

QoS-based multi-domain routing under multiple QoS metrics

Derya Yiltas and Harry Perros

Computer Science Department

NC State University

Raleigh, NC 27511, USA

{dyiltas, hp}@csc.ncsu.edu

Abstract

Applications such as voice and video require network paths that satisfy several different quality of service QoS metrics, such as delay, jitter, packet loss rate, and availability. The calculation of paths under multiple QoS metrics, such as the above four metrics, is a difficult problem since these metrics are in general incompatible. We propose a simple method for combining the above four QoS metrics into a single composite QoS metric which can be used as a link cost in Dijkstra's algorithm in order to calculate a path. We evaluated the proposed method in a multi-domain routing environment where domain reachability information is available through a Service Oriented Architecture paradigm, and we show that it outperforms two commonly used methods. The results are also applicable to routing within a single domain.

1. Introduction

The advent of the Next Generation Networks (NGN) and its emphasis on QoS capable transport networks has brought to prominence the need for connections with QoS guarantees. NGN services will require a seamless connectivity between heterogeneous networks operated by different Internet service providers (ISPs). Consequently, the issue of how to establish a connection with QoS requirements across multiple domains operated by different operators has become an important and challenging research topic.

The issue of multi-domain routing under QoS constraints has been addressed in the open literature. Yannuzzi et al [1] gave an overview of the technical challenges in finding QoS-constrained multi-domain disjoint paths. The primary challenging problems are domain visibility and scalability of domain representation. A possible solution to this problem is to assume some type of topology aggregation, such as PNNI which was defined in the early 1990s for ATM networks. Several recent studies have addressed different aspects of domain aggregation. For instance, Griffin et al [2] and Xiao et al [3] extended BGP to carry aggregated QoS information, Verdi et al [4] proposed link virtualization to facilitate QoS-enabled inter-domain routing, and Sarangan et al [5] studied capacity-aware state aggregation for multi-domain QoS routing. Manvi and Venkataram [6] used a network management approach where network nodes are equipped with agents that provide QoS information including bandwidth, delay, jitter, and packet loss rate for each link to a routing manager. Using this scheme, they showed that bandwidth utilization and call success ratios improved. Yen et al [7] used battery energy along with bandwidth and delay in their multi-constrained QoS routing protocol for mobile ad hoc networks.

IETF established the Working Group “Path Computation Element” (PCE), see Farrel et al [8], in order to address the issues of multi-domain routing. PCE has proposed an architecture where each domain has one or more PCE in charge of constrained path computation within a domain.

PCEs belonging to different domains communicate with each other in order to establish a path across multiple domains. A comparison of different PCE-based routing schemes can be found in Geleji et al [9] and Geleji and Perros [10].

An alternative approach to multi-domain routing under QoS constraints is based on the Service Oriented Architecture (SOA). Network operators post their available connectivity along with price and QoS guarantees in a service directory. A separate entity accesses this directory in order to select a set of appropriate connections for an end-to-end path, see Williams [11], and Verdi et al [4].

The problem of path computation under multiple QoS metrics has been studied in the literature. Let us consider a packet switched network, and let each link i be associated with K different metrics m_{ik} , $k=1,2,\dots,K$. Let $w_k(p)$ be the total value of metric k for a given path p , and let C_k be the bound of the path constraint for metric k , i.e., $w_k(p) \leq C_k$, $k=1,2,\dots,K$. Jaffe [12] considered two different metrics per link and proposed a weighted combination of these two values to form a single metric. Let $w_1(i)$ and $w_2(i)$ be the values of metric 1 and 2 for the i th link. Then, the combined metric per link was obtained using the expression $d_1 w_1(i) + d_2 w_2(i)$, where d_1 and d_2 are two metric-dependent weights given by the expression $d_2/d_1 = (C_1/C_2)^{1/2}$, where C_1 and C_2 are the bounds of the path constraints for metric 1 and 2 respectively. Using this link composite metric, Dijkstra's algorithm is then used to calculate the optimum path. De Neve and Van Mieghem [13] used the following function to describe the total cost of a path

$$\max \left[\frac{w_1(p)}{C_1}, \frac{w_2(p)}{C_2}, \dots, \frac{w_k(p)}{C_k} \right] \quad (1)$$

and proposed a heuristic algorithm that finds a path which has the minimum cost, i.e., minimizes the above function. We shall refer to this scheme of minimizing function (1) over all paths as the *min-max* method. Korkmaz and Krunz [14] proposed a heuristic algorithm that finds a feasible

path that satisfies K additive constraints, i.e., $w_k(p) \leq C_k$, $k=1,2,\dots,K$, which minimizes a cost function. The feasible path is obtained by minimizing the following non-linear cost function over all paths

$$\left(\frac{w_1(p)}{C_1}\right)^\lambda + \left(\frac{w_2(p)}{C_2}\right)^\lambda + \dots + \left(\frac{w_K(p)}{C_K}\right)^\lambda \quad (2)$$

where λ is an integer number. We shall refer to this scheme of minimizing function (2) over all paths as the *minSS* (minimum sum of squares) method. It has been shown that by minimizing this function at least one of the constraints is satisfied, while the remaining constraints are subject to a bound equal to the original bound times the quantity $\sqrt[\lambda]{K}$, where K is the total number of constraints. Also, it was shown that when λ tends to infinity, it becomes compatible to the min-max function (1). The authors considered primarily the case of $\lambda=2$. Barolli et al [15] combined the end-to-end delay and packet loss rate into the function: $w_d(p)/w_s(p)$, where the numerator is the probability that a packet will be successfully transmitted all the way to its destination, calculated by multiplying the packet loss rate for each link along the path. Khavi et al [16] proposed a function for combining metrics that requires defining a parameter ε . Some validations were given for two metrics for different values of ε .

In this paper, we propose a simple scheme for combining four different QoS metrics, namely, delay, jitter, packet loss rate, and availability, into a single composite one. This composite QoS metric is then used as a link cost in Dijkstra's algorithm to select a path in a multi-domain packet-switched environment. The complexity of the proposed algorithm is that of Dijkstra's algorithm. Once the end-to-end path has been calculated, an MPLS label distribution protocol, such as RSVP, can be employed to setup the connection. The actual details of how this can be implemented are beyond the scope of this paper. We use the SOA paradigm for multi-domain routing as a testbed in order to evaluate our scheme, and we show through experimentation that it

gives better results than the two popular functions, namely, min-max and minSS. While our focus has been on inter-domain path selection, the approach is also applicable to multi-metric path selection within a single domain.

The paper is organized as follows. In the following section we briefly review the SOA paradigm for routing in a multi-domain environment and describe the four QoS metrics: delay, jitter, packet loss rate, and availability. In section 3, we give a geometric interpretation of the minSS function and suggest some alternative functions, and in section 4, we describe our proposed scheme for combining these four QoS metrics. Numerical results are provided in section 5, and finally the conclusions are given in section 5.

