Towards Integrated Provisioning of QoS Overlay Networks

Jun-Hong Cui and Swapna Gokhale
University of Connecticut, Storrs, CT 06269

1 Introduction

The Internet is one of the most successful technologies in last century. Within less than four decades (starting from the first packet-switched computer network, ARPAnet), it has evolved into an extremely popular commercial infrastructure, and has a significant impact on almost all aspects of our lives and our society. With the dramatic advances in multimedia technologies and the increasing popularity of real-time applications, recently, quality of service (QoS) support in the Internet has been in a great demand. However, due to many historical reasons, today's Internet primarily provides best-effort connectivity service. To enhance the best-effort service model to provide QoS, researchers have proposed many seminal architectures, represented by IntServ [26] and DiffServ [25]. Unfortunately, due to many critical factors, realizing these QoS architectures in the Internet is unlikely to be feasible in the long run. In addition, there are no right economic models for these proposed architectures: although some ISPs might be interested in providing QoS in their own domains, there are no strong incentives for them to support QoS for users in other domains which are not their customers. Then, a challenging question faced by the researchers in the community is: what would be a practical solution for QoS support in the Internet?

In the past few years, overlay networks have emerged as an alternative mechanism for supporting value-added services such as fault tolerance [9], multicasting [38], and security [72]. Many of these overlays are end-user overlays, namely, overlays are constructed purely among the end hosts without support from any other intermediate nodes. Due to the difficulties of supporting end-to-end QoS purely in end-user overlays, some recent works [47, 58, 92, 144] propose to use backbone overlays for QoS support, where overlays are managed by a third party provider, such as an ISP. In this project, we adopt the approach of backbone overlays, and propose an integrated QoS overlay architecture, referred to as IQORA, for scalable, efficient, and practical QoS support. In IQORA, we advocate the notion of a QoS overlay network (referred to as QSON) as the backbone service domain. A QSON consists of service nodes or proxies, which are strategically deployed by a QSON provider (or a higher tier ISP). The design of QSON relies on well-defined business relationships between the QSON provider, network service providers (i.e., the underlying network domains which we also refer to as lower tier ISPs, or underlying ISPs for short), and end users: the QSON provider dimensions its overlay network according to end user requests, purchases bandwidth from the network service providers based on their service level agreements (SLAs), and sells its QoS services to end users via service contracts. Outside QSON, end hosts in access domains subscribe to QSON by connecting to some special proxies (called edge proxies) advertised by the QSON provider.

The proposed IQORA architecture combines the benefits from overlay networks and QoS-aware IP networks. On the one hand, it does not require the global deployment of QoS-aware routers. On the other hand, it can take advantage of the information obtained from intermediate nodes (proxies or service nodes) to facilitate QoS support. In addition, it offers many other advantages. First, unlike end user overlays (which can only support one application), a QSON provider can support a variety of applications simultaneously. This provides an additional incentive for ISPs to adopt IQORA. Second, it simplifies the management of resources in the underlying networks, since network service providers only need to provide services to limited number of QSON providers instead of millions or billions of individual users. This is facilitated because QSON decouples the end user service management and network resource management and is based on well-defined business relationships via SLAs with network service providers and service contracts with end users. This level of traffic aggregation, in the long run, will make IntServ-like architectures practical. A good analogy to this scenario is the relationship between manufacturers, dealers, and consumers in our daily life.

To make IQORA a reality, effective QSON provisioning which consists of dynamic management of end user QoS flows in an efficient and robust manner based on the SLAs with the underlying ISPs is the key. To accomplish QSON provisioning, many challenging issues need to be addressed. These include: 1. How to efficiently route end user QoS flows along the overlay paths (between proxies)? 2. How to detect in a timely manner overlay path failure and service degradation due to the higher failure probability of overlay paths? 3. How to efficiently restore the failed paths or upgrade a degraded path to a better one? To answer these questions, effective and efficient QoS overlay routing, QoS overlay monitoring, and QoS overlay restoration schemes are desired.

The level of research conducted in the areas of QoS routing, QoS monitoring and QoS restoration varies greatly. QoS routing has been the subject of several research efforts, whereas, relatively few efforts have been devoted to QoS monitoring and QoS restoration. Irrespective of the level of effort, a common drawback of the existing techniques addressing these issues is that they suffer from weak scalability when used in the context of inter-domain connections. In the case of an overlay network, a QoS connection will routinely involve multiple domains, due to which the existing routing, monitoring and restoration techniques cannot be used as is. A yet another drawback is that the research in these three areas has progressed independently, leading to point solutions which are inconsistent and which causes further inefficiencies in the utilization of resources. For QSON provisioning, an integrated efficient solution to QoS routing, QoS monitoring and QoS restoration is in great demand since it is a capital-intensive investment.

The proposed research effort seeks to develop comprehensive, scalable and efficient schemes to provide solutions to integrated QSON provisioning, which comprises of three sub-problems, namely, QoS overlay routing, QoS overlay monitoring and QoS overlay restoration. The schemes for each individual sub-problem have been designed by considering the architecture of IQORA in a unified manner. Additionally, the QoS overlay routing scheme is designed with an eye towards facilitating QoS overlay monitoring and QoS overlay restoration, and the QoS overlay monitoring and QoS overlay restoration schemes are designed to benefit from the support provided by the QoS overlay routing scheme. By considering routing, monitoring and restoration in an integrated manner during the process of designing the schemes, the overheads can be mitigated further [48]. Through extensive simulations and analysis, we propose to evaluate the schemes for their associated overheads and performance measures for various settings of the design parameters and traffic patterns. We also propose to build a testbed of IQORA in Internet2 [1] and PlanetLab [2], in order to study the feasibility and to experimentally investigate the performance in real environments.

The layout of this technical report is as follows: Section 2 reviews the related research in QoS architectures, and QoS routing, monitoring, and restoration techniques. Section 3 describes the service overlay network architecture, IQORA. Section 4 presents the proposed QSON provisioning schemes. Section 5 discusses the preliminary set of performance and efficiency analysis studies. Section 6 finally summarizes the proposed work.

2 Related research

The related research in QoS architectures and QoS routing, QoS monitoring, and QoS restoration techniques is reviewed in this section.

2.1 QoS Architectures

Qos architectures have evolved from IntServ, to DiffServ, to the recently proposed overlay model. In this subsection, we briefly review some representative architectures along with their pros and cons.

Integrated Service Architecture: Integrated Service Architecture (IntServ) [26] is designed to provide different types of QoS guarantees for data flows. It establishes connections through the network using a resource reservation protocol, such as RSVP [164], and an admission control mechanism, such as [65], based on network information. Then, the resource state information for each flow needs to be maintained at routers in order to ensure that sufficient resources are available during the lifetime of the flow. Since the number of data flows in the Internet can be very huge, the main criticism of IntServ is its weak state scalability.

Differentiated Service Architecture: Differentiated Service architecture (DiffServ) [25] is proposed for scalable service differentiation in the Internet. It categorizes data flows into a number of classes. Data in the same class receives same QoS. In a DiffServ domain, packets crossing a link and requiring the same behavior (e.g., scheduling treatment and drop probability, or in other words, in the same class) constitute a Behavior Aggregate (BA). At the ingress nodes, the packets are classified and marked with a Diff-Serv Code Point (DSCP) according to their Behavior Aggregate. At each transit node, the DSCP is used to determine the behavior for each packet. By flow aggregation, DiffServ is much more scalable than IntServ, however, it can only provide very coarse-granularity services.

End-User Overlays for Value-added Services: Several end-user overlays have been designed to support value-added services. Resilient Overlay Networks (RONs) are proposed to detect and recover from Internet path failures [9]. They route packets on paths optimized for application-specific metrics. RON nodes actively monitor the quality of the paths to their neighbors and decide where to route packets based on collected information and application requirements. Amir *et al* propose an overlay architecture, called Spine overlay architecture [7, 8], which uses the hop-by-hop reliability approach on overlay links to reduce the latency and jitter of reliable connections. By applying TCP-like loss recovery and congestion control on each overlay link, this approach can detect packet loss faster and recover the packets locally. Though highly flexible, end-user overlays usually could not provide end-to-end QoS guarantees, since these type of overlays typically cross many intermediate domains, and the uncontrolled domain peering structure is unlikely to provide direct QoS support to the end users. Moreover, it is difficult to design an effective economic model for ISPs to adopt end user overlays.

Backbone QoS Overlays Some recent works [47, 58, 92, 144] propose to use backbone overlays for QoS support, and the backbone overlays are managed by a third party provider. In Service Overlay Networks or SONs [47], bandwidth is provisioned with certain QoS guarantees from individual network domains to build a logical end-to-end service delivery. While SONs rely on underlying networks to provide QoS services, OverQoS [144] presents a Controlled Loss Virtual Link (CLVL) abstraction to provide Internet QoS (e.g., statistical bandwidth and loss rate assurance) using overlay networks and performing bundle loss control on each virtual link. QRON [92] introduced the concept of overlay brokers (OBs) and a general unified framework for an overlay network. Another proposal called QUEST (QoS assured composEable Service InfrasTructure) [58] went further and presented solutions to compose qualified service paths with multiple QoS constraints and load balancing from SLA contracts of individual service components. It should be noted that these backbone overlay research efforts either focus on one aspect of the overlay provisioning (e.g., bandwidth dimensioning [47], or overlay path composition [58]) or dedicate to introducing an overlay architecture (e.g., [92] and [144]). None of these efforts address the issue of integrated provisioning of backbone QoS overlays.

