

Routing Algorithm in Networks on the Globe

Sujit Kumar Bose

S.N. Bose National Centre for Basic Sciences, Kolkata, West Bengal, India

E-mail: sujitkbose1@gmail.com

Technical paper

Keywords: router network algorithm, geodesic, link cost, waiting time in queueing model

Received: June 16, 2020

Packet switching of data in networks is done by either the distance-vector or the link-state routing protocols. These protocols use the Bellman-Ford and the Dijkstra's algorithms respectively for the least cost path from a source base station to a destination station. For inter-network transmission, the path-vector routing protocol is in use. With progress of time, the network topologies are becoming huge in size, requiring large demand on book keeping of routing tables and transmission of the data packets dynamically to several other stations of the network by broadcast, increasing the load on the network. Here, assuming the router stations to be terrestrially located with links along the ground, a large network is assumed to lie on a spherical surface, and so the shortest geodesic path from source to destination becomes a great circular arc. For fast transmission, the cost of a link to a node is multicast to its neighboring nodes only for selection of the path lying as close as possible to the geodesic line between the source and the destination. As the arrival and dispatch of data packets at a nodal station occurs randomly, the cost of a link is estimated in this paper by the waiting time of a queueing process. This process at a router station is thus modeled by the Markovian $M/M/c$ model, where c is the number of servers at the router station. If other commercial fixed charge is involved for the use of a link, then that can be included in the total cost of a link. Finally, a method of search of a mobile destination is also presented using sphericity of the network. Algorithms for the near geodesic path, costs of links as waiting times and destination search in mobile environment are clearly presented.

Povzetek:

1 Introduction

Global digital data transmission networks have become huge in size with passage of time, with ever changing topology. A data network architecture essentially consists of router ground stations as nodes connected by radio/microwave transmitters or preferably by fiberoptic cables along the ground/sea bed that form the links of the network. The essential function of the network is to deliver data packets without loss from a source node to a destination node of the network. The transmission may take place by arbitrary paths, but a fixed consistent path without loops, is preferable.

Routing algorithms at present, are mainly of three types: *distance-vector routing, link-state routing, and path-vector routing*. Essential features of these algorithms are as follows. In *distance-vector routing* each router periodically multicasts its knowledge to its neighbors covering the whole network. The procedure is dynamic and kept in the form of an updated table in each router. A table contains an assigned cost of each link of the router and the least-cost path between any two nodes calculated by the Bellman-Ford algorithm (Bellman [3], Ford [11], Cormen et. al. [9]). This information enables dispatch of data packets

from source node to its destination node along the least-cost path. RIP (Routing Information Protocol) and IGRP (Interior Gateway Routing Protocol) are protocols of this type (Cisco Systems [7] and [8]), that are widely used as a protocol in the TCP/IP environment to route packets between gateways of the internet. The *link-state protocol* was designed to overcome some of the short comings of the distance-vector protocol. In the link-state procedure, initially a router defines the cost of each link and broadcast its information to all other nodes of the network. Data transmission from source to destination along the least-cost path is determined by Dijkstra's algorithm (Bose [4]). However, if a change occurs in the routing table of some node on the path, then all the intermediate nodes are notified accordingly. OSPF (*Open Shortest Path First*) is an elaborate algorithm that carries out this basic feature of link-state routing (Cisco Systems [6]). The distance-vector and link-state routing protocols are suitable for intra-domain autonomous systems. the former tends to become unsuitable for large number of hops to destination node, while the latter needs huge amount of resources to calculate routing tables, creating heavy traffic of router information. Path-vector routing on the other hand is useful for inter-domain routing of autonomous systems. It is similar to the distance-vector rout-

ing protocol, but differs in the sense that a node called the speaker node of each autonomous system acts on behalf of the entire system, creating its own routing table, and advertising it to its neighboring speaker nodes in the neighboring autonomous systems. In this way the destination addresses (and not the costs of links) together with the path descriptions to each node of the destinations is multicast to reach the final destination node. The path selection in a domain is based on routing metric, consisting of information like bandwidth, network delay, hop count, path cost etc. BGP (*Border Gateway Protocol*) belongs to this category.