2. The SOA framework and the QoS metrics

The SOA framework for inter-domain routing with QoS has been discussed in the open literature, see Verdi et al [4], Williams [11], and Bastiaansen et al [17]. We assume that network providers post connectivity to a service repository. This connectivity is in the form of tunnels established between the border routers of a domain controlled by a network provider, along with usage price, available bandwidth, and QoS metrics of delay, jitter, packet loss rate, and availability. A tunnel could be as an LSP setup using the MPLS architecture. Let us assume that user A wants to establish a connection with user B subject to QoS guarantees, where A and B belong to different domains. A service integrator will access the repository and choose among the offerings by the network providers a set of tunnels that make-up an end-to-end path from A to B subject to the required QoS constraints and costs. The implementation aspects of this SOA scheme are beyond the scope of this paper. We now describe the four QoS metrics considered in this paper.

Delay: The end-to-end delay consists of the fixed propagation delay between the sender and the receiver plus a variable delay which is the sum of the queueing delays encountered by the

packets at each router along the path. Assuming that a router is based on the output buffering design, packets may encounter delays at the output ports of the router while they are awaiting transmission on an outgoing link. The end-to-end delay is typically expressed as an average delay, a fixed upper bound, or a statistical upper bound D expressed as a percentile γ . That is, if X is the end-to-end delay then $P[X \leq D] \geq \gamma\%$. In this paper, we will assume a tunnel delay which includes the queuing delay at the output port, transmission time and propagation time. This value is additive over multiple tunnels. This permits the results to be used assuming mean delays, or fixed upper bounds, or percentile delays. In general percentile delays are not additive, but it is possible to construct a function that will yield the percentile of a mixture of distributions if we know the percentile of the individual component distributions. For details see Anjum and Perros [18].

Jitter: There is no specific metric for the jitter, which is used to indicate the variability in the inter-arrival time of packets at the destination. Various measures have been proposed in the literature, such as, the range (i.e. the difference between the largest and the smallest observed inter-arrival time), the variance of the inter-arrival times, and the γ percentile of the inter-arrival time at the destination. In this paper, we will assume the latter, that is, the tunnel jitter is given by the γ percentile of the inter-arrival time at the destination, see [19]. As explained above, using the results in Anjum and Perros [18], percentiles can be added exactly, and consequently, jitter is additive over multiple links.

Packet loss rate: Let p_i be the packet loss rate at the i th tunnel. Then, for a connection established over, say 1 to K tunnels, the arrival loss rate is $\lambda p_1 + \lambda(1-p_1)p_2 + \lambda(1-p_1)(1-p_2)p_3 + \dots + \lambda(1-p_1)\dots(1-p_{K-1})p_K$, where λ is the arrival rate of packets to the connection. Due to the fact that p_i is in general very small, we have that $1-p_i$ is approximately equal to 1, and therefore, the arrival

loss rate is $\lambda(p_1+p_2+\dots+p_K)$. That is, the end-to-end packet loss rate is obtained by adding up all the individual packet loss rates of the tunnels along the path of the connection.

Availability: This is expressed as the probability that a tunnel is available. That is, it is the percent of time that the tunnel is up. Assuming independence between tunnels, the end-to-end availability is obtained by multiplying all the availability probabilities of the tunnels that make up the connection.

We note that the bandwidth requested for a connection is treated as a constraint, in the sense, that all the tunnels that do not satisfy the bandwidth request will not be considered for the path calculation.

3. Geometric interpretation of the minSS function

For each tunnel i let d_i , j_i , l_i and a_i be the delay, jitter, packet loss rate and availability, respectively. For a path p , consisting of a series of interconnecting tunnels, let $w_d(p)$, $w_j(p)$, $w_l(p)$, and $w_a(p)$, be the total end-to-end delay, jitter, packet loss rate and availability, respectively. The following constraints should be satisfied:

$$w_d(p) \leq D, \quad w_j(p) \leq J, \quad w_l(p) \leq L, \quad w_a(p) \geq A$$

where D , J , L and A are the requested constraints in the SLA on the QoS metrics of delay, jitter, packet loss rate, and availability. Note that the delay, jitter, and packet loss are subject to an upper bound, whereas availability is subject to a lower bound. Each path metric is obtained by summing up the metrics of the tunnels that make up the path, except for the availability where $w_a(p)$ is the product $a_1 a_2 \dots a_l$. Since the other metrics have to be minimized, we convert the path availability to path downtime, that is, the percent of time that the tunnel is down, which is given by the constraint $1 - w_a(p) \leq 1 - A$. Consequently, function (2), for $\lambda=2$, used in the minSS method can be written as follows:

$$(w_d(p)/D, w_j(p)/J)$$

$$w_d(p)/D$$

$$d^*/D$$

$$(j^*/J, d^*/D)$$

$$\left(\frac{w_d(p)}{D}\right)^2 + \left(\frac{w_j(p)}{J}\right)^2 + \left(\frac{w_l(p)}{L}\right)^2 + \left(\frac{1-w_a(p)}{1-A}\right)^2 \quad (3)$$

This function is minimized over all paths. We note that by dividing each path metric by its constraint bound, we automatically normalize these metrics to the [0,1] space. This eliminates the problem of the metrics taking values from sets with different ranges. For instance, for a given path p the delay could take values in [1,100] whereas the jitter is a lot less and it may take values in [1,20]. We also note that for $\lambda=2$, expression (3) is the squared distance from zero. For presentation purposes, let us consider only two QoS metrics, namely delay and jitter. Then, the set of paths that satisfy the two metric constraints are within the shaded square in Figure 1.

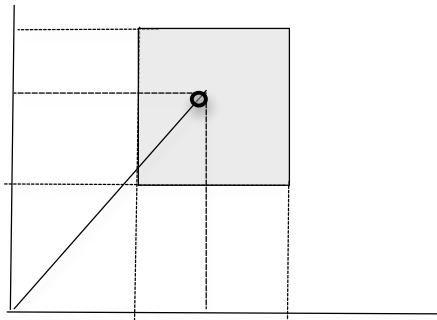


Figure 1: The feasible set of values of function (3)

Since all paths p satisfy the delay and jitter constraint bounds, i.e., $w_d(p)/D \leq 1$ and $w_j(p)/J \leq 1$, all the paths lie within the square bounded by the X and Y axes and the two dotted lines defined by the points $\{(0,1), (1,1)\}$ and $\{(1,0), (1,1)\}$. Now, let us assume that we run Dijkstra's algorithm using delay as a link cost and let d^* be the minimum delay. Likewise, let j^* be the minimum jitter if we ran Dijkstra's algorithm using jitter as a link cost. Then, obviously no paths can lie below the horizontal line crossing the Y axis at the point d^*/D , and left of the vertical line

crossing the X axis at the point j^*/J . As a result the set of feasible paths lie within the shaded box.

