A Probabilistic Scheme for Hierarchical QoS Routing

Donna Ghosh
Department of Computer Science & Eng.
University at Buffalo
State University of New York
Amherst NY 14260
donnag@cse.buffalo.edu

Raj Acharya
Department of Computer Science & Eng.
University at Buffalo
State Univiversity of New York
Amherst NY 14260
acharya@cse.buffalo.edu

ABSTRACT

Quality of Service (QoS) routing consists of two parts: (a) collecting network QoS resource availability information using topology aggregation schemes and (b) computing feasible paths using this information. This paper firstly proposes a probabilistic scheme for topology aggregation in large networks which are QoS sensitive. The proposed scheme is based on information theoretical concepts. Traditional approaches for topology aggregation use either points or line segments in the delay-bandwidth plane for representing each logical link in the aggregation. The aggregated topology is then advertised periodically to every other domain in the network. The QoS resources of the links are fast changing quantities. Hence inevitably the advertised information soon becomes out-of-date or *stale*. The existing approaches do not take into account this staleness and the routers have to compute paths based on information which may be no longer valid. This greatly degrades the performance of the routing algorithms. The problem aggravates as the size of the network increases and definitely cannot be ignored for large networks like the Internet. Our goal is to account for the inherent dynamic nature of the QoS resources by using probablistic measures as opposed to the existing deterministic aggregation schemes. In addition, this paper also proposes a modification to an existing probabilistic QoS routing algorithm for concave QoS parameters. The proposed heuristic algorithm increases the global network resource utilization.

Keywords

QoS Routing, networks, unreliable state information, topology aggregation, probabilistic routing.

1. INTRODUCTION

Multimedia applications such as video-conferencing, web-based television and telemedicine, to name a few, require very stringent *Quality of Service* (QoS) guarantees from the network. The QoS requirements of a connection request can be expressed as a set of constraints e.g. an upper bound on the end-to-end delay, a lower bound on the minimum available bandwidth in the path, an upper bound on the jitter etc. Hence, the QoS constraints are a set of service guarantees required by a connection request from the network. Currently routing schemes in large networks such as the Internet are based on computing shortest paths in terms of number of hops between a source-destination pair. However the path with the least number of hops need not necessarily also satisfy the QoS constraints. Existing routing schemes do not take into account the QoS requirements of the connection requests. With the current explosion in the demand for various multimedia applications, a strong need has been felt for routing based on the QoS requirements of a connection. This is known as QoS routing.

In order to do QoS routing, every router[1] in the network must maintain some information regarding the availability of QoS resources in the network. This information is known as the *state information*. It is said to be *local state* or *global state* depending on whether a node maintains the state information for only its own outgoing links or for the entire network. Depending on how the state information is maintained and how the feasible path is computed, QoS routing[2] can be classified into three strategies, *source routing*, *distributed routing* and *hierarchical routing*. In source routing, every node in the network maintains global state information and the entire feasible path is computed at the source router. Maintaining this global state information up-to-date at every router in the network is a mammoth task and generates a lot of message overhead. In distributed routing schemes, the routers maintain either local or global state information and the feasible path is computed in a distributed manner by exchanging control messages between the routers. Here, the overhead in finding the feasible path can be quite high. Further due to distributed computation, loop free paths cannot be always guaranteed. Further, both source and distributed routing strategies are not scalable and their performance steadily deteriorates as the network size increases [3].

In order to make routing scalable, large networks are hierarchically structured into domains and hierarchical routing

[1] The terms *router* and *node* have been used interchangeably in this paper

[2] Henceforth in the paper, unless mentioned otherwise, *routing* refers to *QoS routing*

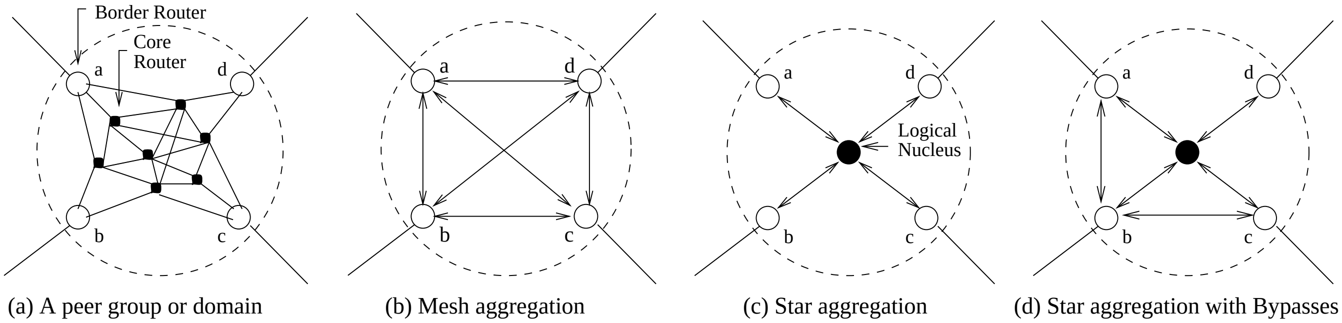

Figure 1: Connectivity aggregation in ATM PNNI standard.

strategy is used. Domains are groups of nodes which belong to the same level of hierarchy in the network. In hierarchical routing, each physical node maintains an aggregated global state information using which the source node computes the feasible path. In aggregated global state information, a node maintains detailed state information regarding its own domain and aggregated state information regarding other domains. The aggregated state information is obtained through topology aggregation. The motivation for maintaining aggregated information for the other domains is as follows. In order to maintain up-to-date global state information at every node in the network, each node periodically advertises its domain topology informaton to every other node in the network. By domain topology information, we mean the domain connectivity as well as the QoS resource availability information. This becomes impossible in large networks due to the enormous amount of bandwidth, time and space required to do this. Furthermore, sometimes due to security reasons, advertising a detailed internal domain topology information may not be desirable. The goal of topology aggregation is to summarize the topology information of a domain in a network in a meaningful fashion. Clearly, an efficient topology aggregation scheme is a precursor to the successful deployment of hierarchical QoS routing in large networks such as the Internet.