The three types of routing protocols require voluminous book keeping at a router and transmission of information to other routers dynamically. The layers of protocol at a router in present day mobile environment is described in detail in Dutta and Schulzrinne [10]. A careful mathematical analysis of routing algorithms is given by Busch and Tirthapura [5]. A dynamic source routing algorithm in such Mobile Ad hoc Network (MANET) is presented in Jayakumar and Chellappan [12] using a new link cache structure maintaining source transparent route. An alternative algorithm is presented by Meghanathan [16] that uses a strategy of energy-efficient broadcast route discovery in the network density and mobility to determine stable routes of transmission. Other approaches such as consensus based networks and multicast pipelined network coding have been suggested in recent years by Arellano-Vazquez et. al. [2] and Li et. al. [13]. On the other hand Li et. al. [14] have presented routing schemes for optimal congestion control for multipath networks having non-congested packet lossess. An earlier investigation by Manvi et. al. [15] suggests a Mobile Agent based Routing (MAR) scheme with the objectives similar to RIP that uses a more flexible, adaptable and distributed mechanism. The present paper addresses the difficulties of data transmission from a geodetic point of view, by presenting algorithms for transmission dynamically, selecting router stations as close to the (shortest) geodesic path between the source and destination nodes on the spherical globe. As different group of data packets arrive at a nodal router through possibly multiple links of the network for forward dispatch, the traffic through the router is modeled as a Markovian $M/M/c$ queueing model, where c is the number of servers employed at the router. The waiting time at such a queue is well known (Bose [4]), and all that is required is to multicast it periodically to the nearest neighboring nodes. From a given node, the next hop is made to that nearest neighbor node, which points towards the destination node, provided the waiting time of that node is not prohibitive. In this way the destination node is reached in quick time, with minimal book keeping of waiting times at a node along the path. The waiting times at the nodes play the role of costs of links used in the standard protocols described earlier. Any fixed charge of a link, as in the case of undersea fiber optic cables may be added to the link cost, if necessary.

2 Selection of linking nodes

A network of stations on the globe is considered, as shown in figure 1 (a). The source and the destination nodes are respectively considered as A_1 and A_n , with respective latitude and longitude (ϕ_1, λ_1) and (ϕ_n, λ_n) , which determine the geographical position of the two nodes. The node $A_i(\phi_i, \lambda_i)$ on the desired path has a nearest neighbor $A_j(\phi_j, \lambda_j)$ that makes the angle $\chi_k = \angle A_k A_i A_n$ the least possible with a favorable waiting time W_k , where $A_k(\phi_k, \lambda_k)$ is any node in the neighborhood of A_i . These nodes are shown separately in figure 1(b). In the spherical triangle $A_i A_k N$, where N is the north pole of the earth, $\angle A_k N A_i =$ difference of latitude of A_k and $A_i = \lambda_k - \lambda_i$, and arc $A_i N =$ colatitude of $A_i = \pi/2 - \phi_i$, arc $r A_k N =$ colatitude of $A_k = \pi/2 - \phi_k$. Let $r := \pi/2 - \phi_i$, ($0 \leq r \leq \pi$), and $\alpha := \angle N A_i A_n$, then from the spherical triangle $N A_i A_k$ (Abramowitz and Stegun [4], p. 79)

$$\frac{\sin(\alpha - \chi_k)}{\cos \phi_k} = \frac{\sin(\lambda_k - \lambda_i)}{\sin r} \tag{4}$$

and

$$\cos r := \sin \phi_i \sin \phi_k + \cos \phi_i \cos \phi_k \cos(\lambda_k - \lambda_i) \tag{5}$$