Consequently, the expression

$$\left(\frac{w_d(p)}{D}\right)^2 + \left(\frac{w_j(p)}{J}\right)^2 \quad (4)$$

can be seen as the squared distance from (0,0), and minimizing it attempts to find a path that is close to the value $(j^*/J, d^*/D)$, which may not always be achievable.

Based on the above observation, alternative schemes can be used, such as, maximize the distance from the worst case which is (1,1), i.e.,

$$G1 = \max_p \left\{ \left(1 - \frac{w_d(p)}{D}\right)^2 + \left(1 - \frac{w_j(p)}{J}\right)^2 \right\}$$

or, minimize the distance from the point $(j^*/J, d^*/D)$, i.e.,

$$G2 = \min_p \left\{ \left(\frac{d^*}{D} - \frac{w_d(p)}{D}\right)^2 + \left(\frac{j^*}{J} - \frac{w_j(p)}{J}\right)^2 \right\}$$

which requires first to calculate d^* and j^* . In the same vein, since d^* and j^* are large numbers (bigger than l), we can maximize the distance from (j^*, d^*) , i.e.,

$$G3 = \max_p \left\{ \left(d^* - \frac{w_d(p)}{D}\right)^2 + \left(j^* - \frac{w_j(p)}{J}\right)^2 \right\}$$

For the case of packet loss rate and downtime, the optimum values l^* and $1-a^*$ are less than l , and therefore $G3$ is expressed as:

$$G4 = \min_p \left\{ \left(l^* - \frac{w_l(p)}{L}\right)^2 + \left((1-a^*) - \frac{1-w_a(p)}{1-A}\right)^2 \right\}$$

or

$$G5 = \max_p \left\{ \left(1 - \frac{w_l(p)}{L}\right)^2 + \left(1 - \frac{1-w_a(p)}{1-A}\right)^2 \right\}.$$

The above functions were implemented in a genetic algorithm and were compared against each other. Numerical comparisons of $G1$, $G2$ and $G3$ with the minSS scheme for delay and jitter only, showed that they produce similar results. Likewise, similar results were obtained when we compared $G4$ or $G5$ with the minSS scheme for packet loss rate and availability only. For the four metrics, we compared the minSS scheme against a metric which was a combination of $G3$ and $G4$ or $G3$ and $G5$. Again, similar results were obtained. The details of these numerical comparisons are omitted since they are not of any particular interest.

4. The new composite QoS metric

In this section, we propose a simple method for combining the four metrics for each tunnel to a single composite metric. The optimum multi-domain path is then calculated using Dijkstra's algorithm where the link cost is the composite metric. This method is also applicable to a single domain, where each link is characterized by the four QoS metrics.

This composite metric was motivated by the notion of the fitness function used in genetic algorithms. The idea behind the fitness value is to rank linearly the values of each metric from 1 and N , and then normalize the rankings by dividing each of them by their sum $1+2+\dots+N = N(N+1)/2$. Specifically, let us consider the delay metric and let the set of all delays be d_1, d_2, \dots, d_N , where d_i is the delay of the i th tunnel. This sequence is first sorted in an ascending order and then the i th element of this ordered list is mapped to integer i , which is then divided by $N(N+1)/2$. Let us assume that the i th element of the sorted list belongs to the j th tunnel. Then, the value $i/[N(N+1)/2]$ is the fitness value of this tunnel for the delay metric. As an example, let us assume that $N=4$, and $\{d_1, d_2, d_3, d_4\} = \{2, 4, 3, 1\}$. Then, the sorted list is $\{1(4), 2(1), 3(3), 4(2)\}$, where the value in the parenthesis indicates the tunnel number. Then, the fitness values for the delay for tunnel 1, 2, 3, 4 are $2/10, 4/10, 3/10, 1/10$ respectively.

The same procedure is used for the jitter and packet loss rate. The availability is treated in the opposite manner. That is, the set of all availability values a_1, a_2, \dots, a_N is sorted out in a descending order and the i th element of this ordered list is matched to the i th integer. The integers are then normalized by dividing them by $N(N+1)/2$. As a result, the highest tunnel availability is matched to the lowest fitness value $1/[N(N+1)/2]$, and the lowest availability is matched to the highest fitness value $N/[N(N+1)/2]$.

Let $fv(d_i)$, $fv(j_i)$, $fv(l_i)$, and $fv(a_i)$ be the fitness values for the delay, jitter and packet loss rate of the i th tunnel. Then, the composite metric cm_i for the i th tunnel is constructed by simply adding these values, that is, $cm_i = fv(d_i) + fv(j_i) + fv(l_i) + fv(a_i)$, $i=1, \dots, N$. The best path is then calculated using Dijkstra's algorithm with the above composite link cost. We shall refer to this scheme as *Dijkstra(CM)*, where CM stands for composite metric.

We note that the linear ranking used in the proposed algorithm is the default method employed in genetic algorithms. We also considered the case where the ranking of the values of each metric is proportional to the size of the metric value, as well as a combination of linear ranking for some metrics and proportional ranking for the rest of them. Neither approaches gave as good results as the linear ranking.

5. Numerical results

In this section we provide numerical comparisons between our method, *Dijkstra(CM)*, and the min-max and minSS schemes. The numerical results were obtained using the multi-domain network shown in Figure 2. This is a fictitious network with sufficient path diversification that permits the generation of paths from 2 to 10 hops, a hop being a tunnel. Randomness is introduced by varying the values of the QoS metrics of the tunnels. In several papers authors generate random topologies which they then use to test their routing algorithms. The problem

with this approach is that there is no control on the complexity of the generated topologies, and as a result the tests carried out on these topologies maybe trivial.

