensions to OSPF. New York, March 1999.

[18] D. Gan T. Li V. Srinivasan G. Swallow D. Awduche, L. Berger. Extensions to RSVP for LSP Tunnels. Request for Comments 3209, Internet Engineering Task Force, December 2001.

[19] Scott Seongwook Lee and Giovanni Pau. Hierarchical Approach for Low Cost and Fast QoS Provisioning. In *Proc. of IEEE Global Communications Conference (GLOBECOM)*, November 2001.

[20] Scott Seongwook Lee. Simulation Environment for Network System Enhancement (SENSE). http://www.cs.ucla.edu/ sslee/sense.