Eq. (4) yields

$$\alpha - \chi_k = \arcsin \left[\frac{\cos \phi_k \sin(\lambda_k - \lambda_i)}{\sin r} \right] \tag{6}$$

where $\sin r = \sqrt{1 - \cos^2 r}$ and can be determined from Eq. (5). Similarly in the spherical triangle $N A_i A_n$, let $a := \text{arc } A_i A_n$ ($0 \leq a \leq \pi$), then since $\angle A_i N A_n = \lambda_n - \lambda_i$ and arc $N A_n = \pi/2 - \phi_n$,

$$\alpha = \arcsin \left[\frac{\cos \phi_n \sin(\lambda_n - \lambda_i)}{\sin a} \right] \tag{7}$$

where

$$\begin{aligned} \cos a &= \sin \phi_i \sin \phi_n \\ &+ \cos \phi_i \cos \phi_n \cos(\lambda_n - \lambda_i) \end{aligned} \tag{8}$$

so that $\sin a = \sqrt{1 - \cos^2 a}$. Using Eq. (7), Eq. (6) yields the angle χ_k in terms of the latitudes and longitudes (ϕ_k, λ_k) , (ϕ_i, λ_i) , (ϕ_n, λ_n) of the nodes A_k , A_i and A_n respectively. If as before, p is the priority assigned to $|\chi_k|$, the goal is to minimize the objective function

$$z_k = p |\chi_k| + (1 - p) \frac{W_k}{W_{max}} \tag{9}$$

where W_k is the waiting time at A_k , normalized by maximum allowable time W_{max} .

3 Waiting time at a node

A router at a node k is assumed to consist of c number of servers. The data packets of information arrive at the router

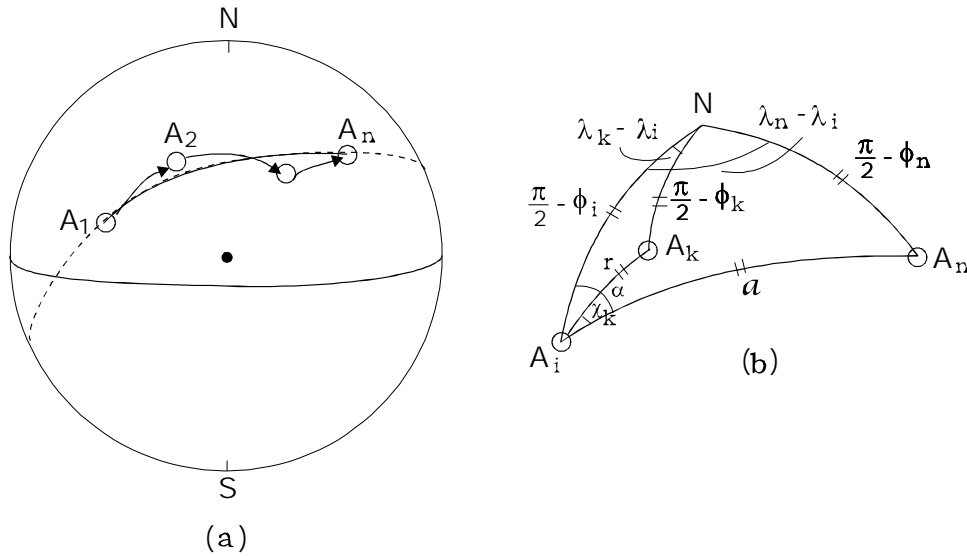


Figure 1: (a) Path $A_1A_2 \dots A_n$ on spherical earth. (b) Configuration of nodes $A_i, A_k,$ and A_n .

at random times, and processed by one of the servers, requiring random serving time and forwarded through a desired link. The serving times are random as the servers of the router may have different specifications and the processing time of a packet for screening by different layers of the router protocol may be different. Thus data forwarding from this point of view is a queueing system that can be modeled by the Markovian $M/M/c$ model (Bose [4], pp. 237-239). According to the model, if $\lambda_k =$ average number of data packets arriving at the router through different links, and $\mu_k =$ average service rate of data packets of one server of the router, then the waiting time W_k is given by

$$W_k = L_k / \lambda_k \tag{10}$$

where L_k is the queue length given by the expression

$$L_k = \frac{\rho}{(1 - \rho)^2} \frac{(c\rho)^c}{c!} / \left[\sum_{n=0}^c \frac{(c\rho)^n}{n!} + \frac{c^c}{c!} \frac{\rho^{c+1}}{1 - \rho} \right] \tag{11}$$

In Eq. (11), $\rho = \lambda_k / c\mu_k$ is the traffic intensity at the router, and must be such that $\rho < 1$ for traffic flow. Otherwise if $\rho \geq 1$, the queue will grow blocking the router altogether.

