Derya YILTAS KAPLAN/ IU-JEEE Vol. 15(2), (2015), 1913-1919

# COST FUNCTIONS FOR TWO-METRIC QUALITY OF SERVICE ROUTING

Derya YILTAS KAPLAN

Istanbul University, Engineering Faculty, Department of Computer Engineering
34320, Avcilar, Istanbul-TURKEY
dyiltas(at)istanbul.edu.tr

***Abstract:*** *Different metrics are used for measuring the goodness of packet routing between the source and destination pairs in a communication network. One approach is to compute the paths from the measured metric values, which are predetermined independently according to the network resources. The combination of some metrics of different natures to get a composite cost function for the routing process is non-trivial. We consider two metrics, namely delay and packet loss rate. The first one takes values from the large numbers and the second from the small. We propose some methods for constructing composite functions of these metrics without any constraints and use them for the shortest path calculations. We give the numerical results of the proposed composite functions versus the Dijkstra's algorithm with the individual metrics. We spot out the best function according to our computer simulation results. Our composite function works for any arbitrary point-to-point networks. To the best of our knowledge, the technique is novel. Our results show an efficient way to balance the effects of the metrics in the context of many-to-many routing.*
***Keywords:*** *Cost function, delay, network communications, packet loss rate, QoS routing.*

## 1. Introduction

One of the most important terms during the transmission of voice or data among the networks is Quality of Service (QoS). This key subject is the basic factor specifying the QoS levels especially in multimedia applications. There are many factors having an influence on QoS of the network. One of them is the route selection or routing, which includes discovering different paths along a network. The metrics in the routing process have been mostly defined in the literature, according to their effects on the whole QoS of the network. Hanzo II and Tafazolli [1] presented a metric classification related to network, link, and physical layers of the OSI model. Some of their metrics are end-to-end delay, packet loss rate, jitter, link stability, and bit error rate. Yin, et. al. [2] gave a similar metric list for performance measurement of routing protocols.

Several researchers have studied end-to-end connection implementations that maintain the required metrics in a reasonable range. QoS renegotiation, which is the decision of the probable QoS values, must be performed between network and application layers to get the necessary performance level [3]. Application layer requires some specific QoS values from the transportation layer and then available QoS values related to the network situation are decided. In various studies, this can be done by QoS negotiation. For example Obreja and Borcoci [4] described an approach in which each domain manager is able to establish QoS values without having the traffic information of other domains. Here, the processes of routing and QoS negotiation run independently. During offline computation, paths are chosen on a network where nodes are referred to domain manages. Therefore any scalability problem is prevented. Iqbal, et. al. [5] proposed another QoS scheme which is dependent on the negotiation between network and application layers. Here, network layer gives the information about network congestion to the application layer. Source node's applications adjust sending data rate according to the congestion level of the traffic flows. Routing is concerned in this scheme.

In the literature, Multi-Constrained Path Problem (MCPP) provides to find multiple metric-constrained paths along a network. MCPP is also considered as an optimality problem [6-7]. Djarallah, et. al. [6] defined the metrics as the vectors of weight parameters. So they performed the metrics as non-composite numbers and computed them discretely. After then they used these metrics to get the optimal paths for the source-destination pairs. They initialized an inter-domain route set on the way of this solution. They obtained several feasible paths which satisfy the predetermined metrics and then eliminated some of them which have larger weights. On the last set, they performed an objective function to select the optimal path.

**Received on: 05.01.2015**
**Accepted on: 20.05.2015**

In another multiple metric method [8], the user requests different QoS values and then each service provider tries to find a suitable domain link based on these requested values. Bertrand, et. al. [9] benefited from some additive metrics and computed the total metric values of the links in a relaxation step. They compared these values with each other with the goal of finding multiple paths and obtaining Virtual Shortest Path Tree which includes the path with the minimum cost.