In the ATM Forum, the PNNI standard was proposed for topology aggregation. In this protocol, nodes are grouped into several clusters, called *peer groups* or domains, in different levels of hierarchy. Within a domain, some nodes would connect to nodes in other domains. They are known as the *border* nodes. The intra-domain nodes are known as the *core* nodes. The standard discusses various topological (connectivity only) representations of the peer group in terms of the border nodes as shown in figure 1. However, the PNNI standard does not deal with the aggregation of the QoS parameters in the links of the domain. In our scheme, we shall be using the mesh form shown in figure 1(b) for domain connectivity representation.

1.1 Sources of Unreliability in Network State Information

As networks grow in size, it becomes impossible for a node to maintain up-to-date network state information. This is because every node in the network cannot be expected to have detailed and instantaneous access to all other nodes and links. Hence, routing algorithms have to use *out-of-date* or *stale* state information and still meet the QoS demands of a connection. The unavoidable origins of uncertainity or unreliability in the state information are further outlined below.

- *Topology Aggregation:* In topology aggregation, the various aggregation steps abstract mutliple physical nodes and links into a much smaller number of logical entities. As a result, the state information of individual nodes and links are lost. The main consequence of this loss of accuracy in network state information is its adverse effect on the path selection algorithms. They now need to consider not only the amount of resources that are available, but also the level of certainity with which these resources are indeed available.
- *Period Update Exchanges:* The state information of the nodes change very frequently due to the inherent dynamic nature of the QoS resource availability parameters. Every time a call is accepted in the network, the QoS resource availabilities in the links change. This state information is maintained up-to-date by frequent exchange of state changes between intra and inter domain nodes, known as *updates.* Each update consumes both, network bandwidth on all the links over which it is sent and processing cycles at all nodes where it is received. Therefore, keeping this overhead to a minimum is desirable, if not mandatory. Typically updates are not propagated throughout the network everytime a resource availability changes. Instead, they are propagated at a frequency which is determined by some underlying thresholding policies [1]. This periodical nature of state exchanges, forces the state information in other nodes to be *always* out-of-date.
- *Inherent Network Nature:* Update exchanges between nodes can never be instantaneous due to propagation, transmission and queueing delays in the links and the intermediate nodes. Hence even if a node were to send updates every time its state changes, the state information at other nodes would still be out-of-date. This is due to the finite amount of time required by the updates to reach the other nodes.

1.2 Motivation

As a result of the issues discussed above, the actual state of a remote node or link can drift away significantly from the value known to other nodes, without them being aware of it. This problem increases with the size of the network and cannot be ignored for large networks like the Internet. The path selection algorithms are now required to compute feasible paths using information which is out-of-date and hence unreliable. It would no doubt, greatly help nodes in their path selection process if, in the topology aggregation updates, there is a measure of the *reliability* or the *certainity* of the advertised information. Clearly for the next generation high speed networks, we need to have topology aggregation schemes which include a measure of the reliability of the information that they are advertising. Additionally, we also need to devise routing algorithms which can take this reliability measure into account while computing feasible paths.

1.3 Goals of the Paper

As explained in section 1.1, the inherent nature of networks and the aggregation problem introduces uncertainity in the advertised information which the existing schemes do not take into account. In this paper, we have proposed a scheme which advertises updates based on probabilistic measures so as to provide the remote nodes with an indication of their chances of indeed finding the advertised resources in the domain. Further, a topology aggregation scheme is incomplete if there is no routing algorithm which uses the aggregated information to compute feasible paths. Probability based routing algorithms have been proposed in [7] and [13]. These algorithms take as input the resource availability probabilities using which they compute feasible paths. These routing algorithms can also be used to compute feasible paths using our proposed aggregated information. In this paper, we have also proposed a modified version of the algorithm for concave parameters in [7] and have shown that indeed our modified heuristic algorithm increases the overall network utilization of the concave QoS resource parameter. We have also compared the performance of the proposed probabilistic aggregation with respect to its deterministic counterpart.

The rest of the paper is organized as follows. In section 2, we have discussed the existing schemes for topology aggregation and outlined their drawbacks. We have also discussed the existing work on probabilistic routing algorithms. In section 3, we have provided a detailed outline of our probabilistic topology aggregation scheme and the rationale behind it. In section 4, we have discussed the original probabilistic routing algorithm proposed in [7] and explained its drawback. We have then explained our suggested modification to the algorithm. In section 5, we have discussed our experimental setup and results. In section 6, we have given a performance analysis of the storage and run time of the our approach and compared it with its deterministic counterpart. Finally we have concluded in section 7 with a brief mention of future work.

2. RELATION TO PREVIOUS WORKS

In the earlier schemes, the QoS resource availability in a logical link was represented by a number, chosen as per some rule ([9], [11]). As an example, let us consider the QoS parameters end-to-end delay and minimum available bandwidth in a path. Let there be three unique paths between a pair of logical nodes with the available path bottleneck bandwidth and end-to-end path delays as the tuples (8,5), (10,10) and (4,2). Now the logical node can advertise (4,10) based on the minimum bandwidth and maximum delay, or (10,2) based on the maximum bandwidth and minimum delay or (7.33,5.67) based on the path averages. However note that these values need not necessarily belong to the same path. Hence if the node optimistically advertises (10,2) and gets a connection request for the same, it cannot support it. This is because there is no physical path between the two logical nodes which indeed has 10 units of bottleneck path bandwidth and 2units of end-to-end delay. Hence, each of these advertisements misrepresent the available network resources. Reference [10] describes some traditional approaches in picking the "best" pair. In [8], comparisons have been made between the star and mesh topology connectivity aggregates of figure 1 with bandwidth as the QoS parameter. However clearly, by using a single pair of numbers which represents a point in the delay-bandwidth plane, it is not sufficient to reflect the available resources in various paths accurately. To deal with this problem, in [10] it has been suggested that along with the numerical values, each logical link also keep an extra parameter which implicitly defines a curve passing the delay-bandwidth point on a delay-bandwidth plane. However, this method has certain drawbacks, such as the curve being very far away from other parameters and being independent of the required QoS. In [14], it has been attempted to deal with this drawback by first plotting the (bandwidth,delay) tuples of all the distinct paths between two border nodes in the delay-bandwidth plane. Then a line segment which is close to all the points is used as the QoS parameter for advertisement. However, whether points or line segements in the delay-bandwidth plane are used, this information is deterministic in nature and soon becomes *out-of-date* at the remote domains. Hence, in this paper we do away with this deterministic representation altogether and instead treat a QoS parameter as a random variable.