It is to be noted that the waiting time at a router station will vary during the course of a day, requiring periodic upgradation and shared accordingly with the nodes of the network.

4 The algorithms

The method developed in sections 2 leads to the following pseudo-code for fast transmission of data through terrestrial networks.

Algorithm 1. Fast Data Transmission Path

1. Input: $\phi_1, \lambda_1, \phi_n, \lambda_n;$ \ \ \ Latitude, Longitude of Source and Destination.

p, W_{max} \ \ \ Priority of deviation from geodesic path, and maximum permitted waiting time.

2. Output: $\phi[], \lambda[]$ \ \ \ Latitude, Longitude of intermediate nodes $i = 2, 3, \dots, n - 1$.

3. $i \leftarrow 1$
 $\phi[1] \leftarrow \phi_1; \lambda[1] \leftarrow \lambda_1$

4. $imax \leftarrow$ Number of stations in the neighbourhood of node i .

$$\cos a \leftarrow \sin \phi[i] \sin \phi_n + \cos \phi[i] \cos \phi_n \cos(\lambda_n - \lambda[i]); \sin a \leftarrow \sqrt{1 - \cos^2 a}$$

5. for $k \leftarrow 1$ to $imax$

$\phi[k]; \lambda[k] \leftarrow$ (Latitude, Longitude) of stations in the neighbourhood of node i (must be known).

$$\cos r \leftarrow \sin \phi[i] \sin \phi[k] + \cos \phi[i] \cos \phi[k] \cos(\lambda[k] - \lambda[i])$$

$$\sin r \leftarrow \sqrt{1 - \cos^2 r}$$

$$\chi[k] \leftarrow | \arcsin\{ \cos \phi_n \sin(\lambda_n - \lambda[i]) / \sin a \} - \arcsin\{ \cos \phi[k] \sin(\lambda[k] - \lambda[i]) / \sin r \} |$$

\ \ \ angle of node k with respect to node n .

$W[k] \leftarrow$ Waiting time at base station k (From Algorithm 2).

if $(W[k] > W_{max})$ exit \ \ \ Blocked Link.

$z[k] \leftarrow p \chi[k] + (1 - p) W[k] / W_{max}$ \ \ \ Objective function to be minimised.

end for

6. for $k \leftarrow 1$ to $imax - 1$ \ \ Sort angles
 $\chi[k]$ to avoid ties.
 for $l \leftarrow k + 1$ to $imax$
 if $(\chi[k] > \chi[l])$ then
 temp $\leftarrow \chi[k]$; $\chi[l] \leftarrow \chi[k]$; $\chi[k] \leftarrow$ temp
 end if
 end for
 end for
7. $kmin \leftarrow 1$
 for $k \leftarrow 2$ to $imax$
 if $(z[kmin] > z[k])$ $kmin \leftarrow k$
 end for
8. $i \leftarrow kmin$ \ \ Next hop to node.
9. if $(i = n - 1)$ stop
10. Go To Step 4.
11. end

The waiting time pseudo-code following Eqs. (10) and (11) required in Algorithm 1 is:

Algorithm 2. Waiting Time following $M/M/c$ Queueing Model

1. Input: λ, μ, c \ \ λ = Average number of arrival of data packets at a Station,

μ = Average Service Rate of data packets by one Server.

c = Number of Servers in parallel at a Station.