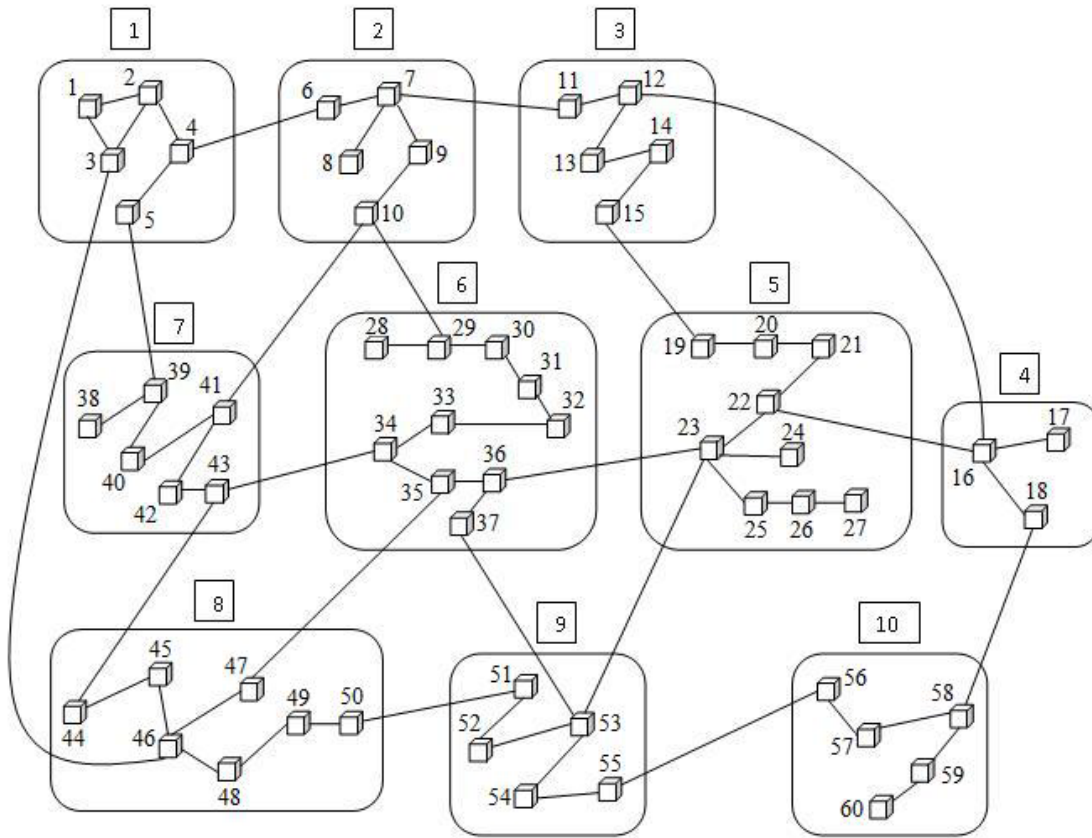


Figure 2: The multi-domain network

The network consists of 10 domains and 60 routers, and the links between routers indicate the tunnels within domains and between domains. Routers that lie along a tunnel are not shown. Each tunnel is associated with a delay, jitter, packet loss rate, and availability. In order to vary the set of QoS metrics, we classified the tunnels within domains into short, medium, and long as follows. All tunnels in domains with 3 or 5 routers, i.e., domains 1, 2, 3, 4, 9, and 10, were classified as short. All tunnels in domains with 6 or 7 routers, i.e., domains 7 and 8, were classified as medium, and all tunnels in domains with 9 or 10 routers, i.e., domains 5 and 6, were classified as long. In addition, the inter-domain tunnels were classified in the same manner. All tunnels

between consecutively numbered domains and between domains 2 and 6 were classified as short. All tunnels between domain pairs of 2-7, 3-5, 5-9, 6-8, and 6-9, were classified as medium, and all tunnels between domain pairs 1-7, 1-8, and 4-10, were classified as long.

Type Name	Delay	Jitter	Packet Loss Rate	Availability
Short	[1,50]	[1,5]	0.000001	[0.9999,0.99999]
Medium	[51,100]	[6,10]	0.00001 or 0.0001	[0.999,0.9999]
Long	[101,150]	[11,15]	0.001	[0.99, 0.999]

Table 1: Tunnel types and related QoS metric ranges

The distribution of the values each QoS metric takes is assumed to be uniform within the range of values shown in Table 1. We assumed that the delay, jitter, and packet loss rate are proportional and the availability is inversely proportional to the tunnel length. The idea behind this tunnel classification is based on the notion that an operator will maintain tunnels with different overall quality.

For the min-max scheme we adapted (1) to the specific four QoS metrics considered in this paper as follows:

$$\max \left[\frac{w_d(p)}{D}, \frac{w_j(p)}{J}, \frac{w_l(p)}{L}, \frac{1-w_a(p)}{1-A} \right] \quad (5)$$

For the minSS scheme we used expression (3) reproduced below for presentation purposes:

$$\left(\frac{w_d(p)}{D} \right)^2 + \left(\frac{w_j(p)}{J} \right)^2 + \left(\frac{w_l(p)}{L} \right)^2 + \left(\frac{1-w_a(p)}{1-A} \right)^2 \quad (6)$$

We recall that function (5) is used in a heuristic algorithm that finds a path that has the minimum cost among all paths; likewise for function (6). In this paper, we calculate the optimum path that minimizes the path cost given by (5) respectively (6) using a genetic algorithm that we implemented. For a given source-destination pair we generated 30 different paths for the initial

generation that satisfied the following four constraints of delay, jitter, packet loss rate, and availability, indicated in section 3:

$$w_d(p) \leq D, w_j(p) \leq J, w_l(p) \leq L, w_a(p) \geq A$$

Then, we regenerated 30 paths using function (5) respectively (6). Subsequently, we carried out regeneration and crossover which also yielded 30 paths of which we selected the best one using function (5) respectively (6). (We note that the bandwidth requested for a connection is treated as a constraint, in the sense, that all the tunnels that do not satisfy the bandwidth request are not considered for the path calculation. In view of this, in our experiments, we assumed that all tunnels satisfy the bandwidth requirements.)

The results in Figures 3, 4, 5, and 6 are given for 2, 4, 6, 8, and 10 hop paths and they were obtained as follows. We first generated 300 different QoS matrices, each consisting of four stochastic variates for the delay, jitter, packet loss rate, and availability, for each tunnel. These stochastic variates were obtained using the uniform distributions reported in Table 1. For the h hop path, $h=2,4,6,8,10$, we selected five different source-destination pairs from Figure 2 whose minimum distance is exactly h hops. For each of these pairs and for each QoS matrix, we calculated the best path using a) the genetic algorithm with function (5) (min-max), b) the genetic algorithm with function (6) (minSS), and c) our method, Dijkstra(CM). As a result, for each number of hops h and for each path calculation scheme we obtained 1500 independent replications (i.e., 300 for 5 paths), based on which we calculated the average and its 95% confidence interval of the delay, jitter, packet loss rate, and availability. Finally, in order to obtain a baseline comparison, we also calculated the best path using Dijkstra's algorithm with each individual QoS metric, i.e., delay, jitter, packet loss rate and availability. We note that the confidence intervals were quite small and therefore they have not been given in the graphs below.

