Study of Routing Algorithms Considering Real Time Restrictions Using a Connectivity Function

Magali Arellano-Vázquez[1], Héctor Benítez-Pérez[2], and Jorge L. Ortega-Arjona[3]

[1] Posgrado en Ciencia e Ingeniería de la computación
[2] Departamento de Ingeniería de Sistemas Computacionales y Automatización
[3] Departamento de Matemáticas, Facultad de Ciencias, UNAM, México D.F., Mexico
arellano_m@uxmcc2.iimas.unam.mx, hector@uxdea4.iimas.unam.mx, jloa@ciencias.unam.mx

The paper focuses on the study of a mobile distributed system that is characterized by frequently changing topology. The routing algorithms [4,5,1] for such a system should be, in general, fully adaptive. Additionally, it is important to know the state of the task scheduler of each node in order to determine whether it acts as a router. Traditionally, existing routing algorithms [3,2] resort to the path discovery process for each modification in topology.

This paper introduces an adaptive routing algorithm based on a connectivity function that evaluates the state of the node's task scheduler as well as the general conditions of the network. The connectivity function assesses the status of the current node and the connection states of its neighboring nodes, thus obtaining the overall state of the system through local data. It is necessary to quantify the cost of the routes, as there may be more than one, considering hops from node to node as a measure. This measure can be bounded, i.e. the number g of hops in a route from node A to node B belongs to the set $g = 0, 1, \ldots, n-1$, where n is the number of nodes.

The construction of the path is performed by using the following five steps:

1. Calculating node's availability.
2. Evaluating connection state of neighboring nodes.
3. Calculating the connectivity function per node.
4. Calculating or updating the adjacency matrix.
5. Calculating node $k+1$ using Floyd-Warshall's algorithm.

The connectivity function is defined as follows: $f(x) = e^{-\frac{{s_d}^2+\delta n^2+hops^2+C_{i,j}^2}{\sigma}}$, where s_d is the space needed to transmit, δn is the data loss in the channel, *hops* is the number of nodes the message passed and $C_{i,j}$ is the load in the data channel between node i and node j.

Let A be a symmetric matrix, the element $A_{i,j}$ indicates whether a node i is connected to the node j. Each row or column of the matrix A represents the node's connections. Therefore, any node can be used as a router, if a node has enough idle time then it will be available for a routing service, this service will be a low priority process, so this node will first complete its own services and then it will respond to external requests. The proposed algorithm does not attempt

R. Groß et al. (Eds.): TAROS 2011, LNAI 6856, pp. 412–413, 2011.

to reserve a channel communication, it assures that the message arrives to its destination despite frequent changes in the network's topology, because the route is rebuilt at each step.

The worst case scenario is that the network topology changes faster than routing itself. This would be reflected in the fact that for the n units time interval we would have n different adjacency matrices. To avoid a common error in adaptive routing algorithms, like RIP [6], which is falling into a cycle (this happens when the routing information is not updated), the proposed algorithm evaluates periodically the conditions of the network, so the matrices are independent and they do not keep a relation to the previous state of the network.

To obtain the full state of the network through local data, the adjacency matrix has been implemented, so that each node possesses information of the nodes to which it connects. Since the adjacency matrix is a mathematical representation of the network's connectivity, we can calculate the cost of the paths between all nodes, adding the restriction mentioned above on the number of hops. To calculate the cost of the paths from each node to the other $n-1$ nodes, the Floyd-Warshall algorithm is used.

Implementation results of our algorithm show that the obtained route is optimal in every transitory state. However, it is possible to get a transitory state in which the destination is unreachable if the node's scheduler is too busy to transmit a data package. In this case a constraint can be added so that upon arrival of a data package enough space is assigned to the node's scheduler. This idea has not yet been implemented and is left for future work. In most cases the routes are not reversible. The run-time results also show that the time is oscillating and not converging to any particular value. A future challenge is to improve the execution time so the time will converge.

References

1. Jacquet, P., Mühlethaler, P., Clausen, T., Laouiti, A., Qayyum, A., Viennot, L.: Optimized link state routing protocol for ad hoc networks. In: Proc. IEEE INMIC 2001, pp. 62–68. IEEE, New York (2001)
2. Liu, Z., Kim, J., Lee, B., Kim, C.: A routing protocol based on adjacency matrix in ad hoc mobile networks. In: Proc. 7th Int. Conf. on Advanced Language Processing and Web Information Technology, pp. 430–436. IEEE Computer Soc., Los Alamitos (2008)
3. Medhi, D., Ramasamy, K.: Network Routing: Algorithms, Protocols, and Architectures. Morgan Kaufmann, San Francisco (2007)
4. Park, V.D., Corson, M.S.: A highly adaptive distributed routing algorithm for mobile wireless networks. In: Proc. IEEE INFOCOM 1997, vol. 3, pp. 1405–1413. IEEE Computer Soc., Los Alamitos (1997)
5. Perkins, C., Royer, E.: Ad-hoc on-demand distance vector routing. In: Proc. of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, pp. 90–100. IEEE Computer Soc., Los Alamitos (1999)
6. Peterson, L.L., Davie, B.S.: Computer Networks: A Systems Approach, 4th edn. Morgan Kaufmann, San Francisco (2007)