Generalized Quality-of-Service Routing With Resource Allocation

Ahmed R. Bashandy, *Member, IEEE*, Edwin K. P. Chong, *Fellow, IEEE*, and Arif Ghafoor, *Fellow, IEEE*

***Abstract*—We present a general framework for the problem of quality-of-service (QoS) routing with resource allocation for data networks. The framework represents the QoS parameters as *functions* rather than static metrics. The formulation incorporates the hardware/software implementation and its relation to the allocated resources into a single framework. The proposed formulation allows intelligent adaptation of QoS parameters and allocated resources during a path search, rather than decoupling the path search process from resource allocation. We present a dynamic programming algorithm that, under certain conditions, finds an optimal path between a source and destination node and computes the amount of resources needed at each node so that the end-to-end QoS requirements are satisfied. We present jitter and data droppage analyzes of various rate-based service disciplines and use the dynamic programming algorithm to solve the problem of QoS routing with resource allocation for networks that employ these service disciplines.**

***Index Terms*—Quality-of-service (QoS), rate-based queueing, resource allocation, routing.**

I. INTRODUCTION

NEXT-GENERATION Internet applications involve accessing large volume of multimedia information from remote servers and databases. Transmission of multimedia data over a broadband network requires special support from the underlying routers and switches. The objective of the network support is to guarantee the *quality-of-presentation (QoP)* required by the multimedia client(s) at the destination(s) [27]. To achieve this objective, the network must provide guaranteed *quality-of-service* (QoS) under diverse network conditions and resource constraints. Several researchers have tackled this problem from a variety of aspects, such as end-to-end strategies [1], individual node resource management strategies [2], QoS-routing strategies without resource consideration [16], [22], [25], [31], [32], and QoS routing with resource consideration [7], [19]–[21], [28].

In a data network, the output interface of a node employs a specific packet scheduling algorithm, commonly known as a *service discipline*. The service discipline determines the relation among the QoS parameters provided by the node and the allocated resources. The existence of these relations changes the perspective about the problem of *QoS routing with resource allocation*. In this paper, we present a general formulation of this problem that employs the knowledge of the service discipline to combine QoS routing and resource allocation into a single framework. The formulation exhibits three important features. First, it takes into consideration the relationship between the QoS metrics and the allocated resources. Second, it allows *independent allocation* of different resources coupled with routing. Third, it captures the interrelationship among various QoS metrics. These features are particularly important if the QoS requirements and the resource constraints are specified in such a way that one stringent requirement can be satisfied at the expense of another one that has some degree of tolerance. Based on the proposed formulation, we present a dynamic programming algorithm to find a route between the source and destination nodes and determine the amount of resources along the intermediate nodes to satisfy the QoS requirements.

The QoS requirements for a path can be specified by four parameters: the maximum variation in end-to-end jitter delay (J_{req}), the minimum reliability or percentage of data droppage (Q_{req}), the minimum long term average bandwidth (R_{req}), and the maximum end-to-end delay (D_{req}). Satisfying these requirements involves two aspects: routing and resource allocation. Traditionally, a network is modeled by a directed graph and the QoS parameters are captured by a *fixed* metric(s) assigned to the edges [7], [16], [19]–[22], [25], [26], [28], [32]. The process of route establishment is usually decoupled from the problem of resource reservation and the ability to vary resources and quality metrics. In reality, such metrics are not fixed and can be changed during the process of path hunting. In this paper, we focus on two key network resources, namely; the *bandwidth* and the *buffer space* used in the queues of service disciplines.

This paper is organized as follows. In the next section, we provide an overview of the recent results in the area of QoS routing. In Section III, we present a network model that allows capturing of relation among the node implementation, allocated resources, and the QoS parameters, and specify conditions when a route satisfies the QoS requirements. In Section IV, we present a general mathematical formulation for the problem of QoS routing with resource allocation, analyze the formulation, and present a dynamic programming algorithm to solve this problem. In Section V, we illustrate the applicability of the framework presented in Sections III and IV by examining some of the well-known service disciplines, and derive the relation

Manuscript received November 1, 2003; revised May 15, 2004. This research was supported in part by the National Science Foundation (NSF) under Grant CISE/EIA-9972883, Grant ECS-0098089, Grant ANI-0099137, and Grant ANI-0207892.

A. R. Bashandy is with Cisco Systems, San Jose, CA 95134 USA (e-mail: bashandy@cisco.com).

E. K. P. Chong is with the Department of Electrical and Computer Engineering, Colorado State University, Fort Collins, CO 80521 USA (e-mail: echong@goku.engr.colostate.edu).

A. Ghafoor is with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907 USA (e-mail: ghafoor@ecn.purdue.edu).

Digital Object Identifier 10.1109/JSAC.2004.839423

between resources and the level of QoS provided by these disciplines. Section VI summarizes the distinction between the research results presented in this paper and earlier work in this area. In Section VII, we conclude the paper and propose some future work.

II. Background

The problem of QoS routing has been the center of attention in both academic and industrial communities for some time. The research in this area can be broadly divided into two areas: QoS routing without resource consideration and QoS routing with resource consideration. In this section, we provide an overview of some of the relevant results in both areas.

In the area of QoS routing without resource consideration [16], [18], [22], [25], [26], [31], [32], [34], the problem is typically formulated as follows. The network is modeled as a directed graph $G=(V,E)$. QoS parameters are captured as functions that map the set of edges E to a set of nonnegative integers, usually referred to as *weights* or *metrics*. These metrics represent the level of QoS provided by each edge. In determining these metrics, interaction among resources is ignored. Instead, each metric is assumed to be an independent numerical value that can be aggregated using ordinary addition. The solution to the QoS routing problem is then a simple path hunting between the source and destination nodes such that the path satisfies all the QoS requirements. The QoS routing problem formulated in this manner is equivalent to a constrained shortest path problem, which is NP-complete [14]. Some researchers have suggested approximate algorithms with time complexity that grows nonexponentially as the solution becomes more accurate. Other have provided pseudopolynomial algorithms with time complexity dependent on the numerical values of the arguments. In some cases, exact algorithms with exponential time complexity have been proposed. In [32], Wang and Crowcroft divided metrics into three categories: additive, multiplicative, and concave (bottleneck). They showed that a combination of one additive or multiplicative metric and one concave metric is not NP-complete, while a combination of more that one additive and/or multiplicative metric is NP-complete. Usually, researchers attempt to solve the optimization version of the problem, which is to find a path that minimizes one of the metrics, while satisfying the constraints on the others.

For the second problem that deals with QoS routing with resource consideration [7], [19], [21], [28], the network is modeled as a directed graph with QoS parameters captured as metrics assigned to each edge. Resources are also defined as numerical values associated with each edge. Based on the available resources, the traffic description, and, in some cases, the employed service discipline, the metrics associated with each edge are computed. In some cases, only a single resource is considered, which is bandwidth [7], [20], [28]. In some studies, the problem of reliability (data droppage) is not addressed, even though a service discipline is assumed to be employed [19], [21]. Most of the existing work in this area applies to special cases and does not address the general formulation of the problem of QoS routing with resource allocation. In addition, in the existing work both the allocated resources and the metrics are assumed to be *static* and cannot be changed during the process of path hunting.

The discussion above highlights the observation that earlier research in the area of QoS routing attempts to solve the problem using one of the following two approaches: the first approach models the problem as a constrained shortest path without considering any resources. In the second approach, fixed metrics are assigned to each link based on the available resources and the traffic description prior to applying a path search algorithm. However, the results of these approaches do not incorporate the effect of multiple routers, connected in tandem, on the QoS parameters of a path, rather addition is used to aggregate the metrics. Also, these results do not include the possibility of tuning the allocated resources *during the path search* in order to establish a route that satisfies all the QoS requirements collectively. For example, suppose that the jitter requirement is very tight while the data droppage requirement has some flexibility. In this case, it may be essential to reduce the queue size to achieve reduced jitter caused by the queueing delay, even though abundant memory may be available. In other words, assigning the metrics prior to the path searching algorithm may preclude finding a path that satisfies the QoS requirements, even though one such path may exist. Varying QoS metrics, if required, is usually carried out prior to a path search. In summary, existing results do not provide a single framework that allows routers and switches to utilize the combined knowledge of hardware/software implementation and the available resources to take intelligent routing decisions.

III. Network Model

As mentioned in Section II, the values of the QoS parameters are sometimes computed based on the available resources, the traffic shape, and service disciplines. These values are computed prior to searching a path and are not allowed to change during the path search. The main drawback of this approach is that it does not fully incorporate the knowledge of the network elements implementation and the traffic shape inside the network in relation to the allocated resources. In this section, we extend the graph theoretic model to incorporate this knowledge. The extension to the graph theoretic model is based on the concept of *jitter graph* proposed in [4]. A jitter graph consists of *jitter nodes* and *jitter links*. We proceed to discuss these two entities and the way they abstract the physical characteristics of a data network. Based on the jitter graph model, we identify the relationships between the QoS parameters and different resources inside a network.

A. Concept of Jitter Graph

A jitter graph consists of a set of *jitter nodes V* and a set of *jitter links E*. The set of nodes V represents the processors and the cross-connections residing in switches, routers, computers, etc. The set of links E represents the service disciplines employed at the output interface of the switches and the physical communication lines to which they are connected. Traffic traverses such networks through established channels. A channel can be viewed as a path through a jitter graph, which is sequence of jitter nodes connected via jitter links. The correspondence between the physical network and proposed model is depicted in Fig. 1.

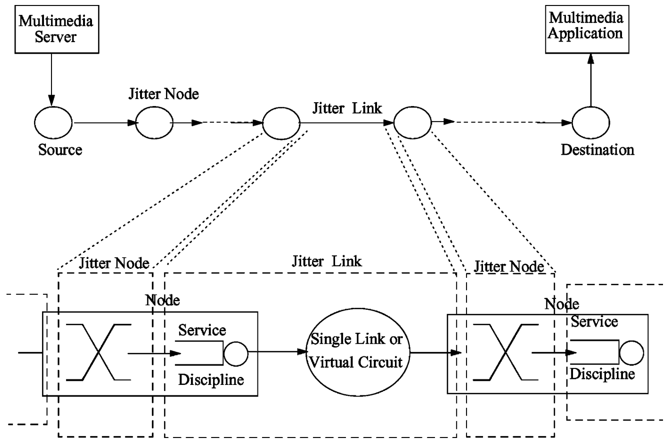


Fig. 1. Network model.

A jitter link (u, v) connecting the two nodes u and v is characterized by five basic attributes: 1) maximum propagation delay D_{uv}; (2) minimum propagation delay d_{uv}; 3) maximum transmission capacity C_{uv}; 4) reliability Q_{uv}; and 5) jitter J_{uv}.

The maximum propagation delay, D_{uv}, is the *maximum* time difference between transmitting a packet by node u and receiving this packet at the node v. This parameter captures the end-to-end delay of point-to-point connections that employ service disciplines with bounded queueing delay, such as PGPS (WFQ) [23], $\mathrm{WF}^2\mathrm{Q}$ [6], and VC [37]. The parameter D_{uv} can also represent a statistical bound for networks with arbitrarily large delays. The attribute d_{uv} is the *minimum* time difference between transmitting a packet by node u and receiving it at the input of node v. This attribute captures the physical propagation delay and the service time, which is usually the size of a packet in bits divided by the physical capacity of the link in bits per second. The third attribute C_{uv} represents the physical transmission capacity of a link in bits per second. The fourth attribute Q_{uv} is the minimum reliability of the jitter link (u, v). Formally, let n_{tran} be the total number of packets transmitted by the jitter node u toward the jitter node v along the jitter link (u, v) so far. Let n_{rec} be the total number of packets received at the jitter node v. The jitter link (u, v) is said to provide a reliability value of Q_{uv} if $n_{\text{rec}}/n_{\text{tran}} \geqslant Q_{uv}$.