Masip-Bruin, et. al. [10] benefited from Enhanced QoS Border Gateway Protocol to provide the paths. In this scheme, a metric for a path segment is chosen and then an assembling function is used to combine all segment values as the whole path value. After the computations, a decision step is applied through a Degree of Preference function for getting the best path related to the metric values, and also all available paths are stored. In a different multi-constrained routing solution, Shin, et. al. [11] initialized the process with some subpaths and then performed their extensions to reach the whole paths. In this scheme, the number of subpaths is limited during the computations, but then raised to find the rest of the paths in a probabilistic way. Xue and Ganz [12] actuated the routing process depending on the metrics of bandwidth and delay consecutively. Similarly, Yuan and Liu [13] considered $k$ different metrics independently. The main drawback of this method is that the authors should store all optimal paths and provide the QoS requirements by using the constraint of each metric.

In the literature, the path weights are commonly computed based on different metrics independently and then a comparison is made between all metrics to get the best path. The challenging problem of combining various metrics to analyze path optimality has been studied within several algorithms as given briefly in [14].

In this paper, we provide an easy computation to find the path of optimal trade-off between two metrics at the same time. These two metrics fall on different number ranges, one is from very small real numbers, and the other from large integers. We give the structures of some composite functions, and use experiments to find the most effective one. Our work differs from the other works in the literature in the way that the metrics are not necessarily in similar natures. For example, Van Mieghem, et. al. [15] implemented their algorithm with all constraints uniformly distributed on [0, 1].

The rest of this paper is designed as follows. Section 2 gives some necessary definitions. In Section 3, we define two basic composite functions and construct a new composite function, of which we select the terms carefully. Then, it is followed by experimental results. Finally, we give the conclusion and future work of this study in Section 4. In this paper, we put aside the unpredictable practical network environmental issues such as user request, admission control, and security.

## 2. Definitions and Notation

The network, or graph, $G=(V,E)$ in question is a connected, and undirected, static point-to-point arbitrary graph, where $V=\{1,2,...,n\}$ is the set of nodes and $E$ the edge set. Each edge $e$ has a two-dimensional cost (-,-), and the two metrics inside are delay ($D$) and packet loss rate ($PL$). The delay of $e$ can be considered as the average time for a message to wait in the outgoing buffer (queue) of $e$ before departure. The packet loss rate of $e$ is the probability of a message being lost during transmission through $e$. We assume two conditions on these two metrics such as general assumptions in the most QoS studies.

**1.** *The delay and packet loss rate are static for each edge.*
**2.** *The delay and packet loss rate of an edge are not directly proportional to each other.*

We argue for assumption 1. Obviously, this assumption may not be accurate as the metrics are affected by the interference between messages. However, their dynamic values are difficult to trace in real time, and even sophisticated methods can only give approximation. More importantly, there is no algorithmic difference between using their actual values and using the approximated ones. In practice, the system can use the latest known values and construct the composite function after a certain period of time.

We now argue for assumption 2. For unreliable protocol like UDP, a lost packet will not be transmitted again, and therefore, there will not cause any longer delay. For reliable protocol like TCP, or the Data Link layer protocol, a lost packet will be re-transmitted; however, such re-transmission will not necessarily be placed back to the waiting queue as a new coming packet. Hence, longer delay will be experienced but it may not take the same amount of time as before.

The values of delays are chosen from the positive integer set, and those of packet loss rates from the positive real numbers much smaller than 1. Followed from traditions, the delay of a routing path $p$ is the sum of all delays over all edges along $p$. (Using product or sum depends solely on the nature of the metric because they are mathematically convertible to each other.) The definition of the packet loss rate of $p$ is similar, because we assume that the product of the packet loss rates of any two edges is negligible. This assumption allows us to simply minimize the sum of packet loss rates in a path. Now, we argue that even without this assumption, we can still translate the packet loss rate into some other quantity and use this additive property again. For every edge with packet loss rate $PL$, define the packet transfer rate to be $1\text{-}PL$, and the packet transfer rate of a path to be the product of all packet transfer rates of the edges along the path. Since each packet transfer rate is at most one, it can be rewritten as a non-positive power of 2 (or e=2.716). Higher packet transfer rate can be achieved by pushing up the sum of all the powers along the path. This can be done by lowering the sum of the absolute values of the powers. The argument is complete. In short, for packet loss rate $PL$ of an edge, we translate it into $|\log(1\text{-}PL)|$. For the simplicity of discussion, we leave this transformation to the engineers, and consider the sum of the packet loss rates along a path as a parameter for optimization.

## 3. Constructing Composite Functions

We divide this section into three subsections: Basic composite functions, a new composite function, and practicing the composite functions without constraints.

### 3.1. Basic Composite Functions

We now study the cost function of a routing path. If either the delay or packet loss rate is considered, we can directly apply Dijkstra's algorithm for finding the shortest path. However, mathematically, it is impossible to achieve optimality for two metrics simultaneously representing the delay and packet loss rate of each edge. Figure 1 is an example, which shows different paths from the source to destination based on different metrics. In other words, we can only find trade-offs between them.

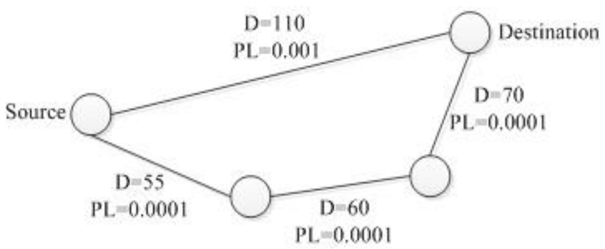


**Figure 1.** The edges with two metrics between the source and destination nodes

A composite function of two metrics, namely $x$ and $y$, can be defined as either an ordered pair $(x,y)$ or a single value of a function of $x$ and $y$. There are different advantages for each of these approaches. For using $(x,y)$, each metric will be considered symmetrically, and the main difficulty of the research work will focus on the relationship between dimensionality and path optimality. Such work is rather algorithmic than practical, and extending the algorithm for more metrics is by no mean trivial. In this paper, we use a single-value composite function, and therefore, the path optimality can be simply done by many existing algorithms, while the research will emphasize on how to construct the function with two metrics of different characteristics. Another advantage of using one single value is the feasibility of allowing more metrics for future research. The goodness of a composite function is measured by the ratio of the resulting cost to the optimal cost, for each metric.

For the construction of a composite function, there are more to consider. First, when one metric is zero (or weighted zero), the format of the composite function must be linear and homogeneous to the cost of a routing path based on the other metric. Second, the weighting of the two metrics can be adjusted. There are two basic formats for the composite function (*CF*):

$$CF = C_D D + C_{PL} PL \qquad (1)$$

$$CF = C_T D * PL + C_D D + C_{PL} PL, \qquad (2)$$

where $C_T$, $C_D$, and $C_{PL}$ are constants and designed by system designers. The notation $CF(p)$ is used for the sum of all $CF$ values of the edges along $p$. In using either equation, we need to minimize $CF(p)$. From the point of constructing the function, we need to design two constants, $C_D/C_T$ and $C_{PL}/C_T$, in Eqn (2), if $C_T \neq 0$; however, only the fraction $C_{PL}/C_D$ is needed in Eqn (1), if $C_D \neq 0$. Therefore, using Eqn (1) is more convenient. For simplicity, we shall refer CF in Eqn (1) as CF1 and that in Eqn (2) as CF2.

A similar function of CF1 was proposed by Jaffe [16] to meet different constraints in MCPP. According to his approximation, the minimization of that function may also give an optimal path that does not satisfy the constraints. CF1 differs in the bound of $D$, which is not in the range [0, 1].

### 3.2. A New Composite Function

We propose a new composite function with considering the Assumptions 1 and 2. With the assignment of the initial values to the delay and packet loss rate of each edge on a sample network, we introduce our composite function, which covers both metrics together. The aggregation of the two metrics from distinct ranges is the bottleneck in the literature. To solve this problem, a common range for the metrics is figured out to get a composite function as the edge cost function for the routing process. This common range is obtained by using ranking on the original metrics. However, it is still not a universal technique for data analysis due to its uncertain performance.

Ranking has been used for removing noises on the original data, and has been verified successfully in many cases. However, it can also remove some important features of a parameter. To the extreme, we cannot judge the goodness of a ranking function; however, we believe that, in many cases, we can apply some statistical techniques to construct a better composite function.

Suppose that we are given sets of data for different parameters, and a good basic composite function $B$ which uses original data. We rank the data and study the relationship between the original values and their ranks. Since the ranking is a strict increasing bijective function of the original data, the inverse function can also be used for describing the relationship. We approximate this inverse function by some simple functions like polynomial and exponential functions. This approximate function is used for substituting the original parameter in function $B$. We do this for all parameters, and the resulting function is then rank-based.

Now, the burden falls on experimental work for finding the approximate inverse function.

Based on the above explanation of the ranking process, the new composite function basically covers the rank, which will be symbolized as $R$, for each metric value. In this subsection, we focus on delay and packet loss rate. Let $D_e$ and $PL_e$ be the values of delay and packet loss rate, and they have ranks $R_{D_e}$ and $R_{PL_e}$, respectively, for edge *e*. In this method we consider same ranks for the same values in each order. For example, if the delay sequence is 5, 10, 15, 15, 47 for five edges, then the delay ranks become 1, 2, 3, 3, 4, respectively. Mathematically, for each metric, our ranking is a bijective function *Rank( )* from the set of values of all edges to an integer set *{1, 2, ..., L}*, for some positive *L* (where *L* is the number of different values in the domain), and $Rank(D_e) < Rank(D_{e'})$ if and only if $D_e < D_{e'}$. After this ranking step, we do not need any other optimization technique for the metric ratings between the edges.

After the computations of the ranks, we construct the composite function of two metrics from distinct ranges as

$$CF_e = Coef * R_{D_e} + Base^{R_{PL_e}} \quad (3)$$

where $CF_e$ is the numerical value of the composite function for *e*. For simplicity, we refer the function in Eqn (3) as CF3. According to the values of the packet loss rates which we use in the domain, each increment is a factor of 10. Ranking can then be considered as a base-10 logarithmic function of the original values. As optimality is measured by the sum of the original values along the routing paths, we use an exponential function of the ranks in the composite function, instead of the ranks themselves. For this reason, the *Base* value in CF3 is selected as 10 according to this relation between the packet loss rate and ranking. Experiments show that with this modification of ranking technique, we can adjust the weighting of delay and packet loss rate efficiently in order to seek for a balance of their performance. Note that *Coef* in CF3 is selected after some practical results, which will be mentioned in Section 3.3.

CF3 gives the cost function of *e* to be used in the selection of the optimal path over a network by Dijkstra's algorithm. The cost function of a path *p* with |*E*| edges can be calculated as

$$CF(p) = \sum_{e=1}^{|E|} CF_e \quad (4)$$

### 3.3. Practicing the Composite Functions

In this subsection, first we get CF1 and CF2 as the cost functions of an edge and then use Dijkstra's algorithm to observe the best paths according to these functions. We refer these methods as Dijkstra(CF1) and Dijkstra(CF2) respectively. After finishing this step, we skip to CF3 practices and select the best *Coef* value to complete CF3 by referring this method as Dijkstra(CF3). Then we extend CF3 to get another function.

In this paper, we computed all experimental results over the network represented in Figure 2. This network is a sample of mesh network and appropriate for the selections of the paths with the specified hop numbers in the comparisons. It covers 75 nodes and 100 full-duplex edges.

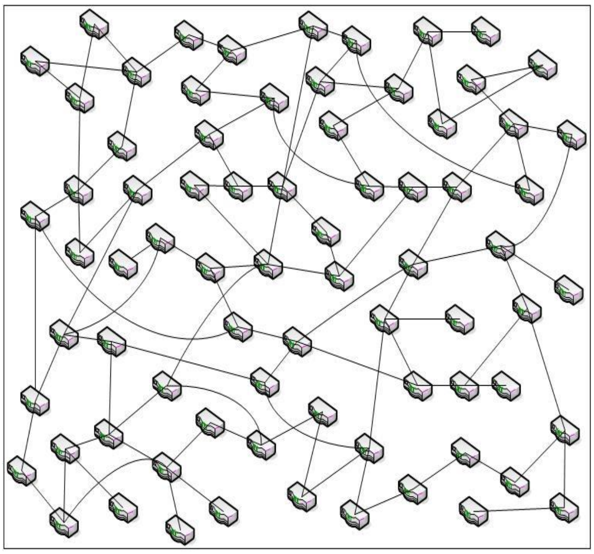

**Figure 2.** The sample network

Each edge *e* in the network was provided with numerical values of delay and packet loss rate. Delay values are between [1, 150] and the packet loss rates should be one of the values 0.000001, 0.00001, 0.0001, and 0.001. We assign the metric values of the whole network as behaving towards the normal distribution.

We selected five source-destination node pairs for each hop number *h*, where *h* =2, 4, 6, 8, 10. *h* represents the minimum edge number obtained manually along a path, which starts from the predetermined source node and ends in the destination. All functions try to find several paths between two end nodes without considering their edge numbers. We executed the application 400 times for each node pair of each hop number. So we used 400 different metric validations of the edges in the network. We computed the average values of delay and packet loss rate of the 2000 different best paths for each hop number.

For fine-tuning the performance, for convenience, we start with $C_{PL} / C_D = 1$ in Eqn (1). If the worst case ratio of delay is greater than that of packet loss rate, we decrease $C_{PL} / C_D$; otherwise, increase it. The worst case ratio of a parameter is referred to the maximum (over all values of *h*) experimental performance ratio averaged over all experiments. The same experiment is also done for the terms in Eqn (2). To speed up, binary search can be used. The binary search can stop if the worst case ratios are less than a threshold, say 5%, apart from each other. We finally chose $C_{PL} / C_D = 185000$ for Dijkstra(CF1), $C_D / C_T = 1/10$ and $C_{PL} / C_T = 185000$ for Dijkstra(CF2) to get the computational results in this subsection.

The worst case ratios of delay and packet loss rate for Dijkstra(CF1) and Dijkstra(CF2) obtained for each hop number are stated as in Table 1. The worst case ratios occur around hop number 6 for both of the delay and packet loss rate values. Both of the functions are in balance according to two metrics.

**Table 1.** Worst case ratios

| Hop Number | Delay | | Packet Loss Rate | |
|---|---|---|---|---|
| | CF1 | CF2 | CF1 | CF2 |
| 2 | 1,101004 | 1,101004 | 1,628099 | 1,628099 |
| 4 | 1,103260 | 1,103289 | 1,600887 | 1,600887 |
| **6** | **1,150929** | **1,151819** | **1,731111** | **1,728889** |
| 8 | 1,136208 | 1,136468 | 1,520000 | 1,520000 |
| 10 | 1,145566 | 1,145517 | 1,713450 | 1,713450 |

We now skip to find the *Coef* value in CF3. Through the use of Eqn (4) as the path cost function in Dijkstra(CF3), the correlated CF3 appears concretely after implementing different *Coef* values and choosing the most effective one, which concludes the paths with the best aggregation of delays and packet loss rates. Figure 3 and Figure 4 show the last delays and packet loss rates provided by using different *Coef*s in CF3 with the constant *Base* value of 10. We practiced several *Coef*s between 100 and 200 in Dijkstra(CF3) and determined that the best values are around 120 and 125 according to the both of the metrics. The metric values measured with the *Coef*s between 120 and 125 are represented in Table 2 for Hop Number=6**.** We chose this hop number because that the main difference between the *Coef*s can be seen clearly at that point. It can be easily extracted from Table 2 that *Coef=120* is the best selection for CF3.

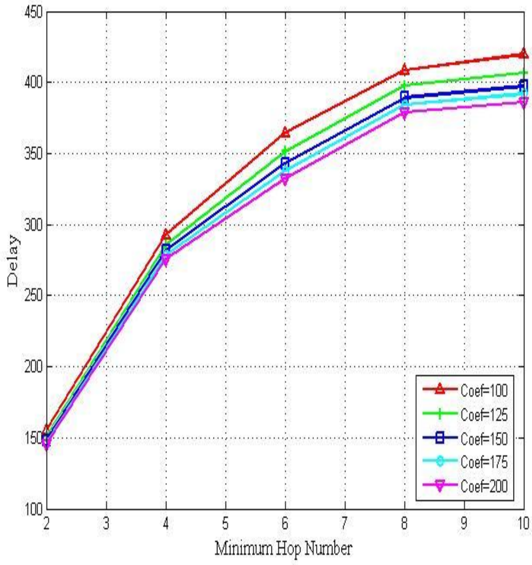


**Figure 3.** Delays for different Coefs in CF3

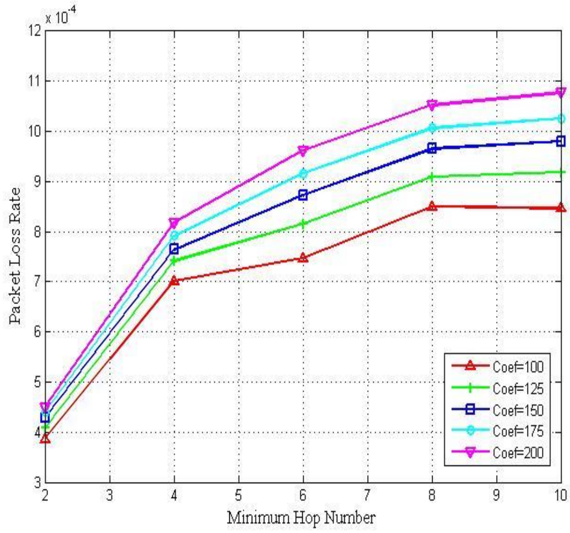


**Figure 4.** Packet loss rates for different Coefs in CF3

**Table 2.** The metric values with *Coef* between [120, 125]

| Coef | Delay | Packet Loss Rate |
|---|---|---|
| 120 | 353,5145 | 0,000805 |
| 121 | 353,3715 | 0,000806 |
| 122 | 353,1195 | 0,000807 |
| 123 | 352,2650 | 0,000811 |
| 124 | 352,0160 | 0,000813 |
| 125 | 351,5465 | 0,000816 |

Consequently, we converted CF3 for each *e* with the selected *Coef* and *Base* values into

$$CF3 = 120 * R_{D_e} + 10^{R_{PL_e}} \quad (5)$$

We now change only $R_{PL_e}$ computation to find another function. We match the packet loss rate values of the set {0.000001, 0.00001, 0.0001, 0.001} to an integer set {$2^0$, $2^1$, $2^2$, $2^3$} sequentially. We used the same *Base* 10 as in Eqn (5) and obtained

$$CF4 = Coef * R_{D_e} + 10^{R_{PL_e}} \quad (6)$$

where CF4 is a new cost function of an edge to be used in Dijkstra's algorithm as Dijkstra(CF4). We practiced several *Coef*s between $10^4$ and $10^8$ in Dijkstra(CF4) and determined that the best value is $10^6$ according to the both of the metrics. After this result, we converted Eqn (6) into

$$CF4 = 10^6 * R_{D_e} + 10^{R_{PL_e}} \quad (7)$$

For any CFx (x=1,2,3 or 4), we use the pseudocode of Dijkstra(CFx) as illustrated in Figure 5.

```
MAIN FUNCTION Dijkstra(CFx)
for hop_number=2:2:10
        for SourceDest_pair=1:5
                for r=1:runs %runs=400
                        Get both metric validations of all edges from the source file.
                        Compute R_{D_e} and R_{PL_e} for each edge.
                        Compute CF_e value of each edge.
                        Apply Dijkstra's Algorithm based on the normalized CF_e values to find the shortest path
                        over the network.
                end
        end
        Calculate average delay values of the shortest paths with the current hop_number.
        Calculate average packet loss rates of the shortest paths with the current hop_number.
end
```

**Figure 5.** Pseudocode of Dijkstra(CFx)

A comparison between Dijkstra(CF1), Dijkstra(CF2), Dijkstra(CF3), and Dijkstra(CF4) is given in Figure 6 and Figure 7 against the related values of each single metric version as Dijkstra(D) or Dijkstra(PL). Dijkstra(CF1) and Dijkstra(CF2) overlap in Figure 6 and Figure 7. The results of Dijkstra(CF1), Dijkstra(CF2), and Dijkstra(CF3) are close to each other. Dijkstra(CF4) improves the PL results. This is very important for any network, especially for multimedia networks.

The worst case ratios of the delay values for CF3 and CF4 are both 1.1x. On the other hand, the worst case ratios of the packet loss rates are 1.7x and 1.6x respectively. It tells that CF4 is better than CF1, CF2, and CF3 according to the balance of two independent parameters.

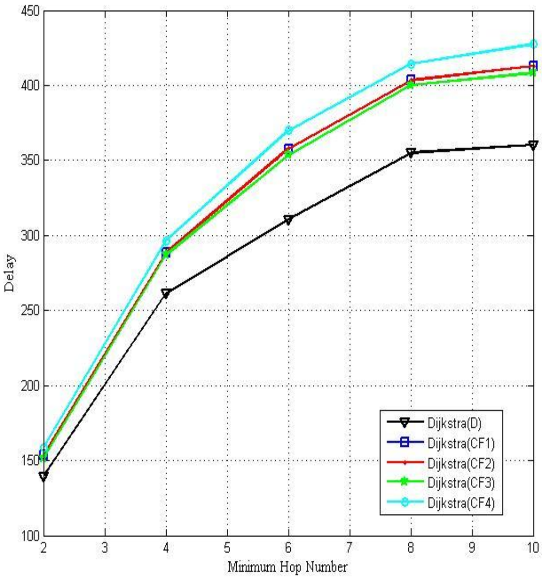


**Figure 6.** Delays for all functions

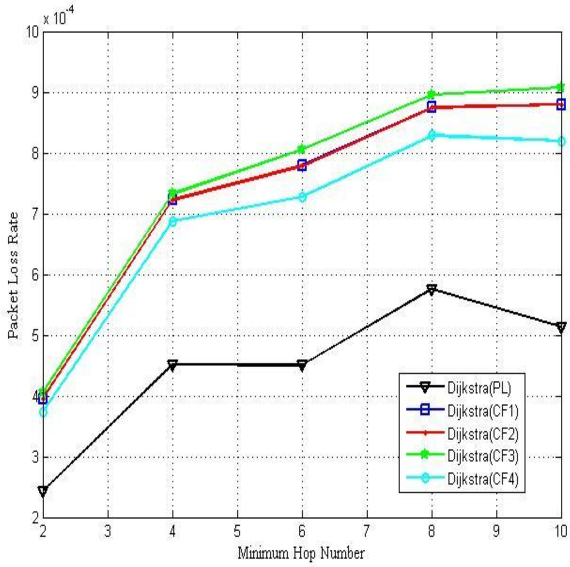


**Figure 7.** Packet loss rates for all functions

## 4. Conclusion

In this study, we proposed some effective composite cost functions correlating two different metrics, namely delay and packet loss rate, and represented the way and rationale of their construction. We obtained the composite functions to find the best path according to the metrics on any arbitrary networks without any constraints. We compared the composite functions against Dijkstra's algorithm with the individual metrics. The numerical results show that our composite functions are good when considering the balance between both metrics. The worst case ratios between two metrics in our functions are well balanced. The functions can also be extended to include more than two metrics. They may also cover bandwidth or jitter with supporting additional operations such as exponentials used in several studies related with network traffic distributions.

## 5. Acknowledgments

The author would like to sincerely thank Savio S.H. Tse for sharing his comments during the preparation of this study.

This work was supported by Scientific Research Projects Coordination Unit of Istanbul University. Project number: 10590.

## 6. References

[1] Hanzo II, L. and Tafazolli, R., "A Survey of QoS Routing Solutions for Mobile Ad Hoc Networks". *IEEE Communications Surveys & Tutorials,* Vol: 9, No: 2, pp. 50-70, 2007.

[2] Yin, C.S., Chaw, L.T., and Yaacob, M., "Hop-by-Hop QoS Routing Using Statistical Distribution-Free Approach". *Malaysian Journal of Computer Science,* Vol: 18 No: 2, pp. 28-37, December 2005.

[3] Brunsting, R.J., "Quality of Service Issues in Wireless Networks", in *An essay presented to the University of Waterloo,* Ontario, Canada, 1995.

[4] Obreja, Ş.G. and Borcoci, E., "Interdomain QoS paths finding based on overlay topology and QoS negotiation approach". *Trends in Telecommunications Technologies,* InTech Publisher, pp. 345-365, 2010.

[5] Iqbal, M., Wang, X., Li, S., and Ellis, T., "QoS scheme for multimedia multicast communications over wireless mesh networks". *IET Communications,* Vol: 4, No: 11, pp. 1312–1324, 2010.

[6] Djarallah, N.B., Sauze, N.L., Pouyllau, H., Lahoud, S., and Cousin, B., "Distributed E2E QoS-Based Path Computation Algorithm Over Multiple Inter-Domain Route", *Int. Conf. on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC),* Barcelona, pp. 169-176, 2011.

[7] Ali, N.B., Molnár, M., and Belghith, A., "Multi-constrained QoS Multicast Routing Optimization", *inria-00271795, version 2 – 10,* 2008.

[8] Yampolskiy, M., Hommel, W., Lichtinger, B., Fritz, W., and Hamm, M.K., "Multi-Domain End-to-End (E2E) Routing with multiple QoS Parameters-Considering Real World User Requirements and Service Provider Constraints", *Second International Conference on Evolving Internet (INTERNET),* Valcencia, pp. 9-18, 2010.

[9] Bertrand, G., Lahoud, S., Texier, G., and Molnár, M., "Computation of Multi-Constrained Paths in Multi-Domain MPLS-TE Networks", *Next Generation Internet Networks,* Aveiro, pp.1-8, 2009.

[10] Masip-Bruin, X., Yannuzzi, M., Serral-Gràcia, R., Domingo-Pascual, J., Enríquez-Gabeiras, J., Callejo, M.Á., Diaz, M., Racaru, F., Stea, G., Mingozzi, E., Beben, A., Burakowski, W., Monteiro, E., and Cordeiro, L., "The EuQoS System: A Solution for QoS Routing in Heterogeneous Networks". *IEEE Communications Magazine,* pp. 96-103, 2007.

[11] Shin, D.-w., Chong, E.K.P., and Siegel, H.J., "Multi-postpath-based lookahead multiconstraint QoS Routing". *Journal of the Franklin Institute,* Vol: 349, No: 3, pp. 1106-1124, 2012.

[12] Xue, Q. and Ganz, A., "Ad hoc QoS on-demand routing (AQOR) in mobile ad hoc networks". *Journal of Parallel and Distributed Computing,* Vol: 63, pp. 154-165, 2003.

[13] Yuan, X. and Liu, X., "Heuristic Algorithms for Multi-Constrained Quality of Service Routing", *IEEE INFOCOM,* Vol: 2, pp. 844-853, 2001.

[14] Yiltas, D. and Perros, H., "Quality of service-based multi-domain routing under multiple quality of service metrics". *IET Communications,* Vol: 5, No: 3, pp. 327-336, 2011.

[15] Van Mieghem, P., De Neve, H. and Kuipers, F., "Hop-by-hop quality of service routing". *Computer Networks,* Vol: 37, pp. 407-423, 2001.

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

### Biography

**Derya Yiltas-Kaplan** received the B.Sc., M.Sc. and Ph.D. degrees in computer engineering from Istanbul University, Istanbul, Turkey, in 2001, 2003 and 2007 respectively. She studied as a post-doctorate researcher at North Carolina State University in USA between 2008 and 2009 for one year. She was granted with scholarships by The Scientific and Technological Research Council of Turkey during her Ph.D. and post-doctorate studies. Now, she is an Assistant Professor in Department of Computer Engineering, at Istanbul University. Her research interests include wireless communications and networking.