In [7] and [13] some probabilistic routing algorithms have been suggested for both concave and convex type of QoS resource parameters. These algorithms solve the problem of finding a path in the network such that the overall probability of actually finding the requested resources in the path is maximized. They assume that the probability distributions of the resource parameters are available as inputs to the algorithms. It naturally follows for us, to use these algorithms for computing the feasible paths, using our probabilistic state information to provide the probability distributions. Further, these algorithms do not aim at maximizing the overall resource utilization in the network. We have suggested a modification of the routing algorithm for concave resources proposed in [7] which increases the overall network resource utilization.

3. PROPOSED AGGREGATION SCHEME

In this section, we have discussed the proposed probabilistic topology aggregation scheme. In section 3.1, we have outlined the setup and given an overview of the aggregation scheme. In section 3.2, we have discussed the details of the scheme. Finally in sections 3.3 and 3.4, we have discussed the rationale for the scheme.

3.1 Setup and Overview

The overall setup that we have used for solving the topology aggregation problem is as follows:

- *Connectivity Representation:* In our study, we have borrowed the full-mesh topology connectivity representation as shown in figure 1(b), from the PNNI standard [6]. In this representation, a fully connected graph with the vertex set as the border nodes is advertised. The border nodes act as the logical nodes or *ports of entry* into the domain with logical links connecting them. We notice that in this representation, the space occupied by the update will be in the order of $O(b^2)$, where b is the number of border nodes in the domain. Further, in the star representation, as shown in figure 1(c), the space occupied by the update is in the order of $O(b)$. Therefore, the star representation scales better than the mesh representation as the number of border nodes in a domain increases. However, the full mesh gives a more accurate representation when compared to the star and hence, we have chosen it for our study here. Our concern in this paper is to investigate our gain by using probabilistic approach in comparison to a deterministic one. Efficacies of the approach will be done as part of future work.
- *Update Frequency Policy:* Various update frequency policies have been discussed in [1]. Clearly, keeping the number of updates to a minimum, irrespective of the traffic conditions is always a desirable protocol property. Hence, we have adopted the constant timer based periodical update frequency approach. In this approach, after every constant time period, the border nodes advertise the aggregated domain updates to all the other domains in the network. In section 5, we have studied the effect of reducing the frequency of updates on the routing performance.

Overview of proposed approach: In order to capture the inherent randomness in the QoS resource parameters, we have modelled them as random variables and have used their probability distributions as the QoS values in the logical links. In cases where there are more than one distinct paths between two logical nodes, we have calculated an optimal distribution which would approximate the distributions in each of these distinct paths in some least distance sense. Using such probabilistic state information, a remote node can then easily calculate the probability of actually finding the requested resources in other domains.

3.2 Details

We represent the QoS parameters of links and paths, such as the available bandwidth, end-to-end delay, end-to-end jitter etc., as continuous random variables with a bounded positive real support, from zero to the maximum available QoS parameter value (e.g. maximum link capacity for available bandwidth in a link). Unfortunately, there is no known traffic model which characterizes the distributions of the various QoS parameters. Therefore we need to come up with an estimation of the true underlying distributions. For this, we treat the random variables as discrete by dividing the support of their distributions into a number of bins [3]. Then every router periodically takes measurements of the QoS resource availability in its own outgoing links and based on the *relative frequency of occurences*, it computes an empirical probability distribution known as the *type* for the random variable. The type of a random variable is a well known concept in Information Theory [4] and is defined below.

Definition 1. The *type* $P(X)$ of a discrete random variable X (henceforth written as $\mathbf{P}$) is defined as:

$$\mathbf{P} \triangleq (\frac{N(a_1|X^n)}{n}, \frac{N(a_2|X^n)}{n}, ..., \frac{N(a_r|X^n)}{n})$$

where, $N(a_i|X^n)$ denotes the number of occurences of an outcome a_i in n number of observations of X.

The various QoS resource parameters can be broadly defined as *additive* and *concave* [17]parameters and are defined below.

Definition 2. Let $d(i,j)$ be a QoS parameter for link (i,j). For any path $p = (i,j,k,....,l,m)$, we say that the QoS parameter d is *additive* or *convex* if,

$$d(p) \triangleq d(i,j) + d(j,k) + \cdots + d(l,m)$$

and that d is *concave* if,

$$d(p) \triangleq min\ \{d(i,j), d(j,k), ..., d(l,m)\}$$

Bandwidth and end-to-end delay are examples of concave and additive parameters respectively. In our scheme, for concave parameters, we take the type of the minimum available QoS resource in the entire path as the QoS parameter of the path. For additive parameters, the routers compute the type for the QoS parameter for the entire path directly, based on path measurements. Various approaches could be adopted for this, for example, a router can periodically query a path with control packets and calculate the additive QoS parameter availability in the path. Then we use this type as the corresponding QoS parameter for the logical link between the pair of logical nodes (border nodes).

Often, there might be more than one distinct paths between a given pair of border nodes. Then we need to come up with a meaningful approximation of the types of the QoS parameters in each of these distinct paths. We shall then use this approximate type as the QoS parameter value for the logical link between the two border nodes. By meaningful, we mean that the approximate type must *optimally fit* all the types in the individual paths. In other words, this is a *Goodness-of-Fit* problem in which the fit has to be optimized such that the approximate type fits each of the types of the individual paths the closest. The objective of goodness-of-fit testing is twofold ([12], [16]):

- It seeks to establish whether data obtained from a real experiment fits a known theoretical model.

[3] Henceforth in the paper, unless otherwise mentioned, by *random variable* we mean *discrete random variable*

- It seeks to establish whether two or more given sets of data have been obtained from the same experiment of the same underlying phenomenon.