2.2 QoS Routing

QoS routing has been researched extensively, and in this section we describe the aspects that are pertinent to the proposed research. *QoS routing* consists of determining a path through the network which has adequate resources to satisfy the QoS requirements of a connection, while simultaneously achieving global efficiency in network resource utilization. QoS routing can be classified into unicast routing and multicast routing [28, 33, 36, 41, 43, 51, 77, 78, 82, 83, 91, 102, 103, 106, 122, 128, 133, 135, 142, 143, 145, 147, 152, 155, 158, 159, 161, 163, 166]. It can be further classified as intra-domain routing and inter-domain routing. To facilitate QoS routing, the state of every link in the network must be expressed in terms of a set of QoS metrics such as delay, bandwidth, jitter and cost [64]. The metrics of the links along a path are composed to obtain QoS metrics of the path. A QoS connection expresses its QoS requirements in the form of constraints on one or more path QoS metrics [33]. These constraints are then compared to the QoS metrics of the paths through the network in order to select one that satisfies the constraints.

QoS routing techniques can be categorized into source routing and distributed routing. In the case of source routing, the source node is responsible for determining a suitable path by applying graph algorithms [24, 29, 45] to the network topology and link states that are stored at the source node [33, 131, 154]. The link state protocol [12, 117, 129, 132, 134] is used to maintain link states. It broadcasts link state information through the network which consumes an enormous amount of resources. As a result, there is a tradeoff between the frequency of link state broadcasts, the staleness of the link states, and the optimality of the paths [11, 14, 15, 18, 69–71, 73, 74, 80, 81, 87, 94, 95, 98, 101, 132, 162]. Another drawback of source routing is that it has to employ heuristic algorithms to solve the k-constrained routing problem [3, 13, 16, 20, 22, 31, 35, 37, 42, 44, 50, 53, 54, 60, 66, 78, 79, 85, 86, 96, 97, 104, 105, 107, 112–115, 120, 122, 136, 139, 150, 153–157, 163]. Distributed routing is achieved by probe flooding, where the source node floods probes through the network towards the destination node searching for suitable paths [32, 33, 57, 62, 119, 137]. The overheads incurred in flooding probes can be reduced by bounded and selective flooding [27, 34, 84, 90, 119]. If multiple suitable paths which satisfy the constraints exist, then the shortest path is chosen to achieve global efficiency in network resource utilization [17, 34, 84, 111, 130, 138, 160], a reduction in delay and a reduction in the probability of path degradation.

The overheads associated with source and distributed routing are aggravated to a large extent when used for inter-domain routing in networks of large size. Aggregation techniques for source routing have been proposed to alleviate this issue [57, 63, 88, 89, 99, 116, 123, 124, 149], but it may lead to inaccuracies, crankback, and reaggregation [21, 30, 52, 59]. The scalability of inter-domain routing via probe flooding can be improved by precomputing only the shortest paths [84, 111, 130]. However, even the number of shortest paths in the case of a large network is likely to be very high.

When a QoS connection is to be routed through an overlay network, it will typically cross multiple domains. For each one of the domains the QSON provider may have a different level of SLA, resulting in a different level of resource availability. When the level of available resources is high, the inefficiencies introduced by aggregation (if aggregation techniques are used) may not be significant. However, in order to maximize the revenue, the QSON provider may be interested in admitting as many connections as possible which may result in a low level of resource availability. In such cases, aggregation will lead to inaccurate and sub-optimal solutions. If distributed routing via probe flooding is to be used, then flooding probes across all the possible paths through the network will consume resources that could be otherwise used for supporting additional connections to increase revenue. Due to these reasons, the existing inter-domain techniques cannot be used for routing in an overlay network.

2.3 QoS Monitoring

Once a suitable path is determined, the required level of resources are reserved along this path for the entire lifetime of the connection, and then this path is used to route the packets of the connection. Despite the reservation of resources, the connection may still experience degradation in the QoS due to the weakening or the failure of one or more nodes/links along the path. *QoS monitoring* is concerned with monitoring an established connection so that QoS degradation can be predicted prior to occurrence, or at least detected as soon as possible after it has occurred

QoS monitoring can be classified into two categories, namely, end-to-end monitoring and distributed monitoring. In end-to-end monitoring, traffic measurements for each monitored connection are recorded only at the source and the destination nodes. Techniques to record and analyze these measurements have been developed [4–6, 55, 56, 68, 108, 109, 126, 140]. End-to-end monitoring, is not capable of localizing the degradation and hence any corrective action such as restoring the connection to an alternate suitable path also needs to be taken end-to-end. End-to-end corrective actions are not scalable, especially for inter-domain connections.

QoS distributed monitoring seeks to localize the degradation by recording measurements at several intermediate nodes along the path. Several issues need to be addressed in the context of QoS distributed monitoring. These include: (i) specification and the number of the monitors, (ii) location of the monitors from which to obtain traffic measurements, (iii) collection and analysis of the measurements, (iv) obtaining synchronized measurements corresponding to the same part of the flow from different monitors, and (v) uniformly distributing the monitors along a connection. A QoS distributed monitoring technique by Jiang *et. al.* [67, 68, 148] seeks to address some of these issues, but it suffers from many limitations. It assumes that the source node has complete information about the identities of all the nodes along an established QoS connection, which is unlikely in the case of inter-domain connections where the node identities may not be revealed to the source node to preserve security. The approach does not consider the issue of how to specify the monitors and how they should be distributed for uniform monitoring of a connection. Also, a separate technique needs to be employed to obtain synchronized measurements from the monitors. Further, additional traffic is generated to facilitate the monitoring process which consumes resources.

In the case of a connection that is routed over an overlay network, it is important to localize the degradation to a single domain. This is necessary because as mentioned in the Section 2.2, each domain in the network may be managed by a potentially different network service provider. Identification of the domain of degradation enables the QSON provider to attribute the degradation to a single network service provider. The SLA with the lower tier ISP who manages the domain in which the degradation has occurred can be renegotiated if necessary. Also, the affected domain can be avoided while routing future connections till the ISP has corrected the cause of degradation. In addition, the QSON provider can apply remedial action to the disrupted connections locally within the affected domain, without causing an impact on the SLAs with the ISPs which manage other domains. It is also important that as few resources as possible be consumed in order to facilitate the monitoring process.

2.4 QoS Restoration

When QoS degradation is expected or its occurrence has been detected, the connection should be restored to an alternate suitable path, and this is referred to as *QoS restoration*. QoS restoration can be performed using a proactive or a reactive approach. Current research has primarily focused on proactive approaches [76], in which resources are reserved along a primary and one or more secondary or backup paths [100, 127]. In the event of failure(s), the connection is restored along one of the secondary paths. To support the proactive approach, the QoS routing technique that is employed has to support multipath routing [39,40,121,151,165]. The secondary paths may extend end-to-end between the source and destination nodes [46,93,110,141,146], and may be totally or maximally disjoint with respect to primary path [61, 118, 146]. Alternatively, the secondary paths or bridges may extend between an upstream and a downstream node of the failed node/link [23, 75, 76, 110], and this is referred to as line restoration. These upstream and the downstream nodes should avoid all the nodes along the original path except the two neighbors of the failed node/link in order to avoid backhauling [10, 162]. Using the proactive approach, a connection can be restored with minimal delay. However, it results in a waste of resources due to duplicate reservations. Also it cannot guarantee restoration, since in the case of line restoration, bridges protect single nodes/links, but cannot protect sets of neighboring nodes/links, and in the case of end-to-end restoration, simultaneous failures could occur along both the primary and the secondary paths. Thus, reactive restoration which consists of rapidly determining and restoring the connection to an alternate suitable path after the occurrence of a failure/degradation is necessary.

Reactive restoration could also be performed end-to-end, or between the two neighbor nodes of the failed node (line restoration). End-to-end restoration incurs a long delay which increases with the distance between the source and the destination nodes. Also, in this case, the impact of the failure is distributed along the failed and the alternate path. Thus, end-to-end restoration is not the best alternative for restoring an inter-domain connection. Reactive line restoration is faster than end-to-end restoration, since it restores only a portion of the disrupted connection. Also, it keeps the impact of the failure in a small area surrounding the failed node/link. However, in this case, it is necessary to determine an alternate path for restoration which does not cause backhauling, which can increase the delay. If all the possible alternate paths cause backhauling, then it precludes the disrupted connection from being restored altogether. It is not clear how the upstream and the downstream nodes for line restoration must be chosen to minimize the probability of backhauling. Also, in the event of multiple node/link failures it is not evident if line restoration should be performed for each one of the failed nodes, or just once covering the entire set of failed nodes. In addition, line restoration may also require higher capacity than end-to-end or path restoration.

From the perspective of QSON provider, proactive approaches waste precious resources which could be used to admit additional QoS connections. Due to this reason, reactive restoration is perhaps the only alternative in an overlay network. Also, the reactive restoration scheme should try to limit the impact of the degradation in one domain from spreading across other domains. Simultaneously, it should also eliminate the possibility of backhauling, since backhauling results in a waste of resources.