Figure 3 gives the delay of the best path computed using min-max, minSS and Dijkstra(CM). We note that our method outperforms min-max and minSS as the length of the path increases. It also gives results which are very close to the optimum results as far as delay is concerned, obtained using Dijkstra's algorithm with delay as link cost. Figure 4 gives similar results for

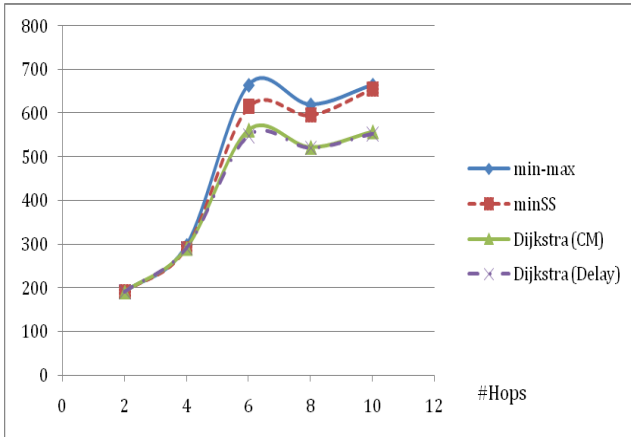


Figure 3: Delay (uniform distribution)

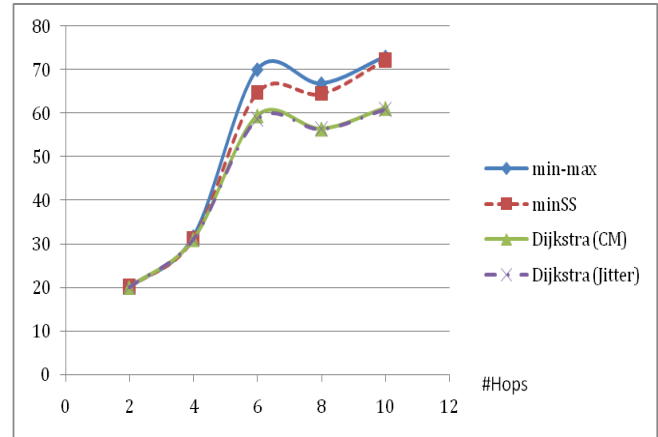


Figure 4: Jitter (uniform distribution)

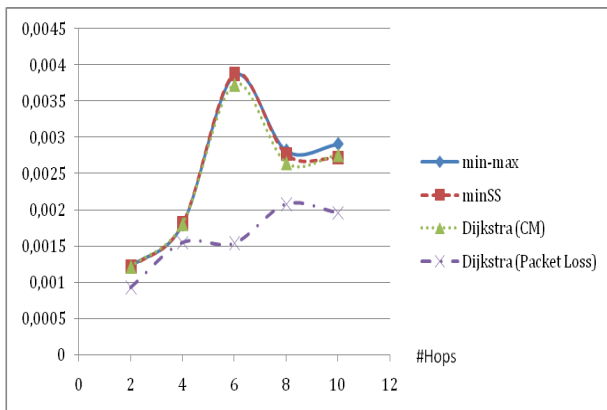


Figure 5: Packet loss rate (uniform distribution)

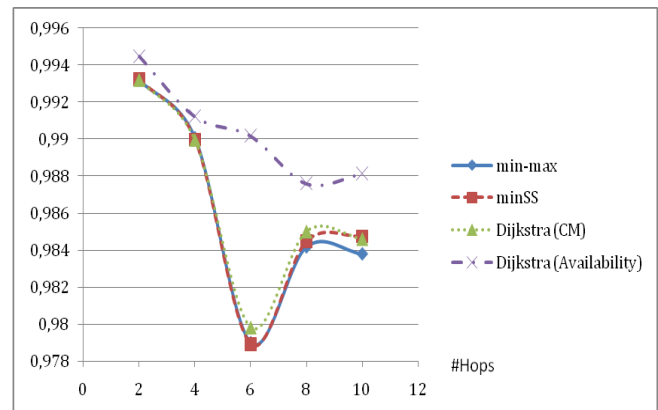


Figure 6: Availability (uniform distribution)

jitter, and similar observations as in Figure 3 hold. Figures 5 and 6 give results for the packet loss rate and availability respectively. We note that all three methods have similar performance. They

all deviate from the optimum solution for hops greater than 4. In general minSS and Dijkstra(CM) outperforms min-max.

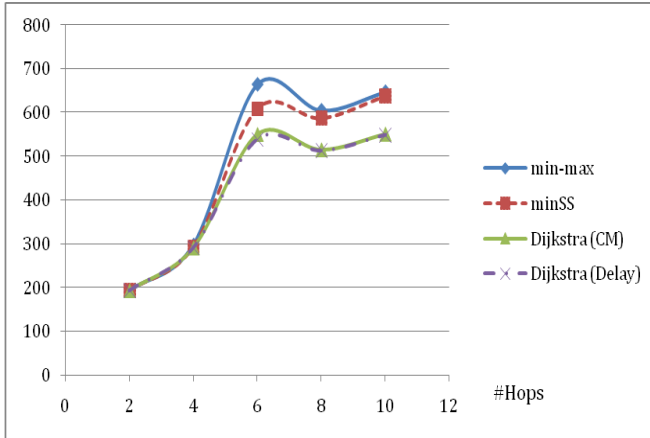


Figure 7: Delay (normal distribution)

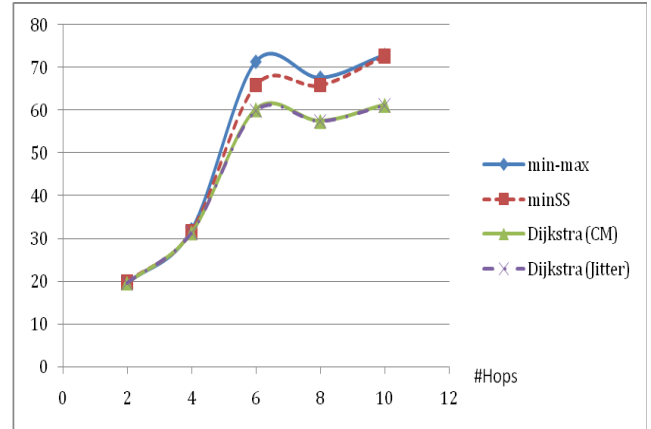


Figure 8 : Jitter (normal distribution)

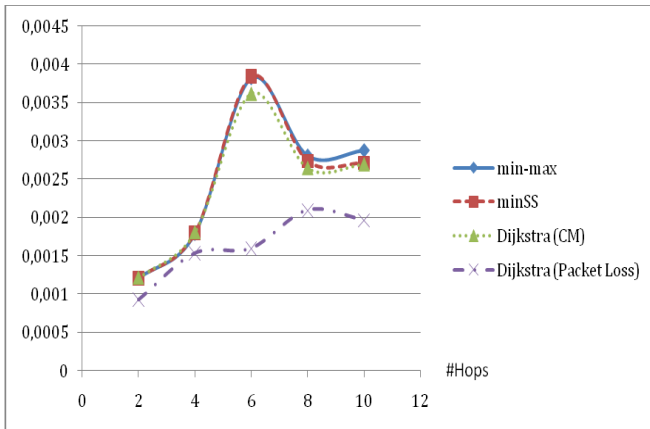


Figure 9: Packet loss rate (normal distribution)

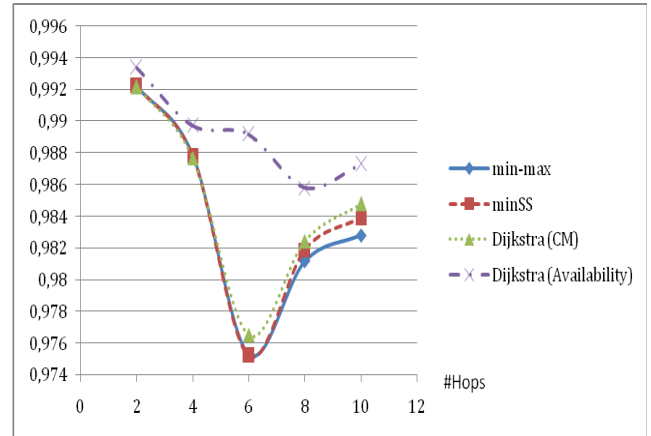


Figure 10: Availability (normal distribution)

A second set of experiments was carried out assuming that the range of values that each metric takes as reported in Table 1 is not uniformly distributed but normally distributed. Each normal distributions was matched to its corresponding uniform distribution by using the same mean and fixing the standard deviation so that the mean +/- 3 standard deviations was equal to the upper/lower bound of the uniform distribution. The results are given in Figures 7 to 10. We note that the results are similar to those given in Figures 3 to 6.

