Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2015, Article ID 510707, 8 pages
http://dx.doi.org/10.1155/2015/510707

Research Article

A Consensus Routing Algorithm for Mobile Distributed Systems

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

[1] *Posgrado en Ciencia e Ingeniería de la Computacion, UNAM, Circuito Escolar s/n, Ciudad Universitaria, Delegación Coyoacán, 04510 México, DF, Mexico*
[2] *Departamento de Ingeniería de Sistemas Computacionales y Automatizacion, IIMAS, UNAM, Apartado Postal 20-726, Delegación A. Obregon, 01000 México, DF, Mexico*
[3] *Departamento de Matematicas, Facultad de Ciencias, Avenida Universidad 3000, Circuito Exterior S/N, Delegación Coyoacán, 04510 México, DF, Mexico*

Correspondence should be addressed to Magali Arellano-Vázquez; magali.arellano@gmail.com

Received 4 March 2015; Revised 21 May 2015; Accepted 11 June 2015

Academic Editor: Chen Peng

This paper introduces a *consensus routing algorithm* based on an availability function. Such a function obtains a consensus among a group of nodes by evaluating the idle time in the scheduler of each node, along with the general conditions of the network and determining the next step of the route. In this regard, the algorithm proposed here makes it possible to avoid flooding while reducing bandwidth use and keeping changes of the network only at a local scale.

1. Introduction

A Mobile Distributed System (MDS) is composed of some wireless devices, acting as nodes. Each node exposes not only a relative position with the rest of the mobile devices, but also a state, and it may act as a router: in a MDS, each node communicates with its neighbours not only as a sender or receiver, but also as a router, forwarding packets to nodes that are not within the transmission range of the sender [1].

Commonly, a MDS is described as a dynamic topology since it changes with the movement, entry, and exit of nodes within the range of the MDS. Such a dynamic behaviour makes it difficult to establish and maintain a communication route between two remote nodes, since all the nodes of such a route may not be available for routing at a later time. Traditional routing algorithms are not useful here, since they require knowledge about the network as a whole in order to determine any optimal route. Using a route discovery algorithm every time the topology changes requires constantly generating a global state of the MDS.

For performance purposes regarding the transmission of messages, routing algorithms need to be bandwidth efficient, spend as few resources as possible in the computation of a route, and reconcile conflicting goals such as dynamic adaptation and low resource consumption [2]. Routing algorithms are commonly described as part of routing protocols.

2. Related Work

There are studies about routing protocols in wireless networks, showing that these protocols can be classified into five main types: reactive, proactive, hybrid, hierarchical, and based on coordinates.

Reactive protocols perform on-demand route discovery by flooding the network and delaying packets until the routes are established. Examples of these protocols are [3], DSR [4], and TORA [5].

Proactive protocols maintain routes between all pairs of nodes. The network is flooded with information packets whenever there are changes in the topology but does not incur delays or overhead to display cover routes on demand. DSDV [6], OLSR [7], and WRP [8] are examples of proactive protocols. In general, proactive protocols present a good performance in static scenarios, while reactive protocols do so in mobile scenarios.

Whether reactive or proactive, in both types of protocols, flooding means that every node in the MDS sends all incoming packets to its neighbourhood, except those packets that must receive. Flooding is easy to implement. Nevertheless, it represents a brute force approach. Furthermore, it is inefficient in the sense that it tends to exponentially increase traffic until consuming all bandwidth of communication channels. Since flooding consumes a substantial amount of bandwidth, it is important to reduce the frequency of route discoveries and so network flooding [1].

Hybrid protocols have been proposed to overcome performance problems due to frequent route discovery attempts in on-demand routing protocols. These routing protocols incorporate both proactive and on-demand protocol features [9]. However, for each route request hybrid protocols do not wide-flood the network. It is difficult to balance between the cost of periodically exchanging route information (i.e., proactivity) and route discovery by controlled network flooding (i.e., reactivity) [10]. Other protocols such a geographical or bypass routing reduce the frequency of flooding by allowing a relay node to start a limited route discovery in the event of a route failure [3, 11], or employ local error recovery mechanisms [12, 13]. Hybrid protocols using either controlled flooding or local error recovery have focused on reducing packet drops and not efficiently using bandwidth during route recovery. However, multipath protocols require additional packet drops and delays due to the dependency on potentially stale routes from caches [1]. Hybrid protocols such as ZRP [14] and SHARP [10] achieve good performance across a wider range of scenarios by combining both reactive and proactive components. They divide the network into zones. Nodes maintain routes proactively within their zone by flooding topology changes within the zone. Routes between zones are discovered on demand with an optimised flooding mechanism [15].