Several researchers [12], [30], [36] define jitter as the difference between the maximum and minimum possible propagation delays across the link (u, v), that is $J_{uv} = D_{uv} - d_{uv}$. Although this is a valid and widely accepted definition—and we use it in Section V—for the sake of generality, we provide a less restrictive definition of jitter. We define jitter for a jitter link (u, v) as the variation in the delay between submitting a packet for transmission at the node u till it is completely received at the input of node v. For a certain class of applications, jitter can be taken to be the maximum difference. For some class of applications, statistical measures, such as the variance of the delay along the link, may be more appropriate.

We make further assumptions to facilitate the analysis.

- The reliability value Q_{uv} of a jitter link (u, v) is affected only by data droppage occurring at the output service discipline as a result of buffer overflow. This assumption is justified by the fact that packet loss due to corruption during signal propagation along optical fibers or coaxial cables tends to be negligible when compared with packet droppage in nodes.
- We assume that a switch is nonblocking [17], that is, the switch fabric has enough processing power to relay packets from the input to the output interface. Hence, we can assume that jitter and delay result only from the queueing, service time, and/or propagation delay.

B. Effect of Resources on the QoS Parameters

Based on the jitter graph model outlined in Section III-A, we identify two types of resources:

- $b(u, v)$: the buffer allocated for a given data stream in the service discipline residing in the node u and transmitting data to the node v;
- $r(u, v)$: the bandwidth allocated for a given data stream along the communication link between the nodes u and v represented by the jitter link (u, v).

The above resources are restricted by the maximum values imposed by the hardware and/or software limitations of a node. These restrictions are specified as follows:

- $B(u, v)$, the total buffer space available for the data stream under consideration to the service discipline transmitting packets from the node u to the node v;
- C_{uv}, the physical link capacity;
- $C_{uv}^{\max} \leqslant C_{uv}$, the available bandwidth or an administrative limit on the maximum amount of bandwidth available for allocation to the data stream under consideration along the jitter link (u, v).

Due to the existence of the service discipline, the jitter, reliability, and end-to-end delay are functions of the resources and the traffic shape. Hence, we have

$$0 < J_{uv} = f_{J_{uv}}\left(b(u, v), r(u, v)\right) < \infty$$
$$0 \le Q_{uv} = f_{Q_{uv}}\left(b(u, v), r(u, v)\right) \le 1$$
$$0 < D_{uv} = f_{D_{uv}}\left(b(u, v), r(u, v)\right) < \infty$$

where the functions $f_{J_{uv}}$, $f_{Q_{uv}}$, and $f_{D_{uv}}$ are specified by the traffic shape and the service discipline employed along the jitter link (u, v). In Section V, we provide a derivation of functions $f_{J_{uv}}$, $f_{Q_{uv}}$, and $f_{D_{uv}}$ for various service disciplines.

The next step is to specify QoS parameters of a path p in a jitter graph. Traditionally, the QoS parameters of a path are assumed to be the sum or the product [9], [32] of the QoS parameters on each individual link. However, in general, a simple summation or product may not provide accurate tight bounds on the QoS parameters of a path. For a given path $p = \langle v_0, \ldots, v_n \rangle$, the jitter $J_p(p)$, the reliability $Q_p(p)$, the maximum delay $D_p(p)$, and the bandwidth $r_p(p)$ should be some functions, with arguments corresponding to the amount of resources allocated along the path. That is

$$\begin{aligned} J_p(p) &= f_J\left(b(v_0, v_1), r(v_0, v_1), \ldots, \right. \\ &\qquad \left. b(v_{n-1}, v_n), r(v_{n-1}, v_n)\right) \\ Q_p(p) &= f_Q\left(b(v_0, v_1), r(v_0, v_1), \ldots, \right. \\ &\qquad \left. b(v_{n-1}, v_n), r(v_{n-1}, v_n)\right) \\ D_p(p) &= f_D\left(b(v_0, v_1), r(v_0, v_1), \ldots, \right. \\ &\qquad \left. b(v_{n-1}, v_n), r(v_{n-1}, v_n)\right) \\ r_p(p) &= f_R\left(b(v_0, v_1), r(v_0, v_1), \ldots, \right. \\ &\qquad \left. b(v_{n-1}, v_n), r(v_{n-1}, v_n)\right) \end{aligned} \tag{1}$$

where the functions f_J, f_Q, f_D, and f_R are specified by the interaction between the service discipline in a path and the traffic shape. As mentioned in Section I, data streams can specify their QoS requirements in terms of four parameters: the maximum jitter $J_{\rm req}$, the minimum reliability $Q_{\rm req}$, the maximum end-to-end delay $D_{\rm req}$, and the minimum longterm bandwidth $R_{\rm req}$. A path p is said to satisfy the QoS requirements of the stream if

$$J_p(p) \leqslant J_{\rm req}$$
$$Q_p(p) \geqslant Q_{\rm req}$$
$$D_p(p) \leqslant D_{\rm req}$$

and

$$r_p(p) \geqslant R_{\rm req}. \tag{2}$$

Earlier studies [6], [13], [15], [22]–[24], [32], [35] have considered the bandwidth $r_p(p)$ as a *concave* or *bottleneck* function of the path. In this paper, we adopt the same convention. Hence

$$r_p(p) = \min_{(u,v)\in p} r(u,v). \tag{3}$$

For convenience, we define the parameter Z_{uv} for a jitter link (u,v) by

$$Z_{uv} = -\ln(Q_{uv}) \tag{4}$$

which represents the log-reliability of the link. Similarly, we define the path log-reliability parameter $Z_p(p)$ for a path p by

$$Z_p(p) = -\ln\left(Q_p(p)\right). \tag{5}$$

Hence, a path p satisfies the reliability requirement $Q_{\rm req}$ if $Z_p(p) \leqslant Z_{\rm req}$, where

$$Z_{\rm req} = -\ln(Q_{\rm req}). \tag{6}$$

This parameter is adopted from the reliability linearization procedure used in [32].

In the rest of the paper, we pay a special attention to information retrieval and broadcast types of applications such as Internet-based radio and TV broadcast, retrieval of preorchestrated multimedia documents, and multimedia Webservices. For this class of applications, *jitter*, J_{uv}, and J_p play a more critical role in terms of end-to-end QoS management, rather than end-to-end delay, D_{uv}, and D_p. Consequently, for the rest of this paper, we focus on the problem of finding a path in a jitter graph for a specific data stream such that the requirements ($J_{\rm req}$, $Q_{\rm req}$, and $R_{\rm req}$) are satisfied.

Next, we discuss the problem of QoS routing with resource allocation and formulate it as a nonlinear program.

IV. QoS Routing With Resource Allocation

In this section, we start by defining the problem of QoS routing with resource allocation in light of the network model discussed in Section III. Then, we present the *path feasibility*, the *subpath addition*, and *nonincreasing reliability* conditions that lead to a polynomial time algorithm for solving the routing problem. At the end of this section, we present a dynamic programming algorithm that solves the problem in polynomial time.

A. Problem Definition

Consider a jitter graph $G = (V, E)$ and two nodes s, $d \in V$ representing the source and destination nodes, respectively. For each jitter link (u,v), we have the following resources: the buffer size $b(u,v)$, where $0 \leqslant b(u,v) \leqslant B(u,v)$, and the bandwidth $r(u,v)$, where $0 \leqslant r(u,v) \leqslant C_{uv}^{\max} \leqslant C_{uv}$.

Definition IV.1: For a given path path p_j, the *path-resource problem* is defined as follows:

$$\begin{aligned} & Z_p^*(p_j) = \min\left\{Z_p(p_j)\right\} \\ \text{subject to} \quad & J_p(p_j) \leqslant J_{\rm req} \\ & 0 \leqslant b(u,v) \leqslant B(u,v)\ \forall (u,v)\in p_j \\ & R_{\rm req} \leqslant r(u,v) \leqslant C_{uv}^{\max}\ \forall (u,v)\in p_j. \end{aligned} \tag{7}$$

Note that the constraint $R_{\rm req} \leqslant r(u,v)$ results from adopting the convention that the bandwidth of a path p_j is a bottleneck function of the path as specified in (3).

A solution to the path-resource problem is represented by numerical values assigned to the decision variables $b(u,v)$ and $r(u,v)$ for all $(u,v) \in p_j$. Specifying these values provides a methodology of *allocating resources* along a path p_j for a given data stream so as to satisfy its QoS requirements. Based on (7), we define a *feasible path* as follows.

Definition IV.2: A path p is said to be *feasible* for the QoS requirements specified by ($R_{\rm req}$, $J_{\rm req}$) if the feasible set in the path-resource problem defined in (7) for the path p is nonempty.

Definition IV.3: Let $P(s,d)$ be the set of all paths between the nodes s and d. The problem of *QoS routing with resource allocation* is defined as follows:

$$\zeta^*(s,d) = \min_{p_j\in P(s,d)} \left\{Z_p^*(p_j)\right\}. \tag{8}$$

This definition of the QoS routing problem is slightly nontraditional as the problem of QoS routing, as defined in the literature, is to find a feasible solution rather than finding an optimized QoS metric subject to a set of constraints specified on the other metrics. However, the optimization problem does provide a solution for the feasibility problem.

The QoS routing problem in (8) has certain properties that makes it distinct from the routing problems addressed in the literature. The key distinct features of our routing problem are listed as follows.

- Our definition does not assume that metrics are combined through addition. Rather, the QoS attributes of a path are *functions* that depend on the QoS resulted from the allocated resources and the individual links.
- Our definition incorporates the relation between metrics and resources in addition to the interaction between routers. A solution to the problem of QoS routing with resource allocation defined by (8), if it exists, depends on the local functions Z_{uv} and J_{uv} for each jitter link (u,v), and the global functions Z_p, Q_p, and r_p constructed by multiple jitter links connected in tandem.

Routing problem in such a general form is intractable. To solve this problem, we need to impose certain restrictions on the parameters Z_{uv}, J_{uv}, Z_p, J_p, and r_p. In the next subsection, we present the *path feasibility*, the *subpath addition*, and the *nonincreasing reliability* conditions, which result in a polynomial time algorithm for solving this problem. In Section V, we analyze various service disciplines to derive instances for the problem of QoS routing with resource allocation based on the properties of these service disciplines.

B. Optimality Conditions

In this subsection, we present sufficient conditions for the algorithm presented in Section IV-C to solve the problem of QoS routing with resource allocation defined in (8). These optimality conditions capture some properties of the hardware and software implementation of the routers and switches.

Definition IV.4: Let $G=(V,E)$ be a jitter graph. Consider the two paths p_1 and p_2 such that the last node in p_1 and the first node in p_2 are identical. The binary operator $\oplus$ is defined as follows: $q=p_1 \oplus p_2$ returns the sequence of nodes and edges q constructed by concatenating the path p_2 to the end of the path p_1.

Note that the operator $\oplus$ produces a *walk* rather than a path or a trail [33], where a walk is a sequence $\langle v_0, e_1, v_1, e_2, \ldots, e_n, v_n \rangle$ such that $e_i=(v_{i-1}, v_i)$, while a path is walk with distinct vertices.