2. Output: $W[]$ \ \ Waiting Time at the Station.

3. $\rho \leftarrow \lambda / (c\mu)$ \ \ Traffic Intensity at the Station.

if $(\rho \geq 1)$ return \ \ Station is blocked.

4. $p_0 \leftarrow 0$

 for $n \leftarrow 0$ to c

$p_0 \leftarrow p_0 + (c\rho)^n / n!$

 end for

$p_0 \leftarrow 1 / \left(p_0 + \frac{c^c}{c!} \frac{\rho^{c+1}}{1 - \rho} \right)$

$W[] \leftarrow \frac{\rho}{(1 - \rho)^2} \frac{(c\rho)^c}{c!} \frac{p_0}{\lambda}$

5. return

6. end

5 Locating destination node

In case the destination node is not known precisely, as in a mobile environment, its location can be determined by adopting a parsimonious method of flooding the search in a restricted zone of terrestrial links whose pole is the source node A_1 . In the method, it is assumed that the

mobile device possesses some *unique label*, named here as “TARGET_DEVICE”. With A_1 as a pole, the spherical surface is first divided in to “latitudinal” strips of width 1° , the angle subtended at the center of the sphere. The search is carried out starting from the innermost strip outwards, until the destination node of the “TARGET_DEVICE” is detected. For further restricting the search area, the strips are bound on the two sides by “longitudinal” great circles of vertical angle $\pi/8$. This means that the polar region is divided in to eight equal sectors.

With A_1 as pole, a node A_k having latitude-longitude (ϕ_k, λ_k) in the searching zone is suppose bounded by of inner and outer circles of “colatitudes” (ψ_0, ψ_1) , then it can be shown from elementary considerations that

$$\phi_1 + \psi_0 \leq \phi_k \leq \phi_1 + \psi_1 \quad (12)$$

where ϕ_1 is the latitude of A_1 . For restricting the “longitudinal” boundary of the search zone, let the geographical meridian through A_1 be taken as the reference circle. If χ be the vertical angle at A_1 subtended by the node A_k with the reference circle, and ψ_k its “colatitude”, then as in section 2, the angle χ is given by the equations

$$\sin \chi = \frac{\sin(\lambda_k - \lambda_1)}{\sin \psi_k} \times \cos \phi_k \quad (13)$$

where

$$\cos \psi_k = \sin \phi_1 \sin \phi_k + \cos \phi_1 \cos \phi_k \cos(\lambda_k - \lambda_1) \quad (14)$$

The angle χ must then lie in the octant of search. The method described above leads to the following:

Algorithm 3. Location of Destination Node

1. Input: ϕ_1, λ_1 ; \ \ Latitude, Longitude of Source Node.

2. Output: ϕ_n, λ_n ; \ \ Latitude, Longitude of Destination Node.

3. $\psi_0 \leftarrow 0$

4. for $i \leftarrow 1$ to 180

$\psi_1 = \leftarrow i * \pi / 180$

 for $j \leftarrow 1$ to 8

$\chi_0 \leftarrow 0$; $\chi_1 = \leftarrow \chi_0 + \pi / 8$

 for $k \leftarrow 1$ to $kmax$; \ \ $kmax$ is max. number of nodes in the search zone.

$\phi[k]; \lambda[k] \leftarrow$ Latitude, Longitude of the stations in k th search zone.

$$\cos(\psi_k) \leftarrow \sin \phi_1 \sin \phi[k] + \cos \phi_1 \cos \phi[k] \cos(\lambda[k] - \lambda_1)$$

$$\sin(\psi_k) \leftarrow \sqrt{1 - \cos^2(\psi_k)}$$

$$\chi \leftarrow \arcsin(\sin(\lambda[k] - \lambda_1) * \cos(\phi[k]) / \sin \psi_k)$$

```

while ( $\psi_0 \leq \phi[k] - \phi_1 \leq \psi_1$  &&
       $\chi_0 \leq \chi \leq \chi_1$ ) do
  { if (DEVICE_LABEL ==
      "TARGET_DEVICE")
    then \\ TARGET_DEVICE located.
       $\phi_n = \phi[k]$ ;  $\lambda_n = \lambda[k]$ 
      stop
    end if }
end for
 $\chi_0 \leftarrow \chi_1$ ,  $\chi_1 \leftarrow \chi_0 + \pi/8$ 
end for
end for
5. end