Hierarchical and coordinate-based protocols do not flood the network. They use location-dependent addresses in routing. These identifiers can change with mobility and, in some protocols, with congestion and failures [16–18]. Therefore, these protocols use both a fixed identifier and a location-dependent address for each node and they usually require mechanisms to look up the location of a node given its fixed identifier [19]. These mechanisms reduce resilience to failures, introduce overhead, and increase complexity [15]. For example, LANMAR [17] and L+ [16] are hierarchical protocols, and GPSR [20] and BVR [18] are coordinate-based protocols.

Protocol VRR [15] combines aspects of proactive routing, establishing and maintaining routes without flood routing by combining the use of a logical ring and a physical map. VRR uses a scheme to identify nodes by a random number between 0 and a finite upper limit. This implies the need to have a mechanism for consensus to assign numbers to each node or the existence of a centralised node that is responsible for global identifiers. This restricts the scalability of the protocol.

Routing algorithms for ad hoc networks, such as AODV [3], TORA [5], and OLSR [7], are not useful to solve the problem of calculating a route between a pair of nodes without having a global state, since these algorithms do not take into consideration the status of the local scheduler within each node. Wormhole [21] and vector-based distance and link state [22, 23] algorithms cannot be used here, since they reserve a channel and keep it, but due to the dynamic topology, they do not seem to be a time-efficient option for route construction [24–28].

The WARP algorithm [29] focuses on repairing paths. WARP is reactive and is based on distance vector and detects when the position of the destination node has changed and it makes a spiral search engine around the old position, thus seeking a neighbour node that still maintains communication with the destination node. This algorithm requires a vector distance, which could be inconsistent.

Researches have developed consensus mechanisms for routing protocols. For example, in [30], the authors separate the delivery of the package into two different logical modes: a stable mode, which ensures that the route is selected only after arriving at an agreement that is reached by a state consisting of a global view, and a transient mode, which ensures high availability when a package finds a router that does not have a stable route, due to a fail in the link or because the algorithms are still calculating a stable route. The consensus algorithm is used to calculate a global state of the system, staring from the distributed snapshots that each autonomous system sent.

Opportunistic networks are similar to MDS. However, the routing algorithm used in this kind of networks is the Epidemic Routing [31]. This routing algorithm uses the mobility of nodes to propagate messages over the network: every time a node physically finds another node there is an exchange of distance vectors. Both nodes then compare and update their messages and vectors. This is a costly operation each time, since the network normally tends to have a large number of nodes. Thus, this protocol maximises the rate of message delivery, while minimising latency delivery. Nevertheless, this is costly in terms of bandwidth utilisation and processing on each node.

The algorithm "Adaptive Routing in Mobile Ad Hoc Networks Based on Decision Aid Approach" [32] chooses which routing algorithm is suitable depending on the current network conditions. This algorithm provides three algorithms to choose from: OLSR [7], DSR [4], and AODV [3]. It uses a voting algorithm. This assigns a weight to each node depending on the number of neighbour nodes that have a majority vote.

Consensus algorithms in distributed systems have been used successfully to problems where a team of agents must cooperatively compute logical vector function that returns a set of decisions depending on a set of input events. Agents have partial accessibility to the input events, very simple local computation and broadcast communication capabilities, and may be affected by failures. This issue involves reliable network information diffusion that can be achieved making use of techniques based on Robust Flooding [33]. For dynamic networks, [34] proposes a synchronous consensus model, in which at each round after the network topology it changes completely. It uses a variant connection model and a global clock to synchronise the rounds.

In [35] a consensus algorithm for opportunistic networks is proposed. Each node carries messages and propagates them through the network, using Epidemic Routing [31]. Disconnection of a node is not considered as a fault, but as an expected behaviour. It is expected to reach immediately the neighbourhood of the destination node. The algorithm requires a global view of the network.

Different modelling schemes may be approximated through bounded time delays. This effect can be modelling, for instance, by bounding sampling data to guarantee a transmission rate over a trigger protocol event. For example, the reactive protocol in [36] allows coping several varying time multiple sampling periods, such as current condition in this paper, by entering an artificial induced local delay which is directly related to a known sample data error that can be locally modelled through a proper local control design. In [37] the authors propose an event-triggered communication scheme. This generates the transmission events, utilising information about the system's outputs at a time-varying sampling period and the output error. In this work sampling is performed using a known and bounded time function for a local system on a global network guaranteeing the stability in an exponential response. The variable sampling function is global and its effects are local to each node to interact on this global network. The idea of local influence is used in CRA.

3. Contributions

Our goal here is to propose an algorithm for routing discovery, whose main objective is to reduce flooding. A *Consensus Routing Algorithm* (CRA) takes into consideration characteristics of the MDS, such as idle time in the scheduler, network's availability, mobility, and the distance between nodes. Idle time in the scheduler node is an important feature because if there is not enough time in node's scheduler to transmit a message then the path is broken. The proposal includes versions for static nodes and for mobile nodes.