The correctness of the proposed algorithm is based on the *path feasibility*, the *subpath addition*, and the *nonincreasing reliability* conditions defined below.

Definition IV.5: Let G be a jitter graph with the parameters Z_{uv}, J_{uv}, and $r(u,v)$ defined for every edge $(u,v) \in E$ and $Z_p(p)$, $J_p(p)$, and $r_p(p)$ defined for every path. Let the QoS requirements for the bandwidth and jitter be defined by be the *positive* pair $(J_{\mathrm{req}}, R_{\mathrm{req}})$. The graph G is said to satisfy the *path feasibility* condition if, for each path p such that $r_p(p) \geqslant R_{\mathrm{req}}$, the requirement $J_p(p) \leqslant J_{\mathrm{req}}$ can be satisfied for arbitrarily small positive values of J_{req}.

The path feasibility condition may appear to be unsatisfiable. However, the intuition behind it is as follows. For most queueing systems, the queueing delay, which is the main source of jitter, can be reduced by increasing the bandwidth or decreasing the queue size. The bandwidth cannot be increased arbitrarily due to the physical and the administrative limits of a node. However, in most of the cases, the queue length can be reduced to an arbitrarily small size to satisfy an arbitrarily small jitter requirement.

Definition IV.6: Let G be a jitter graph with the parameters Z_{uv}, J_{uv}, and $r(u,v)$ defined for every edge $(u,v) \in E$ and $Z_p(p)$, $J_p(p)$, and $r_p(p)$ defined for every path p. Consider two paths $p_1(s,w)$ and $p_2(s,w)$ such that $Z_p^*(p_1(s,w)) \leqslant Z_p^*(p_2(s,w))$, where $Z_p^*(p)$ is the value of the optimum cost function of the path-resource problem defined in (7). Consider $(w,x) \in E$. Let $q_1(s,x)=p_1(s,w) \oplus (w,x)$ and $q_2(s,x)=p_2(s,w) \oplus (w,x)$. Let both paths $q_1(s,x)$ and $q_2(s,x)$ be feasible. The jitter graph G is said to satisfy the *subpath addition condition* if

$$Z_p^*\left(q_1(s,x)\right) \leqslant Z_p^*\left(q_2(s,x)\right).$$

The intuition behind the subpath addition condition is that, in some cases, concatenating the same edge to two paths results in the same reduction in reliability.

Definition IV.7: Let G be a jitter graph with the parameters Z_{uv}, J_{uv}, and $r(u,v)$ defined for every edge $(u,v) \in E$ and $Z_p(p)$, $J_p(p)$, and $r_p(p)$ defined for every path p. Let $p(s,w)$ be a path from s to w. Let $q(w,x) \in E$ be a path from w to x. The jitter graph G is said to satisfy the *nonincreasing reliability condition* if

$$Z_p^*\left(p(s,w)\right) \leqslant Z_p^*\left(p(s,w) \oplus q(w,x)\right). \tag{9}$$

The nonincreasing reliability condition defined by the relation (9) is satisfied for almost all practical networks. This is because the probability of data droppage in the additional path $q(w,x)$ increases the probability of data droppage for the whole path, unless such a path employs an error correction technique, which rarely occurs for high volume multimedia traffic.

C. QoS Routing Algorithm

In this section, we present a dynamic programming algorithm to solve the problem of QoS routing with resource allocation defined in (8). The algorithm defined by the following recursive relation:

$$z^{k+1}(s,x)=\min\Big\{z^k(s,x), \min_{w \in \mathrm{adj}^{-1}(x)} \left\{Z_p^*\left(p^k(s,w) \oplus (w,x)\right)\right\}\Big\} \tag{10}$$

where $p^k(s,x)$ is the path computed at the k^{th} iteration and $z^k(s,x)=Z_p^*(p^k(s,x))$. The derivation of (10) can be found in Appendix A-1.

Based on (10), the algorithm **QoS-Routing** specified in Algorithm IV.1 finds a solution to the problem of QoS routing with resource allocation.

```
Algorithm IV.1: QoS-Routing(G, s, J_req, Z_req, R_req)
Pre-Process(G)
▶ Initialization Loop
For each x ∈ adj(s)  z^1(s,x) := Path-Opt(G,x,s)
For each x ∈ adj^-1(s)  π^1(x) := x
For each x ∉ adj(s)  z^1(s,x) := ∞
▶Main Loop
for k := 1 → |V| − 2
  for each node x ∈ V − {s}
    z^{k+1}(s,x) := z^k(s,x)
     π^{k+1}(x) := π^k(x)
     for each node w ∈ adj^-1(x)
       π_old := π^{k+1}(x)
       π^{k+1}(x) := w
       temp := Path-Opt(G,x,s)
       if temp < z^{k+1}(s,x)
         z^{k+1}(s,x) := temp
     else
         π^{k+1}(x) := π_old
```

Before we analyze this algorithm, we point out an important issue. The algorithm is given in a general form, and relies on the subalgorithms **Pre-Process** and **Path-Opt**. Prior to executing the main loop, the subalgorithm **Pre-Process** performs some preliminary processing such as deleting edges with insufficient resources. The subalgorithm **Path-Opt** solves the path-resource problem defined in (7). Implementations of these subalgorithms depend on the characteristics of the network elements (routers and switches) employed by the network and the interaction between these elements when they are connected in tandem. The properties and interaction are completely specified by the functions Z_{uv}, J_{uv}, Z_p, J_p, and r_p. In Section V, we examine some of the rate-based service disciplines to derive the functional forms for Z_{uv}, J_{uv}, Z_p, J_p, and r_p and, hence, produce concrete implementations for the subalgorithms **Pre-Process** and **Path-Opt**.

The analysis of the proposed **QoS-Routing** algorithm is as follows.

1) *Time Complexity*: Assume that the time complexity of **Path-Opt** is K and that of **Pre-Process** is I.
   In the initialization loop, all the edges are scanned once. In the main loop, all the edges are scanned $|V|-2$ times. Hence, the running time of the algorithm is $O(K|V||E|+I)$.
2) *Correctness of the QoS Routing Algorithm*: To prove the correctness of the algorithm, we need to show that the application of this algorithm on a jitter graph yields a solution to the problem of QoS routing with resource allocation as defined by (8), given that the path feasibility, the subpath addition, and the nonincreasing reliability conditions are satisfied. Such a proof involves two aspects. First, we need to prove that the subgraph $G_\pi = (V, E_\pi)$ with $E_\pi = \{(\pi(x), x) : \pi(x) \neq \text{Nil}\}$ is acyclic. Second, we need to prove that, at the termination of the algorithm, $z^{|V|-1}(s,x) = \zeta^*(s,x)$ for all nodes $x \in V$ that are reachable from s.

The next theorem provides a formal statement of the correctness of Algorithm IV.1.

Theorem IV.1: Given the jitter graph $G = (V, E)$ with parameters J_{uv}, Z_{uv}, J_p, Z_p, r_p as defined in (1) and (3). Assume that the path feasibility, the subpath addition, and the nonincreasing reliability conditions given in Definitions IV.5, IV.6, and IV.7, respectively, are satisfied. Assume that the algorithm **Path-Opt** solves the path-resource problem defined in (7) if a path exists or returns ∞ otherwise. Then, at the termination of Algorithm IV.1, the following statements are true:

1) $z^{|V|-1}(s,x) = \infty$ for every node $x \in V$ such that there is no path from the source node s to x;
2) $\pi^{|V|-1}(s,x) = \text{Nil}$ for every node $x \in V$ such that there is no path from the source node s to x;
3) $z^{|V|-1}(s,x) = \zeta^*(s,x)$ for every node $x \in V$ such that there exists a path from the source node s to x;
4) the graph $G_\pi = (V, E_\pi)$ such that $E_\pi = \{(\pi^{|V|-1}(x), x) : \pi^{|V|-1}(x) \neq \text{Nil}\}$ contains the path $p^*(s,x)$ for every node $x \in V$ if there exists a path from the source node s to x.

Proof: Due to space limitations, we only provide an outline of the proof. Refer to [3, Sec. 5.7] for details of the proof. The proof is carried out by induction on the variable k used in Algorithm IV.1. In the induction basis, the theorem is proved for $k = 1$. That is, the theorem is proved for one edge only. In the induction step, we prove the theorem for $k+1$ assuming that it is correct for k.

In this section, we have presented an algorithm that solves the problem of QoS routing with resource allocation based on the path feasibility, the subpath addition, and the nonincreasing reliability conditions, which, in effect, impose certain relationships between the different QoS parameters and the available resources. In the next section, we examine some of the rate-based service disciplines to derive relationships between the QoS parameters and the available resources. We illustrate the conditions under which the path feasibility, the subpath addition, and the nonincreasing reliability conditions are satisfied.

V. Applications to Rate-Based Service Disciplines

In the previous sections, we have presented a general framework that captures the effect of service disciplines, network resources, and the interaction of nodes (connected in tandem), on the level of QoS provided by a network. The proposed framework provides a general formulation of the QoS routing problem with resource allocation. The key feature of the framework is the representation of the QoS parameters as functions of resources rather than fixed value metrics. In this section, we demonstrate the applicability of the framework by examining some of the rate-based service disciplines and deriving functional forms for J_{uv}, Q_{uv}, J_p, Q_p, and r_p. The end-to-end delay D_{uv} and D_p can be easily derived from the jitter J_{uv} and J_p.

By studying several service disciplines, such as the ones discussed in [6], [13], [15], [23], [24], and [37], it can be noticed that the jitter and data losses in a network depend on three factors: 1) the traffic shape; 2) the interaction between service disciplines connected in tandem; and 3) the allocated resources. To elaborate, let the *arrival function* at the output queue of the node u be denoted by $A^{(i)}_{\text{out}}(t,u)$. Then, for the session i, the jitter link (u,v) has well defined functions $J^{(i)}_{uv}$ and $Q^{(i)}_{uv}$ that depend on the function $A^{(i)}_{\text{out}}(t,u)$, the resources allocated at the node u, and the properties of the communication link between the nodes u and v. *Leaky bucket* constrained sources are commonly used to model deterministically bounded data sources [6], [8], [15], [23], [29]. The analysis and results obtained in the literature are generally based on the assumption that data sources are leaky bucket constrained. In the usual leaky bucket notation, with σ and ρ representing the bucket size, and average traffic rate, respectively, an arrival function $A^{(i)}_{\text{out}}(t,u)$ is said to conform to (σ,ρ), or $A^{(i)}_{\text{out}}(t,u) \sim (\sigma,\rho)$, if, for any interval $(\tau,t]$, $A^{(i)}_{\text{out}}(t,u) - A^{(i)}_{\text{out}}(\tau,u) \leqslant \sigma + \rho(t-\tau)$. For the rest of this section, we use leaky bucket constrained data sources to model the traffic entering a network.

We examine four service disciplines in this section: generalized processor sharing (GPS) [10], [23], packet by packet generalized processor sharing (PGPS) [23], [24], worst case fair weighted fair queueing (WF^2Q) [6], and self-clocked fair queueing (SCFQ) [15]. It is shown in [13], [35] that the resource requirements and QoS behavior of virtual clock (VC) [37] is identical to PGPS. Hence, we can apply the results for networks employing the PGPS discipline to the ones employing the VC technique. The GPS, PGPS, WF^2Q, and SCFQ service disciplines have some common characteristics. For example, they all provide an upper bound on the queueing delay, which is equivalent to the link jitter J_{uv}. Also, the analysis and results obtained for these service disciplines assume the leaky bucket constrained data sources. Note, the aforementioned service disciplines are analyzed to illustrate the generality and applicability of the definitions and algorithm defined in Section IV.