The goodness-of-fit problem can be framed as a general hypothesis testing problem in the following manner. Let us observe an experiment $\mathbf{A}$. Then we have the partition $\mathbf{A} = [\mathcal{A}_1, \mathcal{A}_2, ..., \mathcal{A}_m]$, consisting of m events $\mathcal{A}_i$. We wish to test the hypothesis H_0 that their probabilities $p_i = P(\mathcal{A}_i)$ have m given values p_{0i}:

$$H_0 : p_i = p_{0i}, \text{ all } i \quad \text{against} \quad H_1 : p_i \neq p_{0i}, \text{ some } i.$$

In our case, the hypothesis that we want to test is the closeness of fit of the approximate type to each of the types of the individual distinct paths. For this, we use **Kullback Leibler distance** (KL distance) [4] as the measure for the goodness-of-fit test.

Definition 3. The *Relative Entropy* or *KL distance* between two probability mass functions $p(x)$ and $q(x)$ is defined as:

$$\begin{aligned} D(p||q) &\triangleq \sum_{x \in \mathcal{X}} p(x) \log \frac{p(x)}{q(x)} \\ &= E_p \ \log \frac{p(X)}{q(X)}. \end{aligned}$$

We use the convention (based on continuity arguments) that $0 \log \frac{0}{q} = 0$ and $p \log \frac{p}{0} = \infty$.

The KL distance is a measure of the distance between two distributions. In other words, $D(p||q)$ is a measure of the inefficiency of assuming that the distribution is q when the true distribution is p. Some of its important properties, which we will use later in sections 3.3 and 3.4, has been listed in the APPENDIX.

We use KL distance to calculate the optimal approximate type in the following manner. Let $\mathcal{P}_\mathbf{n}$ denote the set of all possible types of the random variable X based on sequences of observations of length n. Let there be s number of distinct paths between a given pair of border nodes. Hence, by using the method described before, we can come up with a type for the QoS parameter of each of these s number of paths, say $\mathbf{P}^1, \mathbf{P}^2, ...,$ and $\mathbf{P}^s$.

Problem Statement 1: Find a type $\mathbf{P}^*$ from the set $\mathcal{P}_\mathbf{n}$, such that $D(\mathbf{P}^i||\mathbf{P}^*)$ $\forall i$ is minimized. This can be expressed as the equation:

$$\mathbf{P}^* = \underset{\mathbf{P} \in \mathcal{P}_n}{argmin} [D(\mathbf{P}^1||\mathbf{P}) + D(\mathbf{P}^2||\mathbf{P}) + \cdots + D(\mathbf{P}^s||\mathbf{P})] \tag{1}$$

Problem Statement 2: Find a type $\mathbf{P}^*$ from the set $\mathcal{P}_\mathbf{n}$, such that $D(\mathbf{P}^*||\mathbf{P}^i)$ $\forall i$ is minimized. This can be expressed as the equation:

$$\mathbf{P}^* = \underset{\mathbf{P} \in \mathcal{P}_n}{argmin} [D(\mathbf{P}||\mathbf{P}^1) + D(\mathbf{P}||\mathbf{P}^2) + \cdots + D(\mathbf{P}||\mathbf{P}^s)] \tag{2}$$

This is a constrained based optimization problem and we have used the method of Lagrange Multipliers to solve for the optimal type $\mathbf{P}^*$. The solutions that we have obtained are as follows:

Solution for Problem Statement 1: The optimal type P^* which solves Problem Statement 1 is as follows:

$$\mathbf{P}^*(x) = \frac{\sum_{i=1}^{s} \mathbf{P}^i(x)}{\sum_{x \in \mathcal{X}} \sum_{i=1}^{s} \mathbf{P}^i(x)} \tag{3}$$

Solution for Problem Statement 2: The optimal type P^* which solves Problem Statement 2 is as follows:

$$\mathbf{P}^*(x) = \left\{ \frac{\prod_{i=1}^{s} \mathbf{P}^i(x)}{\sum_{x \in \mathcal{X}} \prod_{i=1}^{s} \mathbf{P}^i(x)} \right\}^{\frac{1}{s}} \tag{4}$$

Here, $\mathbf{P}^*(x)$ denotes the relative frequency of occurence for outcome x of random variable X in the type $\mathbf{P}^*$ and $\mathcal{X}$ denotes the set of all possible outcomes of the random variable X. Note that since KL distance is not symmetric and neither does it satisfy the triangle inequality, it is not a true metric. Hence we have individually solved for minimizing both $D(\mathbf{P}^i||\mathbf{P}^*)$ and $D(\mathbf{P}^*||\mathbf{P}^i)$.

3.3 Rationale for Using Types

As mentioned at the beginning of this section, unfortunately currently we do not know any theoretical model for the probability distributions of the QoS resource availability parameters. Hence we have used local measurements made by the routers to compute the type for the resource which is an empirical probability mass function (pmf). Let a router take n number of independent measurements of the QoS resource parameter X, denoted by $X_1, X_2, ..., X_n$. Let the true underlying probability mass function of X be denoted by $\mathbf{F}$ and the type of X based on the n measurements be denoted by $\mathbf{P}$. Then there is a well known theorem which has been stated below. Its proof is beyond the scope of this paper but can be found in [4].

THEOREM 1. *Let $X_1, X_2, ..., X_n$ be i.i.d. $\sim \mathbf{F}$. Then, $D(\mathbf{P}||\mathbf{F}) \rightarrow 0$ with probability 1 as $n \rightarrow \infty$.*

We know that the KL distance of two probability mass functions is zero if and only if they are the same. Hence, the above theorem implies that for a very large number of measurements, the type of the random variable converges to its true probability mass function in a probabilistic sense.

3.4 Rationale for Using KL Distance

As is clear from the properties of KL distance mentioned in the APPENDIX, that convergence in the KL distance sense is stronger than convergence in the $L1$ norm sense or the χ^2 statistic sense, two of the most frequently used distance measures for discrete random variables. Further, here we give an Information Theoretic argument for choosing KL distance as the measure for the goodness-of-fit test.