```

6 Conclusion and future scope

Data transmission networks require large tables at the router stations for lossless steady transmission. Moreover, the information at a router node is dynamically shared by other nodes as well. In present day dynamic networks, the information sharing tends to increase by huge amounts. To mitigate the work load of the routers, a data transmission algorithm is presented here, in which transmission takes place from source to destination dynamically along a path as close as possible to the terrestrial geodesic joining the two nodes, taking in to account the costs of the intervening links. The cost of a link at a connecting node is estimated here by the waiting time at the node router according to the queueing $M/M/c$ model. All that is required is to share the information of the waiting times periodically among all the nodes of the network. The algorithm however does not preclude use of other methods of estimating the costs of the different links. An improvement in this direction could be to find a model that incorporates sudden unsteady bursts observed in actual data transmission in fiber cables, and raising the tail of the Markov queueing model. Finally, an algorithm is presented for searching the destination node, if it is unknown, such as in the case when the device is in a mobile environment.

7 Acknowledgement

The author is thankful to the S.N. Bose National Center for Basic Sciences, Kolkata for providing necessary facilities for undertaking this research. Special thanks also go to Dr. Sagar De for useful discussions and inputs.

8 References

References

- [1] M. Abramowitz and I.A. Stegun (1972), Handbook of Mathematical Functions, Dover Publications, New York.
- [2] M. Arellano-Vazquez, M. Benitez-Perez, J. Ortega-Arjono (2015), A consensus routing algorithm for mobile distributed systems, Int. J. of Distributed Sensor Networks, **11**, <https://doi.org/10.1155/2015/510707>.
- [3] R.E. Bellman (1958), On a routing problem, Quart. Appl. Math. **16**, 87-90.
- [4] S.K. Bose (2012), Operations Research Methods, Narosa Publishing, New Delhi.
- [5] C. Busch, S. Tirthapura (2005), Analysis of Link Reversal Routing Algorithms, SIAM J. Comput. **35**, 305-326, <https://doi.org/10.1137/S0097539704443598>.
- [6] Cisco Systems (2011), IOS IP Routing, RIP Configuration Guide.
- [7] Cisco Systems (2005), An Introduction to IGRP.
- [8] Cisco Systems (2011), IOS IP Routing, OSPF Command Reference.
- [9] T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein (2009), Introduction to Algorithms, McGraw-Hill, 651-655.
- [10] A. Dutta, H. Schulzrine (2014), Mobility Protocols and Handover Optimization, John Wiley, New York.
- [11] L.R. Ford, Network Flow Theory (1956), RAND Corporation paper P-923, Santa Monica, California.
- [12] C. Jayakumar, C. Chellappan, Optimized on demand routing protocol of mobile ad hoc network, Informatica, **17** (2006), 481-502.
- [13] P. Li, S. Guo, S. Yu, A.V. Vasilkos (2014), Reliable Multicast with Pipelined Network Coding using Opportunistic Feeding and Routing, IEEE Transactions on Parallel and Distributed Systems, **25** (2014), 3264-3273, <https://doi.org/10.1109/TDPS.2013.2297105>.
- [14] S. Li, W. Sun, Y. Zhang, Y. Chen, Optimal congestion control and routing for multipath networks with random losses, Informatica, **26** (2015), 313-334, <http://dx.doi.org/10.15388/Informatica.2015.50>.
- [15] Manvi, S. Sunilkumar, P. Venkataram, An agent-based best effort routing technique for load balancing, Informatica, **17** (2006), 407-426.

- [16] N. Meghanathan, Performance comparison study of multicast routing protocols for mobile ad hoc networks under default flooding and density and mobility aware energy-efficient (DMEF) broadcast strategies, *Informatica*, **35** (2011), 165-184.