Let

- $q^{(i)*}(u,v)$ be the maximum number of backlogged bits from the session i in the output service discipline queue of the node u during the life time of the session i;
- $q^{(i)*}(n)$ be the maximum number of bits backlogged inside the path $p(n) = \langle v_0, \ldots, v_n \rangle$;
- K_{uv} be the number of sessions traversing the link (u,v);
- $b^{(i)}(u,v)$ be the size of the buffer in bits allocated for the i^{th} session at the output service discipline of the node u, which transmits packets to the subsequent node v;
- $r^{(i)}(u,v)$ be the bandwidth (in bits per second) allocated for the i^{th} session along the communication link (u,v);
- C_{uv} be the capacity (in bits per seconds) of the link (u,v) connecting the nodes u and v;
- L be the maximum packet size in bits.

The values of $b^{(i)}(u,v)$ and $r^{(i)}(u,v)$ represent the resources allocated for the i^{th} session at the output service discipline of the node u and along the jitter link (u,v). Hence, for each session i, the parameters $J_{uv}^{(i)}$ and $Q_{uv}^{(i)}$ of the jitter link (u,v), representing the service discipline, are functions of the resources $b^{(i)}(u,v)$ and $r^{(i)}(u,v)$. Also, for a path p, the parameters $J_p^{(i)}$, $Q_p^{(i)}$, and $r_p^{(i)}$ are functions of the resources $b^{(i)}(u,v)$ and $r^{(i)}(u,v)$ for each link $(u,v) \in p$.

For the GPS, PGPS, WF^{2}Q, and SCFQ disciplines, we perform the following tasks.

- Analyze the service discipline to determine $J_{uv}^{(i)}$, $Q_{uv}^{(i)}$, $J_p^{(i)}$, $Q_p^{(i)}$, and $r_p^{(i)}$, as functions of the resources.
- *Specialize* the path-resource problem defined in (7) to reflect the properties of the service discipline. The resulting path-resource and QoS routing problems are considered to be *instances* of the general problems defined in (7) and (8), respectively.
- Solve the instance of the path-resource problem.
- Derive a solution for the instance of the path resource problem for the path $p(n+1) = \langle v_0, v_1, \ldots, v_n, v_{n+1} \rangle$ given that we have a solution for the subpath $p(n) = \langle v_0, \ldots, v_n \rangle$. The objective is to reduce the time complexity of the subalgorithm **Path-Opt**.
- Discuss the optimality conditions and describe how to use Algorithm IV.1 to solve the *instance* of the problem of QoS routing with resource allocation.
- Compute the time complexity needed to solve the problem of QoS routing with resource allocation corresponding to the service discipline under consideration.

In the following sections, we analyze the above mentioned service disciplines.

A. Generalized Processor Sharing (GPS)

Generalized processor sharing (GPS) was first proposed in [10] under the name weighted fair queueing (WFQ). It is also known as the fluid fair queueing (FFQ) [15], [35]. GPS has also been analyzed in [23] and [24], where the authors provide guaranteed bandwidth and worst case queueing delay on a session basis. In GPS, packets are assumed to be infinitesimally divisible and the server can serve multiple packets simultaneously.

1) The Instance of the Path-Resource Problem: The derivation of the QoS parameters for the ith session along the path p consisting of GPS servers connected in tandem is given in Appendix A-2. Given $Q_p^{(i)}$, $J_p^{(i)}$, and $r_p^{(i)}$, we can define the path-resource problem for a path consisting of GPS servers connected in tandem by the following optimization problem:

$$Q_p^{(i)*}(p(n)) = \max \left\{ \min \left\{ \frac{\min_{1 \leqslant j \leqslant n} \left\{ b^{(i)}(v_{j-1}, v_j) \right\}}{\sigma^{(i)}}, 1 \right\} \right\}$$

subject to

$$\begin{aligned} &J_p^{(i)}(p(n)) = \frac{\min \left\{ \sum_{j=1}^{n} \left\{ b^{(i)}(v_{j-1}, v_j) \right\}, \sigma^{(i)} \right\}}{r_p^{(i)}(p(n))} \leqslant J_{\text{req}} \\ &0 < b^{(i)}(v_{j-1}, v_j) \leqslant B^{(i)}(v_{j-1}, v_j), \quad j \in \{1, \ldots, n\} \\ &\rho^{(i)} \leqslant r_p^{(i)}(p(n)) \leqslant \min_{1 \leqslant j \leqslant n} C_{v_{j-1}, v_j}^{\max} \end{aligned} \tag{11}$$

where $B^{(i)}(v_{j-1}, v_j)$ represents the maximum buffer size allowed for the i^{th} session on the link (v_{j-1}, v_j). The decision variables of this problem are $b^{(i)}(v_{j-1}, v_j)$, $j \in \{1, \ldots, n\}$, and $r_p^{(i)}(p(n))$. The bandwidth is the minimum bandwidth along the path. If we choose the real number $\phi^{(i)}(u,v)$ representing the fraction of the link bandwidth assigned to session i (see Appendix A-2), such that $g^{(i)}(u,v) = C_{uv}^{\max}$, then

$$r_p^{(i)}(p(n)) = \min_{1 \leqslant j \leqslant n} C_{uv}^{\max}. \tag{12}$$

It can easily be shown that if the value of the function $Q_p^{(i)}(p(n))$ is optimum, then the value of $r_p^{(i)}(p(n))$ is at least $\min_{0 \leqslant j \leqslant n} C_{uv}^{\max}$ as given in (12). After finding the optimum value of the decision variable $r_p^{(i)}(p(n))$, we are left with the decision variables $b^{(i)}(v_{j-1}, v_j)$, $j \in \{1, \ldots, n\}$.

2) Solving the Instance of the Path-Resource Problem Given the Solution for $p(n)$: Define the variable $maxQ = \min\{\sigma^{(i)}, \sum_{j=1}^{n} B^{(i)}(v_{j-1}, v_j)\}$. The solution to the optimization problem in (11) has three cases.

Case 1)

$$J_{\text{req}} \geqslant maxQ / r_p^{(i)}(p(n)).$$

Case 2) $J_{\text{req}} < \max Q / r_p^{(i)}(p(n))$ and $\min_{1 \leqslant j \leqslant n} B^{(i)}(v_{j-1}, v_j) \geqslant \sigma^{(i)}$.

Case 3) $J_{\text{req}} < \max Q / r_p^{(i)}(p(n))$ and $\min_{1 \leqslant j \leqslant n} B^{(i)}(v_{j-1}, v_j) < \sigma^{(i)}$.

In Case 1), the number of bits accumulated in the queues inside the path cannot possibly increase the jitter beyond the maximum required value J_{req}. Hence, an optimum solution can be obtained by having $b^{(i)}(v_{j-1}, v_j) = \min\{\sigma^{(i)}, B^{(i)}(v_{j-1}, v_j)\}$ for all $j \in \{1, \ldots, n\}$ and $Q_p^{(i)*}(p(n)) = \min_{1 \leqslant j \leqslant n} (b^{(i)}(v_{j-1}, v_j) / \sigma^{(i)})$.

For Cases 2) and 3), the maximum number of queued bits can result in increasing the jitter along the path beyond the maximum allowed value $J_{\rm req}$. Hence, the amount of buffer space $b^{(i)}(v_{j-1}, v_j)$, $j \in \{1, \dots, n\}$, must be reduced to avoid violation of the jitter requirement. Therefore, we need to maximize the value of $Q_p^{(i)}(p(n))$ such that $J_{\rm req} r_p^{(i)}(p(n)) = \sum_{j=1}^{n} b^{(i)}(v_{j-1}, v_j)$. The optimum solution for this case is given by

$$b^{(i)}(v_{j-1}, v_j) = \min\left\{B^{(i)}(v_{j-1}, v_j), \frac{J_{\rm req} r_p^{(i)}(p(n))}{n}\right\}, \quad j \in \{1, \dots, n\} \tag{13}$$

$$Q_p^{(i)*}(p(n)) = \frac{\min\limits_{1 \leqslant j \leqslant n} b^{(i)}(v_{j-1}, v_j)}{\sigma^{(i)}}. \tag{14}$$

To show that the solution provided in (13) and (14) indeed yields the optimum value of $Q_p^{(i)}(p(n))$ for Cases 2) and 3), we need to consider two cases. The first case is when $\min_{1 \leqslant j \leqslant n} B^{(i)}(v_{j-1}, v_j) < J_{\rm req} r_p^{(i)}(p(n))/n$. Let $k = \arg\min_{1 \leqslant j \leqslant n} b^{(i)}(v_{j-1}, v_j) < (J_{\rm req} r_p^{(i)}(p(n))/n)$. In this case, nothing can be done because increasing any other value $b^{(i)}(v_{j-1}, v_j)$, $j \neq k$, will not affect the value of $b^{(i)}(v_{k-1}, v_k)$, which is dictated by $B^{(i)}(v_{k-1}, v_k)$. The second case is when $B^{(i)}(v_{j-1}, v_j) > (J_{\rm req} r_p^{(i)}(p(n))/n)$ for all $j \in \{1, \dots, n\}$. In this case, if we increase the value of $b^{(i)}(v_{m-1}, v_m)$ for some $m \in \{1, \dots, n\}$, we must decrease the value of $b^{(i)}(v_{l-1}, v_l)$ for another $l \in \{1, \dots, n\}$, otherwise the total number of queued bits may result in violating the jitter constraint. This results in having

$$b^{(i)}(v_{l-1}, v_l) < \frac{J_{\rm req} r_p^{(i)}(p(n))}{n} \tag{15}$$

which leads to a smaller value of $Q_p^{(i)}(p(n))$.

3) Solving the Path-Resource Problem for $p(n+1)$ *Given the Solution for* $p(n)$*:* Suppose we solve the path-resource problem for the path $p(n) = \langle v_0, \dots, v_n \rangle$. In addition, suppose we concatenate a jitter link (v_n, v_{n+1}) to the end of $p(n)$ to produce $p(n+1)$. We can use the solution of the pathresource problem for $p(n)$ to solve the path-resource problem for $p(n+1)$. This technique is valid due to the following two observations. First, if the path-resource problem for $p(n)$ belongs to either Case 2) or 3), then the path resource problem for $p(n+1)$ must also belong to Case 2) or 3) because adding a jitter link representing a GPS server can never reduce the overall jitter. In other words, the variable $maxQ$ can only get larger or stay unchanged. Second, in solving the path-resource problem for $p(n+1)$, there is no need to recompute the values of the QoS parameters of the required resources for each link in $p(n)$. We only need to recompute the global parameters which include $r_p^{(i)}(p(n+1))$, $maxQ$, $\min_{1 \leqslant j \leqslant n+1}\{B^{(i)}(v_{j-1}, v_j)\}$, and $\arg\min_{1 \leqslant j \leqslant n+1}\{B^{(i)}(v_{j-1}, v_j)\}$. In addition, we need to determine whether the path-resource problem for $p(n+1)$ belongs to Case 1), 2), or 3). We can update these global parameters every time an edge is concatenated to a path. The values of $b^{(i)}(u, v)$ for each individual link can be computed after the optimum path between the source node s and each node $x \in V$ in the graph is computed. Based on these observations, we can solve the problem for $p(n+1)$ using the solution of $p(n)$ by applying the following steps.