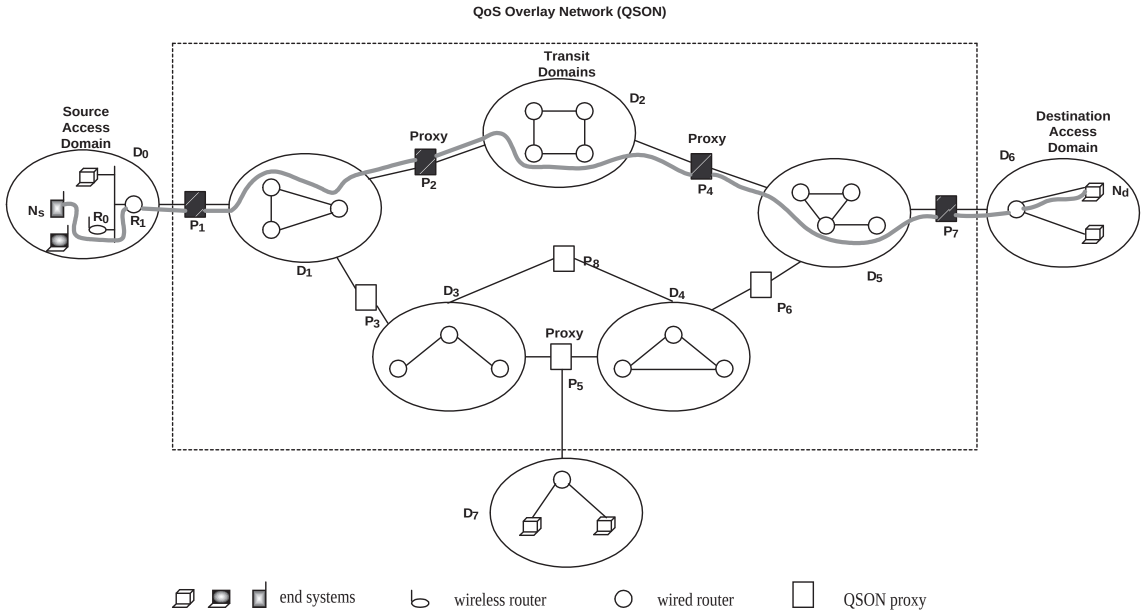

Figure 1: The Architecture of IQORA

3 The Integrated QoS Overlay Architecture

We propose an integrated QoS overlay architecture (IQORA) for scalable, efficient, and practical QoS support. In IQORA, a QSON (QoS overlay network) is constructed as the backbone service domain, which consists of many strategically deployed proxies by the QSON provider. The overlay paths between proxies are composed based on the service level agreements (SLAs) between the QSON provider and the underlying ISPs. Outside QSON, end hosts in access domains subscribe to QSON by proper paths connecting to some edge proxies advertised by the QSON provider. A high level illustration of IQORA is shown in Figure 1. Though QSON is an overlay network across multiple domains, from the end user point of view, it is actually a single logical domain. End-user flows are managed by QSON, and the underlying ISPs only see "aggregated" flows traversing overlay paths.

To realize IQORA, effective QSON provisioning is necessary. First, end users subscribe to QSON dynamically, and it is critical to efficiently route end-user QoS flows along overlay paths in QSON and maximize the bandwidth utilization since QSON is a capital-intensive investment. Secondly, due to the higher failure rate of overlay paths, timely detection on overlay path failure and service degradation is desired. Lastly, once paths fail or services degrade, QSON has to be equipped with restoration schemes to recover the failed paths or switch degraded paths to better ones. Stated succinctly, we want to address the issue of efficient QoS overlay routing, QoS overlay monitoring, and QoS overlay restoration schemes for providing an integrated service to end users.

In the proposed research project, we will develop and evaluate QoS overlay routing, QoS overlay monitoring, and QoS overlay restoration schemes. These schemes will be designed by considering the architecture of the IQORA in a unified manner. Towards this end, we first introduce some concepts and terminology, which will be used in the description of the preliminary versions of the QoS overlay schemes in the next section.

3.1 Physical Network Structure

The underlying physical network structure from which QSON will be constructed is consistent with the structure of the present Internet, which is composed of domains. We assume that each one of the domains is managed by a network service provider (also referred to as underlying ISP in this technical report). QSON proxies are strategically deployed between domains. Each proxy may belong to multiple domains. In each domain, we refer to QSON proxies as border nodes, and all other nodes as internal nodes. An internal node can only be linked to internal nodes or QSON proxies within its domain, whereas a QSON proxy can be linked to any node (internal nodes or QSON proxies) in the domains it belongs to. For example, in Figure 1, QSON proxy P_2 belongs to both domain D_1 and domain D_2. We refer to domains hosting end users as access domains, such as D_0, D_6, D_7, and other domains used for data delivery as transit domains, such as D_1 through D_5. The (QSON) proxies in access domains are called (QSON) edge proxies, which are usually advertised to end users by the QSON provider. In addition, we refer to the (QSON) edge proxy in the source (or destination) access domain as source (or destination) (QSON) edge proxy. Furthermore, the two proxies in a transit domain along a path from the source to the destination, are referred to as ingress proxy and egress proxy in the forward direction. For example, in Figure 1, if a connection originates in access domain D_0 and terminates in access domain D_6, P_1 is a source QSON edge proxy, and P_7 is a destination QSON edge proxy. Also, in the above example P_1 and P_2 are respectively the ingress and egress proxies for domain D_1.

3.2 Logical Network Structure

In order to route end-user requests through the overlay network, for each domain the QSON provider needs to know about the possible alternate paths between ingress/egress proxy pairs, and the amount of bandwidth available on these paths. It does not need to know the topology of the underlying ISP domain. The QSON provider thus enters into a SLA with an underlying ISP, where the SLA specifies the set of alternate paths between all the pairs of proxies that belong to ISP's domain, and the amount of bandwidth allocated by the ISP to the QSON provider on each one of the paths. The initial amount of bandwidth purchased by the QSON provider can be adjusted based on the demands of the end users of the QSON by modifying the SLA. The logical view of the QSON thus consists of paths between pairs of proxies in each domain, and each one of these paths may be annotated by the bandwidth allocated by the ISP to the QSON. For each path through the domain, the ISP may also provide the QSON information about the number of hops along the path. Information regarding the number of hops along paths can be used to guide the selection of one path if multiple suitable paths exist, so that higher efficiency in resource utilization, lower delay, and lower probability of path degradation can be achieved. Each path will also be annotated with the number of hops

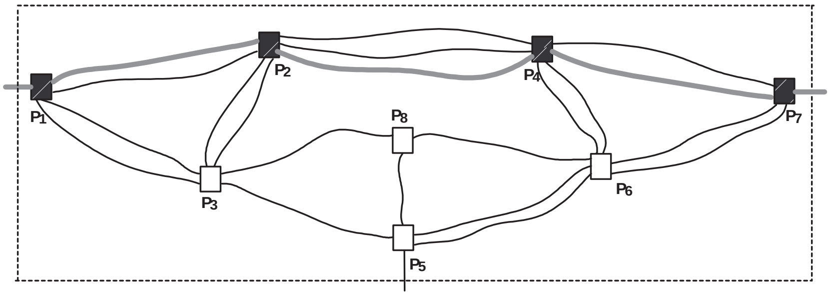

Figure 2: A Logical View of QSON

along the path in addition to the bandwidth. The logical network structure from the point of view of the QSON is shown in Figure 2.

3.3 Network State in QSON

The QSON uses the bandwidth allocated along the paths between the proxies to route end-user connections. As a result, as end-user connections arrive and depart, the amount of available bandwidth along each one of the paths between the proxies changes dynamically. In the QSON, each proxy maintains the amount of available bandwidth on all the paths to other proxies in the domain[1] that are included in the SLA. Each proxy also maintains a list of (logical) paths to the other proxies which are in the network but which do not belong to the same domain(s). A (logical) path between the pair of proxies which do not belong to the same domain consists of a sequence of proxy nodes. For example, in Figure 1, proxy P_1 has a list of (logical) paths $P_1P_2P_4$, $P_1P_2P_4P_7$, $P_1P_3P_5$, $P_1P_3P_5P_6$, $P_1P_3P_5P_6P_7$, $P_1P_3P_8$, $P_1P_3P_8P_6$, and $P_1P_3P_8P_6P_7$.

4 Proposed QSON Provisioning Schemes

In this section we describe the preliminary versions of QoS overlay routing, QoS overlay monitoring and QoS overlay restoration schemes. During the proposed research project the preliminary versions of the schemes will be refined based on the results obtained from the performance and efficiency analysis studies (See Section 5).

4.1 QoS Overlay Routing

An end-user QoS connection originates at the source node in the source access domain, traverses one or more QSON proxies in the transit domains, and terminates at the destination node in the destination access domain. In order to route such a QoS connection, an end-to-end path which satisfies the QoS constraints is obtained by composing the paths through the source and destination access domains and one or more

---

[1] Here we assume all the domains participating in IQORA are QoS aware. The case of non-QoS aware domains or incremental deployment, is discussed in Section 5.

transit domains in the QSON as explained below. We assume that the end-user connection expresses its QoS constraints in terms of bandwidth, since the QSON provider can only provide the bandwidth requested by a user in a *guaranteed* manner.

To route a QoS connection, the source node forwards probes along all the existing paths to the source QSON edge proxy within its domain. These probes are loaded with the bandwidth constraints of the connection. Each probe computes the bottleneck bandwidth of the path it traverses in the forward direction. Therefore, for each path between the source node and the source edge proxy, a probe will reach the source edge proxy. The source edge proxy then selects a suitable path among the possibly multiple paths. A suitable path between the source node and the source edge proxy is the one that has sufficient bandwidth to satisfy the constraints of the connection. If multiple suitable paths are available, then one can be selected either randomly, or based on other criteria such as the number of hops along the path, or the actual bottleneck capacity of the path.