Let there be s number of distinct physical paths between two border nodes. Let the random variable Y_i denote the QoS resource availability in path i. We have assumed that the Y_i s are independent of each other. We shall be using the concepts of *entropy* and *mutual information* which have been defined in the APPENDIX. We can state our objective as follows:

Objective: To approximate the vector of random variables $\mathbf{Y} = [Y_1, Y_2, ..., Y_s]$ with a single random variable, say X, such that the *mutual information* between X and $\mathbf{Y}$, denoted by $I(\mathbf{Y}; X)$, is maximized. This would ensure that knowing X will give us maximum information regarding each of $Y_1, Y_2, ..., Y_s$.
Now,

$$\begin{aligned} I(\mathbf{Y}; X) &= I(Y_1, Y_2, ..., Y_s; X) \\ &= \sum_{i=1}^{s} I(Y_i; X | Y_{i-1}, ..., Y_1), \quad \text{by using chain rule} \\ &= \sum_{i=1}^{s} I(Y_i; X), \quad \text{assuming Y}_\text{i}\text{ s to be independent} \\ &= \sum_{i=1}^{s} H(Y_i) - \sum_{i=1}^{s} H(Y_i | X) \end{aligned} \tag{5}$$

Note that $H(Y_i)$ $\forall i$ is fixed depending on the underlying true pmf of the QoS resource parameter. Further, mutual information and entropy are always positive quantities. Hence, from equation 5, in order to maximize $I(\mathbf{Y}; X)$, we need to chose X such that $\sum_{i=1}^{s} H(Y_i|X)$ is minimized. Now, $\sum_{i=1}^{s} H(Y_i|X) = 0$ when $Y_i = g_i(X)$ $\forall i$, where $g_i(X)$ denotes any function of X. This is because, if Y_i is a function of X, then X gives all the information about Y_i too. Hence, if $i = 1$, then the solution is $Y_1 = X$. Intuitively, we would want to chose X, such that knowing the pmf of X would also give us information about the pmf of Y_i. We suggest in choosing X such that its *J divergence* from each of Y_i is minimized. We have defined J divergence in the APPENDIX. However, this problem is a non-linear optimization problem and to the best of our knowledge, there is no closed form solution for it. Hence, we have individually solved for minimizing $D(\mathbf{P}^i||\mathbf{P}^*)$ and $D(\mathbf{P}^*||\mathbf{P}^i)$.

4. PROBABILISTIC ROUTING SCHEME

In this section, we have described the probabilistic hierarchical routing scheme that has been studied in the paper. In section 4.1, we have described the overall setup that we have studied. Note that the topology aggregation scheme described in section 3 can be applied for both concave as well as additive QoS parameters. For the purpose of studying the gain obtained by using the probabilistic aggregation scheme, we also have to decide on the corresponding routing algorithm. In [7] and [13], probabilistic routing algorithms have been suggested, which are different for concave and additive parameters. We shall for the rest of the paper, concentrate on concave parameters. Also, for ease of understanding, we shall chose bandwidth as an example of concave parameter and present our studies in terms of bandwidth as the QoS parameter. In sections 4.2 and 4.3, we have explained the routing algorithm for concave parameters suggested in [7] and discussed its drawback. In 4.4, we have discussed our proposed modification to the algorithm.

4.1 Setup and Overview

In this section we have described the overall routing architecture and setup that we have used in our paper. It is based on the framework for QoS routing in the Internet as suggested by the IETF in RFC 2386 [5]. Specifically, our setup is as follows:

- *Link State Routing:* Network state information updates are periodically flooded throughout the network, using which each router maintains its own routing table. Connections are routed based on their QoS requirements and the network state information maintained at the routers.
- *On-Demand Source Routing:* The feasible paths are computed on-demand (as and when a connection request is received) at the source routers. If the source and destinations are in different domains, then the source router computes a "skeleton" path which specifies the entire path in terms of the border nodes of the various domains. A detailed path in each intermediate domain is chosen when the connection request enters the domain.
- *Intra and Inter Domain Update Exchanges:* The core routers in a domain maintain state information only regarding their own domain through periodical flooding of intra-domain updates, as is typical of link state protocols. The border routers in a domain also maintain detailed state information regarding their own domain in the same way as the core routers. Additionally the border routers also maintain an aggregated state information regarding all the other domains in the network through periodical flooding of inter-domain aggregated topology updates.

4.2 Existing Algorithm - MRP

Let us represent a given network by a graph $G(V, E)$. Let us assume that every node knows the complete graph connectivity information. In addition, let every node also know the quantity $p_l(x)$ for every link $l \in E$. $p_l(x)$ denotes the probability that link l can accomodate a connection request which requires x units of bandwidth. It can be calculated from the types in the aggregated state information. Now, let a node receive a connection request for b units of bandwidth. Then the routing problem is to find a path from the source node to the destination node such that the total probability of finding *at least* b units of bandwidth in every link in the path is maximized. Let $p_l = p_l(b)$ denote the probability of finding b units of bandwidth in link l. Assuming that the random variables in different links are independent of each other, for a path $\mathbf{K}$, $\prod_{l \in \mathbf{K}} p_l$ is the probabiilty that at least b units of bandwidth is available in *all* the links of the path $\mathbf{K}$. This problem can be stated as follows:

Problem Statement 3: For a given bandwidth requirement b, find a path $\mathbf{K}^*$ such that, for any path $\mathbf{K}$:

$$\prod_{l \,\in\, \mathbf{K}^*} p_l(b) \;\geq\; \prod_{l \,\in\, \mathbf{K}} p_l(b).$$

Solution for Problem Statement 3: The solution to this problem has been proposed in [7] as the algorithm *Most Reliable*

Path (MRP).
Algorithm(**MRP**):

1. Let $w_l = -\log p_l$, $\forall\, l \in E$.
2. Find the shortest path according to the metric w_l. This can be done by using any standard shortest path algorithm such as the Dijkstra's algorithm.

4.3 Drawback in MRP

The solution provided by algorithm **MRP** gives the *safest path* i,e. a path which has the maximum probability of satisfying the requested bandwidth. However, it makes no attempt at trying to maximize the total bandwidth admitted into the network, which is a measure of the overall network resource utilization. Further, **MRP** does not take into account the length of the path in terms of the *number of hops* at all. Now consider the following scenario. Let us have a connection request for b units of bandwidth. Let there be two paths K_1 and K_2 between the source-destination pair, such that path K_1 has 2 hops and path K_2 has 10 hops. Let $p_l = p_l(b) = 1$ for every $l \in K_1$, $l \in K_2$. Then $w_l = 0$ for every $l \in K_1$ and $l \in K_2$. As a result the total length of both the paths will be *zero* in any shortest path algorithm that we use with w_l as the link weights. Therefore the path returned by **MRP** in this case will correspond to the path discovered *first* by the shortest path algorithm and both K_1 as well as K_2 are equally likely solutions.