1) Set $r_p^{(i)}(p(n+1)) = \min\{r_p^{(i)}(p(n)), r(v_n, v_{n+1})\}$.
2) If $p(n)$ belongs to Case 2) or 3), then
   a) If $p(n)$ belongs to Case 2) and $B^{(i)}(v_n, v_{n+1}) \geqslant (J_{\rm req} r_p^{(i)}(p(n+1)))/(n+1)$, then $p(n+1)$ belongs to Case 2). Hence, $Q_p^{(i)*}(p(n+1)) = (J_{\rm req} r_p^{(i)}(p(n+1)))/(n+1)/\sigma^{(i)}$.
   b) If $B^{(i)}(v_n, v_{n+1}) < (J_{\rm req} r_p^{(i)}(p(n+1)))/(n+1)$, then $p(n+1)$ belongs to Case 3). Hence, $Q_p^{(i)*}(p(n+1)) = \min\{Q_p^{(i)*}(p(n)), (B^{(i)}(v_n, v_{n+1})\})/\sigma^{(i)}\}$.
3) If the path-resource problem for $p(n)$ belongs to Case 1), then $p(n+1)$ may belong to Case 1), 2), or 3). If $p(n+1)$ belongs to Case 1), then $Q_p^{(i)*}(p(n+1)) = \min\{Q_p^{(i)*}(p(n)), (B^{(i)}(v_n, v_{n+1}))/\sigma^{(i)}\}$. If $p(n+1)$ belongs to Case 2) or 3), then apply steps 2a) or 2b), respectively.

4) Optimality Conditions: A GPS network satisfies the path feasibility condition defined in Definition IV.5 because the solutions to the instance of the path-resource problem presented in (27), (13), and (14) are valid for any positive value of $J_{\rm req}$. Also, it can be noticed that concatenating a jitter link to path does not increase the overall reliability, which means that GPS networks satisfy the nonincreasing reliability condition.

The subpath addition condition defined in Definition IV.6 is satisfied if $r_p^{(i)}(p(n))$, for a path $p(n)$, remains constant after concatenating the link (u, v). However, if $r_p^{(i)}(p(n+1))$ decreases because the new link has a smaller available bandwidth, then the subpath addition condition may not be satisfied. To solve the problem of QoS routing with resource allocation under this condition, we can apply the technique used in [19] and [21] in conjunction with the algorithm **QoS-Routing** specified in Algorithm IV.1. In this technique, the set of links E is partitioned into M disjoint subsets $E_1, \dots, E_M$, where $M \leqslant |E|$, such that the links belonging to the same subset E_l have the same value of $C^{\max}$. Denote the value of $C^{\max}$ for the subset E_l by C_l. The subsets are sorted such that $C_1 \leqslant C_2 \leqslant \cdots \leqslant C_M$. The algorithm **QoS-Routing** is applied M times such that at the lth time all the links (u, v) having $C_{uv}^{\max} < C_l$ are deleted and $r(u, v) = C_l$ for all the remaining edges.

An implementation of the subalgorithm **Path-Opt** can use the steps in Section V-A3 to solve the path-resource problem for a path consisting of GPS servers. We call this implementation **GPS-Path-Opt**.

An implementation for the subalgorithm **Pre-Process** used at the beginning of the algorithm **QoS-Routing** specified in Algorithm IV.1 scans the set of links E and deletes any link (u, v) if $C_{uv}^{\max} < R_{\rm req}$ or if $B^{(i)}(u, v) = 0$, where $R_{\rm req}$ is the minimum required bandwidth for the session under consideration. The time complexity of such an implementation is $O(|E|)$.

5) Time Complexity: It can be noticed from Algorithm IV.1 that while solving the path-resource problem for the path $p(n+1)$ from the solution the path $p(n)$, we can find the value of $Q_p^{(i)*}(p(n+1))$ in $O(1)$ steps. This value is used by the algorithm for updating the attribute π while searching for an optimum path between the source node s and any node $x \in V$.

After finding the optimum path to every node, we compute the value of the resources for the path from the source node s to the destination node d using the **GPS-Path-Opt** algorithm. The time complexity of the **GPS-Path-Opt** algorithm is $O(|V|-1)$ because the longest path in a graph has at most $|V|-1$ edges. Hence, the time complexity of the **QoS-Routing** algorithm is $O(|V||E|+|E|)$. As we apply the **QoS-Routing** algorithm M times, where $M \leqslant |E|$, the time complexity for solving the problem of QoS routing with resource allocation for GPS networks is $O(|V||E|^2)$.

B. *PGPS and* $\mathrm{WF^2Q}$

The PGPS [23], [24] and $\mathrm{WF^2Q}$ [6] service disciplines are based on the GPS discipline. The GPS service discipline is not a realistic service discipline because it assumes that packets are infinitesimally divisible and that the server can serve multiple packets simultaneously. For the PGPS service discipline, the authors proposed a server that serves packets from the backlogged sessions in the same order as their finish service time under the GPS discipline. For the $\mathrm{WF^2Q}$ discipline, the server does not consider all the backlogged packets; rather it considers only those packets that have already started service, and possibly finished, under GPS. Details on how to choose the next packet to serve can be found in [6], [23], and [24].

1) The Instance of the Path Resource Problem: The derivation of the QoS parameters for the ith session along the path p consisting of PGPS or $\mathrm{WF^2Q}$ servers connected in tandem is given in Appendix A-3. Given $Q_p^{(i)}$, $J_p^{(i)}$, and $r_p^{(i)}$, we can specialize the path-resource problem defined in (7) for a path consisting of PGPS or $\mathrm{WF^2Q}$ servers as follows:

$$Q_p^{(i)*}(p(n)) = \max\left\{\min\left\{\min_{1\leqslant j\leqslant n}\left\{\frac{b^{(i)}(v_{j-1},v_j)}{\sigma^{(i)}+jL}\right\},1\right\}\right\}$$

subject to

$$\begin{aligned} &J_p^{(i)}(p(n)) = \frac{q^{(i)*}(n)}{r_p^{(i)}(p(n))} \leqslant J_{\text{req}} \\ &0 < b^{(i)}(v_{j-1},v_j) \leqslant B^{(i)}(v_{j-1},v_j), \quad j\in\{1,\ldots,n\} \\ &\rho^{(i)} \leqslant r_p^{(i)}(p(n)) \leqslant \min_{1\leqslant j\leqslant n} C^{\max}_{v_{j-1},v_j}. \end{aligned} \tag{16}$$

2) Solving the Instance of the Path-Resource Problem: As in the case of GPS, $r_p^{(i)}(p(n))$ is given the same value as in (12). Thus, the decision variables for this problem are $b^{(i)}(v_{j-1},v_j)$, $j\in\{1,\ldots,n\}$.

Let the variable $\max Q = \min\{\sigma^{(i)}+nL, \sum_{j=1}^{n} B^{(i)}(v_{j-1},v_j)\}$. Just like the path-resource problem for GPS defined in (11), the solution to the optimization problem in (16) has three cases.

Case 1)

$$J_{\text{req}} \geqslant \max Q / r_p^{(i)}(p(n)).$$

Cases 2 and 3)

$$J_{\text{req}} < \max Q / r_p^{(i)}(p(n)).$$

In Case 1), the number of bits accumulated in the queues inside the path $p(n)$ cannot possibly increase the jitter beyond the maximum required value J_{req}. Hence, an optimum solution can be achieved by having $b^{(i)}(v_{j-1},v_j) = \min\{\sigma^{(i)}+jL, B^{(i)}(v_{j-1},v_j)\}$ for all $j\in\{1,\ldots,n\}$, and $Q_p^{(i)*}(p(n)) = \min_{1\leqslant j\leqslant n}(b^{(i)}(v_{j-1},v_j))/(\sigma^{(i)}+jL)$.

In Cases 2) and 3), the accumulation of bits inside the queues can cause the jitter constraint to be violated. Hence, we need to reduce the maximum number of queued bits $q^{(i)*}(n)$ by reducing the maximum queue size $b^{(i)}(v_{j-1},v_j)$ for each jitter link $(v_{j-1},v_j)\in p(n)$. Note that we cannot decrease jitter by increasing the bandwidth $r_p^{(i)}(p(n))$ because the value $r_p^{(i)}(p(n))$ is already as large as possible. Thus, the problem reduces to maximizing $Q_p^{(i)}(p(n))$ subject to having $q^{(i)*}(n) = J_{\text{req}} r_p^{(i)}(p(n))$. The solution to this problem is specified by assigning values to the variables $b^{(i)}(v_{j-1},v_j)$ for $j\in\{1,\ldots,n\}$.

Optimum values of $b^{(i)}(v_{j-1},v_j)$, $j\in\{1,\ldots,n\}$ are given by

$$\begin{aligned} &b^{(i)}(v_{j-1},v_j) \\ &\quad= \min\left\{\frac{J_{\text{req}} r_p^{(i)}(p(n))}{n\left[\sigma^{(i)}+\frac{n+1}{2}L\right]}\left(\sigma^{(i)}+jL\right), B^{(i)}(v_{j-1},v_j)\right\}, \\ &\quad j\in\{1,\ldots,n\} \end{aligned} \tag{17}$$

and the corresponding optimum value $Q_p^{(i)*}(p(n))$ is given by

$$Q_p^{(i)*}(p(n)) = \min\left\{\frac{J_{\text{req}} r_p^{(i)}(p(n))}{n\left[\sigma^{(i)}+\frac{n+1}{2}L\right]}, \min_{1\leqslant j\leqslant n}\frac{B^{(i)}(v_{j-1},v_j)}{\sigma^{(i)}+jL}\right\}. \tag{18}$$

The derivation of the solution given in (17) and (18) is omitted due to space limitations. Detailed derivation can be found in [3, Sec. 5.5.2].

3) Solving the Path-Resource Problem for $p(n+1)$ Given the Solution for $p(n)$: Similar to the case of GPS, we need to show that if we have the solution of the path-resource problem for the path $p(n)$, we do not need to resolve the problem for the path $p(n+1)$ constructed by concatenating the jitter link (v_n, v_{n+1}) to the end of $p(n)$. This can be proved by using the same arguments used for the case of GPS discipline. Thus, we can solve the problem for $p(n+1)$ using the solution of $p(n)$ by applying the following steps.

1) Set $r_p^{(i)}(p(n+1)) = \min\{r_p^{(i)}(p(n)), r(v_n, v_n+1)\}$.
2) If $p(n)$ belongs to Case 2) or 3), then
   a) If $p(n)$ belongs to Case 2) and $B^{(i)}(v_n,v_{n+1}) \geqslant (J_{\text{req}} r_p^{(i)}(p(n+1)))/((n+1)[\sigma^{(i)}+(n+2/2)L])[\sigma^{(i)}+(n+1)L])$, then $p(n+1)$ belongs to Case 2). Hence, $Q_p^{(i)*}(p(n+1)) = (J_{\text{req}} r_p^{(i)*}(p(n+1)))/((n+1)[\sigma^{(i)}+(n+2/2)L])$.
   b) If $B^{(i)}(v_n,v_{n+1}) < (J_{\text{req}} r_p^{(i)}(p(n+1)))/((n+1)[\sigma^{(i)}+(n+2/2)L])[\sigma^{(i)}+(n+1)L]$, then $p(n+1)$ is belongs to Case 3). Hence, $Q_p^{(i)*}(p(n+1)) = \min\{Q_p^{(i)}(p(n)), (B^{(i)}(v_n,v_{n+1}))/(\sigma^{(i)}+(n+1)L)\}$.
3) If the path-resource problem for $p(n)$ belongs to case (1), then $p(n+1)$ belong Case 1), 2), or 3). If $p(n+1)$ belongs to Case 1), then $Q_p^{(i)*}(p(n+1)) = \min\{Q_p^{(i)*}(p(n)), (B^{(i)}(v_n,v_{n+1}))/(\sigma^{(i)}+(n+1)L)\}$. If $p(n+1)$ belongs to Case 2) or 3), then apply steps 2a) and 2b), respectively.