Departing from the source access domain, the probe is forwarded by the source edge proxy to other proxies in the same domain as the source edge proxy. The proxies chosen to forward the probes are such that they lie along the possible multi-domain (logical) paths leading to the destination edge proxy. To choose the possible multi-domain paths, we suggest a criteria of coarse-grained delay threshold. This delay threshold can be expressed in terms of the number of proxies along the path. A possible multi-domain (logical) path is thus a path where the number of proxies along a path is less than a pre-specified limit. Before forwarding the probe to the next proxy, the source edge proxy composes the bandwidth of the path carried by the probe with the bandwidth of a suitable path between itself and the next proxy, and loads the probe with this bandwidth. A suitable path in a transit domain can be selected based on criteria similar to those described for the selection of a path in the source access domain. Additional administrative factors included in the SLA between the QSON and the underlying ISP (for the domain) may also be considered in the selection. For example, some paths through the domain may have a higher cost, in which case these paths may not be chosen unless it is absolutely necessary. If multiple possible paths to the destination edge proxy exist through a single egress proxy, then the probe is forwarded only once to the egress proxy. If no path between the chosen ingress/egress proxy pair has sufficient bandwidth to satisfy the bandwidth requirements, the probe is pruned and not forwarded further or pruned. The probe is then forwarded to the next proxy along the selected path. After the next proxy receives the probe, it repeats the same process as that of the source edge proxy. This continues until the probe reaches the destination edge proxy. In summary, for each domain along a possible logical path, the ingress proxy forwards the probe to the egress proxy in the domain. Before forwarding the probe, the ingress proxy updates the bandwidth of the path carried by the probe with the bandwidth of a suitable path between itself and the egress proxy.

When a probe reaches the destination edge proxy, it carries the bandwidth of a path between the source node and itself. The destination edge proxy then forwards the probe to the destination node along all the paths (probe flooding as in the source access domain). Thus, each probe that reaches the destination node carries the bandwidth of a path between the source and the destination nodes. The destination node then compares the QoS metrics of the path(s) (i.e., the bottleneck bandwidth) with the bandwidth requirement of the connection and selects one. The final path selection may also be based upon other administrative factors enforced by the QSON.

Renegotiation: During the routing of an end-user connection, if the QSON provider encounters a situation where the available bandwidth along all the possible logical paths between the source and the destination edge proxies is less than the bandwidth requirements of the connection, the provider has two options. In the first option it can choose to block the connection. This option though simple is not desirable. In the second option, it chooses a possible path and determines the domain(s) in which the available bandwidth is less than the bandwidth requested by the connection. It can then negotiate with the ISPs to upgrade the bandwidth in those domains.

An Example: We explain the above QSON routing scheme with the help of an example. Referring to Figure 1, we consider a connection originating at the source node N_s in domain D_0 which is to be routed to the destination node N_d in domain D_6. The QoS constraints of the connection are expressed in terms of the required bandwidth, say b units. In order to route this connection, the source node N_s floods probes along the path $N_sR_0R_1$ towards source edge proxy P_1. At node N_s, the bandwidth of the path N_sR_0 is compared with b units, and since this bandwidth is higher than b units, the probe is forwarded to node R_0. At node R_0, the bandwidth of the path $N_sR_0R_1$ is composed, and upon determining that it is greater than b units, the probe is forwarded to node R_1. The same process is repeated at node R_1, and the probe is forwarded to the source edge proxy P_1. At proxy P_1, three possible paths which satisfy the pre-specified limit on the number of proxies (delay-constrained threshold) exist to proxy P_7, which is the destination edge proxy. These three paths are $P_1P_2P_4P_7$, $P_1P_3P_5P_6P_7$, and $P_1P_3P_8P_6P_7$. For the first path, proxy P_1 composes the bandwidth of the path $N_sR_0R_1P_1$ with the bandwidth of the suitable path between itself and P_2, finds that the bandwidth of the entire path $N_sR_0R_1P_1P_2$ to be greater than b units and hence forwards the probe to proxy P_2. The probe is forwarded to proxy P_2 along the chosen suitable path. The destination edge proxy can be reached along the second and the third path by a single egress proxy, namely, As a result, a single probe is forwarded to proxy P_3, by proxy P_1 by composing the bandwidth of the path $N_sR_0R_1P_1$ with that of the suitable path between P_1 and P_3 and upon determining that it is greater than b units. When the probe reaches proxy P_2, the same process is repeated for the path between P_2 and P_4, and upon determining that the overall path between N_s and P_4 satisfies the bandwidth constraints, the probe is forwarded to proxy P_4. At proxy P_3, however, it is determined that the bottleneck bandwidth of the paths between P_3 and the egress proxies P_5 and P_8 is not sufficient to satisfy the bandwidth constraints of the connection. As a result, the probe that reaches proxy P_3 is pruned and not forwarded any further. The probe that reaches proxy P_4 continues to traverse to the destination proxy P_7 since the bottleneck bandwidth of the path traversed by the probe satisfies the QoS constraints. When a probe reaches the destination proxy P_7, probes are once again flooded to the destination node N_d along all the existing paths.

Flow State: For an end user connection, once a path is determined by the proposed routing scheme, resources (i.e., bandwidth) will be reserved along the path (using RSVP-like resource reservation protocol [164]). To route packets, obviously, each proxy needs to retain information of its next logical hop (i.e., next proxy). In addition, to facilitate monitoring and restoration, each proxy is also expected to maintain the flow state information which consists of the proxy that is two hops away, the source edge proxy, and the destination edge proxy in the path. For example, proxy P_3 along the path $P_1P_3P_8P_6P_7$, will maintain the following state: next proxy P_8, two-hop proxy P_6, source edge proxy P_1 and destination edge proxy P_7. We will discuss how this additional flow state information is used to facilitate the design of QSON monitoring and QSON restoration schemes in next two sections.

Advantages: The routing scheme described above offers several advantages over prevalent inter-domain routing schemes. First, on-demand uncontrolled probe flooding via all the existing paths is used only in the source access and destination access domain. Since the access domains are smaller in size, uncontrolled flooding is affordable. In a transit domain, probes are not flooded along all the existing paths between every ingress/egress proxy pair that lies along a possible logical path. Instead, a probe is sent only along the selected path between each possible ingress/egress proxy pair. This significantly mitigates the overheads of conventional inter-domain routing via probe flooding where probes are forwarded along all the inter-domain paths that exist between the source and destination proxies. For example, if there are three transit domains between the source and destination proxies, such that the first domain has three alternate paths, the second one has four paths and the third domain has five paths. In the conventional schemes, probes would be flooded across all the sixty paths that result from the combination of paths in the three domains, whereas in the proposed scheme a probe would traverse across only one path in each one of the domains. *Probe pruning* mitigates the overheads further through the transit domains. Another significant advantage of this scheme is that it facilitates the determination of multiple candidate paths for routing a QoS connection without incurring any additional overhead. Thus, if resource reservation fails along the selected path, it can be attempted along another path which is readily available.

4.2 QoS Overlay Monitoring

In this section we describe the proposed QoS overlay monitoring scheme, and also discuss how the monitoring scheme is designed to leverage information from routing. The monitoring scheme is mainly concerned with distributed monitoring with an explicit goal to the detect the domain in which degradation occurs.

Each QSON proxy along an established QoS overlay connection is designated to be a monitor. As discussed in QoS overlay routing scheme, during resource reservation, the QSON proxies along the path are instructed to record the identity of the source edge proxy to which the traffic measurements are to be periodically forwarded. For example, proxies P_2 and P_4 will be designated to monitor a connection routed along the path $P_1P_2P_4P_7$.

Advantages: Designating the QSON proxies along a connection eliminates the issue of having to specify and locating the monitors. Also, in each domain, it is ensured that the monitors are the two end points of the connection through the domain. If a degradation is detected by analyzing the traffic measurements collected by the monitors which represent two end points of the connection through a domain, it can be localized to the domain with higher confidence. For example, referring to Figure 1, if a degradation is detected by analyzing the traffic measurements recorded by proxies P_1 and P_2, then the cause is likely to lie in domain D_1. Thus, the identity of the domain where degradation occurs can be determined while preserving the security of the domain. Since the QSON proxies are pre-designated as monitors, unlike in the conventional techniques where any node could potentially serve as a monitor, the monitors are much smaller in number since the number of QSON proxies will be much fewer compared to internal nodes. Thus they may be equipped with special purpose hardware, to obtain accurate time from the Universal Time Coordination (UTC) [19, 49, 125] without incurring undue costs. Hence, once the QSON proxies are synchronized, the source edge proxy may be able to obtain synchronized information from them without any special synchronization technique. The QoS overlay restoration scheme described in Section 4.3, also charges the QSON proxies with the responsibility of restoring the connection in the event of QoS degradation. Thus by designating the QSON proxies to be monitors and assigning them the responsibility of restoration, the issue of remedying the degradation can be handled efficiently.

4.3 QoS Overlay Restoration

In this section we describe the proposed reactive restoration scheme and also discuss how it couples with the proposed QoS overlay routing scheme to achieve better restoration. The QSON provider is only responsible for the portion of the path between the source edge proxy and the destination edge proxy. As a result, the proposed restoration scheme only addresses restoration of the path through the transit domains, between the edge proxies. It does not address restoration of the paths through access domains.

Since proxies along the overlay path are designated to be the monitors, we assume proxies are robust. However, the paths between proxies are prone to failure. If a path in any one of the transit domain fails, then line restoration for a connection routed along the failed path is performed between the ingress and egress proxies in the transit domain. The alternate paths between the proxies included in the SLA, and the bandwidth available along the alternate paths is used to select a suitable one to route the connection. If a suitable alternate path does not exist between the original ingress/egress proxy pair, then the ingress proxy checks whether a path is available from itself to the proxy that is one hop away from the egress proxy (next-hop proxy). Each proxy retains information about the next-hop proxy for each flow in the resource reservation phase as described in the QoS overlay routing scheme. The path between the ingress proxy and the next-hop proxy should not be through the domains (except for the domain in which the disruption lies, and the domain to which the egress proxy belongs) that are already along the path. This is necessary to avoid backhauling.

