A QoS Routing Algorithm Based on Three Different Metrics

Luís Henrique M. K. Costa [1,2]
Luis.Costa@lip6.fr

Serge Fdida [1]
Serge.Fdida@lip6.fr

Otto Carlos M. B. Duarte [2]
otto@gta.ufrj.br

[1] LIP6 – Université Pierre et Marie Curie
4, place Jussieu
75252 Paris Cedex 05 - France

[2] Grupo de Teleinformática e Automação – GTA
Universidade Federal do Rio de Janeiro – COPPE/EE
P.O. Box 68504 - 21945-970 - Rio de Janeiro - RJ - Brasil *

Abstract

Quality of Service (QoS) based routing is one of the main component of a new Integrated Services Internet Architecture. In this paper a method for constructing scalable QoS-based routing algorithms that considers three QoS requirements is developed. The proposed approach defines three metrics: bandwidth consumption, transmission delay and loss probability and its complexity is similar to that of algorithms that are limited to two metrics. Its simplicity is its major advantage. The key idea is the use of a simple mathematical transformation that turns additive the loss metric together with a heuristic that guarantees all metrics. The average performance gain of the proposed approach when compared to a classical approach achieves 40%.

1 Introduction

Due to the emerging multimedia applications, the Internet tends to receive a greater quantity of traffic with QoS (Quality of Service) requirements. The best-effort service offered by the IP layer is no longer appropriate for that kind of traffic because of its QoS requirements such as bounded delay or packet loss probability.

*This work was sponsored by UFRJ, FUJB, PROTEM-CC, CNPq, FAPERJ, CAPES, REENGE and COFECUB.

To carry multimedia traffic in the Internet some new characteristics and services must be added. New classes of service must be offered to provide some guarantees to applications. A mechanism must exist to allow applications to notify their QoS requirements to the network, and some traffic management mechanisms must be added to hosts and routers. The IETF (Internet Engineering Task Force) is working on two different approaches for QoS provision in the Internet, the Integrated Services [1] and the Differentiated Services model [2]. There is also a working group concerned with the definition of a Resource Reservation Protocol [3] used by applications to signal their QoS requirements to the network.

Another important area is related to the definition of a routing architecture that considers the QoS requirements of applications. In traditional routing, packets are delivered using a route based on its source and destination addresses. In QoS-based routing, the route for each flow considers also its traffic requirements whose QoS needs will be mapped into several routing metrics.

A main issue in the definition of a QoS-based routing architecture is to guarantee its scalability [4]. The elected scheme must exhibit a complexity that does not grow indefinitely with the number of nodes. This characteristic is even more important with the Internet's growth. There is also another point that must be considered when guaranteeing scalability in the routing scheme: the size of the routing tables. This issue is very important even for traditional routing schemes.

For such reasons, the definition of routing algorithms that consider multiple metrics is very important. We can divide the QoS routing approaches in algorithms that use heuristics to find a feasible path (in general through the knowledge of the entire topology) [5, 6, 7, 8] and algorithms that try to find routes in the "classical" fashion, usually based on well-known routing algorithms. In general these algorithms can be constructed in two ways. One approach is to consider the dependency between the used metrics [9]. This solution puts the complexity on the definition of the relations between metrics, imposing minor changes to traditional algorithms.

The second approach is to use independent metrics that will be considered separately by the route computation algorithm [10, 11]. This solution transfers the difficulty to the construction of efficient algorithms, in such a way that the time spent to find the routes does not preclude its implementation. Some authors have shown that the computation of routes with more than two metrics is very difficult because of its complexity; however simple algorithms can be constructed in the case of route computations with only two metrics [10]. For independent metrics, the multiple metrics can also be considered as a single mixed metric. This can lead to the utilization of traditional algorithms with few changes, without any increase in computational complexity.

In this work, we consider independent metrics where the two ways to construct the algorithm mentioned above are combined. A new routing algorithm is proposed that considers three metrics in order to better characterize the applications. This algorithm handles the three metrics by means of a simple mathematical transformation that turns additive the loss metric and, as a consequence, it is solvable in polynomial time since it has the same structure of traditional algorithms.

The algorithm was implemented as an additional module to the Berkeley NS-2 (Network Simulator Version 2) where a complete routing scheme was constructed with the routing protocol and an IP-like network layer. The proposed algorithm and another algorithm that considers the three independent metrics separately were simulated. A comparison on the time spent to discover routes was achieved and the results were sound to be up to 40% better with the proposed solution.