An implementation of the subalgorithm **Path-Opt** can follow the steps presented in this subsection to solve the instance of the path-resource problem for PGPS and $\mathrm{WF}^2\mathrm{Q}$. We can refer to such an implementation as **PGPS-WF^{2}Q-Path-Opt**.

4) Optimality Conditions and Time Complexity: Just like GPS, jitter graphs representing PGPS and $\mathrm{WF}^2\mathrm{Q}$ service disciplines satisfy the path feasibility and nonincreasing reliability conditions defined in Definition IV.5 and IV.7, respectively. The subpath addition condition is satisfied only if the bandwidth $r_p^{(i)}$ of a path does not decrease when a link is concatenated to its end. Thus, we can apply the technique used in solving the problem of QoS-routing with resource allocation for GPS networks to solve the same problem for PGPS or $\mathrm{WF}^2\mathrm{Q}$ networks. An implementation for the algorithm **Pre-Process** for PGPS or $\mathrm{WF}^2\mathrm{Q}$ performs the same tasks as those for GPS networks. The time complexity is also the same as that of GPS networks.

C. Self-Clocked Fair Queueing (SCFQ)

SCFQ service discipline is presented as an example of a service discipline that can be incorporated into the framework presented in Sections III and IV but applying Algorithm IV.1 to a network that employs such a service discipline may not yield an optimum solution for the problem of QoS routing with resource allocation. Instead, we provide a suboptimal solution for this service discipline.

In the PGPS and WF^2Q, computing the tag for selecting the next packet to be transmitted may be too complex for high speed networks. For SCFQ, an approximate method for computing such a tag has been proposed in earlier literature. Due to such approximation, packets under SCFQ service discipline can suffer more queueing delay than PGPS and WF^2Q. Zhang analyzed SCFQ in [35] and provided bounds on the buffer space requirements and the queueing delay. As mentioned in Sections V-A and V-B, we assume that the arrival function $A(i)_{\mathrm{out}}^{(i)}(t,u) \sim (\sigma^{(i)}, \rho^{(i)})$.

1) The Instance of the Path-Resource Problem: The derivation of the QoS parameters for the i^{th} session along the path p consisting of SCFQ servers connected in tandem is provided in Appendix A-4. Given the functional forms of $J_p^{(i)}(p(n))$, $Q_p^{(i)}(p(n))$, and $r_p^{(i)}(p(n))$ for a path $p(n)$, the path-resource problem for SCFQ can be defined as follows:

$$Q_p^{(i)*}(p(n)) = \max\left\{\min\left\{\min_{1\leqslant j\leqslant n}\left\{\frac{\{b^{(i)}(v_{j-1},v_j)}{\sigma^{(i)}+jL}\right\},1\right\}\right\}$$

subject to

$$J_p^{(i)}(p(n)) = \frac{q^{(i)*}(n)}{r_p^{(i)}(p(n))} + \sum_{j=1}^{n}\frac{\left(K_{v_{j-1}v_j}-1\right)L}{C_{v_{j-1}v_j}} \leqslant J_{\mathrm{req}}$$
$$0 < b^{(i)}(v_{j-1},v_j) \leqslant B^{(i)}(v_{j-1},v_j), \quad j\in\{1,\ldots,n\}$$
$$\rho^{(i)} \leqslant r_p^{(i)}(p(n)) \leqslant \min_{1\leqslant j\leqslant n} C^{\max}_{v_{j-1},v_j}. \tag{19}$$

2) Solving the Instance of the Path-Resource Problem: Assuming that $\sum_{j=1}^{n}((K_{v_{j-1}v_j}-1)L)/(C_{v_{j-1}v_j}) < J_{\mathrm{req}}$, we can define the variable $maxQ$ to be the same as that used for PGPS and $\mathrm{WF}^2\mathrm{Q}$. By replacing J_{req} with J'_{req} such that

$$J'_{\mathrm{req}} = J_{\mathrm{req}} - \sum_{j=1}^{n}\frac{\left(K_{v_{j-1}v_j}-1\right)L}{C_{v_{j-1}v_j}} \tag{20}$$

we can apply the procedure used in solving the path-resource problem for PGPS and $\mathrm{WF}^2\mathrm{Q}$ to solve the path-resource problem for SCFQ.

3) Optimality Conditions: The subpath addition is satisfied if the bandwidth $r_p^{(i)}$ of a path does not change when a link is concatenated to its end. Also, the nonincreasing reliability condition is satisfied. However, due to the existence of the term $\sum_{j=1}^{n}((K_{v_{j-1}v_j}-1)L)/(C_{v_{j-1}v_j})$ in (40) (Appendix A-4), the path jitter constraint may not always be satisfied for small values of J_{req}. Hence, the path feasibility condition is not satisfied for SCFQ networks. An implementation of the subalgorithm **Pre-Process** for SCFQ networks performs the same tasks as those performed for PGPS and $\mathrm{WF}^2\mathrm{Q}$ networks. In addition, it needs to perform an extra task because the path feasibility condition is not satisfied. This extra task is to run a shortest path algorithm [5], [11] on the jitter graph with $(K_{uv}-1)L/C_{uv}$ assigned as a metric for each link (u,v). If the sum of the metrics of the edges of the shortest path from the source node s to the destination node d is not less than J_{req}, then there are no feasible paths in the current network and the algorithm **QoS-Routing** terminates without any success. Otherwise, to solve the instance of the problem of QoS routing with resource allocation for SCFQ networks, we apply the same procedure used for PGPS or $\mathrm{WF}^2\mathrm{Q}$ networks, but use J'_{req} req defined in (20) instead of J_{req}. The fact that the path feasibility condition is not satisfied can lead to a suboptimal solution to the instance of the problem of QoS routing with resource allocation for SCFQ networks.

VI. Comparison With Previous Research

In Section II, we gave an overview of the previous research in the area of QoS routing. In this section, we emphasize some of the aspects that make this paper distinct from the previous research. Some of these aspects are: the network model, the solution approach, and combining queueing analysis with routing.

As elaborated in Section III, the proposed model combines QoS routing and resource allocation into a single framework. It captures the relationship between the QoS metrics and the allocated resources as well as the relationship between the metrics themselves. We model the QoS metrics of each link in the network as *functions* of the allocated resources rather than fixed valued weights. The functional forms are dependent on the traffic description and the hardware/software implementation of the network element. For a given path p, a QoS parameter is also modeled as a function of the resources allocated along the path. Unlike previous models, the functional form depends on the interaction between the network elements when connected in tandem, rather assuming simple aggregation functions, such as addition or multiplication. In short, this model is more realistic, makes less assumption about the network, and captures the intelligence derived from the knowledge about network elements implementation.

Based on the proposed model, we have presented a novel formulation of the problem of QoS routing with resource allocation in (7) and (8). Unlike previous research, the solution to the problem yields both a feasible path and the amount of resources needed to provide the level of QoS along this path. We provide an algorithm that yields an exact solution under certain condition. We show, in Section V, that these conditions are satisfiable for some packet scheduling algorithms.

It is quite common to consider queueing analysis and QoS routing as separate fields of research. In Section V, we show that, by combining these two apparently different research areas, one can provide more insight into both the problems and use this insight to solve the problem of QoS routing. The use of queueing analysis of various service disciplines is the first step toward employing the knowledge of the hardware and software implementation to solve the problem of QoS routing with resource allocation.

VII. Conclusion

In this paper, we presented a generalized formulation for the problem of QoS routing combined with resource allocation. The key feature in the new formulation is that the QoS parameters of the links are taken to be functions, rather than fixed value metrics. Another important aspect is that the formulation captures the fact that the different metrics and resources are interrelated and modifying one of the resources leads to the change of more than one metric. The new formulation has the advantage of using the knowledge of the hardware and software of the nodes to intelligently satisfy the multiple conflicting QoS requirements of the end users.

Based on the new formulation, we presented a dynamic programming algorithm that solves the problem of QoS routing with resource allocation under certain conditions. The correctness of the algorithm relies on the *path feasibility* and the *subpath addition* conditions. To show the universality of the algorithm, we analyze some of the well-known service disciplines to derive instances of the general formulation that reflects the properties of these service disciplines.

The contribution in this paper can be summarized as follows.

- A novel generalized formulation for the QoS routing with resource allocation problem.
- Analysis of the problem and the dynamic programming algorithm for solving this problem.
- Applying the formulation and the dynamic programming algorithm to existing service disciplines, which shows the generality of the proposed formulation and QoS routing algorithm.
- The jitter and data droppage analysis of some of the rate-based service discipline under insufficient resources condition.

The proposed formulation can be extended in several directions. One direction is to analyze the interaction between different service disciplines connected in tandem to derive functional forms for jitter, reliability, and bandwidth along a path. Another direction is to develop and analyze a parallel implementations to Algorithm IV.1, which can lead to an effective QoS routing with resource allocation protocol. A third direction is to derive general functional forms for jitter, reliability, and bandwidth so as to accommodate a large number of the commonly used service disciplines.

We believe that the results presented in this paper provide a new perspective to the problems of QoS routing and resource allocation. It also presents a foundation for further research in the quest of designing intelligent networks.

Appendix

A. Derivation of the QoS Routing Algorithm

In this appendix, we derive the dynamic programming algorithm to solve the problem of QoS routing with resource allocation defined in (8). The development of the algorithm is similar to that of the Bellman–Ford shortest-path algorithm [5].

Let $\mathrm{adj}^{-1}(x)$ be the set of all nodes w such that $(w,x) \in E$. Let $p^*(s,x)$ be a path that solves the QoS routing with resource allocation problem defined in (8) for a node $x \neq s$. The QoS problem can be solved using the following recursive equation:

$$\zeta^*(s,x) = \min_{w \in \mathrm{adj}^{-1}(x)} \left\{ Z_p^* \left(p^*(s,w) \oplus (w,x) \right) \right\}. \tag{21}$$

For an acyclic jitter graph, an algorithm can be derived directly from the recursive relation given in (21). However, (21) is not given in a sequential form and, hence, we cannot use it directly to derive an algorithm for an arbitrary jitter graph. Equation (21) needs to be modified as described below.

Let $P^k(s,x)$ be the set of simple paths with at most k edges between the node s and the node x. Define $\zeta^k(s,x)$ by

$$\zeta^k(s,x) = \min_{p_j \in P^k(s,x)} \left\{ Z_p^*(p_j) \right\}. \tag{22}$$

If there is no path from s to x with at most k edges, we define $\zeta^k(s,x)$ to be ∞. Note that $P^{|V|-1}(x,s) = P(s,x)$ since a simple path between any two nodes has at most $|V|-1$ edges. Hence, $\zeta^{|V|-1}(s,x) = \zeta^*(s,x)$. Equation (10) follows from (22).