If a path is available between the ingress proxy and next-hop proxy which satisfies this criteria, then the original routing scheme is used to determine if this path has sufficient bandwidth to satisfy the requirements of the connection. If the available bandwidth along this path is adequate, then the connection between the ingress/egress proxy pair is restored to the path between the ingress proxy and the next-hop proxy. If no suitable path exists between the ingress proxy and the next hop proxy, end-to-end restoration between the source edge proxy and destination edge proxy is performed.

Example: We explain the QoS restoration scheme with the help of an example. In Figure 1, consider a connection routed from proxy P_1 to proxy P_7 along the path $P_1P_3P_5P_6P_7$. Suppose for this connection the path between proxies P_3 and P_5 through domain D_3 fails. In this case, to restore the disrupted connection an alternate path with sufficient bandwidth between proxies P_3 and P_5 needs to be obtained. If a suitable alternate path is found, then the connection is restored to that path. On the other hand, if a suitable alternate path does not exist between proxies P_3 and P_5, then it is determined whether there exists a path from ingress proxy P_3 to next-hop proxy P_6 through domains that are not already on the path. Referring to Figure 1, it can be observed that a path exists from proxy P_3 to next-hop proxy P_6 through proxy P_8. Proxy P_8 belongs to domain D_3 (in which degradation lies) and D_4 (to which the egress proxy P_5 belongs), but it does not belong to any other domains that lie along the original path. If sufficient bandwidth is available to restore the connection to the path $P_3P_8P_6$, then the connection is restored to that path. If sufficient bandwidth is not available along this path, then the connection is restored end-to-end between the source edge proxy P_1 and the destination edge proxy P_7.

Advantages: The above QoS restoration approach offers many advantages. The most important advantage is that it eliminates backhauling altogether while simultaneously trying to limit the impact of the restoration on the rest of the network in the following manner. Line restoration is initially attempted directly between the ingress/egress proxy pair which is affected by failure. In this case the impact is limited to the domain in which the failure has occurred. Also, in this case, line restoration is essentially used to restore the portion of the path through the domain between the two end points, and is equivalent to end-to-end restoration within the domain. As a result, backhauling is eliminated. If the connection cannot be restored between the original ingress/egress proxy pair in the first step, then in the next step, line restoration is attempted between the ingress proxy and the next-hop proxy, in which case a few additional domains may be impacted. However, by choosing a path between the ingress proxy and next-hop proxy through domains that do not already lie along the connection (other than the domain in which the failure has occurred, and the domain(s) to which the egress proxy belongs), backhualing can be eliminated. The nodes between which line restoration is performed are pre-designated, and not chosen arbitrarily. Also, by restoring the failed portion of the path between its two end points, multiple failures in the domain can be considered. End-to-end restoration is used only as the last resort to restore a connection.

5 Performance Analysis and Evaluation

We plan to extensively evaluate the proposed QoS overlay schemes for their associated overheads and performance measures via simulation and experimentation. For each scheme, we will identify a set of design parameters and analyze the influence of these design parameters on the overheads and the performance measures of the scheme. Different topologies of the overlay network and end-user traffic patterns will be considered during evaluation. The proposed schemes will also be compared with the prevalent techniques in the literature.

5.1 Evaluation via Simulation

The preliminary set of simulation experiments are described below. For each one of the simulation experiments, we describe the objective of the experiment, the initial set of design parameters, metrics to estimate the associated overheads and the performance measures. Subsequently, additional simulations will be identified based on the analysis of the results obtained from this preliminary set.

5.1.1 QoS Overlay Routing Scheme

The proposed QoS overlay routing scheme will be compared with prevalent inter-domain routing schemes which include source routing with and without aggregation, and distributed routing via probe flooding. The schemes will be compared with respect to their overheads during the process of routing connections. Since overheads in each one of the schemes are caused by different factors, objective metrics to quantify these overheads for each one of the schemes which allow for a fair comparison among them will be determined. Performance measures that will be used for comparing the schemes include the connection blocking rate, connection bandwidth blocking rate and the delay incurred in determining a suitable path or the routing delay.

5.1.2 QoS Overlay Monitoring Scheme

In the proposed QoS overlay monitoring scheme, no additional traffic is generated to manage the monitoring process. Thus, initially we seek to obtain a quantitative estimate of the additional traffic that is avoided in proposed technique as compared to the prevalent distribution monitoring techniques [67, 68, 148]. Further, we will investigate the ability of the monitoring scheme to consistently detect the domain in which the degradation occurs. We will then seek to investigate the overheads associated with transmitting recorded traffic measurements from the monitors to the source node for the purpose of monitoring. A design parameter which will have an impact on the detection capability of the monitoring scheme as well as on its overhead, is the frequency with which the monitors record and send traffic measurements, and the influence of this parameter and the associated tradeoffs will be investigated.

5.1.3 QoS Overlay Restoration Scheme

The proposed QoS overlay restoration scheme which is a hybrid of end-to-end and line restoration will be compared to schemes which use pure end-to-end and line restoration. When a portion of a path along an established QoS overlay connection fails, the number of alternate suitable paths (which do not cause back-hauling) available for line restoration determines the probability that the disrupted connection can actually be restored using line restoration. Thus, one of the performance metrics used for comparison of these schemes will be the average number of alternate paths available for line restoration. When a connection is disrupted, an alternate suitable path must be determined and the connection must be restored to that path as soon as possible. The delay incurred in determining an alternate suitable path, and restoring the connection to that path is termed as restoration delay. The schemes will also be compared with respect to their restoration delay.

5.1.4 Integration of Routing, Monitoring and Restoration Schemes

Initially, the routing, monitoring and restoration schemes will be evaluated individually. Subsequently, the three schemes will be evaluated in an integrated manner, to analyze the interplay between these schemes. Integrated evaluation will seek to assess how the design parameters of one scheme influence the overheads and performance of other schemes.

5.2 Experimental Evaluation

We propose to implement IQORA in PlanetLab [2] and Internet2 [1]. PlanetLab is essentially a loosely structured overlay network, which does not provide guaranteed QOS services, while Internet2 enables many advanced networking features, such as MPLS and DiffServ support. We will deploy QSON proxies in Internet2 and PlanetLab, and using measurements, we will compose several overlay paths between any two QSON proxies. In our testbed, we will treat PlanetLab and Internet2 as two neighboring transit domains, in order to deploy a QSON logical network. Access domains will comprise the UCONN network and networks of other collaborating universities. We will implement the proposed overlay routing, monitoring, and restoration schemes in the testbed, and evaluate the performance of these schemes via experimentation.

5.3 Incremental Deployment

Due to its formidable size, it is feasible to employ a QoS scheme in the Internet only if it can be deployed in an incremental manner. In the previous sections, we assume each domain including source access domain and destination access domain are QoS-aware. However, it is possible that some domains may not be capable of providing QoS services. For the domain which is not QoS-aware, the QSON proxies in the domain could use measurements to obtain QoS metrics, such as bandwidth capacity, delay, etc. In such situations, although IQORA cannot provide guaranteed services, significantly enhanced services may be possible.

6 Conclusions

In this technical report, we have described our proposed research for integrated provisioning of QoS overlay networks. The contributions of the proposed work could be summarized as follows:(i) A practical overlay architecture IQORA for scalable and efficient QoS support, (ii) Effective and efficient schemes for QSON integrated provisioning, which involves QoS overlay routing, QoS overlay monitoring and QoS overlay restoration, and taking into consideration the synergies in these issues, (iii) Extensive evaluation of the schemes to obtain quantitative estimates of the overheads and performance measures for various network topologies and traffic characteristics, (iv) A real testbed of IQORA in Internet2 and PlanetLab for the experimental evaluation of the schemes, and (v) A network design tool which encapsulates the schemes in a user-friendly manner which could be used by a QSON provider for network design and planning.

References

[1] Internet 2. *http://www.internet2.edu/.*

[2] Planetlab. *http://www.planet-lab.org/.*

[3] E. Aboelela and C. Douligeris. "Fuzzy inference system for QoS routing in B-ISDN". In *Proc. of IEEE Canadian Conference on Electrical and Computer Engineering*, pages 141–144, May 1998.

[4] M. Aida, N. Miyoshi, and K. Ishibashi. A scalable and lightweight QoS monitoring technique combining passive and active approaches – On the mathematical formulation of CoMPACT monitor –. In *Proc. of INFOCOM*, 2003.

[5] E. Alshaer, H. Abdel-Wahab, and K. Maly. A scalable monitoring architecture for managing distributed multimedia systems. In *Proc. of MMNS*, 1997.

[6] E. Alshaer, H. Abdel-Wahab, and K. Maly. Hierarchical filtering–based monitoring system for large–scale distributed applications. In *Proc. of the 10th Intl. Conference on Parallel and Distributed Systems*, October 1997.

[7] Y. Amir and C. Danilov. Reliable communication in overlay networks. In *Proceedings of the IEEE International Conference on Dependable Systems and Networks, June 2003.*, 2003.

[8] Y. Amir, C. Danilov, and C. Nita-Rotaru. High performance, robust, secure and transparent overlay network service. In *Proceedings of International Workshop on Future Directions in Distributed Computing (FuDiCo), June 2002.*, 2002.

[9] D. G. Andersen, H. Balakrishnan, M. F. Kaashoek, and R. Morris. Resilient overlay networks. In *Proceedings of Symposium on Operating Systems Principles*, pages 131–145, 2001.

[10] J. Anderson, B. Doshi, S. Dravida, and P. Harshavardhana. Fast restoration of ATM networks. *IEEE Journal on Selected Areas in Communications*, 12(1):128–138, January 1994.