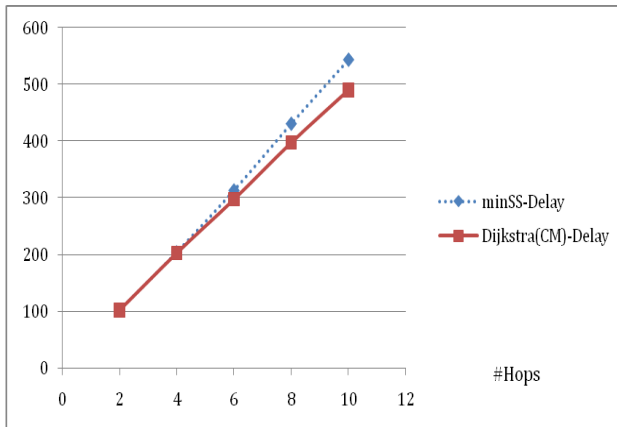


Figure 11: Delay (identical distribution)

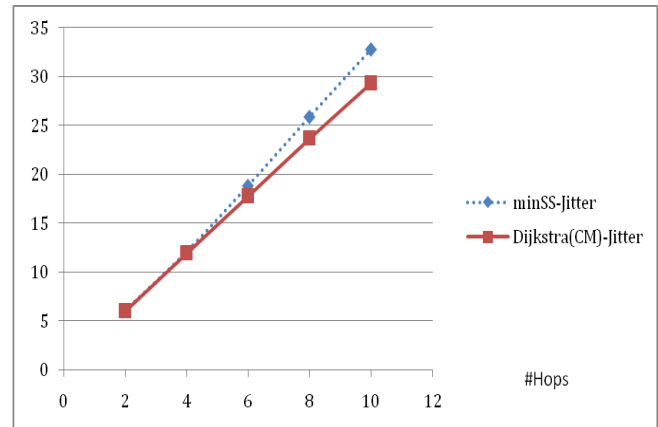


Figure 12: Jitter (identical distribution)

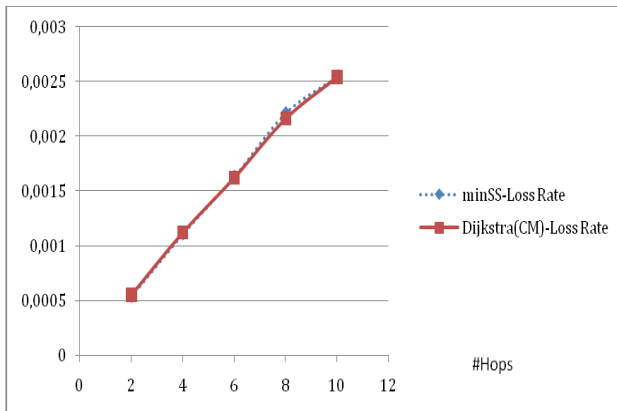


Figure 13: Packet loss rate (identical distribution)

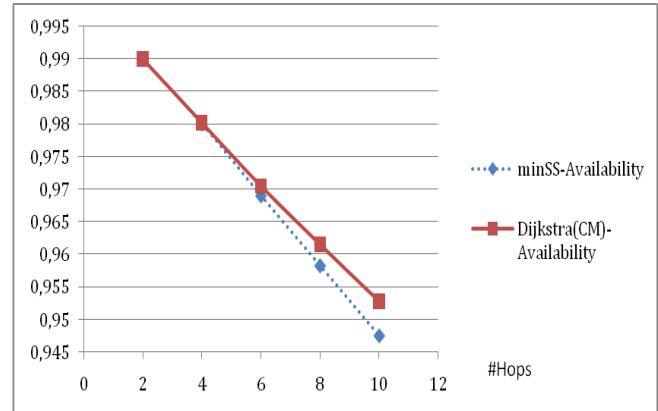


Figure 14: Availability (identical distribution)

A further comparison between minSS and Dijkstra(CM) is given in Figures 11 to 14 under the assumption that all tunnels are identical as far as the four QoS metrics are concerned. (The min-max was not included in these graphs and also in the subsequent comparisons, since it does not perform as well as minSS.) Specifically, the delay, jitter, packet loss rate, and availability for each tunnel is uniformly distributed in the range $[1,100]$, $[1,5]$, $\{0.000001,0.00001,0.0001, 0.001\}$ and $[0.99,0.99999]$ respectively. Again, Dijkstra(CM) calculates paths with lower delay

and jitter and higher availability as the number of hops increases, than those calculated using the minSS function. Both methods are similar for the packet loss rate.

The results given in graphs 3 to 6 and 7 to 10, were obtained assuming three groups of tunnels, i.e., short, medium, and long with QoS metrics as indicated in Table 1. We assumed that the delay, jitter, and packet loss rate were proportional and the availability was inversely proportional to the tunnel length. We also experimented with different combinations of the distributions for the QoS metrics given in Table 1 for each group of tunnels, of which we report on the following two representative cases.

In [19] it was mentioned that the jitter is quite large in access networks as opposed to wide area networks, because the path from a subscriber to the edge router serving the access network is fixed, as opposed to a wide area network where the operator has more paths to choose from so that a low jitter can be provided. Based on this observation, we switched the distributions for the jitter between the short and the long tunnels, while all other distributions remained as shown in Table 1. That is, we assumed that the jitter is uniformly distributed in the range of [11,15] for short tunnels and [1,5] for long tunnels. In graphs 15 to 18, we give results for the minSS, and Dijkstra(CM). For comparison purposes we also provide the optimum solution using Dijkstra's algorithm with each individual QoS metric. As can be seen Dijkstra(CM) gives better results than minSS for all metrics except jitter for which minSS outperforms Dijkstra(CM) for 10 hops.

The second set of results are given in graphs 19 to 22. They were obtained by switching the distributions for the jitter, packet loss rate, and availability between the short and long tunnels, with the remaining values being the same as in Table 1. In this case, Dijkstra(CM) and minSS have comparable performance, with Dijkstra(CM) performing slightly better than minSS for jitter and packet loss rate, as the number of hops increased.