B. QoS Parameters for GPS

Each session i traversing the communication link between nodes u and v is assigned a real number $\phi^{(i)}(u,v)$ corresponding to the fraction of the link bandwidth assigned to that session. Suppose that the session i is backlogged at some instance t. Then, the rate of serving the session i at the instance t is given by $(\phi^{(i)}(u,v))/(\sum_{j\in B(t)} \phi^{(j)}(u,v))C_{uv}$ b/s, where $B(t)$ is the set of sessions that are backlogged at the instance t. Define $g^{(i)}(u,v)$ by

$$g^{(i)}(u,v) = \frac{\phi^{(i)}(u,v)}{\sum_{j=1}^{K_{uv}} \phi^{(j)}(u,v)} C_{uv}. \tag{23}$$

$g^{(i)}(u,v)$ represents the minimum bandwidth guaranteed for the i^{th} session at any time instance where data from the ith sessions is backlogged in the queue of the output service discipline of the node u. Thus, the value of $g^{(i)}(u,v)$ is equivalent to the resource $r^{(i)}(u,v)$, which the bandwidth allocated for the ith session. If the real numbers $\phi^{(i)}(u,v)$ are chosen such that $\rho^{(i)} \leqslant g^{(i)}(u,v)$ for all $i \in \{1,\ldots,K_{uv}\}$ and $\sum_{j=1}^{K_{uv}} g^{(j)}(u) < C_{uv}$, then $q^{(i)*}(u,v) = \sigma^{(i)}$ and the maximum queueing delay is bounded by $\sigma^{(i)}/g^{(i)}(u,v)$. Recall, that queueing delay at the service discipline of a node is represented by the jitter $J_{uv}^{(i)}$.

If $b^{(i)}(u,v) < q^{(i)*}(u,v)$, then part of the data may be dropped. Hence, the minimum reliability along the jitter link (u,v) is less than 1. Thus, the parameters of the jitter link (u,v) representing GPS are given by

$$Q_{uv}^{(i)} = \min\left\{\frac{b^{(i)}(u,v)}{\sigma^{(i)}}, 1\right\} \tag{24}$$

$$J_{uv}^{(i)} = \min\left\{\frac{\sigma^{(i)}}{r^{(i)}(u,v)}, \frac{b^{(i)}(u,v)}{r^{(i)}(u,v)}\right\} \tag{25}$$

$$r^{(i)}(u,v) = g^{(i)}(u,v) \tag{26}$$

where $Q_{uv}^{(i)}$ is the reliability for the ith session along the jitter link (u,v), $J_{uv}^{(i)}$ is the jitter along the jitter link (u,v), $r^{(i)}(u,v)$ is the bandwidth allocated for the ith session in bits per second, $g^{(i)}(u,v)$ is defined in (23), and $b^{(i)}(u,v)$ is the buffer space allocated for the output queue for the ith session.

The next step is to compute the parameters $Q_p^{(i)}$, $J_p^{(i)}$, and $r_p^{(i)}$ for a path $p(n) = \langle v_0,\ldots,v_n\rangle$ consisting of GPS servers in tandem. Referring to [24], we can see that the maximum number of bits $q^{(i)*}(n)$ queued in the jitter links $(v_0,v_1),\ldots,(v_{n-1},v_n)$, which represents the GPS servers at the output of the nodes $v_0,\ldots v_{n-1}$, is given by $\sigma^{(i)}$. Also, the bandwidth $r_p^{(i)}(p(n))$ of the path $p(n)$ is taken as

$$r_p^{(i)}(p(n)) = \min_{0\leqslant j\leqslant n} g^{(i)}(v_{j-1},v_j). \tag{27}$$

Thus, the maximum queueing delay along the path $p(n)$ is given by $\sigma^{(i)}/r_p^{(i)}(p(n))$. By closely examining the GPS service discipline, we can see that under the worst case conditions, the maximum number of bits $q^{(i)*}(j)$ backlogged in the subpath $p(j) = \langle v_0,\ldots,v_j\rangle$ can be buffered at the output queue of the node v_{j-1} represented by the jitter link (v_{j-1},v_j). If the total buffer available at this node is less than $q^{(i)*}(j)$, then some of the bits may be dropped. Hence, the minimum reliability $Q_p^{(i)}(p(n))$ for the path $p(n)$ is given by

$$Q_p^{(i)}(p(n)) = \min\left\{\frac{\min_{1\leqslant j\leqslant n}\left\{b^{(i)}(v_{j-1},v_j)\right\}}{\sigma^{(i)}}, 1\right\}. \tag{28}$$

As for the queueing delay, although the sizes of the queues in individual nodes may be less than $\sigma^{(i)}$, there may be bits simultaneously buffered inside multiple nodes. As long as no node has zero buffer space, the total number of bits backlogged inside the path $p(n)$ can be as large as $\sigma^{(i)}$. Thus, the maximum jitter $J_p^{(i)}(p(n))$ along the path $p(n)$ is given by

$$J_p^{(i)}(p(n)) = \frac{\min\left\{\sum_{j=1}^{n} b^{(i)}(v_{j-1},v_j)\right\}, \sigma^{(i)}\Big\}}{r_p^{(i)}(p(n))}. \tag{29}$$

C. QoS Parameters for PGPS

Let $A_{\text{out}}^{(i)}(t,u)$ be the number of bits fed into the output queue of the PGPS or WF^{2}Q service discipline during the interval $(0,t]$ for the session i. Assume that $A_{\text{out}}^{(i)}(t,u) \sim (\sigma^{(i)},\rho^{(i)})$ with $\rho^{(i)} \leqslant g^{(i)}(u,v)$ and $\sum_{j=1}^{K_{uv}} g^{(j)}(u) < C_{uv}$, where $g^{(i)}(u,v)$ is defined in (23) and K_{uv} is the number of session traversing the jitter link (u,v). For both PGPS and WF^{2}Q, if the real numbers $\phi^{(i)}(u,v)$, $i \in \{1,\ldots K_{uv}\}$, are chosen proportional to the long term average $\rho^{(i)}$, we have $q^{(i)*}(u,v) = \sigma^{(i)} + L$ and the maximum queueing delay is bounded by $\sigma^{(i)} + L/g^{(i)}(u,v)$. Recall that the queueing delay at the service discipline of a node u transmitting data to node v is represented by the jitter $J_{uv}^{(i)}$. If $b^{(i)}(u,v) < q^{(i)*}(u,v)$, then some packets may be dropped. Hence, the minimum reliability along the jitter link (u,v) is less than 1. Thus, the parameters of the jitter link (u,v) representing PGPS or WF^{2}Q are given by

$$Q_{uv}^{(i)} = \min\left\{\frac{b^{(i)}(u,v)}{\sigma^{(i)}+L}, 1\right\} \tag{30}$$

$$J_{uv}^{(i)} = \min\left\{\frac{\sigma^{(i)}+L}{r^{(i)}(u,v)}, \frac{b^{(i)}(u,v)}{r^{(i)}(u,v)}\right\} \tag{31}$$

$$r^{(i)}(u,v) = g^{(i)}(u,v) \tag{32}$$

where $Q_{uv}^{(i)}$ is the reliability for the i^{th} session along the jitter link (u,v), $r^{(i)}(u,v)$ is the bandwidth allocated for the ith session in bits per second, $b^{(i)}(u,v)$ is the buffer space allocated for the output queue for the ith session, $g^{(i)}(u,v)$ is defined in (23), and L is the maximum packet size in bits.

The next step is to compute the bandwidth $r_p^{(i)}$, the jitter $J_p^{(i)}$, and the reliability $Q_p^{(i)}$ for a path $p(n) = \langle v_0,\ldots,v_n\rangle$ consisting of n hops of PGPS or WF^{2}Q servers connected in tandem. For $r_p^{(i)}$, we can see from [6], [24], and [35] that we can use (27) used for GPS.

For the reliability, denote by $q^{(i)*}(j)$ the maximum number of bits that can be queued in the subpath $p(j) = \langle v_0,\ldots,v_j\rangle$. Assume that, for all jitter links $(u,v) \in E$, the bandwidth $r^{(i)}(u,v)$ allocated for each session i traversing the link (u,v) is proportional to its session long term average bandwidth $\rho^{(i)}$. Referring to [6], [23], [24], and [35], we can see that the maximum number of bits queued in the subpath $p(j)$ is given by

$$q^{(i)*}(j) = \sigma^{(i)} + jL \tag{33}$$

where L is the maximum packet size. Under the worst case conditions, the maximum number of queued bits $q^{(i)*}(j)$ inside the subpath $p(j)$ can be queued at the output service discipline of the last node v_{j-1} represented by the jitter link (v_{j-1},v_j). Hence, the reliability $Q^{(i)}(p(n))$ of the path $p(n)$ is given by

$$Q_p^{(i)}(p(n)) = \min\left\{\min_{1\leqslant j\leqslant n}\left\{\frac{b^{(i)}(v_{j-1},v_j)}{\sigma^{(i)}+jL}\right\}, 1\right\}. \tag{34}$$

The number of bits queued inside a subpath $p(j)$ depends of the number of hops from v_0 to v_j. If some of the jitter links along the path $p(n)$ do not have enough buffer space to hold the maximum possible queue size, then the maximum number of bits queued inside the subpath $p(j) = \langle v_0,\ldots,v_j\rangle$ is given by

$$q^{(i)*}(j) = \min\left\{q^{(i)*}(j-1) + b^{(i)}(v_{j-1},v_j), \sigma^{(i)} + jL\right\} \tag{35}$$

and the initial condition is given by

$$q^{(i)*}(0) + b^{(i)}(v_0,v_1) = \min\left\{b^{(i)}(v_0,v_1), \sigma^{(i)} + L\right\} \tag{36}$$

where $b^{(i)}(v_{j-1}, v_j)$ is the buffer allocated for the ith data stream at the output of the node v_{j-1}.

Even though some or all of the jitter links may not have enough buffer space to hold the maximum possible number of queued bits $q^{(i)*}(j)$, $j \in \{1, \ldots, n\}$, the maximum jitter $J_p^{(i)}(p(n))$ can occur if $q^{(i)*}(n)$ bits are distributed among the jitter links of the path $p(n)$. Hence, $J_p^{(i)}(p(n))$ is given by

$$J_p^{(i)}(p(n)) = \frac{q^{(i)*}(n)}{r_p^{(i)}(p(n))}. \tag{37}$$

D. QoS Parameters of SCFQ

The parameters $J_{uv}^{(i)}$ and $Q_{uv}^{(i)}$ for the jitter link (u, v) representing SCFQ can be computed as follows:

$$Q_{uv}^{(i)} = \min\left\{\frac{b^{(i)}(u,v)}{\sigma^{(i)} + L}, 1\right\} \tag{38}$$

$$J_{uv}^{(i)} = \min\left\{\frac{\sigma^{(i)} + L}{r^{(i)}(u,v)} + (K_{uv} - 1)\frac{L}{C_{uv}} \quad \frac{b^{(i)}(u,v)}{r^{(i)}(u,v)} + (K_{uv} - 1)\frac{L}{C_{uv}}\right\} \tag{39}$$

where L is the maximum packet size in bits, $b^{(i)}(u, v)$ is the size of the output queue allocated for the ith session, $r^{(i)}(u, v)$ is defined in (32) and K_{uv} is the number of session traversing the link (u, v).

For the path parameters $J_p^{(i)}(p(n))$, $Q_p^{(i)}(p(n))$, and $r_p^{(i)}(p(n))$ for a path $p(n)$, we use the analysis of SCFQ presented in [35]. First, $r_p^{(i)}(p(n))$ for SCFQ is the same as that for GPS, PGPS, and $\mathrm{WF^2Q}$ given in (27). Also, the reliability $Q_p^{(i)}(p(n)$ is the same as that of PGPS and $\mathrm{WF^2Q}$ given in (34).