This paper is organized as follows. In section 2, the main issues related to routing algorithms are discussed, considering QoS requirements that are important to the majority of applications. Sections 3 and 4 present the proposal for a scalable QoS-based routing architecture, being able to handle three different metrics. The proposed approach was implemented and tested, as shown in section 5. Section 6 presents some final considerations.

2 QoS-based Routing Algorithms

The main difference between QoS-based routing and classical IP routing is that the routing decision is based not only on source and destination addresses for each packet, but also on the traffic characteristics of the related flow. This fact has two major effects on the routing mechanisms (i.e. routing protocol and IP layer).

First, the packet classifier inside the IP layer, which in fact stores the routing table, has to be changed. In a general way, the routing table will now contain several entries per destination instead of just one. The selected entry will depend upon the traffic characteristics of each flow. A first approach could be to keep one entry per flow in the routing table, but this solution would lead to very large tables and the routing protocol would have to be queried every time a new flow is setup. A more reasonable approach seems to be the maintenance of one entry per class of service [1, 12]. Thus, each flow is mapped into a class, which in turn corresponds to a specific entry in the routing table. This is the approach considered in this work. Nevertheless, here the definition of class of service does not suppose necessarily that a different queueing treatment at routers is implemented for each class of service. Each service is simply characterized by a different set of QoS requirements, e.g. different acceptable maximum delay.

Second, the routing algorithm used by the routing protocol will have to consider several metrics in route computation at the same time, instead of just one as in traditional routing. This leads to difficulties in the construction of such QoS-based routing algorithms. From the point of view of path finding algorithms, these metrics are translated into paths with multiple restrictions instead of just one.

In this section, we analize the decision factors in the selection of metrics that are adequate to the routing protocol implementation. The different ways that metrics can be combined are discussed.

This combination can be done by the construction of a unique measure which represents all the metrics, or by the analysis of each metric separately, by means of specific algorithms. Then we present a source routing algorithm used as an example for the construction of routing algorithms that consider three metrics (QoS requirements).

2.1 Routing computation for several metrics

The metric selected for routing defines the way the network is represented. This decision not only influences the complexity of the routing algorithm, but also the set of QoS parameters that this scheme supports. Several aspects must be taken into account when choosing the metrics that constitute a routing decision factor [4].

No matter which metrics are used, a routing algorithm should be efficient and scalable and, as a consequence, it must be simple. Furthermore, the used metrics must also represent the network characteristics to support basic QoS requirements. The negotiated QoS parameters will be translated into the metric set used by the routing protocol to make route decisions.

The route computation for multiple QoS restrictions is a complex problem. The metrics chosen can be treated by the routing algorithm in different ways. The first difference between the possible approaches is related to the correlation between the metrics. Metrics like transmission delay, bandwidth and loss probability can be considered dependent or not. It is clear that when there is not enough bandwidth for a flow, the end-to-end delay grows as well as the probability of packet losses, since packets begin to be dropped at a certain point. Algorithms based on the dependence of metrics were proposed by Ma and Steenkiste [9].

Another way to make routing decisions with multiple metrics is to consider that the metrics are independent from each other. Metrics can be considered independent if some assumptions are made. For example, consider once again the three metrics mentioned above. Packet loss probability due only to the transmission link characteristics (not considering packet dropping by routers), delay due only to the propagation time (not considering queuing delays) and the available bandwidth for new flows on a link are metrics independent from each other. In fact, the terms not considered in the first two metrics are taken into account in the available bandwidth. One can argue that since propagation delay is in general static and small, it may be ignored. However it is not always the case: if satellite links are present, the propagation delay is certainly not negligible. In the case of independent metrics, the computational complexity of the route discovery algorithm is the main problem. Wang and Crowcroft [10] have shown that for certain combinations of QoS requirements the problem is NP-complete. They have proved that only bandwidth and another metric (such as delay, jitter or loss probability), in the total of two metrics, can be handled by algorithms of reasonable complexity. Although, the simplest mathematical solution is the single mixed metric approach, it was not considered due to severe limitations as described in section 2.2.

The mixed metric approach and the algorithms proposed in [10] are discussed in the next sections. According to this analysis the proposal of a new routing algorithm that considers three metrics is made.

2.2 Single mixed metric

Algorithms for a single metric route computation are well known and used by most of today's routing protocols. A direct approach then would be to map the user's QoS requirements into a single measure, which could be used to make the routing decision compatible with traditional algorithms. The single mixed metric is used in [13] where it is called an "overall cost".

The single mixed metric can be defined as a mathematical function that receives as parameters a user's set of QoS requirements and gives as a result a single combined measure. A function that calculates a measure proportional to the desired bandwidth and inversely proportional to delay and loss probability can be defined, as shown in equation 1.

$$f(p) = \frac{B(p)}{D(p) \times L(p)}, \tag{1}$$

where p is the computed path, $B(p)$ its bandwidth, $D(p)$ the transmission delay associated with this path and $L(p)$ the associated packet loss probability. The path that results in the greater value of $f(p)$ would be the best route accordingly to this measure.