5. Conclusions

We proposed a simple function for combining four QoS metrics, delay, jitter, packet loss rate, and availability, into a single composite QoS metric. This composite metric can then be used as the link cost in Dijkstra's algorithm to calculate the shortest path. We compared the proposed QoS function against two popular schemes, min-max [13] and minSS [14], and against the individual optimum solutions obtained using Dijkstra's algorithm for each individual QoS metric, for different combinations of values of the QoS metrics. Our proposed method outperforms the min-max and minSS schemes, of which the minSS gives better results than min-max, and it often gives results which are very close to the individual optimum solutions. Our method gives paths that have lower delay and jitter than the minSS scheme as the number of hops of the path increases, and similar packet loss rate and availability as minSS. In addition, our proposed method is extremely easy to use and it can be incorporated in other path calculation algorithms.

6. Acknowledgment

This work has been supported by the Scientific and Technological Research Council of Turkey.

References

- [1] Yannuzzi, M., Masip-Bruin, X., Sanchez, S., Domingo-Pascual, J., Oreda, A., and Sprintson, A., "On the challenges of establishing disjoint QoS IP/MPLS paths across multiple domains", *IEEE Communications Magazine*, Dec. 2006, 44, (12), pp. 60-66.
- [2] Griffin, D., Spencer, J., Griem, J., Boucadair, M., Morand, P., Howarth, M., Wang, N., Pavlou, G., Asgari, A., and Georgatsos, P., "Interdomain Routing Through QoS-Class Planes", *IEEE Communications Magazine*, Feb. 2007, 45, (2), pp. 88-95.
- [3] Xiao, L., Wang, J., Lui, K.-S., and Nahrstedt, K., "Advertising Interdomain QoS Routing Information", *IEEE Journal on Selected Areas in Communications*, Dec. 2004, 22, (10), pp. 1949-1964.
- [4] Verdi, F. L., Magalhães, M. F., Cardozo, E., Madeira, E. R. M., and Welin, A., "A Service Oriented Architecture-based Approach for Interdomain Optical Network Services", *Journal of Network and Systems Management*, June 2007, 15, (2), pp. 141-170.
- [5] Sarangan, V., Ghosh, D., and Acharya, R., "Capacity-Aware State Aggregation for Interdomain QoS Routing", *IEEE Transactions on Multimedia*, Aug. 2006, 8, (4), pp. 792-808.
- [6] Manvi, S.S., and Venkataram, P., "Mobile Agent Based Approach for QoS Routing", *IET Communications*, 2007, Vol. 1, No. 3, pp. 430-439.

- [7] Yen, Y.S., Chang, R.S., and Chao, H.C., “Flooding-limited for multi-constrained quality-of-service routing protocol in mobile ad hoc networks”, IET Communications, 2008, Vol. 2, No. 7, pp. 972-981.
- [8] Farrel, A., Vasseur, J. P., and Ayyangar, A., “Framework for Inter-Domain Multiprotocol Label Switching Traffic Engineering”, IETF RFC 4726, Nov. 2006.
- [9] Geleji, G., Perros, H., Xin, Y., and Beyenne, T., “A Performance Analysis of Inter-Domain QoS Routing Schemes Based on Path Computation Elements”, HONET 2008.
- [10] Geleji, G. and Perros, H., “Multi-domain PCE-based QoS routing”, Technical Report, Computer Science, NC State University, 2010.
- [11] Williams, M., “End-to-End Network Services: What is Really Missing?”, 22nd Asia-Pacific Advanced Network Meeting, Singapore, July 2006.
- [12] Jaffe, J.M., “Algorithms for finding paths with multiple constraints”, Networks, 1984, 14, (1), pp. 95–116.
- [13] De Neve, H., and Van Mieghem, P., “A Multiple Quality of Service Routing Algorithm for PNNI”, Proc. IEEE ATM Workshop, Fairfax, VA, USA, May 1998, pp. 324-328.
- [14] Korkmaz, T., and Krunz, M., “Multi-constraint Optimal Path Selection”, Proc. 20th Annual Joint Conf. of the IEEE Computer and Communications Societies (INFOCOM), Anchorage, AK, USA, April 2001, pp. 834-843.
- [15] Barolli, L., Koyoma, A., Matsumoto, K., Saganuma, T., and Shiratori, N., “A Genetic Algorithm Based Routing Method Using Two QoS Parameters”, Proc. 13th International Conf. and Workshop on Database and Expert Systems Applications (DEXA’02), Aix-En-Provence, France, Sept. 2002, pp. 7-11.
- [16] Khadivi, P., Samavi, S., Todd, T.D., and Saidi, H., “Multi-constraint QoS Routing Using a New Single Mixed Metric”, Proc. IEEE Int. Conf. Communications (ICC 2004), Paris, France, June 2004, Vol. 4, pp. 2042-2046.
- [17] Bastiaansen, H., and Hermans, P., “Managing Agility Through Service Orientation in an Open Telecommunication Value Chain”, IEEE Communications Magazine, Oct. 2006, 44, (10), pp. 86-93.
- [18] Anjum B. and Perros, H., “Adding percentiles for end-to-end quality of services”, Technical Report, Computer Science, NC State University, 2009.
- [19] “Inter-provider Quality of Service”, White paper draft 1.1 November 17, 2006, a white paper prepared by the Quality of Service Working Group MIT Communications Futures Program (CFP)

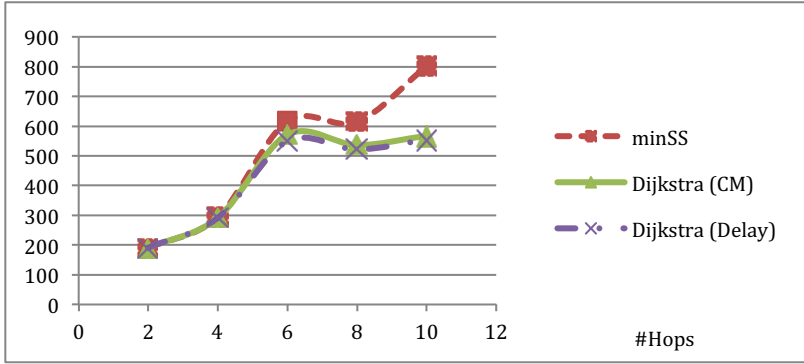


Figure 15: Delay (switched jitter distributions)

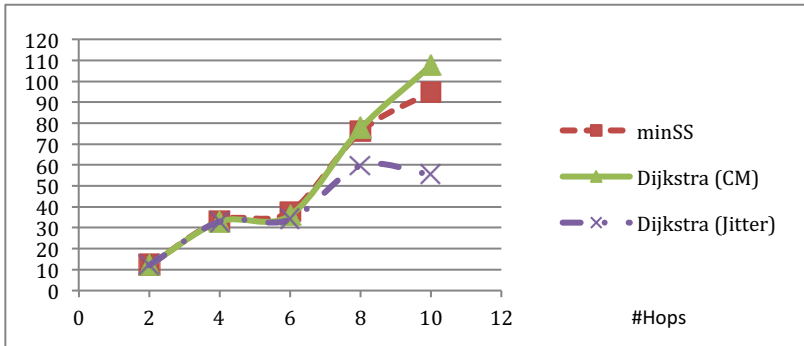


Figure 16: Jitter (switched jitter distributions)