However intuitively, it is always better to reserve resources along the *shortest feasible path*, when there are more than one candidate feasible paths. This helps in admitting a greater number of admission requests. This has been verified by our simulation results, mentioned in section 5. The reason for the increase in the number of calls admitted is that for a given connection request, resources are reserved in the least number of links possible. This leads to a better utilization of the link capacities in the entire network. This observation has also been reported in [15] and [2]. In [15] it has been shown that in general algorithms which along with finding feasible paths that satisfy the QoS requirements of the calls, also aim at minimizing the number of hops in the path admit a greater number of requests than those algorithms which do not consider the number of hops in the path at all. In [2], a comparitive study has been done for the *safest-shortest* and *shortest-safest* heuristics. In safest-shortest routing, all the minimum hop paths between the source and destination are determined and the one with the largest safety (i,e. product of probablities in individual links of the entire path) is used. In shortest-safest routing, all the safest paths between the source and the destination are found and the shortest one is used. They have shown that shortest-safest performs better than the safest-shortest. This can be explained intuitively, as the shortest path need not always be the safest one too. Hence, if the safety of the shortest path is very low, safest-shortest will reject the call but shortest-safest will admit it.

In the next section we have proposed a modification to **MRP** to convert it to the shortest-safest approach. In section 5 we have compared the performance of the *safest* approach of **MRP** with the *shortest-safest* approach of the heuristic algorithm **SMRP**.

4.4 Proposed Modification - SMRP

In order to convert the safest approach of **MRP** into the shortest-safest approach of **SMRP**, we have proposed a change in the definition of the link weights w_l. We give the modified heuristic algorithm below:

Shortest Most Reliable Path (SMRP).
Heuristic Algorithm(**SMRP**):

1. Let $w'_l = -\log p_l \;+\; 1$, $\forall\, l \in E$.
2. Find the shortest path according to the metric w'_l. This can be done by using any standard shortest path algorithm such as the Dijkstra's algorithm.

The link weights defined in this manner can be thought of as the sum of the safety of the link and an unit cost for traversing the link. This unit cost discourages paths with more number of hops when the safety of the paths are the same.

5. EXPERIMENTAL RESULTS

We have done extensive simulations on networks of various sizes in order to verify the consistency of the nature of the network performance. We have reported in this paper the results that we have obtained in a network of 10 domains, with 18 nodes each. Each domain has between 2 to 4 border nodes. The simulations have been done using OPNET, a commercial network simulation tool. The results reported here are the averages over multiple simulation runs. Each simulation has been run for 25,000 connection requests, all of which are inter-domain requests. The source-destination pairs are generated randomly. All links in the network are duplex. The link capacities of the inter-domain links has been kept twice that of the intra-domain links. Connection requests arrive as per a Poisson distribution. The connection durations are exponentially distributed. The bandwidth requests are uniformly distributed between 1 and 2 Mbps.

We have compared the following schemes:

1. Probabilistic topology aggregation with SMRP as the routing algorithm (refered to henceforth as PROB-SMRP).
2. Probabilistic topology aggregation with MRP as the routing algorithm (refered to henceforth as PROB-MRP).
3. Deterministic topology aggregation with SF as the routing algorithm (refered to henceforth as BW-SF). In this scheme, in the topology aggregation part, instead of sending the type, the actual value of the available bottleneck bandwidth is used. In cases where there are multiple distinct physical paths between two border nodes, the average bottleneck bandwidth value is used as the parameter for the logical link. In the routing algorithm, we use the *Shortest Feasible* path approach

(SF). In this we first prune all links which cannot support the bandwidth requested and then run a shortest path algorithm with the link weights as 1. We favor the shortest-feasible path approach as opposed to finding *any* feasible path for the reason explained in section 4.3.

We have also compared all the above three schemes using the solution for both *Problem Statement 1* and *Problem Statement 2,* for the probabilistic aggregation (as mentioned in section 3.2). However since there is no significant change in the results, we have reported here the results obtained by using the solution for *Problem Statement 1.* The *call admission ratio* is defined as the ratio of calls admitted over all the arrivals. However, when the calls can request for different amounts of bandwidths, as in our experimental setup, a high call admission ratio does not necessary reflect *high efficiency.* Thus we use *bandwidth admission ratio* instead for the performance comparisons, which is defined as the ratio of total bandwidth admitted into the network over all the requested bandwidths [15].

5.1 Sensitivity to Routing Update Interval

In figure 2, we have plotted the bandwidth admission ratio of the three methods with varying inter-domain update exchange intervals. The intra-domain update interval has been kept fixed in all the three schemes. We notice that the bandwidth admission ratio of both PROB-SMRP and PROB-MRP is always more than that of BW-SF. Infact, as the update intervals are increased, we see that BW-SF suffers very badly and there is a steep decline in its bandwidth admission ratio. However, the bandwidth admission ratio of PROB-SMRP and PROB-MRP remain nearly constant. Hence we note that PROB-SMRP and PROB-MRP are robust to less frequent exchange of updates. This is a very desirable property for any hierarchical routing scheme as it avoids clogging the backbones with protocol control messages. This occurs because with less frequent update exchanges, the state information in the BW-SF scheme remains out-of-date for a much longer time. On the other hand, the probability based schemes are not much affected as they aim at learning the underlying distributions. In figure 3, we have plotted the percentage increase in the total bandwidth admitted in the network by PROB-SMRP and PROB-MRP in comparison to BW-SF, as calculated from figure 2. As was mentioned before, we can clearly notice that as the update interval increases, the percentage increase in bandwidth admitted also increases steeply. Infact we expect this increasing trend to be steady with further increase of the update interval due to the steady deterioration in performance of BW-SF. PROB-SMRP gives about 14% to 21% improvement and PROB-MRP gives about 0.8% to 6% improvement in comparison to BW-SF, when the update intervals are between 55 to 385 seconds. Further, the bandwidth admission ratio of PROB-SMRP is about 14% to 15% more than that of PROB-MRP. This proves our claim in section 4.3 that in general the *shortest feasible path* approach performs better than *any feasible path* approach.