[11] G. Apostolopoulos, R. Geurin, S. Kamat, and S. K. Tripathi. "Improving QoS routing performance under inaccurate link state information". In *Proc. of the 16th International Teletraffic Congress (ITC'16)*, Edinburgh, UK, June 1999.

[12] G. Apostolopoulos, R. Guerin, S. Kamat, A. Orda, and S. K. Tripathi. "Intra-domain QoS routing in IP networks: A feasibility and cost/benefit analysis". *IEEE Network*, 13(5), September/October 1999.

[13] G. Apostolopoulos, R. Guerin, S. Kamat, and S. Tripathi. On reducing the processing cost of on–demand QoS path computation. In *Proc. of ICNP'98*, pages 80–89, October 1998.

[14] G. Apostolopoulos, R. Guerin, S. Kamat, and S. Tripathi. Quality of Service based routing: A performance perspective. In *Proc. of SIGCOMM*, pages 17–28, 1998.

[15] G. Apostolopoulos, R. Guerin, S. Kamat, and S. K. Tripathi. Server based QoS routing. In *Proc. of GLOBECOM*, 1999.

[16] G. Apostolopoulos and S. Tripathi. On the effectiveness of path pre-computation in reducing the processing cost of on–demand QoS path computation. In *Proc. of IEEE Symposium on Computer and Communication*, June 1998.

[17] G. Apostolopoulos and S. K. Tripathi. "On the effectiveness of path pre–computation in reducing the processing cost of on–demand QoS path precomputation". In *Proc. of Third IEEE Symposium on Computers and Communications*, pages 42–46, July 1998.

[18] A. Ariza, E. Casilari, and F. Sandoval. QoS routing with adapative updating of link states. *Electronic letters*, 37(7):604–606, April 2001.

[19] P. Ashton. Algorithms for off–line clock synchronisation. Technical Report TR-COSC 12/95, Department of Computer Science, University of Canterbury, December 1995.

[20] B. Awerbuch, Y. Azar, S. Plotkin, and O. Waarts. Throughput–competitive on–line routing. In *Proc. of 34th Annual Symposium on Foundations of Computer Science*, November 1993.

[21] B. Awerbuch, Y. Du, B. Khan, and Y. Shavitt. Routing through networks with hierarchical topology aggregation. *Journal of High–Speed Networks*, 7(1), 1998.

[22] L. Barolli, A. Koyama, H. Sawada, T. Suganuma, and N. Shiratori. A new QoS routing approach for multimedia applications based on genetic algorithms. In *Proc. of First Intl. Symposium on Cyber Worlds*, pages 289–295, November 2002.

[23] Y. Bejerano, Y. Breitbart, A. Orda, R. Rastogi, and A. Sprintson. Algorithms for computing QoS paths with restoration. In *Proc. of INFOCOM*, 2003.

[24] R. Bellman. *Dynamic Programming*. Princeton University Press, Princeton, N.J., 1957.

[25] S. Blake, D. Black, and et al. An architecture for Differentiated Services. *IETF RFC 2475*, 1998.

[26] R. Braden, D. Clark, and S. Shenker. Integrated Services in the internet architecture: an overview. *IETF RFC 1633*, 1994.

[27] C. Caixia, W. Shiquan, and W. Jianfang. "A tunable QoS routing algorithm". In *Proc. of the 3rd World Congress on Intelligent Control and Automation*, pages 2360–2364, July 2000.

[28] K. Carlberg and J. Crowcroft. Building shared trees using a one–to–many joining mechanism. *ACM Computer Communication Review*, pages 5–11, January 1997.

[29] C. Casetti, R. L. Cigno, M. Meilla, and R. Munafo. A new class of QoS routing strategies based on network graph reduction. In *Proc. of INFOCOM*, pages 715–722, June 2002.

[30] B. Chang and R. Hwang. “Dynamic update of aggregated routing information for hierarchical routing in ATM networks”. In *Proc. of Eigth Intl. Conference on Parallel and Distributed Systems*, pages 653–660, June 2001.

[31] J. Chen. “Efficient heuristic algorithms for finding multi–constrained paths”. In *Proc. of Intl. Conference on Communications*, pages 1074–1079, May 2002.

[32] S. Chen and K. Nahrstedt. Distributed QoS routing. Technical Report UIUCDCS-R-97-2017, University of Illinois at Urbana-Champaign, 1997.

[33] S. Chen and K. Nahrstedt. An overview of Quality–of–Service routing for the next generation high–speed networks: Problems and solutions. *IEEE Network, Special Issue on Transmission and Distribution of Digital Video*, 1998.

[34] S. Chen and K. Nahrstedt. Distributed Quality–of–Service routing in high–speed networks based on selective probing. In *Proc. of Local Computer Networks*, October 1998.

[35] S. Chen and K. Nahrstedt. On finding multi–constrained paths. In *Proc. of Intl. Conference on Communications*, pages 874–879, June 1998.

[36] S. Chen, K. Nahrstedt, and Y. Shavitt. A QoS–aware multicast routing protocol. In *Proc. of INFOCOM*, 2000.

[37] Y. Chen and R. Hwang. “QoS routing algorithms for multiple traffic classess”. In *Proc. of Intl. Conference on Communications*, pages 2217–2221, May 2002.

[38] Y.-H. Chu, S. G. Rao, and H. Zhang. A case for end system multicast. In *Proceedings of ACM Sigmetrics*, June 2000.

[39] I. Cidon, R. Rom, and Y. Shavitt. Multi–path routing combined with resource reservation. In *Proc. of INFOCOM*, 1998.

[40] I. Cidon, R. Rom, and Y. Shavitt. “Analysis of multi–path routing”. *IEEE/ACM Trans. on Networking*, 7(6):885–896, December 1999.

[41] J. Cui, J. Kim, D. Maggiorini, K. Boussetta, and M. Gerla. Aggregated multicast – A comparative study. Technical Report UCLA-CSD-020011, UCLA, 2002.

[42] Y. Cui, J. Wu, and M. Xu. Precomputation for finding paths with two additive weights. In *Proc. of Intl. Conference on Communications*, 2003.

[43] J. Dai, H. Pung, and T. Angchuan. “A multicast routing protocol supporting multiple QoS constraints”. In *Proc. of 10th Intl. Conference on Networks*, pages 31–36, 2002.

[44] H. de Neve and P. van Mieghem. "A multiple quality of service routing algorithm for PNNI". In *ATM Workshop Proceedings*, pages 324–328, 1998.

[45] E. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, pages 269–271, 1959.

[46] R. Doverspike and J. Yates. Challenges for MPLS in optical network restoration. *IEEE Communications Magazine*, 39(2):89–96, 2001.

[47] Z. Duan, Z.-L. Zhang, and Y. T. Hou. Service overlay networks: SLAs, QoS, and bandwidth provisioning. *IEEE/ACM Transactions on Networking*, 11(6):870–883, 2003.

[48] H. Elazhary and S. Gokhale. An integrated approach to QoS provisioning and monitoring. In *Proc. of IASTED Intl. Conference on Parallel and Distributed Computing and Networks*, Innsbruck, Austria, February 2004.

[49] J. Elson, L. Girod, and D. Estrin. Fine–grained network time synchronization using reference broadcasts. In *Proc. of the 5th Symposium on Operating Systems Design and Implementation*, 2002.

[50] F. Ergun, R. Sinha, and L. Zhang. QoS routing with performance–dependent costs. In *Proc. of INFOCOM*, March 2000.

[51] M. Faloutsos, A. Banerjea, and R. Pankaj. QoSMIC: Quality of Service sensitive multicast Internet protocol. In *Proc. of SIGCOMM*, pages 144–153, 1998.

[52] E. Felstaine, R. Cohen, and O. Hadar. Crankback prediction in hierarchical ATM networks. In *Proc. of INFOCOM*, 1999.

[53] G. Feng, C. Douligeris, K. Makki, and N. Pissinou. "Performance evaluation of delay–constrained least–cost QoS routing algorithms based on linear and nonlinear lagrange relaxation". In *IEEE Intl. Conference on Communications*, volume 4, pages 2273–2278, May 2002.

[54] G. Feng, K. Makki, N. Pissinou, and C. Douligeris. "An efficient approximate algorithm for delay–cost constrained QoS routing". In *Proc. of Tenth Intl. Conference on Computer Communications and Networks*, pages 395–400, October 2001.

[55] L. Franken and B. Haverkort. The performability manager. *IEEE Network*, pages 24–32, Jan/Feb 1994.

[56] M. Fry, V. Witana, P. Ray, and A. Seneviratne. Managing QoS in multimedia services. *JNSM*, 5(3), 1997.

[57] D. Ghosh, V. Sarangan, and R. Acharya. "Quality–of–service routing in IP networks". *IEEE Trans. on Multimedia*, 3(2):200–208, June 2001.

[58] X. Gu, K. Nahrstedt, R. Chang, and C. Ward. Qos-assured service composition in managed service overlay networks. In *Proceedings of IEEE 23rd International Conference on Distributed Computing Systems, Providence, May 2003.*, 2003.

[59] R. Guerin and A. Orda. QoS–based routing in networks with inaccurate information: Theory and algorithms. In *Proc. of INFOCOM*, 1997.

[60] R. Guerin and A. Orda. "Computing shortest paths for any number of hops". *IEEE/ACM Trans. on Networking*, 10(1):102–114, February 2002.

[61] Y. Guo, F. Kuipers, and P. Mieghem. Link–disjoint paths for reliable QoS routing. *Intl. Journal of Communication Systems*, 2003.