However, a single mixed metric as defined above would have problems when used in real situations. It does not carry enough information to guarantee that all user's QoS requirements will be respected. Guaranteeing that the $f(p)$ function is satisfied does not ensure that each QoS requirement is respected.

Another problem is related to the different composition rules for various QoS parameters. There are three common composition rules: additive, multiplicative and concave. If the metric for a path is the addition of the metrics of each link composing the path, then the metric is additive. If it is the multiplication of all link metrics, then the rule is multiplicative. Finally, if the metric is the minimum value between all the link metrics, the composition is concave. Delay transmission has an additive composition rule, but link bandwidth is concave. Therefore, the single mixed metric function defined above does not follow these composition rules.

2.3 Multiple metrics

The use of multiple metrics is able to model the network in a more precise way, but the problem of finding a path given multiple constraints can become very complex [9, 10]. The larger the number of metrics the more complex the route algorithm. It must be also considered that the appropriate QoS requirements depend directly upon the specific application. The algorithm's complexity is a function of the metrics' composition rules.

The most usual user's QoS requirements are transmission delay, jitter, cost, packet loss probability and bandwidth consumption. The link bandwidth has a concave composition rule. The composition rule for loss probability is more complex. Let p be a path, $lp(p)$ the path metric and $lp(i,j)$ the metric associated with the link between node i and node j. The composition metric for loss probability is shown in equation 2.

$$lp(p) = 1 - ((1 - lp(i,j) \times (1 - lp(j,k)) \times ... \times (1 - lp(l,m))). \tag{2}$$

However, if transmission success probability is used instead of loss probability, the equivalent composition rule is multiplicative. In [10] the complexity for algorithms which consider additive and multiplicative metrics is discussed, showing that the problem is NP-complete. Any combination of n additive metrics with m multiplicative metrics is also a NP-complete problem.

As a result, between the metrics cited above, delay, jitter, cost, loss probability and bandwidth, the only tractable combination is to associate bandwidth and another metric. The reason is that bandwidth has a concave composition rule, while the others are all additive or multiplicative.

2.4 Source routing with two restrictions

In this section, an algorithm proposed in [10] for route computation given two restrictions is presented. One of the requirements is bandwidth, the other can be transmission delay, jitter or cost (all of them have the same composition rule). This is an algorithm for source routing based on Dijkstra's algorithm. It supposes the knowledge of the entire topology, but the principles presented later to support a third metric are directly applicable to distributed routing computation algorithms. In fact, the algorithm simulated within NS is a distributed algorithm based on distance-vectors.

In the following analysis, the link bandwidth is the residual transmission rate available to new flows at each link. The bandwidth metric is defined as the minimum bandwidth available through all links in the route. The transmission delay has two basic components, the queueing delay and the propagation delay. The queueing delay is influenced by the available bandwidth and traffic characteristics. Therefore, the transmission delay is considered to be only the part due to propagation time. Queueing delay is supposed to be associated to available bandwidth. Thus the two metrics are independent.

The minimum available bandwidth on a path can be considered as its width, and the delay as its length. So the routing problem is to find a path in the network which length is smaller than a given value and width larger than a minimum.

Let $G = (N, A)$ be a graph representing the network, with N nodes and A arcs. Two real numbers are associated to each link (i, j): the available bandwidth b_{ij} and the propagation delay d_{ij}. For an arc (i, j) not present in the graph, it is considered that $b_{ij} = 0$ and $d_{ij} = \infty$. Given a path $p = (i, j, k, ..., l, m)$, the path width, $w(p)$, will be $min[b_{ij}, b_{jk}, ..., b_{lm}]$ and its length $l(p) = a_{ij} + a_{jk} + ... + a_{lm}$, the sum of propagation delays. Given two constants, B for the bandwidth and D for the maximum propagation delay, the problem is to find a path p between i and m such that $w(p) \geq B$ and $l(p) \leq D$.

First, the algorithm consists on discarding all arcs whose $b_{ij} < B$. All the possibles paths within this new graph will respect the bandwidth restriction. The algorithm must then point out a path that satisfies the maximum acceptable delay, if one exists. The following algorithm finds a path between node 1 and node m with available bandwidth not smaller than B and propagation delay not larger than D. D_i is the estimated length of the path between node 1 and node i.

```
Step 1:  ∀i,j : d_ij = ∞ if b_ij = 0.
Step 2:  L = {1}; ∀i, i ≠ 1 : D_i = d_1i.
Step 3:  Find k ∉ L | D_k = min_{i∉L} D_i.
           If D_k > D, the path cannot be found and the algorithm finishes.
```

If L contains node m, a path was found and the algorithm finishes.

$L := L \cup \{k\}$.

Step 4: $\forall i \notin L : D_i := \min[D_i, D_k + d_{ki}]$.

Go to step 3.

Step 1 eliminates all arcs whose bandwidth is smaller than B. Steps 2 to 4 find the path with minimum delay to node m using Dijkstra's algorithm [10]. The algorithm finishes if node m is in the path or if the delay reaches D before m is included. Each step in the algorithm needs $O(N)$ iterations, and in the worst case the steps are repeated $N-1$ times. Then the total number of operations in the worst case is $O(N^2)$.

3 Constructing an Algorithm with Three Constraints

In the previous sections, the main problems in the definition of a QoS-based routing protocol were presented. The use of a single mixed metric would be discarded, due to the difficulties imposed by the different composition rules of each metric. Then, with the assumption of metrics independence [10], an algorithm that considers two metrics was presented. The algorithm could be constructed because one metric is the bandwidth, which has a concave composition rule. The other metric can be the delay, jitter or cost (all of them additive). Another important feature when considering routing protocol scalability is related to the size of routing tables. In this work the tables are considered to have one entry per destination per class of service, instead of one entry per flow.

Given the observations made above, an hybrid scheme is proposed. The mechanism proposed is a tradeoff between the algorithm simplicity and route decision accuracy given the user's QoS parameter set. Some changes are made to the algorithm presented in the previous section by means of a careful utilization of a single mixed metric. This metric takes into account the propagation delay and the packet loss probability, which in addition to available bandwidth provide an algorithm that is likely to support a major set of applications. These three specific metrics may be treated without augmenting the complexity of the algorithm.

3.1 Metric combination

The utilization of a single mixed metric is an interesting approach because classical routing algorithms could be used without major changes. However there are two problems with this solution. The first problem is related to the definition of a mathematical function compatible with the composition rule of each metric. The other is that with a measure that is the combination of two metrics, it would not be possible to guarantee that each single metric is respected. In other words, choosing a route that satisfies the single mixed metric does not guarantee that the isolated QoS requirements are respected.

The most common metrics are propagation delay, jitter, bandwidth and loss probability. Propagation delay and jitter have both an additive composition rule. Loss probability has a more complex composition rule, as shown in section 2. However, loss probability can be represented in a different way, as the transmission success probability. In this case, the composition rule is multiplicative. Let p be a path, $S(p)$ the transmission success probability in this path, and $S(i,j)$ the metric associated with the link between node i and node j. The composition rule for the transmission success probability will be as shown in equation 3.

$$S(p) = (S(i,j) \times S(j,k) \times ... \times S(l,m)). \tag{3}$$

If the logarithm of this metric is calculated, its composition rule will be additive. In this way it is possible to combine this metric with other metrics that have an additive composition rule, such as delay. Therefore, the logarithmic transmission success probability, $Slog$, is defined by:

$$Slog(p) = log(S(p)) = (log(S(i,j)) + log(S(j,k)) + ... + log(S(l,m))). \tag{4}$$

It is important to note the effects of this transformation. The additive composition is desirable, but other issues are identified. The metric value for the loss probability is always between 0 and 1. So when the logarithm is calculated, this interval is transformed into $-\infty$ to 0. Numbers with a large absolute value could constitute a problem for the route computation. But if the interval for the transmission success probability is limited, say from 0.1 to 1, the logarithmic interval will be bounded from -1 to 0. This limitation is reasonable, since it means that the transmission success probability will not be less than 10%. So the absolute value for the logarithmic success probability will range from 0 to 1. The same interval is valid for the path metric (composed of various links). This logarithmic probability will be smaller than the smallest metric associated with each link on the path, and will approximate zero as the number of links increase. The composition rule of the two used metrics is additive, so is the composition rule of the single mixed metric. The restriction on the range of loss probability, while limiting the application of the proposed scheme, permits the construction of three-metric algorithms as complex as the two-metric algorithms proposed in [10].

The single mixed metric as defined above eliminates the other drawback of this approach. The transmission success probability will be represented by real numbers from 0 to 1, and the delay will be an integer. So it is possible to distinguish the contribution from both metrics in the single mixed metric (the other disadvantage mentioned above).

With the definition of a single mixed metric for delay and loss probability, by means of a scheme similar to that presented previously, it is possible to construct algorithms that also take into account the available bandwidth. The idea is the same, that is, to execute a previous search in the network graph to eliminate all arcs that do not meet the bandwidth restriction. The next step is to use an algorithm similar to Dijkstra's algorithm with little changes. These changes are made to ensure that the requirements on each of the combined metrics are individually respected, a disadvantage of the use of a single mixed metric.

4 The SMM Algorithm

In this section, a new algorithm that considers three metrics in route computation is proposed. This algorithm is called SMM (from Single Mixed Metric routing algorithm). It is constructed with the careful utilization of a single mixed metric. This metric combines the transmission delay and loss probability. Together with the available bandwidth, this algorithm defines a three-metric routing scheme. This is a source routing algorithm, as the one presented earlier in section 2.4. The algorithm for distributed route computation based on distance-vectors (implemented in the simulations) is constructed based on the same principles, and is in fact simpler than the source routing one.

Let $G = (N, A)$ be a representative graph for the network, with N nodes and A arcs. Two real numbers are associated with each arc, b_{ij} representing its available bandwidth and u_{ij} (equation 5) being the single mixed metric defined as the addition of $Slog(i, j)$, the logarithmic transmission success probability (real number between 0 and 1) with $D(i, j)$, the propagation delay (an integer).

$$u_{i,j} = Slog(i, j) + D(i, j). \tag{5}$$

When an arc (i, j) is not present in the graph, it is considered that $b_{ij} = 0$ and $u_{ij} = \infty$. Given a path $p = (i, j, k, ..., l, m)$, the path width, $w(p)$, is computed by $min[b_{ij}, b_{jk}, ..., b_{lm}]$ and its length is the addition of the single mixed metric related to each of its component links, $l(p) = u_{ij} + u_{jk} + ... + u_{lm}$. Given two constants B for the minimum bandwidth and U for the maximum acceptable value for the single mixed metric, the problem is to find a path p between i and m such that $w(p) \geq B$ and $l(p) \leq U$, as it was done previously.

`Step 1:` $\forall i, j : u_{ij} = \infty$ `if` $b_{ij} = 0$.

`Step 2:` $L = \{1\}; \forall i, i \neq 1 : U_i = u_{1i}$.

`Step 3:` `Find` $k \notin L \mid U_k = \min_{i \notin L} U_i$.

If $U_k > U$, a path cannot be found and the algorithm finishes.

If L contains node m, a path was found and the algorithm finishes.

$L := L \cup \{k\}$.

Step 4: $\forall i \notin L : U_i := \min[U_i, U_k + u_{ki}]$.

Go to step 3.

The algorithm shown above does not ensure that the transmission success probability will be larger than the desired minimum (or smaller, in log). The comparison between two real numbers is dominated by the integer part of each number. Because the delay is represented by an integer, it will be always guaranteed.

A minor change to the algorithm can be made in such a way that it will also guarantee success transmission probability. The loss probability QoS requirement is mapped into a minimum absolute value for the logarithmic transmission success probability. In other words, the addition of these metrics for each link on the path should be smaller than a known bound, defined by an application's QoS requirement.

Let P be the known limit for logarithmic transmission success probability and $Slog(ij)$ the logarithmic transmission success probability for arc ij. Given a path p with x hops, if $Slog(ij)$ for each link is smaller than P/x, then $Slog(p)$ will be smaller than P. However, when a path is being computed, its length (x) is not known yet. A first idea would be to restrict $Slog(ij)$ to P/C, where C is the length of the longest path on the network graph. If $Slog(ij)$ is always limited to P/C, at each step, then it is guaranteed that the path p, $Slog(p)$ will be smaller than P. However this limit can be too restrictive, since the route between two nodes can have in the general case a length smaller than C.

So an alternative scheme is proposed to relax this restriction. At each algorithm step, a hop "remainder" (SS) and a logarithmic probability "remainder" (SP) are defined. The hop remainder is the difference between the length of the longest path in the graph and the length of the present computed path. The probability remainder is the difference between the required limit and the present value for the logarithmic transmission success of the path at the actual step. Then the maximum logarithmic probability that can be added at each step will be no more P/C, but SP/SS. This value can be larger than P/C, in the case where SP is smaller than $SP \times (P/C)$. So the probability of finding a path in the network graph grows. Thus, the algorithm is modified as follows:

`Step 1:` $\forall i, j : u_{ij} = \infty$ `if` $b_{ij} = 0$.

$SS = C; SP = P$.

`Step 2:` $L = \{1\}; \forall i, i \neq 1 : U_i = u_{1i}$.

`Step 3:` `Find` $k \notin L \mid (U_k = \min_{i \notin L} U_i)$ `and` $((U_k - \lfloor U_k \rfloor) < (SP/SS))$.

`If` $U_k > U$`, a path cannot be found and the algorithm finishes.`

`If` L `contains node` m`, a path was found and the algorithm finishes.`

$L := L \cup \{k\}$.

$SS := SS - 1$.

$SP = SP + (U_k - \lfloor U_k \rfloor)$.

`Step 4:` $\forall i \notin L : U_i := \min[U_i, U_k + u_{ki}]$.

`Go to step 3.`

In this way, the transmission success probability restriction is stronger than the real QoS requirement, but this restriction is relaxed and can guarantee that all QoS requirements are achieved, without increasing the algorithm's complexity. This imprecision in route discovery is a tradeoff with the algorithm complexity.

The SMM algorithm takes the same number of iterations to find a route as the original algorithm for two metrics [10]. The computational complexity of the single mixed metric is low and is computed before the route discovery algorithm. Thus the scalability of the proposed routing architecture is improved through the utilization of the SMM algorithm in addition with the class of service based routing tables. The proposed scheme has the advantage of route computation with three metrics, while other algorithms used two metrics. The use of bandwidth, delay and loss probability can characterize a larger set of applications than those limited to bandwidth and another single metric.

Routing tables may be dynamic. In fact, the choice of metrics made in our approach favors the implementation of dynamic routing protocols. The delay and loss probability metrics are characteristics intrinsic to the transmission media and topology used, so they are likely not to change in short periods of time. The other metric is quite dynamic: the bandwidth available in the network links. The update of this metric may be made locally in routers through the utilization of some kind of call admission control (CAC). The CAC process should inform the routing process of the available bandwidth in links. If a CAC mechanism is not present, then a traffic measurement element should be implemented which to inform the routing process. It is clear that the time scale used in the communication between this "traffic measurer" and the routing process is an important issue. Too high update rates may lead to oscillations and/or important processing charge at routers. The same effects can be produced if routers announce changes in the available bandwidth too frequently. A kind of "hysteresis" should be employed in both bandwidth announcements (intra-router and inter-router).

5 Implementation on Network Simulator

The proposed scheme was implemented in the kernel of the Berkeley Network Simulator (NS) [14]. It provides three different unicast routing protocols that implement the Dijkstra's routing algorithm and the Distributed Bellman-Ford algorithm. The routing mechanism provided by NS is dynamic only for the Bellman-Ford algorithm, which is the algorithm implemented by distance-vector routing protocols such as RIP [15].

The QoS-based routing scheme was constructed into NS by adding two new routing protocol modules. Other aspects from NS's network implementation were also modified to support QoS-based routing. The NS's link implementation was modified. Now each link has three metrics associated with it, instead of one. The routing protocol considers these three metrics in making route decisions, as previously stated. The three elected metrics in this implementation are propagation delay, available bandwidth and packet loss probability.

A packet classifier that takes into account the type of packet was created. Each packet has to be classified into a set of classes of service. This approach was based on the proposal from the Integrated Services IETF Working Group [1]. This set was defined, as an initial approach, with three classes. Each class was defined as having one metric as the most important: delay, bandwidth or loss probability. This definition is based on the approach used in [10]. In that work a precedence between the two used metrics had to be defined. Here, the precedence between metrics is the differentiation between the three classes of service. The routing table now has three entries per destination, instead of one, as in classical protocols. When a packet arrives, the classifier has to check its type and destination to decide to which interface it will be forwarded.

The proposed algorithm could be evaluated by means of the implementation of two protocols. In the first one, the non-scalable approach, the route is searched by checking each metric separately.

This approach is referenced as $3M$. If finds optimal routes. For example, given a specific class of service, it finds the route between two points with minimum delay; if more than one route was found, then it chooses the one with maximum bandwidth; again, if more than one path was found, the one with minimum loss probability is chosen. The three metrics have its optimal values (in a certain precedence), although a class of service may be defined by, e.g., a maximum delay - one does not have to find a route with minimum delay. The second approach is to use the SMM algorithm, that is, with a single mixed metric. This approach does not guarantee that the routes have each of the three metrics with an optimal value, instead it can guarantee just that the metrics are within the limits imposed by the QoS requirements of each class of service. This approach is likely to spend less time to find routes especially when there are multiple equal-metric feasible paths. The routes found by the two approaches are not necessarily the same to all source and destination pairs, but they comprise always the QoS requirements defined for each class of service. Two different modules were implemented.

NS's routing mechanism does not have a real-time support, so some changes were done to measure the time spent to find routes. This time depends on the number of iterations performed by the algorithm to find the route. A "probe element" was added at each node. This "probe" keeps track of the number of steps needed for route computation and produces a delay in route updates proportional to these measures.

Several topologies were constructed, all of them with a single traffic source. This source sends packets at a constant bit rate (CBR). By using NS's dynamic scheme, a link in the topology goes down at a specific time. This link is one of the component arcs of the route followed by the traffic. After that, the routing protocol has to adapt to the network changes. While the routes are not updated, the source continues to send traffic on a failed route. Some packets will be lost, at a quantity proportional to the time the routes are not updated, since the traffic generated is CBR.

The topologies used in the simulations were produced by GT-ITM (Georgia Tech Internetwork Topology Models), a freeware software that generates network topologies randomly with some constraints [16]. The used topologies were created with the flat random model with 10, 15, 25, 30, 35, 40 and 50 nodes. Three different topologies were produced with 10 nodes. The difference between them is the connectivity (quantity of links between the nodes). These links are generated randomly by GT-ITM. The link probability for the three ten-node topologies is respectively $p = 20\%$, $p = 50\%$ and $p = 80\%$. For larger topologies, this probability is smaller, because of the greater number of nodes. With 50 nodes and $p = 20\%$ problems were encountered with the simulator due to the system size (memory consumption).

By convention, the CBR source was always set at node 0 sending packets to the "last" node in the topology (e.g. node 14 in the fifteen node topology). So the used route was randomly produced, and its size (in hops) is variable in the different experiments. Given a route, we show the packet loss related to the failure of the various links in the route. These packet losses are proportional to the algorithm complexity, because of the delay we introduced at the route updates. This delay is proportional to the number of steps spent by the algorithm. This measure represents an evaluation of the complexity (number of iterations needed) of each of the two algorithms implemented.

The packet losses measured for each topology are shown in the tables 1 and 2 (separated by the number of nodes). The values presented in the first two columns refer to the percentage of packets lost on a CBR transmission of fixed duration, for each of the algorithms. Each line correspond to a link failure in the topology, each link being identified by the nodes it connects. The third column represents the percentile difference between the first two measures: the advantage of one algorithm over the other. The packet losses (two first columns of the tables) present a growing tendency with the topology size. Here, the topology size refers not only to the number of nodes but also to the quantity of links that exist between these nodes. The packet losses are not directly related to the quantity of nodes in the topology. This is because in the distance-vector model the quantity of neighbors of a router plays a more important role in the processing charge (route computation is distributed, and routers do not have knowledge of the entire topology). The amount of messages processed by a router is proportional to the quantity of neighbors it is connected to. Nevertheless, when the number of nodes grows, it is likely that there will exist nodes with larger quantities of neighbors, so this tendency of increasing processing charges is expected.

If a link disconnects the graph, all packets are lost. This is what happens link 0_1 fails in the first ten-node topology, where the route used was 0_1_2_9. These cases were omitted from the tables.

As the topology size grows, the packet losses for the $3M$ classical approach (non-scalable) grows at a rate larger than for the SMM approach. Thus the third column of the tables presents values that have an increasing tendency. It is important to observe also that in the experiments made the SMM approach never has a performance worse than the $3M$ approach. The average performance gain when considering all the experiments is about 40%.

6 Final Considerations

QoS-based routing must be considered as a main component to provide new service classes on the Internet. This routing scheme has several requirements and each of its components has specific questions to be considered. Protocol scalability is a major issue of such a routing mechanism, and it is directly related to the route discovery algorithm under utilization.

In this work, after studying the solutions proposed in the literature, an algorithm that takes into account three metrics on route computation, independently, and with the same complexity as two-metric algorithms was constructed. This is done with simple mathematical calculations, which

Table 1: Packet losses for the ten, fifteen, twenty and twenty-five node topologies.

	Algorithm		Percentile Difference
Failure	Three metric ($3M$)	Single metric (SMM)	%
10 nodes – 12 links – route 0_1_2_9			
Link 1-2	8.1	5.6	31
Link 2-9	6.8	4.9	28
10 nodes – 16 links – route 0_9			
Link 0-9	13.3	10.9	18
10 nodes – 34 links – route 0_2_3_9			
Link 0-2	10.5	6.7	36
Link 2-3	9.0	4.7	48
Link 3-9	12.9	9.4	35
15 nodes – 18 links – route 0_9_6_1_14			
Link 0-9	8.6	4.1	52
Link 9-6	11.3	6.0	47
20 nodes – 35 links – route 0_2_6_19			
Link 0-2	18.2	7.7	58
Link 2-6	18.2	8.6	53
Link 6-19	23.3	14.3	39
25 nodes – 54 links – route 0_21_24			
Link 0-21	23.6	12.8	46
Link 21-24	25.7	17.4	32

Table 2: Packet losses for the thirty, thirty-five, forty and fifty node topologies.

	Algorithm		Percentile Difference
Failure	Three metric ($3M$)	Single metric (SMM)	%
30 nodes – 68 links – route 0_29			
Link 0-29	33.6	16.7	50
35 nodes – 61 links – route 0_19_17_34			
Link 0-19	33.8	17.6	48
Link 19-17	32.6	19.9	39
Link 17-34	41.8	26.2	37
40 nodes – 53 links – route 0_3_2_22_1_17_39			
Link 22-1	25.9	11.3	56
Link 1-17	32.6	15.9	51
50 nodes – 65 links – route 0_27_11_49			
Link 0-27	17.1	7.5	56
Link 27-11	16.5	6.9	58
Link 11-49	59.7	13.7	77

can be performed in advance to route discovery. The utilization of three metrics guarantees that a number of applications larger than with just two metrics may be well characterized. The metrics used favor the implementation of dynamic routing tables, since two of these metrics are likely to be "static". This fact, combined with the low complexity of the proposed algorithm and the definition of routing tables per class of service (instead of a per-flow basis), improve the protocol scalability.

With the implementation of the proposed routing scheme inside the NS simulator, it was observed that the proposed algorithm provides route computations in a scalable way. The simplicity of the proposed scheme is its most important characteristic. Scalable QoS-based routing seems to be feasible only by the utilization of schemes whose complexity compares with today's routing algorithms. The experiments together with the analysis made showed that the SMM algorithm is a promising approach to provide scalable QoS routing. Compared to a classical approach, the average performance gain is about 40%.

References

[1] R. Braden, D. Clark, and S. Shenker, *RFC1633: Integrated Services in the Internet Architecture: An Overview.* Network Working Group, June 1994.

[2] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, *RFC2475: An Architecture for Differentiated Services.* Network Working Group, Dec. 1998.

[3] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin, *RFC2205: Resource Reservation Protocol (RSVP) - Version 1 Functional Specification.* Network Working Group, Sept. 1997.

[4] E. Crawley, R. Nair, B. Rajagopalan, and H. Sandick, *RFC2386: A Framework for QoS-Based Routing.* Network Working Group, Aug. 1998.

[5] W. C. Lee, M. G. Hluchyj, and P. A. Humblet, "Routing subject to quality of service constraints in integrated communication networks," *IEEE Network Magazine*, vol. 9, pp. 46–55, July-Aug. 1995.

[6] R. Vogel, R. G. Herrtwich, W. Kalfa, H. Wittig, and L. C. Wolf, "QoS-based routing of multimedia streams in computer networks," *IEEE Journal on Selected Areas in Communications*, vol. 14, pp. 1235–1244, Sept. 1996.

[7] H. F. Salama, D. S. Reeves, and Y. Viniotis, "A distributed algorithm for delay-constrained unicast routing," in *IEEE INFOCOM'97*, 1997.

[8] S. Chen and K. Nahrstedt, "On finding multi-constrained paths," in *IEEE International Conference on Communications*, (Atlanta), June 1998.

[9] Q. Ma and P. Steenkiste, "Quality of service routing for traffic with performance guarantees," in *IFIP Fifth International Workshop on Quality of Service*, (New York), pp. 115–126, May 1997.

[10] Z. Wang and J. Crowcroft, "Quality of service routing for supporting multimedia applications," *IEEE Journal on Selected Areas in Communications*, vol. 14, pp. 1228–1234, Sept. 1996.

[11] D. H. Lorenz and A. Orda, "Qos routing in networks with uncertain parameters," in *IEEE INFOCOM'98*, 1998.

[12] P. P. White, "RSVP and integrated services in the internet: A tutorial," *IEEE Communications Magazine*, vol. 35, pp. 100–106, May 1997.

[13] L. L. H. Andrew and A. A. N. A. Kusuma, "Generalised analysis of a QoS-aware routing algorithm," in *IEEE GLOBECOM'98*, pp. 118–123, 1998.

[14] K. Fall and K. Varadhan, *NS notes and documentation.* University of California at Berkeley/Lawrence Berkeley National Laboratory, Mar. 1999. Available at http://www-mash.cs.berkeley.edu/ns.

[15] G. Malkin, *RFC1721: RIP Version 2 Protocol Analysis.* Network Working Group, Nov. 1994.

[16] E. W. Zegura, K. Calvert, and S. Bhattacharjee, "How to model an internetwork," in *IEEE INFOCOM'96*, vol. 2, pp. 594–602, Mar. 1996.