5.2 Sensitivity to Resource Availability

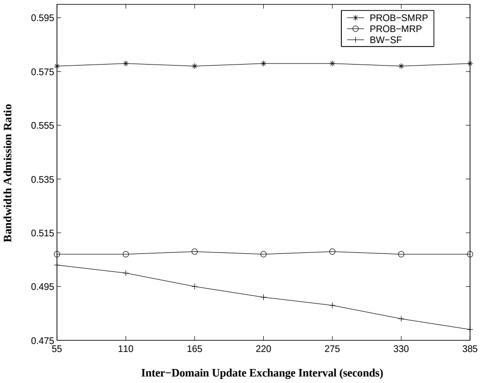

Figure 2: Bandwidth admission ratio with varying inter-domain update exchange intervals (Capacity of intra and inter-domain links are 20 and 40Mbps respectively).

In figure 4, we have plotted the bandwidth admission ratio of the three methods with varying intra and inter domain link capacities. Here again we notice that the probabilistic schemes consistently perform much better when compared to BW-SF. We also notice that the bandwidth admission ratio rises with an increase in the link capacities for all the three schemes. This is because with more resource available, naturally more bandwidth can be admitted. In figure 5, we have plotted the percentage increase in the total bandwidth admitted in the network by PROB-SMRP and PROB-MRP in comparison to BW-SF, as calculated from figure 4. Here, we notice that the percentage increase obtained by the probabilistic schemes decrease with an increase in the link capacities. This is as expected because with more resource availability, more bandwidth is admitted regardless of the underlying scheme. Hence all three schemes perform well. We notice that PROB-SMRP gives between 21% to 8% improvement and PROB-MRP gives between 5% to 0.45% improvement in comparison to BW-SF, when the intra-domain link capacities are between 5 to 40 Mbps. Note that since the ratio of intra and inter domain link capacities have kept constant at $\frac{1}{2}$, an intra-domain link capacity of 5 Mbps implies an inter-domain link capacity of 10 Mbps. Again for the same reason as explained in section 4.3, PROB-SMRP performs better than PROB-MRP.

5.3 Protocol Message Overhead

The number of messages exchanged in all the three schemes to maintain the state information in every router is the *same.* However, the size of the messages are different in the various schemes. The size of the message in PROB-SMRP and PROB-MRP is equal to the size of the type. If there are k number of bins in the type, then the message will be an array of size k. On the other hand, the message in BW-SF is a single number i,e. an array of size 1. However, by using the probabilistic schemes, we can afford to have very less update exchanges without affecting the routing performance at all, as shown in our simulation results.

6. PERFORMANCE ANALYSIS

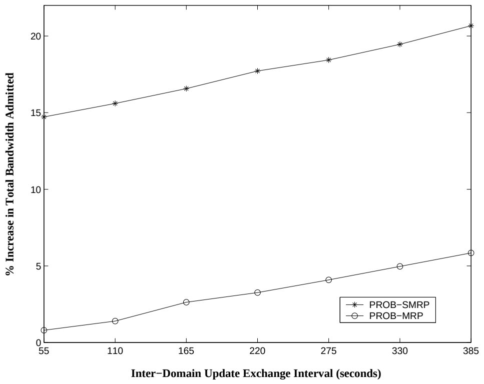

Figure 3: % Increase in bandwidth admitted by PROB-SMRP and PROB-MRP over BW-SF (calculated from figure 2).

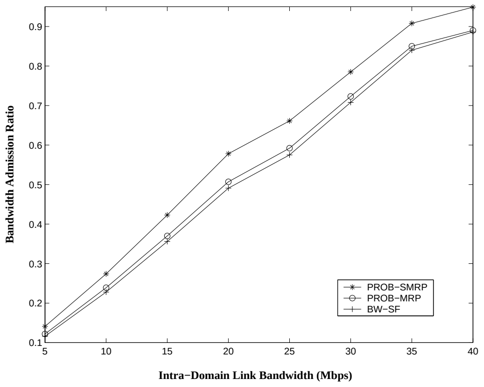

Figure 4: Bandwidth admission ratio with varying intra and inter-domain link capacities (inter-domain update interval is 220 seconds, ratio of intra to inter-domain link capacities=$\frac{1}{2}$).

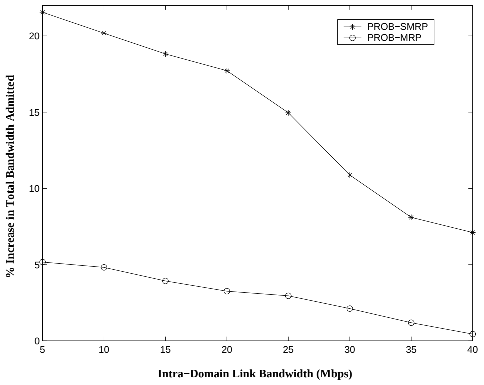

Figure 5: % Increase in bandwidth admitted by PROB-SMRP and PROB-MRP over BW-SF (calculated from figure 4).

In this section we have given an analysis of the storage and run time for the topology aggregation schemes used in PROB-SMRP and PROB-MRP (henceforth referred to as PROB) and have compared it with that used in BW-SF (henceforth referred to as BW).

Storage Space: We shall use the following notation. Let a domain i be represented as a graph $G_i\ (V_i, B_i, E_i)$, where V_i, B_i and E_i denote the total number of nodes, border nodes and edges respectively in the domain i. Let there be n domains. Then every border node in the domain maintains two state tables for the intra and the inter domain information. Let ${N^i}_{intra}$ and ${N^i}_{inter}$ denote the number of entries in the intra and inter domain tables maintained at a border node in domain i. These numbers remain the same for both PROB and BW. Now, let the type of the QoS parameter have k number of bins. Then the storage space required at the border node to maintain the two tables in the scheme PROB is $O(\ k\cdot({N^i}_{intra} + {N^i}_{inter})\ )$. In the BW scheme, the space required is $O(\ {N^i}_{intra} + {N^i}_{inter}\ )$.

Run Time: We shall stick to the same notation as above. The run time for PROB can be calculated as follows:

- Maintaining type for all links in domain i: $O(k{\cdot}|E_i|)$.
- Finding bottleneck type for logical link between two border nodes: $O(k{\cdot}|E_i|) + O(|E_i|)$.
- Total time for finding one logical link in the mesh: $O(k{\cdot}|E_i|) + O(k{\cdot}|E_i|) + O(|E_i|) = O((2k+1){\cdot}|E_i|)$.
- Total time for calculating the entire mesh: $O((2k+1){\cdot}|E_i|{\cdot}|B_i|^2)$.

Proceeding similarly, we can show that the total time for calculating the mesh at a border node B_i in domain i is $O(|E_i|{\cdot}|B_i|^2)$. Hence, the run time for PROB is increased by an order of k when compared to BW. This is not surprising because the instead of dealing with a single number as in BW, PROB deals with an array of size k for every link. Also note that the contrained minimization problem solution is obtained off line. The scheme PROB only has to plug in values into the solution every time it has to compute the approximate type for a logical link.

7. CONCLUSION AND FUTURE WORK

In this paper we have proposed a novel probabilistic scheme for hierarchical quality of service routing. We have justified the need for probabilistic schemes as opposed to their deterministic counterparts due to the inherent nature of the routing problem. We have shown that our scheme gives between 14 to 21% increase in the total bandwidth admitted into the network when the update interval is between 55 to 385 seconds. Further, we have also explained that we expect this increasing trend to remain steady with further increase in update intervals. Hence we have shown that the probabilistic schemes are much more robust to less update exchanges and perform much better in comparison with their deterministic counterparts.

The only price that we pay for this robustness and improved performance is in terms of messages of greater but constant size and some comuptation overhead at the individual routers to maintain the types. We believe that it is a better philosophy to infrequently exchange longer messages which include information regarding their reliability, rather than very frequently exchange shorter messages which do not contain any reliability information. Also, we believe that maintaining the traffic characteristic information at the routers in terms of the types can help the routers in predicting various catastrophic events and taking diagnostic and even preventive measures. Infact our aim is to further investigate these possibilities. In the future we would like to make our scheme more adaptive to catastrophic events like congestion or router break downs.

8. REFERENCES

[1] G. Apostolopoulos, R. Guerin, S. Kamat, and S. Tripathi. Quality of service based routing: A performance perspective. *ACM SIGCOMM '98*, September 1998.

[2] G. Apostolopoulos, R. Guerin, S. Kamat, and S. Tripathi. Improving qos routing performance under inaccurate link state information. *ITC-16*, June 1999.

[3] S. Chen and K. Nahrstedt. An overview of qos routing for the next generation high-speed networks- problems and solutions. *IEEE Network Magazine*, 12(6):64–79, November-December 1998.

[4] T. M. Cover and J. A. Thomas. *Elements of Information Theory.* Wiley Series in Telecommunications, New York, 1991.

[5] E. Crawley, R. Nair, B. Rajagopalan, and H. Sandick. A framework for qos based routing in the internet. *Technical Report, RFC 2386*, August 1998.

[6] T. A. Forum. Private network-to-network interface specification, version 1.0. *Technical Report af-pnni-0055.000*, March 1996.

[7] R. Guerin and A. Orda. Qos routing in networks with inaccurate information: Theory and algorithms. *INFOCOMM '97*, 1997.

[8] F. Hao and E. W. Zegura. On scalable qos routing-performance evaluation of topology aggregation. *INFOCOMM '00*, 2000.

[9] A. Iwata, H. Suzuki, R. Izmailow, and B. Sengupta. Qos aggregation algorithms in hierarchical atm networks. *IEEE Proceedings of the ICC'98*, pages 243–248, 1998.

[10] T. Korkmaz and M. Krunz. Source-oriented topology aggregation with multiple qos parameters in hierarhical atm networks. *IEEE*, pages 137–146, 1999.

[11] W. C. Lee. Topology aggregation for hierarchical routing in atm networks. *ACM SIGCOMM'95, Computer Communications Review*, pages 82–92, 1995.

[12] B. W. Lindgren. *Statistical Theory.* Macmillan and Macmillan, New York, 1976.

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

[14] K. S. Lui and K. Nahrstedt. Topology aggregation of bandwidth-delay sensitive networks. *IEEE GLOBECOM '00*, 2000.

[15] Q. Ma and P. Steenkiste. On path selection for traffic with bandwidth guarantees. *ICNP*, October 1997.

[16] A. Papoulis. *Probability and Statistics.* Prentice Hall, New Jersey, 1990.

[17] Z. Wang and R. Crowcroft. Quality of service routing for supproting multimedia applications. *IEEE Journal of Selected Areas of Communications*, September 1996.

APPENDIX

A. PROPERTIES OF KL DISTANCE

Some of the important properties of the Kullback Leibler distance between two pmfs p and q are as follows [4]:

- $D(p||q) \geq 0$
- $D(p||q) = 0$, iff $p = q$
- $D(p||q) \neq D(q||p)$
- $D(p||q) \leq D(p||r) + D(r||q)$ is not true always.
- Convergence in KL sense implies convergence in the $L1$ norm sense but no proof is known for the reverse.
- The χ^2-statistic is twice the first term in the Taylor series expansion of the KL distance.

Clearly, the KL distance is not a true metric because it does not satisfy the triangle inequality and is not symmetrical. However, the **J divergence** is a true metric and is defined as follows:

Definition 4. The *J divergence* between two probability mass functions $p(x)$ and $q(x)$ is defined as:

$$J(p(x)||q(x)) \triangleq D(p(x)||q(x)) + D(q(x)||p(x))$$

We use the convention (based on continuity arguments) that $0 \log \frac{0}{q(x)} = 0$ and $p \log \frac{p(x)}{0} = \infty$.

B. ENTROPY AND MUTUAL INFORMATION

Definition 5. The *entropy* $H(X)$ of a discrete random variable X is defined as [4]:

$$H(X) \triangleq -\sum_{x \in \mathcal{X}} p(x) \log p(x).$$

where, $p(x)$ denotes the probability mass function of X.

Definition 6. Consider two discrete random variables X and Y with a joint pmf $p(x, y)$ and marginal pmfs $p(x)$ and $p(y)$. The *mutual information* $I(X; Y)$ is defined as [4]:

$$I(X; Y) \triangleq D(p(x, y)||p(x)p(y)).$$