[62] A. Hafid, G. Bochmann, and R. Dssouli. Distributed multimedia application and Quality of Service: A review. *Electronic Journal on Networks and Distributed*, pages 1–50, 1998.

[63] F. Hao and E. Zegura. On scalable QoS routing: Performance evaluation of topology aggregation. In *Proc. of INFOCOM*, 2000.

[64] J. Hong, S. Kim, and K. Lee. QoS routing schemes for supporting load balancing. In *Proc. of 5th Intl. Conference on High Speed Networks and Multimedia Communications*, 2002.

[65] S. Jamin, P. Danzig, S. Shenker, and L. Zhang. A measurement-based admission control algorithm for integrated services packet networks. *Proceedings of ACM SIGCOMM'95*, Sept. 1995.

[66] H. Jeon, S. Kim, Y. Kim, H. Kim, and J. Lee. Performance analysis of QoS routing with network dependent delay cost. In *Proc. of the 15th Intl. Conference on Information Networking*, pages 689–693, February 2001.

[67] Y. Jiang, C. Tham, and C. Ko. A QoS distribution monitoring scheme for performance management of multimedia networks. In *Proc. of GLOBECOM*, 1999.

[68] Y. Jiang, C. Tham, and C. Ko. Challenges and approaches in providing QoS monitoring. *Intl. Journal of Network Management*, 10:323–334, 2000.

[69] W. Jianxin, W. Weiping, C. Jianer, and C. Songgiao. A randomized QoS routing algorithm on networks with inaccurate link–state information. In *Proc. of Intl. Conference on Communication Technology*, pages 1617–1622, August 2000.

[70] H. Jiuchuan, Z. Hongke, and Z. Sidong. Heuristic QoS routing based on possibility distribution and soft computing. In *Proc. of Intl. Conference on Computer Networks and Mobile Computing*, pages 399–403, October 2003.

[71] Y. Ke and J. A. Copeland. Scalability of routing advertisement for QoS routing in an IP network with guaranteed QoS. In *Proc. of GLOBECOM*, 2000.

[72] A. Keromytis, V. Misra, and D. Rubenstein. Sos: Secure overlay services. In *Proceedings of ACM SIGCOMM'02, (Pittsburgh, PA), August 2002.*, 2002.

[73] S. Kim and M. Lee. "Server based QoS routing with implicit network state updates". In *Proc. of Ninth Intl. Conference on Networks*, pages 511–516, October 2001.

[74] S. Kim and M. Lee. "Server based QoS routing with implicit network state updates". In *Proc. of Ninth Intl. Conference on Networks*, pages 511–516, October 2001.

[75] M. Kodialam and T. Lakshman. Dynamic routing of bandwidth guaranteed tunnels with restoration. In *Proc. of INFOCOM*, 2000.

[76] M. Kodialam and T. V. Lakshman. "Restorable dynamic quality of service routing". *IEEE Communications Magazine*, 40(6):72–81, June 2002.

[77] V. Kompella, J. Pasquale, and G. Polyzos. Multicast routing for multimedia communication. *IEEE/ACM Transactions on Networking*, 1(3):286–292, 1993.

[78] V. Kompella, J. Pasquale, and G. Polyzos. Two distributed algorithms for multicasting multimedia information. In *Proc. of 2nd Intl. Conference on Computer Communications and Networks*, pages 343–349, June 1993.

[79] K. Korkmaz and M. Krunz. Multi-constrained optimal path selection. In *Proc. of INFOCOM*, 2001.

[80] T. Korkmaz and M. Krunz. "Hybrid flooding and tree–based broadcasting for reliable and efficient link–state dissemination". In *Proc. of GLOBECOM*, pages 2400–2404, November 2002.

[81] T. Korkmaz and M. Krunz. "Bandwidth–delay constrained path selection under inaccurate state information". *IEEE/ACM Trans. on Networking*, 11(3):384–398, June 2003.

[82] L. Kou, G. Markowsky, and L. Berman. A fast algorithm for Steiner tree. *Acta Informatica*, pages 141–145, 1981.

[83] F. Kuipers and P. Mieghem. MAMCRA: A constrained–based multicast routing algorithm. *Computer Communications*, 25:801–810, 2002.

[84] S. Kweon and K. Shin. Distributed QoS routing with bounded flooding for real–time applications. Technical Report CSE-TR-388-99, University of Michigan, 1999.

[85] C. Lee, J. C. Cui, R. Chow, and R. Newman. Altruistic QoS routing with multimedia dispersion. In *Proc. of Intl. Conference on Multimedia and Expo*, pages 487–490, August 2001.

[86] C. Lee, J. C. Cui, R. Chow, and R. Newman. An altruistic QoS routing for multimedia communications. In *Proc. of 2001 Intl. Symposium on Intelligent Multimedia, Video and Speech Processing*, pages 316–319, May 2001.

[87] C. Lee, J. C. L. Liu, R. Chow, and R. Newman. “Altruistic QoS routing with multimedia dispersion”. In *Proc. of Intl. Conference on Multimedia and Expo*, pages 487–490, August 2001.

[88] S. S. Lee, S. Das, G. Pau, and M. Gerla. A hierarchical multipath approach to QoS routing: Performance and evaluation. In *Proc. of Intl. Conference on Communications*, May 2003.

[89] W. Lee. Topology aggregation for hierarchical routing in ATM networks. In *Proc. of SIGCOMM*, pages 82–92, 1995.

[90] W. Lee and B. Lee. “Pre–computation based selective probing (PCSP) scheme for distributed QoS routing”. In *Proc. of GLOBECOM*, pages 1778–1782, December 2000.

[91] Z. Li and P. Mohapatra. QoS–aware multicasting in DiffServ domains. In *Proc. of Global Internet Symposium*, 2002.

[92] Z. Li and P. Mohapatra. QRON: QoS-aware routing in overlay networks. *IEEE Journal on Selected Areas in Communications, January*, 2004.

[93] H. Long and S. Shen. A mechanism for restorable QoS routing. In *Proc. of Computer Networks and Mobile Computing (ICCNMC) 2003*, pages 60–67, October 2003.

[94] D. H. Lorenz and A. Orda. “QoS routing in networks with uncertain parameters”. In *Proc. of INFOCOM*, pages 3–10, March 1998.

[95] D. H. Lorenz and A. Orda. QoS routing in networks with uncertain parameters. *IEEE/ACM Transactions on Networking”*, 6(6):768–778, December 1998.

[96] Q. Ma and P. Steenkiste. Quality–of–Service routing for traffic with performance guarantees. In *Proc. of Intl. Workshop on Quality of Service*, 1997.

[97] Q. Ma and P. Steenkiste. Quality–of–service routing with performance guarantees. In *Proc. of the 4th Intl. IFIP Workshop on Quality of Service*, May 1997.

[98] Z. Ma, P. Zhang, and R. Kantola. Influence of link state updating on the performance and cost of QoS routing in an Internet. In *Proc. of IEEE Workshop on High Performance Switching and Routing*, 2001.

[99] V. Marbukh. “QoS routing under adversarial binary uncertainty”. In *Proc. of IEEE Intl. Conference on Communications*, pages 2268–2272, May 2002.

[100] J. L. Marzo, E. Calle, C. Scoglio, and T. Anjali. “Adding QoS protection in order to enhance MPLS QoS routing”. In *Proc. of Intl. Conference on Communications*, pages 1973–1977, 2003.

[101] X. Masip-Bruin, S. Sanchez-Lopez, J. Sole-Pareta, and J. Domingo-Pascual. “QoS routing algorithms under inaccurate routing information for bandwidth constrained applications”. In *Proc. of Intl. Conference on Communications*, pages 1743–1748, 2003.

[102] I. Matta and M. Eltoweissy. “A scalable QoS routing architecture for real–time CSCW applications”. In *Proc. of Fourth IEEE Real-Time Technology and Applications Symposium*, pages 137–142, June 1998.

[103] P. Mieghem, G. Hooghiemstra, and R. Hofstad. On the efficiency of multicast. *IEEE/ACM Transactions on Networking*, 9(6):719–732, December 2001.

[104] P. Mieghem and F. Kuipers. On the complexity of QoS routing. *Computer Communications*, 2001.

[105] S. Ming-Hong, W. Si-Bing, and B. Ying-Cai. “A bandwidth constrained QoS routing optimization algorithm”. In *Proc. of Intl. Conference on Communication Technology*, pages 491–494, April 2003.

[106] D. Mitzel and S. Shenker. Asymptotic resource consumption in multicast reservation styles. In *Proc. of SIGCOMM*, pages 226–233, 1994.

[107] M. Moh, B. Wei, and J. H. Zhu. “Supporting differentiated services with per–class traffic engineering in MPLS”. In *Proc. of Tenth Intl. Conference on Computer Communications and Networks*, pages 354–360, October 2001.

[108] A. Moore. Measurement–based management of network resources. Technical report, Computer Science Dept., University of Tennessee, January 1993.

[109] K. Mourelatou, A. Bouloutas, and M. Anagnostou. An approach to identifying QoS problems. *Computer Communications*, pages 563–570, 1994.

[110] K. Murakami and H. Kim. Comparative study on restoration schemes of survivable ATM networks. In *Proc. of INFOCOM*, 1997.

[111] S. Norden and J. Turner. Inter–domain QoS routing algorithms. Technical Report WUCS-02-03, Department of Computer Science, WUCS, 2002.

[112] A. Orda and A. Sprinston. “A scalable approach to the partition of QoS requirements in unicast and multicast. In *Proc. of INFOCOM*, pages 685–694, March 2002.

[113] A. Orda and A. Sprintson. “QoS routing: A precomputation perspective”. In *Proc. of INFOCOM*, pages 128–136, March 2000.