The maximum possible number of bits queued in a subpath $p(j)$ is given by $\sigma^{(i)} + jL$, which is the same as that of PGPS and $\mathrm{WF^2Q}$. Thus, if some of the jitter links along the path $p(n)$ do not have enough buffer space, the maximum number of bits queued inside a subpath $p(j)$ is given $q^{(i)*}(n)$ defined in (35) and (36). The jitter $J_p^{(i)}(p(n))$ is given by

$$J_p^{(i)}(p(n)) = \frac{q^{(i)*}(n)}{r_p^{(i)}(p(n))} + \sum_{j=1}^{n} \frac{\left(K_{v_{j-1}v_j} - 1\right)L}{C_{v_{j-1}v_j}} \tag{40}$$

REFERENCES

[1] S. Baqai, M. F. Khan, M. Woo, S. Shinkai, A. Khokhar, and A. Ghafoor, "Quality-based evaluation of multimedia synchronization protocols for distributed multimedia information systems," *IEEE J. Sel. Areas Commun.*, vol. 14, no. 7, pp. 1388–1403, Sep. 1996.

[2] R. Barden, L. Zhang, S. Berson, S. Herzog, and S. Jamin. (1997, Sep.) Resource reservation protocol (RSVP) version 1 functional specifications. Request For Comments 2205. [Online]. Available: http://www.cis.ohiostate. edu/bin/rfc/rfc2205.html

[3] A. R. Bashandy, "Quality of service routing and traffic regulation for distributed multimedia systems," Ph.D. dissertation, Purdue Univ., West Lafayette, IN, Aug. 1999. [Online]. Available: http://multimedia.ecn.purdue.edu/~bashandy/papers/thesis.ps.

[4] A. R. Bashandy, E. K. P. Chong, and A. Ghafoor, "Network modeling and jitter control for multimedia communication over broadband network," in *Proc. IEEE Int. Conf. Comput. Commun. INFOCOM*, vol. 2, New York, Mar. 1999, pp. 559–566.

[5] R. Bellman, "On a routing problem," *Quart. Appl. Math.*, vol. 16, no. 1, pp. 87–90, 1958.

[6] J. C. R. Bennett and H. Zhang, "$\mathrm{WF^2Q}$: Worst case fair weighted fair queueing," in *Proc. IEEE Int. Conf. Comput. Commun. INFOCOM*, vol. 1, 1996, pp. 120–128.

[7] I. Cidon, R. Rom, and Y. Shavitt, "Multipath routing combined with resource reservation," in *Proc. IEEE INFOCOM*, Kobe, Japan, Apr. 1997, pp. 92–100.

[8] R. L. Cruz, "A calculus for network delay, Part I: Network elements in isolation," *IEEE Trans. Inf. Theory*, vol. 37, no. 1, pp. 114–121, Jan. 1991.

[9] ——, "A calculus for network delay, Part II: Network analysis," *IEEE Trans. Inf. Theory*, vol. 37, no. 1, pp. 114–121, Jan. 1991.

[10] A. Demers, S. Keshav, and S. Shenker, "Analysis and simulation of a fair queueing algorithm," *Internetworking, Res. Experience*, vol. 1, no. 1, pp. 3–26, Sep. 1990.

[11] E. W. Dijkstra, "A note on two problems in connection with graphs," *Numerische Mathematik*, vol. 1, pp. 269–271, 1959.

[12] D. Ferrari, "Distributed delay jitter control in packet-switching Internetworks," *Internetworking: Res. Experience*, vol. 4, no. 2, pp. 1–20, Mar. 1993.

[13] N. R. Figueira and J. Pasqual, "An upper bound on the delay for the virtual clock service discipline," *IEEE/ACM Trans. Netw.*, vol. 3, no. 4, pp. 399–408, Aug. 1995.

[14] M. R. Garey and D. S. Johnson, *Computers and Intractability, A Guide to the Theory of NP-Completeness*. San Francisco, CA: Freeman, 1979.

[15] S. J. Golestani, "A self clocked fair queueing scheme for broadband applications," in *Proc. IEEE INFOCOM*, Toronto, ON, Canada, Jun. 1994, pp. 636–646.

[16] J. M. Jaffe, "Algorithms for finding paths with multiple constraints," *Networks*, vol. 14, pp. 95–116, 1984.

[17] M. Karol, M. Hluchyj, and S. Morgan, "Input versus output queueing on a space-division packet switch," *IEEE Trans. Commun.*, vol. COMM–35, pp. 1347–1356, Dec. 1987.

[18] T. Korkmaz and M. Kruz, "Multiconstrained optimal path selection," in *Proc. IEEE Int. Conf. Comput. Commun. INFOCOM*, vol. 2, Anchorage, AK, Mar. 2001, pp. 834–843.

[19] Q. Ma and P. Steenkiste, "Quality of service routing for traffic with performance guarantees," in *Proc. Int. Workshop on Quality of Service'97 (IWQOS'97)*. New York, 1997.

[20] ——, "Supporting dynamic interclass resource sharing: A multiclass QoS routing algorithm," in *Proc. IEEE Int. Conf. Comput. Commun. INFOCOM*, vol. 1, New York, Mar. 1999, pp. 649–660.

[21] A. Orda, "Routing with end to end QoS guarantees in broadband networks," in *IEEE Proc. Int. Conf. Comput. Commun. INFOCOM*, vol. 1, San Francisco, CA, Apr. 1998, pp. 27–34.

[22] A. Orda and A. Sprinnston, "Per-computation schemes for QoS routing," *IEEE/ACM Trans. Netw.*, vol. 11, no. 4, pp. 578–591, Aug. 2003.

[23] A. K. Parekh and R. G. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: The single-node case," *IEEE/ACM Trans. Netw.*, vol. 1, no. 3, pp. 344–357, Jun. 1993.

[24] ——, "A generalized processor sharing approach to flow control in integrated services networks: The multilenode case," *IEEE/ACM Trans. Netw.*, vol. 2, no. 2, pp. 137–150, Apr. 1994.

[25] H. F. Salama, D. R. Reeves, and Y. Viniotis, "A distributed algorithm for delay-constrained unicast routing," in *Proc. IEEE INFOCOM*, Kobe, Japan, Apr. 1997.

[26] J. L. Sobrinho, "Algebra and algorithms for QoS path computation and hop-by-hop routing in the Internet," *IEEE/ACM Trans. Netw.*, vol. 10, no. 4, pp. 541–592, Aug. 2002.

[27] R. Steinmetz, "Human perception of jitter and media synchronization," *IEEE J. Sel. Areas Commun.*, vol. 14, no. 1, pp. 61–72, Jan. 1996.

[28] The ATM Forum Technical Committee. (1996, Mar.) Private network-network interface specification version 1.0 (PNNI). af-pnni-0055.000. [Online]. Available: ftp://ftp.atmforum.com/pub/approved-specs/af-pnni-0055.000

[29] J. Turner, "New directions in communication, or which way to the information age," *IEEE Commun. Mag.*, vol. 24, pp. 8–15, 1986.

[30] D. Verma, H. Zhang, and D. Ferrari, "Delay jitter control for real-time communication in a packet switching network," in *Proc. TriComm'91*, 1991.

[31] R. Vogel, R. G. Herrtwich, W. Kalfa, H. Wittig, and L. C. Wolf, "QOS-based routing of multimedia streams in computer networks," *IEEE J. Sel. Areas Commun.*, vol. 14, no. 7, pp. 1235–1244, Sep. 1996.

[32] Z. Wang and J. Crowcroft, "Quality-of-service routing for supporting multimedia applications," *IEEE J. Sel. Areas Commun.*, vol. 14, no. 7, pp. 1228–1234, Sep. 1996.

[33] D. B. West, *Introduction to Graph Theory*. Upper Saddle River, NJ: Prentice-Hall, 1996.

[34] X. Yuan, "Heuristic algorithms for multiconstrained quality-of-service routing," *IEEE/ACM Trans. Netw.*, vol. 10, no. 2, pp. 244–256, 2002.
[35] H. Zhang, "Service disciplines for guaranteed performance service in packet switching networks," *Proc. IEEE*, vol. 83, no. 10, pp. 1373–1396, Oct. 1995.
[36] H. Zhang and D. Ferrari, "Rate controlled static priority," presented at the INFOCOM, vol. 1, San Francisco, CA, Apr. 1993.
[37] L. Zhang, "Virtual clock: a new traffic control algorithm for packet switching networks," in *Proc. ACM SIGCOM'90*, Philadelphia, PA, Sep. 1990, pp. 19–29.

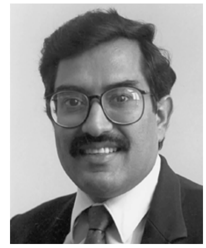
Ahmed R. Bashandy (S'93–M'95) received the B.S. degree in electronics engineering from Cairo University, Cairo, Egypt, in 1990, the M.S. degree in computer science and the Ph.D. degree in computer engineering from Purdue University, West Lafayette, IN, in 1995 and 1999 respectively.

He joined Cisco Systems Inc., San Jose, CA, in 1999, where he has worked on various projects in unicast and multicast routing. His current interests are network security and routing.

Edwin K. P. Chong (S'87–M'91–SM'96–F'04) received the B.E. (honors) degree with First-Class Honors from the University of Adelaide, South Australia, in 1987, and the M.A. and Ph.D. degrees both from Princeton University, Princeton, NJ, in 1989 and 1991, respectively, where he held an IBM Fellowship.

He joined the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, in 1991, where he was named a University Faculty Scholar in 1999, and was promoted to Professor in 2001. Since August 2001, he has been a Professor of Electrical and Computer Engineering and a Professor of Mathematics at Colorado State University, Fort Collins. He coauthored the recent best-selling book *An Introduction to Optimization* (New York, NY: Wiley, 2001) 2nd edition. His current interests are in communication networks and optimization methods.

Dr. Chong received the NSF CAREER Award in 1995 and the ASEE Frederick Emmons Terman Award in 1998. He was a corecipient of the 2004 Best Paper Award for a paper in the *Computer Networks Journal*. He was on the Editorial Board of the IEEE TRANSACTIONS ON AUTOMATIC CONTROL and is currently an Editor for *Computer Networks*. He served as an IEEE Control Systems Society Distinguished Lecturer.



Arif Ghafoor (S'83–M'83–SM'86–F'99) received the B.Sc. degree in electrical engineering from the Engineering University of Science and Technology, Lahore, Pakistan, in 1976, the M.Sc., M.Phil., and the Ph.D. degrees from Columbia University, New York, in 1977, 1980, and 1984, respectively.

He is currently a Professor in the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, and is the Director of Distributed Multimedia Systems Laboratory. He has been actively engaged in research areas related to multimedia systems, information security, and parallel and distributed computing. He has coedited *Multimedia Document Systems in Perspectives* and has coauthored *Semantic Models for Multimedia Database Searching and Browsing* (Norwell, MA: Kluwer, 2000). He has published over 150 technical papers in leading journals and conferences. He has served on the Editorial Boards of various journals including *ACM/Springer Multimedia Systems Journal*, the *Journal of Parallel and Distributed Databases*, and the *International Journal on Computer Networks*. He has served as a Guest/Co-Guest Editor for various special issues of numerous journals including *ACM/Springer Multimedia Systems Journal*, the *Journal of Parallel and Distributed Computing*, and the *International Journal on Multimedia Tools and Applications*.

Dr. Ghafoor has received the IEEE Computer Society 2000 Technical Achievement Award for his research contributions in the area of multimedia systems. He has served as a Guest/Co-Guest Editor for the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS and the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING.