# Mathematica Balkanica

Mathematical Society of South-Eastern Europe
A quarterly published by
the Bulgarian Academy of Sciences – National Committee for Mathematics

---

Mathematica
Balkanica

New Series Vol.11, 1997, Fasc. 3-4

# One Novel Strategy for Link-State Shortest-Path Routing Algorithms

*Iskra Djonova-Popova*

*Presented by Bl. Sendov*

One novel strategy for the link-state shortest-path routing algorithms is proposed. It combines the features of the minimum-hop and the shortest-path strategies and is based on calculation of minimum-hop flow-augmenting path for each destination. This path contains minimum number of links while avoiding those that are congested. The strategy introduces a unique property of being able to trigger the congestion control scheme when necessary. Its performance was investigated upon the flow model of a sample network. The results obtained are encouraging and favor the proposed strategy in terms of lower resource usage, higher reliability, smaller processing time and possibility for tradeoff between delay performance and throughput.

## 0. Introduction

The function of a routing algorithm in a wide area network is to select the best set of available links to transfer packets between a source and destination. The objective is to minimize the average delay at lower network loads and keep it reasonable at high throughput and at the same time to make the efficient use of the expensive long-distance links. It is very difficult to achieve all of these requirements at the same time, particularly in situations, where changes in traffic statistics are difficult to predict.

Various dynamic routing techniques have been conceived to provide different levels of adaptation to network traffic, as well as to efficient link usage. Their primary goal is to determine the path for every source-destination pair. The routes assigned must be adaptive to survive network failures and to accommodate distributions of traffic that differ from those for which the network was designed. Moreover, effective and fair usage of network resources should be provided. To achieve these prerogatives, all adaptive routing techniques process the information about the network state in periodical time intervals and recalculate the paths. The frequency with which a route is modified can vary. A new route can be selected when the network topology changes, for a source-destination session, for all the packets in the message or for each packet. The choice of frequency for the routes updates is mainly dependent on the statistical parameters of the network traffic.

## 1. Shortest-path algorithms

A number of dynamic routing algorithms is known as a class of shortest-path routing algorithms. Naturally, these algorithms select a single shortest-path for every source-destination pair, i. e. the path with the minimum "length". The "length" of the path is computed as the sum of the "lengths" of the links comprising that path. In such a case, the routing problem is equivalent to that of finding the shortest path through a graph. The metrics chosen or the "lengths" for the links differ from algorithm to algorithm. The most common metrics used are:

1. the "length" of each link is considered to be 1, which means that the minimum-hop paths are actually the shortest paths, [1];
2. the "length" of each link is configurable by the network administrator, [2];
3. the "length" is equal to the average delay experienced in the measuring interval, [3].

The advantages of the first strategy that uses the hop-count metrics are:

- minimum network resources are used,
- the reliability of the transmission is higher,
- the traffic that informs every node about the network topology is low since it is broadcasted only in case of topology changes,
- the processing time for obtaining minimum-hop tree is smaller.

The main disadvantage is the inability to spread the traffic over the other links of the network when some of the links contained in the minimum-hop path are overloaded. In such cases loss of packets might be experienced unless additional congestion control scheme is implemented. This is especially evident in cases when the high traffic in the network is generated from a small number of source-destination pairs.

The other two strategies try to avoid this disadvantage and to adapt the routing paths to any situation that might appear. They have variable link metrics that require additional traffic to be propagated through the network quite frequently. The purpose of these updates is to inform all the nodes about the link states in terms of the metric chosen by the network administrator (in case of algorithm 2) or the delay measured upon the link (in case of algorithm 3). The more frequent these updates are, the more additional update traffic is conveyed through the network. At the same time the additional calculation of shortest-path trees is needed in the nodes that obviously requires significant processing time.

However, the better performance is obtained in terms of keeping delay at reasonable level at higher throughput, even when the source-destination pairs are unevenly spread throughout the network. These kinds of strategies provide avoidance of congestion to a certain level of input traffic, yet need additional technique that will find congested ink and throttle the sources that cause the excess traffic. In addition to the higher processing time and the large quantity of additional update traffic, another disadvantage of these algorithms are the calculation of source-destination paths with unnecessarily large number of hops. The appearance of such paths is particularly frequent in presence of lower network loads.

The set of shortest paths from a source to all the destinations forms a tree with the source as root node. The two algorithms most commonly used in communication networks shortest- path calculation are Dijkstra's algorithm, [4] and Bellman-Ford algorithm, [5]. The former is usually associated with the link-state algorithms (also called topology-broadcast algorithms), and the latter with the distance-vector algorithms. In a link-state algorithm, every node is informed about the entire network topology, as well as the state of each link represented by its "length". To reduce the update traffic every node broadcasts the information about its adjacent links only. In a distance-vector algorithm, the "length" from the current node to all another nodes is updated based on the information received from the adjacent nodes and the state of the adjacent links.

## 3. One novel strategy

It seems natural to try to combine the strengths of the two kinds of shortest-path strategies: the minimum-hop strategy and the shortest-path strategy where the links have variable "length". The idea behind the new strategy is to keep the minimum-hop path whenever the state of the links in it satisfy a certain level of "goodness". When some of the links in the network are not "good", the routing algorithm should try to find the minimum-hop path that avoid the links that are not "good". In the text the shortest-path tree obtained in this way will be called a minimum-hop flow-augmenting tree, since all the paths in it are actually the minimum-hop paths that allow additional flow of traffic, [6]. At higher network loads, situations where it is not possible to construct the flow-augmenting tree without including some of the links that are not "good" might appear. Then the links that are not "good" have to be included in the tree. They are a potential danger to become bottlenecks and to cause unpleasant congestion effects. That is why the inclusion of such links in the tree should always trigger the congestion control scheme. Hence, this strategy has a unique property that none of the above mentioned strategies has, and that is the property to use the parameters that represent the state of the links twice: first for purpose of routing, and second for activating the congestion control scheme.

The term of a "good" link, as used in the previous text as a broad term. The parameters that characterize the links can be defined in different ways and should be dependent on the requirements of the network. The following criteria consider that a link is "good" as long as:

- the average delay upon the link does not exceed a certain limit,
- the cost assigned to the link by the network administrator is below some upper limit,
- the percent of occupied buffers for the link is less than some specified value,
- the utilization of the link is limited with some upper bound.

The choice of the parameter that characterize the state of the links can be made variable and dependent on the certain requirements for the network performance. For example, if the low average delay is the most important, then the upper limit of the delay over the link should be small. If higher throughput is of primary concern then the bound on the link utilization should be at its maximum possible value.

The detailed description of the novel strategy is given through the presentation of the $\alpha$-routing algorithm that was developed upon the flow model of a network.

### 3.1. The $\alpha$-routing algorithm

The algorithm belongs to the class of link-state algorithms. It is distributive in nature and constructs the routing tables according to global knowledge about the network state. The link states are updated in regular measuring intervals, but they are broadcasted through the network only when there is a change in a link state. Each node broadcasts the information about the state of its outgoing links. That way the update traffic in each measuring interval is kept small. The update of databases in all the nodes provides avoidance of loops. As with other link-state algorithms, loops may appear only during the short updating interval. Consider the graph $G(NFL)$ with $n$ nodes and l links that symbolize the network. Furthermore, let the average traffic of the link $(i, j)$ in some measuring interval be represented as a flow on that link. If the capacity of the link, $C_{ij}$, and the flow, $f_{ij}$, are expressed in the same units, than we define the residual capacity as:

$$C_{ij}^{r} = C_{ij} - f_{ij}. \tag{1}$$

The residual capacity is representative of the additional flow that can be pushed through the link before the congestion appears. It is assumed that when it approaches 0, the probability of the link getting congested is very high. In order to avoid this critical point when the congestion will be certainly manifested, we proclaim that additional flow is admissible or the link is "good" when the following condition holds:

$$C_{ij}^{r} > \alpha C_{ij}, \tag{2}$$

where $\alpha$ is a parameter that can take values between 0 and 1, $0 < \alpha < 1$. If the condition (2) is satisfied, the link is considered under loaded or "good". Otherwise it should be avoided in the flow augmenting tree calculations. If a link that does not satisfy this condition has to be included in the tree, it is considered as a bottleneck.

The parameter $\alpha$ provides the flexibility in the traffic management. The situation when $\alpha$ is close to 0 provides maximal utilization of the links, but promotes a high risk of congestion appearance. Hence, with this value of a it is reasonable to expect large average delay in the network. For greater values of $\alpha$ the throughput might be smaller, but the delays are also kept small.

Once all the nodes in the network have the identical topological data base that contains the states of all the links, each of them performs the bread first search on a first label, first scanned (FLFS) principle to obtain the minimum-hop flow-augmenting tree with itself as a root, [6]. The fact that the data base is identical in each of the nodes, and the search is made on the first labeled first scanned principle, provides consistency in routing tables at different nodes and avoids the possibility of loops to appear. It is assumed that at the beginning, when there is no traffic in the network, the routing table based on the minimum-hop metrics exists at each of the nodes.

For calculation of the minimum-hop flow-augmenting tree, the algorithm considers every node $j$ to be in one of the three states:

- unlabelled, which means that there is no label associated with node $j$,
- labelled, but not scanned, which means that label $(i, k)$ is associated with node $j$. The first term in the label denotes the previous node in the tree, and the second denotes the order in which the node $j$ was labeled,

- scanned, which means that all nodes $i$ adjacent to $j$ for which condition (2) holds have been labeled.

The steps of the $\alpha$-algorithm are:

**Step 1:** Initialization

Label the root $(0, k)$, $k = 1$.

At this point the root is labeled, but not scanned and all the other nodes are unlabelled.

**Step 2:** Put all labeled nodes on tree in order in which they were labeled.

If all nodes are on the tree, go to Step 5.

**Step 3:** If there are nodes that are labeled, but not scanned, go to Step 4. Otherwise, from the previous rooting table find the next node $j$ for the last node put on tree $i$. Label it by $(i, k)$, $k \mapsto \max(k) + 1$ and consider the link $(i, j)$ as a bottleneck.

**Step 4:** Find the node $i$ that is labeled, but not scanned with the maximum value of $k$-term in the label.

For each link $(i, j)$ that is "good", i.e. $C^r_{ij} > \alpha C_{ij}$, label $j$ with $(i, k)$, $k \mapsto max(k) + 1$.

At this point the node $i$ is considered scanned. Go to Step 2.

**Step 5:** If there are bottlenecks included in the tree activate the congestion control scheme.

Exit.

The algorithm presented, is actually an extension of the Edmonds-Karp's algorithm, [6] for obtaining minimum-hop flow augmenting path between a source and a single destination.

**A simple example:** Figure 1a represents a graph of a simple network with 6 nodes and 9 links. The capacities of all the links are equal, $C = 5$. For the node 1 as a root, the minimum-hop path tree is presented on Figure 1b, the shortest-path tree with the flow as a metrics on Figure 1c, and the minimum-hop flow-augmenting tree for $\alpha = 0.2$ on Figure 1d.

The numbers besides the links on Figure 1a and Figure 1c represent the flow on the links. On Figure 1d they are representatives of the residual capacities of the links. When $\alpha = 0.2$, only the links with the residual capacities that are bigger than 1 can be included in the minimum-hop tree flow-augmenting tree. The minimum-hop tree as well as minimum-hop flow augmenting tree are obtained using FLFS principle.

## 4. Performance comparison

The strategy proposed here was developed with intention to combine the good features from the minimum-hop and shortest-path strategies. The minimum-hop flow-augmenting path is a path with the minimum number of network resources used, yet it avoids the over loaded parts of the network. At the point at which a congested link is included in the tree, the congestion control mechanism is to be activated. Hence, the unique feature of the flow-augmenting path strategy is that it is tightly connected with the congestion control scheme. The point at which the congestion control mechanism is triggered is variable and depends on the choice of the parameter $\alpha$ used.

To get better insight into the performance of the $\alpha$-algorithm in terms of the average delay, the three routing strategies without any congestion control scheme implemented, were examined upon the flow model of a sample network. An analytical simulation method and the 19-node network from [8], were used. The network was subjected to two different matrices with 32 and 98 one-way traffic sessions, respectively, [8]. The M/D/1 statistics for each queue throughout the network was chosen.

The total delay function for each link $(i,j)$, $D_{ij}(f_{ij})$, is defined as the expected delay/packet times the expected number of packets/second and for the M/D/1 statistics it is expressed as:

$$D_{ij}(f_{ij}) = \frac{f_{ij}(2C_{ij} - f_{ij})}{2C_{ij}(C_{ij} - f_{ij})}. \tag{3}$$

The expected end-to-end delay per packet is determined as $T = D_T/\gamma$, where $D_T = \sum_{i,j} D_{ij}(f_{ij})$ and $\gamma$ is the expected number of packets/second entering the network. The network examined is presented on Figure 2. The capacities of all the links are equal, $C_{ij} = C$, $j = 1, 2, \ldots, 19$.

Figure 3a and 3b shows the curves obtained for the minimum-hop, shortest-path and minimum-hop flow-augmenting strategy with different values for the parameter $\alpha$. The influence of the parameter on the performance of the algorithm was as expected. The higher values of $\alpha$ provide better delay performance, while the lower values move the knee of the curves on the left and provide reasonable delays for higher throughput. The knees of the curves are the points at which the congestion control should be activated, thus protecting the network from the excess input traffic.

## 5. Conclusion

One novel shortest-path routing strategy was presented in the paper. The strategy suggest a routing algorithm which provides paths with minimum number of links, but avoids the links that do not obey to a certain condition. The strategy was described through the $\alpha$-algorithm. The parameter $\alpha$ was introduced to offer a possibility for tradeoff between the delay and the throughput. The results presented on Figure 3 show that the algorithm is sub optimal in terms of the average delay for high network loads. However, it has other advantages:

- saves network resources,
- increases the reliability of the transmission,
- triggers the congestion control scheme,
- offers a possibility for tradeoff between the delay and the throughput,
- requires lower update traffic.

The $\alpha$-algorithm proposed here is only a theoretical version of the implementation of the novel strategy. For practical implementation, specification of terms such as flow and residual capacity are required, which is the subject of our further research.

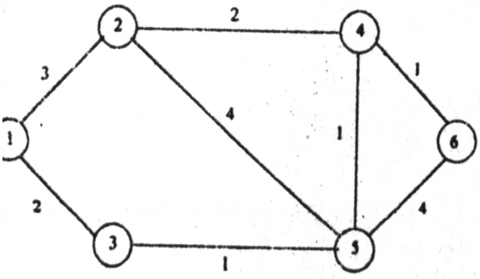

Figure 1a Example network

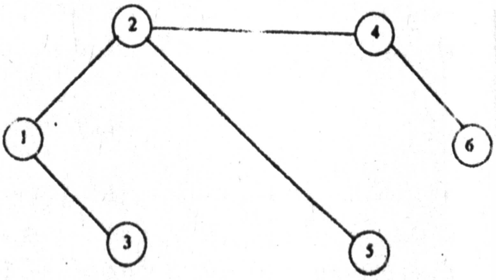

Figure 1b Minimum-hop tree

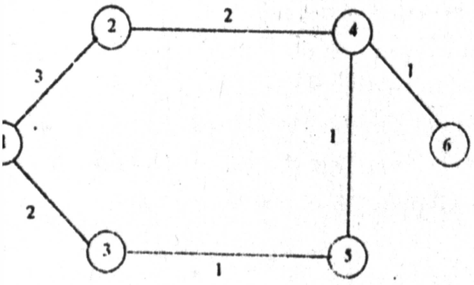

Figure 1c Shortest-path tree

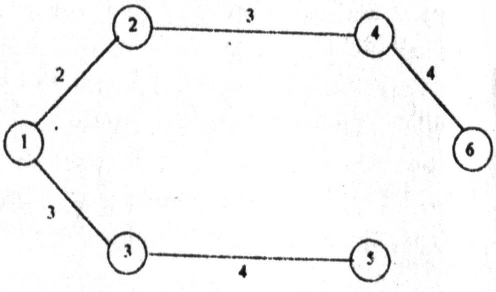

Figure 1d Minimum-hop flow-augmenting tree

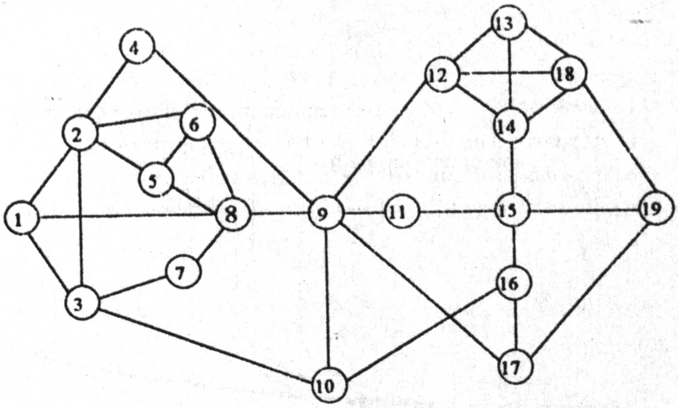

Figure 2 Topology of the examined network

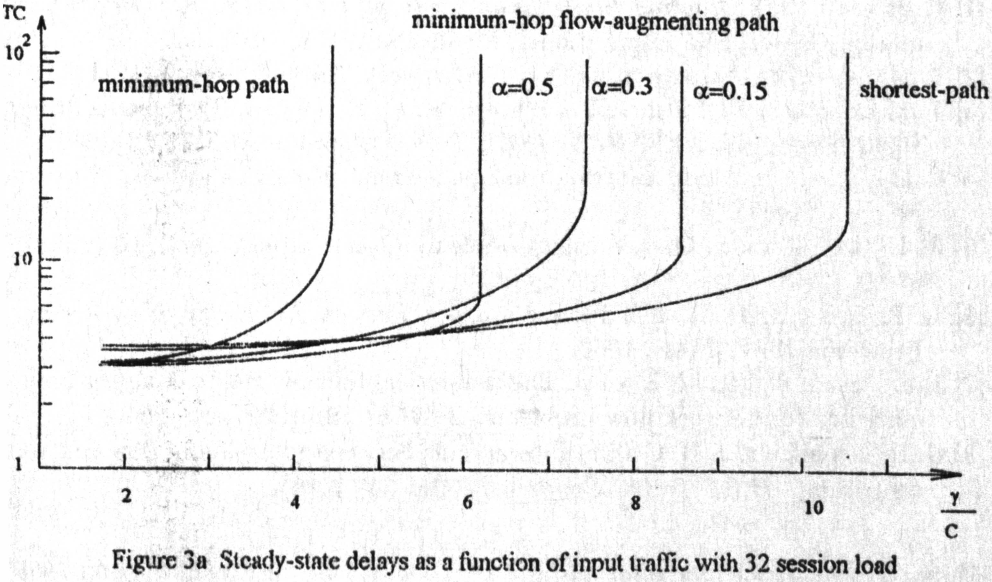

Figure 3a Steady-state delays as a function of input traffic with 32 session load

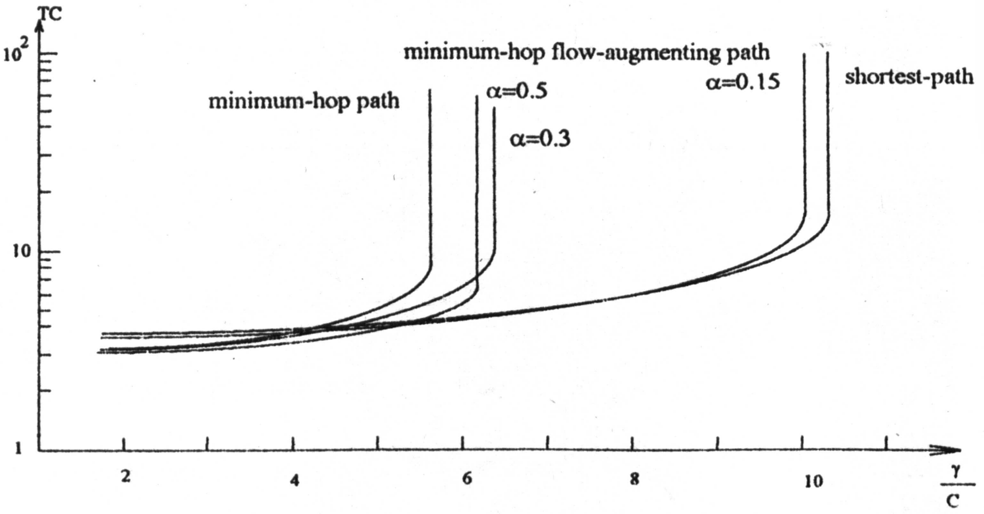

Figure 3b Steady-state delays as a function of the input traffic with 98-session load

## References

[1] C. Hedrick, *Routing Information Protocol*, RFC 1058. Network Information Center, SRI International, Menlo Park, CA, 1988.

[2] J. Moy, *OSPF Version 2*, RFC 1247. Network Working Group, 1991.

[3] J. M. McQuillan, I. Richer, E. C. Rosen, The new routing algorithm for the ARPANET. *IEEE Trans. Commun.* COM-28 (1980).

[4] E. Dijkstra, A note on two problems in connection with graphs. *Numer. Math.* **1** (1959).

[5] R. E. Bellman, On a routing problem. *Quart. Appl. Math.* **16** (1958), 87-90.

[6] L. R. Ford, D. R. Fulkerson, *Flows in Networks*. New Jersey, Princeton Univ. Press, 1962.

[7] J. Edmonds, R. M. Karp, The theoretical improvements in algorithmic efficiency for network flow problems. *J. ACM* **19** (1972), 248-264.

[8] G. H. Thaker, J. B. Cain, Interactions between routing and flow control algorithms. *IEEE Trans. Commun.* COM-34 (1986).

*Center of Energetics and Informatics*
*Macedonian Academy of Sciences and Arts*
*Krste Misirkov 2, Skopje,*
*Republic of MACEDONIA*

*Received 01.02.1995*