[114] A. Orda and A. Sprintson. QoS routing: The precomputation perspective. In *Proc. of INFOCOM*, pages 128–136, 2000.

[115] A. Orda and A. Sprintson. Precomputation schemes for QoS routing. *IEEE/ACM Transactions on Networking*, 11(4), 2003.

[116] W. Pak and S. Bahk. Partial optimization method of topology aggregation for hierarchical QoS routing. In *Proc. of Intl. Conference on Communications*, 2002.

[117] L. Peterson and B. Davie. *Computer Networks: A Systems Approach*, chapter Internetworking. Morgan Kaufman, New York, NY, 2000.

[118] N. Plotkin, B. Bellur, and R. Ogier. Quality–of–Service routing using maximally disjoint paths. In *Proc. of the Intl. Workshop on Quality of Service*, pages 119–128, May/June 1999.

[119] H. K. Pung, J. Song, and L. Jacob. "Fast and efficient flooding based QoS routing algorithm". In *Proc. of Eigth Intl. Conference on Computer Communications and Networks*, pages 298–303, October 1999.

[120] N. Rao and S. Batsell. QoS routing via multiple paths using bandwidth reservation. In *Proc. of INFOCOM*, pages 11–18, 1998.

[121] N. S. V. Rao and S. G. Batsell. "QoS routing via multiple paths using bandwidth reservation. In *Proc. of INFOCOM*, pages 11–18, March 1998.

[122] G. Rouskas and I. Baldine. Multicast routing with end–to–end delay and delay variation constraints. *IEEE Journal on Selected Areas in Communications*, 15(3), April 1997.

[123] V. Sarangan and R. Acharya. "Using network flows in hierarchical QoS routing". In *Proc. of Ninth Intl. Conference on Networks*, pages 422–427, 2001.

[124] V. Sarangan and R. Acharya. Using network flows in hierarchical QoS routing. In *Proc. of Ninth Intl. Conference on Networks*, pages 422–427, 2001.

[125] U. Schmid and K. Schossmaier. Interval–based clock synchronization. *Journal of Real–Time Systems*, 12(2), 1997.

[126] C. Schmidt and R. Bless. QoS monitoring in high performance environments. In *Proc. of the 4th IFIP Intl. Workshop on Quality of Service*, pages 199–207, 1996.

[127] K. Scott, G. Jimenez, and D. Bernstein. Finding alternatives to the best path. In *Proc. of 76th Annual Meeting of The Transportation Research Board*, January 1997.

[128] A. Shaikh, S. Lu, and K. Shin. Localized multicast routing. In *Proc. of GLOBECOM*, pages 1352–1356, November 1995.

[129] A. Shaikh, J. Rexford, and K. Shin. Dynamics of Quality–of–Service routing with inaccurate link–state information. Technical Report CSE-TR-350-97, Dept. of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, 1997.

[130] A. Shaikh, J. Rexford, and K. Shin. Efficient precomputation of Quality–of–Service routes. In *Proc. of NOSSDAV*, pages 15–27, July 1998.

[131] A. Shaikh, J. Rexford, and K. Shin. Evaluating the overheads of source–directed Quality–of–Service routing. Technical Report CSE-TR-381-98, Univ. of Michigan, 1998.

[132] A. Shaikh, J. Rexford, and K. Shin. Evaluating the impact of stale link state on Quality-of-Service routing. *IEEE/ACM Transactions on Networking*, 9(2):162–176, 2001.

[133] A. Shaikh and K. Shin. Destination–driven routing for low–cost multicast. *IEEE Journal of Selected Areas in Communications*, 15(3):373–381, 1997.

[134] A. Shankar, C. Alaettinoglu, K. Zieger, and I. Matta. Transient and steady–state performance of routing protocols: Distance–vector versus link–state. *Internetworking: Research and Experience*, 7(1), March 1996.

[135] S. Shi and J. Turner. Routing in overlay multicast networks. In *Proc. of INFOCOM*, June 2001.

[136] D. Shin, E. K. P. Chong, and H. J. Siegel. A multiconstraint QoS routing scheme using the depth–first search method with limited crankbacks. In *Proc. of 2001 IEEE Workshop on High Performance Switching and Routing*, pages 385–389, March 2001.

[137] K. Shin and C. Chou. A distributed route–selection scheme for establishing real–time channel. In *Proc. of 6th IFIP Intl. Conference on High Performance Networking*, pages 319–329, Sertember 1995.

[138] K. Shin, C. Chou, and S. Kweon. Distributed route selection for establishing real–time channels. *IEEE Transactions on Parallel and Distributed Systems*, 11(3), March 2000.

[139] S. Siachalou and L. Georgiadis. Efficient QoS routing. In *Proc. of INFOCOM*, 2003.

[140] J. Souza and R. Boutaba. Managing QoS in multimedia networks and services. In *Proc. of MMNS*, 2000.

[141] A. Srikitja, D. Tipper, and D. Medhi. On providing survivable QoS services in the next generation Internet. In *Proc. of MILCOM*, October 1999.

[142] A. Striegel and G. Manimaran. A survey of QoS multicasting issues. *IEEE Communications*, pages 82–87, June 2002.

[143] A. Striegel and G. Manimaran. Managing group dynamics and failures in QoS multicasting. *IEEE Communications magazine*, June 2002.

[144] L. Subramanian, I. Stoica, H. Balakrishnan, and R. H. Katz. Overqos: An overlay based architecture for enhancing internet qos. *In Proceedings of USENIX NSDI'04*, pages 71–84, 2004.

[145] Q. Sun and H. Langendorfer. A new distributed routing algorithm with end–to–end delay guarantee. In *Proc. of 2nd Workshop Protocols Multimedia Systems*, pages 452–458, October 1995.

[146] B. Szviatovszki, A. Szentesi, and A. Juttner. On the effectiveness of restoration path computation methods. In *Proc. of High Speed Networks Symposium*, 2002.

[147] H. Takahashi and A. Matsuyama. An approximate solution for the Steiner tree problem in graphs. *Mathematica Japonica*, 1980.

[148] C. Tham, Y. Jiang, and C. Ko. Monitoring QoS distribution in multimedia networks. *Intl. Journal on network Management*, pages 75–90, March/April 2000.

[149] A. Vasilakos, M. P. Saltouros, A. F. Altassis, and W. Pedrycz. Optimizing QoS routing in hierarchical ATM networks using computational intelligence techniques. *IEEE Trans. on Systems, Man and Cybernetics*, 33(3):297–312, August 2003.

[150] S. L. Vieira. "Efficient routing with quality–of–service requirements". In *Proc. of Sixth IEEE Symposium on Computers and Communications*, pages 326–331, July 2001.

[151] S. Vutukury and J. Garcia-Luna-Aceves. A distributed algorithm for multipath computation. In *Proc. of GLOBECOM*, 1999.

[152] B. Wang and J. C. Hou. "An efficient QoS routing algorithm for quorumcast communication". In *Proc. of Ninth Intl. Conference on Network Protocols*, pages 110–118, November 2000.

[153] Z. Wang and J. Crowcroft. Bandwidth–delay based routing algorithms. In *Proc. of GLOBECOM*, pages 2129–2133, 1995.

[154] Z. Wang and J. Crowcroft. Quality of service routing for supporting multimedia applications. *IEEE Journal on Selected Areas in Communications*, 14(7):1228–1234, 1996.

[155] R. Widyono. The design and evaluation of routing algorithms for real–time channels. Technical Report TR-94-024, University of California at Berkeley, June 1994.

[156] W. Xiao, Y. Luo, B. Song, A. Xu, C. Law, and K. Ling. "An efficient heuristic algorithm for multi–constrained path problem". In *Proc. of Vehicular Technology Conference*, pages 1317–1321, September 2002.

[157] W. Xinhong and W. Guangxing. "An algorithm for QoS routing to optimize network resource utilization". In *Proc. of Intl. Conference on Info-Tech and Info-net*, pages 474–479, October 2001.

[158] S. Yana, M. Faloutsos, and A. Banerjea. QoS–aware multicast routing for the Internet: The design and evaluation of QoSMIC. *IEEE/ACM Transactions on Networking*, February 2002.

[159] D. Yang, W. Liao, and Y. Lin. MQ: An integrated mechanism for multimedia multicasting. In *Proc. of IEEE Intl. Conference on Multimedia and Expo*, pages 1063–1066, August 2000.

[160] X. Yuan and A. Saifee. Path selection methods for localized Quality of Service routing. In *Proc. of the 10th Intl. Conference on Computer Communications and Networks*, pages 102–107, 2001.

[161] X. Yuan and G. Yang. "Empirical probability based QoS routing". In *Proc. of Intl. Conference on Communications*, pages 1713–1717, 2003.

[162] X. Yuan, W. Zheng, and S. Ding. A comparative study of QoS routing algorithms that tolerate imprecise state information. In *Proc. of Intl. Conference on Computer Communications and Networks*, 2002.

[163] D. Zappala and D. Zhou. "Performance evaluation of path searching heuristics for multicast routing". In *Proc. of Eleventh Intl. Conference on Computer Communications and Networks*, pages 248–254, October 2002.

[164] L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala. RSVP: a new resource reservation protocol. *IEEE Network*, Sept. 1993.

[165] Y. Zhong and X. Yuan. Impact of resource reservation on the distributed multi–path Quality of Service routing scheme. In *Proc. of Intl. Workshop on Quality of Service*, June 2000.

[166] Q. Zhu, M. Parsa, and J. Garcia-Luna-Aceves. A source–based algorithm for delay–constrained minimum–cost multicasting. In *Proc. of INFOCOM*, March 1995.