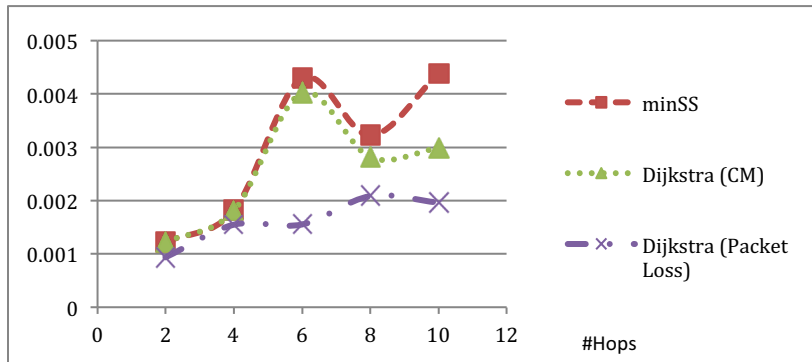


Figure 17: Packet loss rate (switched jitter distributions)

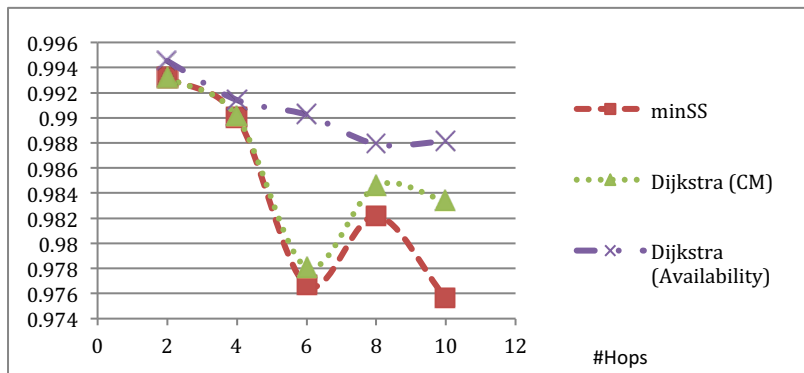


Figure 18: Availability (switched jitter distributions)

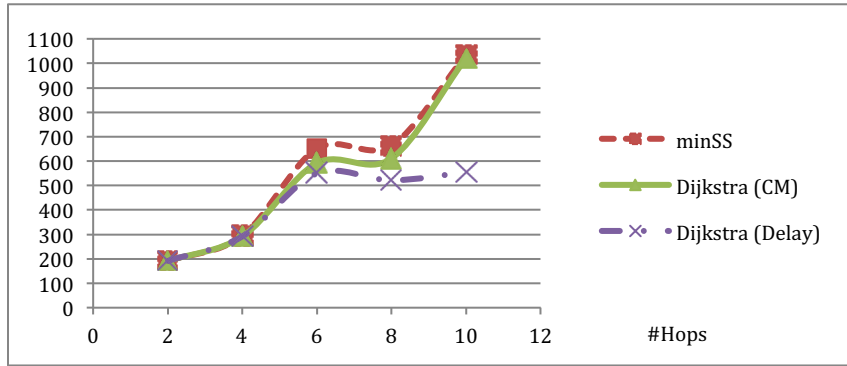


Figure 19: Delay (switched jitter, packet loss rate, and availability distributions)

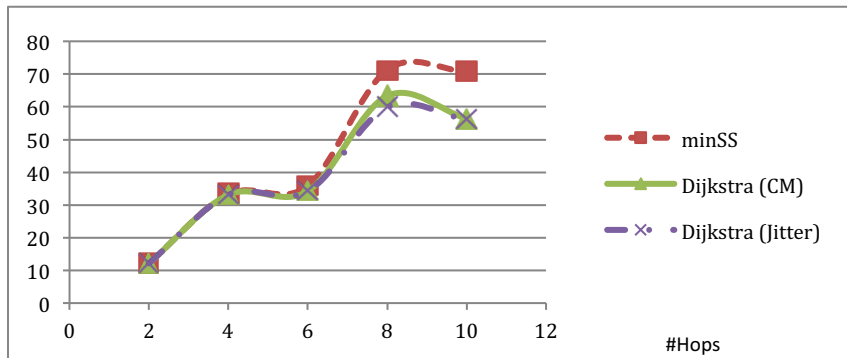


Figure 20: Jitter (switched jitter, packet loss rate, and availability distributions)

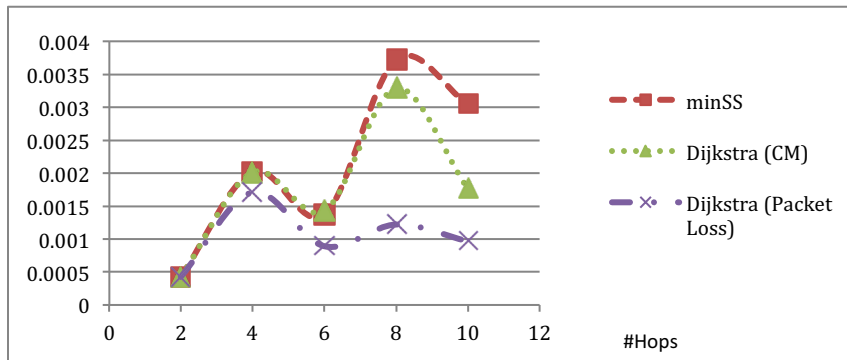


Figure 21: Packet loss rate (switched jitter, packet loss rate, and availability distributions)

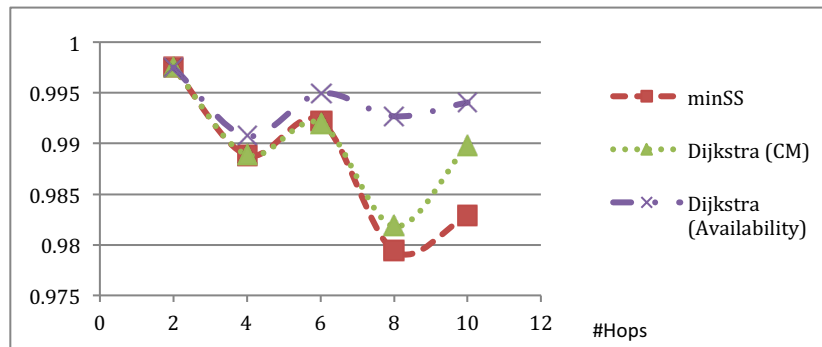


Figure 22: Availability (switched jitter, packet loss rate, and availability distributions)