The algorithm obtains a route through a dynamic topology, by establishing successive consensus among nodes. In a local group, each node obtains routing information from its neighbours while they have an active connection. This provides resilience to local changes of the dynamic topology: any change is kept at a local scale within the MDS; when some change occurs (a mobile node moving, entering, or exiting), it does not affect the overall state of the MDS, but it only affects the local group of the involved nodes. Thus, it is not necessary to recalculate the whole route, but only those groups of nodes affected by the change. The union of the local states of the nodes forms the local network status at a given time T.

In order to determine the cost of sending a packet between two nodes, routing metrics are defined in terms of how long it takes to deliver a packet over a given route. Normally, each routing algorithm has its own metrics, which assign a value to a route, based on a factor such as hop count, bandwidth, and link reliability [38].

The rest of this paper is organised as follows. Section 2 presents an overview of the *CRA*; Section 3 describes the experiments based on the case studies for one and two hops. Section 4 presents our results. And Section 5 provides conclusions.

4. The Consensus Routing Algorithm (CRA)

For the purposes of this paper, the CRA is analysed considering that the topology is "mobile"; that is, the topology changes through time.

For modelling a distributed system (MDS) with n nodes, consider the following parameters and variables:

(i) Let *node* be a data structure per node that includes the following fields:

  (1) Unique ID per node.

  (2) Axes x and y.

  (3) Letting β be the field corresponding to idle time in the scheduler of a node when it acts as a router.

  (4) Letting λ be the field corresponding to the load of every link between all pairs of directly connected nodes.

  (5) A field that indicates whether a node can move through network.

(ii) Let n_s be the *source node*, from which the route discovery process starts, and let n_d be the *destination node*, to which the route discovery process ends.

(iii) Let $d_G(i)$ be the *degree of node* i, that is, the number of links incident to the node i.

(iv) Let k be the *total number of routing steps* of a given route.

(v) Let l be *neighbourhood* with respect to a node, whose extension is presented in number of hops.

(vi) Let J_i be the set of nodes belonging to neighbourhood of node i in radial neighbourhood radius ρ. Where ρ is defined by the wireless range of each node, every node into radius ρ is a neighbour of i and belongs to J_i. The set J_i can be defined as in

$$J_i = \left\{ j : \left| x_i - x_j \right| \wedge \left| y_i - y_j \right| \leq \rho \right\} \qquad i = 1, \ldots, n, \quad j = 1, \ldots, n. \tag{1}$$

(vii) Let $|J_i| = m$ be the number of entries for CRA for each round.

(viii) Let $f^k_{i,J_i}(\rho) \mid i = 1, \ldots, n$ be the *function values of link availability* [39]. These values are used as input for the algorithm.

(ix) Let y^k_i be the output of the consensus operation.

(x) A small percentage of nodes are mobile during the route search.

The CRA makes use of an *availability function* $(f^k_{(i,J^h_i)}(\rho))$ to assess the local conditions of the connection between a pair of nodes. The values of an availability function are used as input to the consensus algorithm, which is based on [40].

Equation (2) describes such an availability function, which is calculated for each node of the MDS:

$$f^{k}_{(i,J_i)}(\rho) = e^{-((\lambda(J^h_i)-\Gamma)^2/\sigma_1^2+\text{hops}^2/\sigma_2^2+(\tau-\beta(J^h_i))^2/\sigma_3^2)} \qquad i = 1,\ldots,n;\ h = 1,\ldots,m. \tag{2}$$

This availability function takes into consideration not only the parameters and variables described above, but also other relevant parameters of the MDS, such as the current available bandwidth per link (τ), the idle time in the scheduler of a node (Γ), and the number of hops between a particular pair of nodes (hops). The value obtained from this function is between 0 and 1, which indicates whether such a link is available to be part of the route. Notice that commonly $\lambda(J^h_i)$ is given in μs, while $\beta(J^h_i)$ is given in k bps. Thus, to avoid inconsistencies, both parameters are scaled to 10^1. Let $\sigma_i \mid i = 1, 2, 3$ be standard deviations of λ, τ and hops, respectively.

The algorithmic complexity of the CRA depends on the size of neighbourhood (l) and the degree of a node, that is, the number of nodes with which each node is directly connected ($D_G(n)$).

The implementation of the CRA is based on a search tree, in which each branch is evaluated to choose the best candidate. As the number of nodes grows, the neighbourhood l of a node tends to grow. Thus, the search tree also grows in depth as the number of neighbours per node increases.

The CRA is developed for the MDS, through the following steps:

(1) $i = n_s$ starts searching for n_d by initially searching in a radial neighbourhood of radius ρ. n_s gets information about load and idle time in node's scheduler.

For this, n_s verifies the values, for its direct neighbours; for information about their link load; and (c) λ, checking for the time available in the scheduler of every node in the neighbourhood. In turn, n_s informs to its neighbours to look for n_d.

Notice that while n_s performs a consensus (described in the following steps), the nodes included in J_i simultaneously seek n_d in its immediate neighbours. If n_d is found at this point, the routing finishes. Otherwise, the neighbours start a new consensus, now with their own direct neighbours, by calculating the availability function, using (2). Then values that are proposed to begin the consensus are the values obtained by the availability function on each pair consisting of n_s and its neighbours.

(2) Calculate the numerical distances $m(m-1)/2$ input pairs. The numerical distances are calculated using

$$d_{(J^h_i,J^{h+1}_i)} = \left| f^{k}_{(i,J^h_i)}(\rho) - f^{k}_{(i,J^{h+1}_i)}(\rho) \right|, \quad h = 1,\ldots,m. \tag{3}$$

(3) Determine the "degree of closeness," or "level of agreement," of all $m(m-1)/2$ input pairs based on

$$s_{(J^h_i,J^{h+1}_i)} = \frac{1}{1 + p d^{q}_{(J^h_i,J^{h+1}_i)}}, \quad h = 1,\ldots,m. \tag{4}$$

This function (known as "agreement indicator") provides a value for the agreement of inputs J^h_i and i, based on their numerical distance. The numerical distance is then considered as the comparison of the availability of each data link between two neighbouring nodes.

In this function, the parameters p and q are used only for tuning purposes. The parameter q tunes the rate of roll-off of this function, while the parameter p is used to set the midpoint value of the function. This function is continuous and generates $s_{(i,J^h_i)} = 1$ for numerically equivalent input pairs. For differing inputs, it produces a real value in the range $(0, 1)$ such that, as $d_{(i,J^h_i)}$ gets larger, $s_{(i,J^h_i)}$ tends towards zero [40].

(4) After obtaining the agreement indicator values for all of the input pairs (i, J^h_i), a weight value is assigned for each input pair based on (5). An input value that differs greatly from the other input values is assigned a smaller weight than an input value that is close to any of the other input values [40]:

$$w_{(i,J^{\omega}_i)} = \frac{1}{1 + \prod^{m}_{h\neq\omega,h=1} s_{(J^{\omega}_i,J^h_i)}}, \qquad 0 < w_{(i,J^{\omega}_i)} < 1,\ \omega = 1,\ldots,m. \tag{5}$$

This type of weighting directly indicates the influence of the agreement between any particular input and the other inputs on its weight value [40].

(5) The output of the consensus is obtained using

$$y^k_i = \frac{\sum^{m}_{h=1} w_{(i,J^h_i)} f^{k}_{(i,J^h_i)}(\rho)}{\sum^{m}_{h=1} w_{(i,J^h_i)}}. \tag{6}$$

(6) The value of y^k_i is compared with the input values, so the node in J corresponding to the input associated with (7) is chosen as the winner of the consensus:

$$\min\left(\left| y^k_i - f^{k}_{(i,J^h_i)}(\rho) \right|\right) = G^k, \quad h = 1,\ldots,m. \tag{7}$$

The node is named G^k and acts as the original source node n_s for the next route discovery. Then, the value decided by CRA is G^k. Moreover, notice that this new route discovery is carried out in a more efficient way, since the nodes that participated in the previous consensus now look up their own $y^k_{J^h_i}$ and may use this value for another route discovery of consensus in a kth future step. This is so since all nodes in J_i have calculated their $y^k_{J^h_i}$ regarding their immediate neighbours.

(7) Finally, after several route discoveries, some node should have n_d in its J_i, and a route has been found between n_s and n_d. This route is made up by all the winners of every consensus. So, a path of k steps

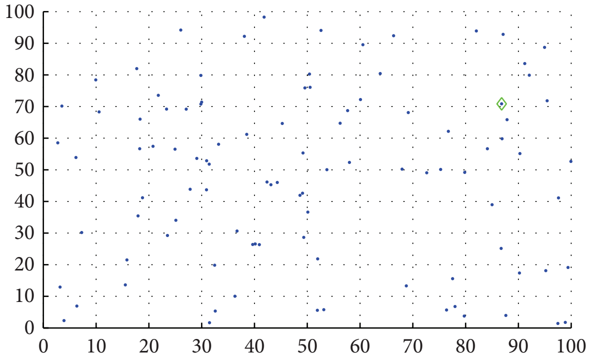


Figure 1: Network used in this example.

would be made up by the corresponding nodes, which can be represented as a sequence:

$$n_s \longrightarrow G^1 \longrightarrow G^2 \longrightarrow \cdots \longrightarrow G^k \longrightarrow n_d. \qquad (8)$$

Every G^k is a local minimum, to be maintained if the values of nodes β and λ do not change. This is the case for a mobile topology. The choice of a local minimum provides the most suitable node to transmit among reachable nodes in the kth step.

The objective of this routing algorithm is to consider local conditions. This will allow the nodes to decide a route based on such local, bounded conditions.

This routing algorithm makes use of a consensus algorithm to evaluate such local conditions, allowing "deciding" which neighbour node to use as router and thus going on building a route aiming to the destination node.

The CRA makes use of the concept of l-neighbourhood and achieves the purposes of avoiding a complete flooding of the MDS. The algorithm has been tested considering the number of hops as a parameter for defining the reach of the l-neighbourhood.

5. Experiments

Simulation experiments have been carried out with the CRA for a mobile topology, using a MDS of 100 nodes. These nodes were distributed based on a normal distribution.

All simulations have been implemented using Matlab. Note that, for each step of the route, the CRA is applied until the destination node is found or when a time-out condition is declared. Figure 1 shows CRA's behaviour in a network, where the source node (n_s) is placed at coordinates (87, 71) and the destination node (n_d) is placed at coordinates (95, 89).

Using this network, both algorithms (CRA and flooding) were executed with the following results. The results of CRA are presented in Table 1 and Figure 2. Table 1 shows (in time) construction step by step of a route between node A and node B. The first column displays the times CRA has been used before reaching the destination node. The second column shows location of each source node to the start of each execution of CRA. The winner path is shown in columns 2, 3, and 4; it is composed of 3 pairs of coordinates: the first pair of coordinates is the source node for corresponding round, and the second and third pairs of coordinates correspond to source node's direct neighbour and a neighbour of this, which has been evaluated before by CRA as the most suitable path. In round 4 destination node is reached. The total cost of this search is 5.93 units of availability function, and the total distance covered is 64.61 units.

Table 1: CRA's results for searching a route from node placed at (87, 71) to destination node placed at (95, 89).

Round	n_s		n_d	Cost	Distance
1	(87, 71)	(84, 57)	(90, 55)	1	20.78
2	(90, 55)	(87, 60)	(84, 57)	1.97	9.99
3	(84, 57)	(88, 67)	(92, 80)	1.97	24.57
4	(92, 80)	(95, 89)		0.99	9.26
				5.93	**64.61**

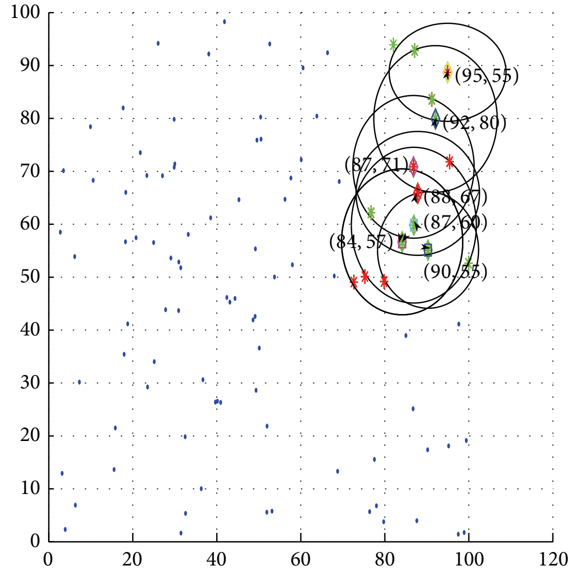


Figure 2: CRA's results for searching a route from node placed at (87, 71) to destination node placed at (95, 89).

The data reported in Table 1 are depicted in Figure 2.

For comparison purposes, using the same network, source node, and destination node, the experiment was replicated using the flooding algorithm. The results are displayed in Table 2, where column 1 displays the number of rounds in which the result was obtained for the corresponding row. In this experiment, the flooding algorithm was run 2 times and 43 results were obtained, of which 7 are of round 1 and the rest of the latter. The route reaching the destination node consists of three nodes, but there were 41 extra parallel transmissions. The total cost of this search is 41.74 units in the availability function, and the total distance covered is 464.78 units. The data displayed in Table 2 are shown in Figure 3.

Table 2: Flooding algorithm results for searching a route from node placed at (87, 71) to destination node placed at (95, 89).

Round	n_s	n_d	Cost	Distance
1	**(87, 71)**	(88, 67)	0.98	5.119
2	(88, 67)	(54, 50)	0.9956	6.98
2	(88, 67)	(58, 52)	0.9914	9.30
2	(88, 67)	(45, 65)	0.9627	10.08
2	(88, 67)	(44, 46)	0.992	10.54
2	(88, 67)	(42, 46)	0.8929	11.42
2	(88, 67)	(56, 65)	0.7344	11.72
2	(88, 67)	(43, 45)	0.986	11.74
2	(88, 67)	(39, 61)	0.986	12.16
2	(88, 67)	(49, 43)	0.9961	12.75
2	(88, 67)	(49, 42)	0.9854	13.42
1	**(87, 71)**	(95, 72)	0.99	8.663
2	(95, 72)	(16, 22)	0.99	7.87
2	(95, 72)	(6, 7)	0.94	11.39
2	(95, 72)	(3, 13)	0.99	12.38
1	**(87, 71)**	(92, 80)	0.98	10.46
2	(92, 80)	(95, 18)	0.99	4.34
2	(92, 80)	(90, 17)	0.98	9.38
2	(92, 80)	(87, 25)	0.99	14.02
1	**(87, 71)**	(87, 60)	0.99	11.02
1	**(87, 71)**	(76, 62)	0.99	13.31
2	(77, 62)	(82, 94)	0.99	5.18
2	(77, 62)	**(95, 89)**	0.99	8.84
1	**(87, 71)**	(91, 84)	0.95	13.47
2	(91, 84)	(80, 4)	0.99	7.88
2	(91, 84)	(78, 7)	0.98	10.02
2	(91, 84)	(97, 1)	0.99	10.19
2	(91, 84)	(76, 6)	0.90	11.36
2	(91, 84)	(99, 2)	0.97	11.48
2	(91, 84)	(90, 17)	0.98	13.64
1	**(87, 71)**	(84, 57)	0.98	14.42
2	(84, 57)	(49, 42)	0.98	5.51
2	(84, 57)	(49, 43)	0.99	6.06
2	(84, 57)	(49, 29)	0.99	8.02
2	(84, 57)	(44, 46)	0.99	11.02
2	(84, 57)	(43, 45)	0.98	11.15
2	(84, 57)	(42, 46)	0.89	12.29
2	(84, 57)	(41, 26)	0.82	13.79
2	(84, 57)	(54, 50)	0.99	13.90
2	(84, 57)	(40, 27)	0.99	14.14
2	(84, 57)	(40, 26)	0.97	14.61
2	(84, 57)	(37, 31)	0.98	14.68
2	(84, 57)	(52, 22)	0.97	14.87
			41.74	**464.78**

6. Results

In the simulations that have been performed, the flooding algorithm established 44 connections while searching exhaustively the destination node, evaluating each link using an "availability function." The total cost of the search was 41.74 units and the distance covered was 464.78 units. Instead, CRA established only 7 links having 5.93 units of cost, and the distance covered was 64.61 units.

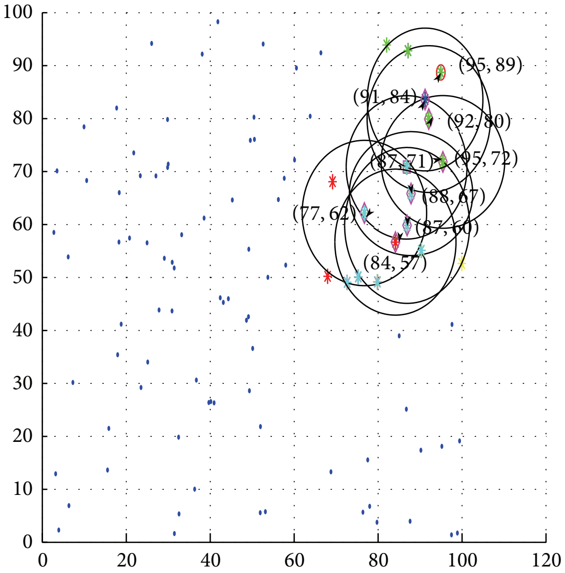


Figure 3: Flooding algorithm results for searching a route from node placed at (87, 71) to destination node placed at (95, 89).

It often happens that CRA presents a routing-loop, so that the destination node is not reached. In order to prevent the algorithm from reaching a state of completion, a number ξ that will limit the time that a node can be in a cycle before stopping the process is defined. Therefore, a flag is sent and the algorithm will terminate correctly.

The flooding algorithm does not consider the local conditions of a node, so that the choice of nodes belonging to the route is inefficient, in the sense that it is not ensured that these nodes are available to transmit, and this increases the probability of delays during the routing process.

7. Conclusions

The route obtained by the CRA is represented as a sequence of nodes with the best conditions for transmitting like a router and thus reducing the communication efficiency of such a route. By using a limited neighbourhood, the number of active connections is significantly reduced, particularly when compared with flooding. However, the flooding algorithm obtains a route in less time, but this route is more likely to be inefficient compared with the route calculated by the CRA.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors would like to thank the financial support of PAPIIT UNAM INI00813 and CONACYT 176556 and technical assistance of Ing. Adrian Durán and Ing. Joel Durán.

References

[1] C. Sengul and R. Kravets, "Bypass routing: an on-demand local recovery protocol for ad hoc networks," *Ad Hoc Networks*, vol. 4, no. 3, pp. 380–397, 2006.

[2] M. K. Marina and S. R. Das, "Routing in mobile ad hoc networks," in *Ad Hoc Networks*, P. Mohapatra and S. V. Krishnamurthy, Eds., pp. 63–90, Springer, New York, NY, USA, 2005.

[3] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," in *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA '99)*, pp. 90–100, February 1999.

[4] D. Johnson, Y. Hu, and D. Maltz, "The dynamic source routing protocol (DSR) for mobile ad hoc networks for IPv4," RFC 4728 (Experimental), IETF, 2007.

[5] V. D. Park and M. S. Corson, "A highly adaptive distributed routing algorithm for mobile wireless networks," in *Proceedings of the 16th Annual Joint Conference of the IEEE Computer and Communications Societies, Driving the Information Revolution (INFOCOM '97)*, p. 1405, 1997.

[6] C. E. Perkins and P. Bhagwat, "Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers," *ACM SIGCOMM Computer Communication Review*, vol. 24, no. 4, pp. 234–244, 1994.

[7] P. Jacquet, P. Mühlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot, "Optimized link state routing protocol for ad hoc networks," in *Proceedings of the IEEE International Multi Topic Conference*, pp. 62–68, 2001.

[8] S. Murthy and J. J. Garcia-Luna-Aceves, "An efficient routing protocol for wireless networks," *Mobile Networks and Applications*, vol. 1, no. 2, pp. 183–197, 1996.

[9] M. R. Pearlman and Z. J. Haas, "Determining the optimal configuration for the zone routing protocol," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, pp. 1395–1414, 1999.

[10] V. Ramasubramanian, Z. J. Haas, and E. G. Sirer, "SHARP: a hybrid adaptive routing protocol for mobile ad hoc networks," in *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '03)*, pp. 303–314, ACM, New York, NY, USA, June 2003.

[11] C.-K. Toh, "Associativity-based routing for ad hoc mobile networks," *Wireless Personal Communications*, vol. 4, no. 2, pp. 103–139, 1997.

[12] M. Spohn and J. J. Garcia-Luna-Aceves, "Neighborhood aware source routing," in *Proceedings of the 2nd ACM International Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc '01)*, pp. 11–21, 2001.

[13] I. D. Aron, S. K. S. Gupta, and E. K. S. Gupta, "Analytical comparison of local and end-to-end error recovery in reactive routing protocols for mobile ad hoc networks," in *Proceedings of the ACM Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, pp. 69–76, 2000.

[14] Z. J. Haas, M. R. Pearlman, and P. Samar, "The zone routing protocol (zrp) for ad hoc networks," draft-ietf-manet-zone-zrp-04.txt, 2002.

[15] M. Caesar, M. Castro, E. B. Nightingale, G. O'Shea, and A. Rowstron, "Virtual ring routing: network routing inspired by DHTs," in *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM '06)*, pp. 351–362, ACM, 2006.

[16] B. Chen and R. Morris, "Scalable landmark routing and address lookup for multihop wireless networks," Tech. Rep. 837, 2002, Scalable landmark routing and address lookup for multihop wireless networks,.

[17] P. Pei, M. Gerla, and X. Hong, "Lanmar: landmark routing for large scale wireless ad hoc networks with group mobility," in *Proceedings of the 1st ACM International Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc '00)*, pp. 11–18, IEEE Press, Piscataway, NJ, USA, 2000.

[18] R. Fonseca, S. Ratnasamy, D. Culler, S. Shenker, and I. Stoica, "Beacon vector routing: scalable point-to-point in wireless sensornets," Tech. Rep. IRB-TR-04-012, Intel Research, 2004.

[19] S. M. Das, H. Pucha, and Y. C. Hu, "Performance comparison of scalable location services for geographic ad hoc routing," in *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '05)*, vol. 2, pp. 1228–1239, March 2005.

[20] B. Karp and H. T. Kung, "Gpsr: greedy perimeter stateless routing for wireless networks," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MOBICOM '00)*, pp. 243–254, ACM, New York, NY, USA, August 2000.

[21] A. Y. Seydim, "Wormhole routing in parallel computers," 1998.

[22] T. H. Cormen, C. Stein, R. L. Rivest, and C. E. Leiserson, *Introduction to Algorithms*, McGraw-Hill Higher Education, 2001.

[23] D. Medhi and K. Ramasamy, *Network Routing: Algorithms, Protocols, and Architectures*, Morgan Kaufmann, San Francisco, Calif, USA, 2007.

[24] R. O. Saber, S. Richard, and R. M. Murray, "Consensus protocols for networks of dynamic agents," in *Proceedings of the 2003 American Control Conference*, June 2003.

[25] I. Broustis and M. Faloutsos, "Routing in vehicular networks: feasibility, modeling, and security," *International Journal of Vehicular Technology*, vol. 2008, Article ID 267513, 8 pages, 2008.

[26] B. Mohar, "The Laplacian spectrum of graphs," in *Graph Theory, Combinatorics, and Applications*, pp. 871–898, John Wiley & Sons, New York, NY, USA, 1991.

[27] A. Jorstad, D. DeMenthon, I.-J. Wang, and P. Burlina, "Distributed consensus on camera pose," *IEEE Transactions on Image Processing*, vol. 19, no. 9, pp. 2396–2407, 2010.

[28] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[29] B. Doerr, L. A. Goldberg, L. Minder, T. Sauerwald, and C. Scheideler, "Stabilizing consensus with the power of two choices," in *Proceedings of the 23rd ACM Symposium on Parallelism in Algorithms and Architectures (SPAA '11)*, pp. 149–158, June 2011.

[30] J. P. John, E. Katz-Bassett, A. Krishnamurthy, T. Anderson, and A. Venkataramani, "Consensus routing: the internet as a distributed system," in *Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation (NSDI '08)*, pp. 351–364, USENIX Association, Berkeley, Calif, USA, 2008.

[31] A. Vahdat and D. Becker, *Epidemic Routing for Partially-Connected Ad Hoc Networks*, 2000.

[32] Y. O. Yazir, R. Farahbod, A. Guitouni, S. Ganti, and Y. Coady, "Adaptive routing in mobile ad hoc networks based on decision aid approach," in *Proceedings of the 8th ACM International Symposium on Mobility Management and Wireless Access (MobiWac '10)*, pp. 1–10, October 2010.

[33] A. Fagiolini and A. Bicchi, "On the robust synthesis of logical consensus algorithms for distributed intrusion detection," *Automatica*, vol. 49, no. 8, pp. 2339–2350, 2013.

[34] F. Kuhn, R. Oshman, and Y. Moses, "Coordinated consensus in dynamic networks," in *Proceedings of the 30th Annual ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing (PODC '11)*, pp. 1–10, 2011.

[35] A. Benchi, P. Launay, and F. Guidec, "Solving consensus in opportunistic networks," in *Proceedings of the 16th International Conference on Distributed Computing and Networking (ICDCN '15)*, Goa, India, January 2015.

[36] C. Peng and Q.-L. Han, "A novel event-triggered transmission scheme and L_2 control co-design for sampled-data control systems," *IEEE Transactions on Automatic Control*, vol. 58, no. 10, pp. 2620–2626, 2013.

[37] C. Peng and J. Zhang, "Event-triggered output-feedback $\mathscr{H}_\infty$ control for networked control systems with time-varying sampling," *IET Control Theory & Applications*, 2015.

[38] J. Macfarlane, *Network Routing Basics*, John Wiley & Sons, New York, NY, USA, 2005.

[39] M. Arellano-Vàzquez, H. Benítez-Pérez, and J. L. Ortega-Arjona, "Study of routing algorithms considering real time restrictions using a connectivity function," in *Towards Autonomous Robotic Systems: 12th Annual Conference, TAROS 2011, Sheffield, UK, August 31-September 2, 2011. Proceedings*, vol. 6856 of *Lecture Notes in Computer Science*, pp. 412–413, Springer, Berlin, Germany, 2011.

[40] G. R. Latif-Shabgahi, "A novel algorithm for weighted average voting used in fault tolerant computing systems," *Microprocessors and Microsystems*, vol. 28, no. 7, pp. 357–361, 2004.

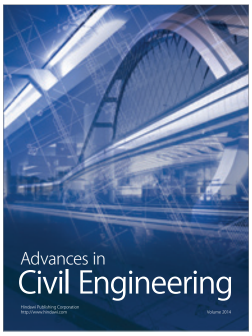
Journal of
Engineering

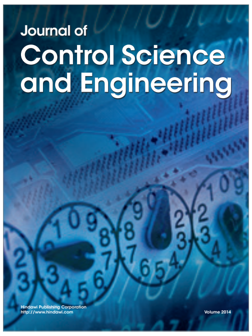
The Scientific World Journal

International Journal of
Rotating Machinery

Journal of
Sensors

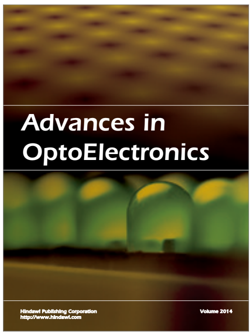
International Journal of
Distributed Sensor Networks


Advances in
Civil Engineering


Journal of
Control Science and Engineering

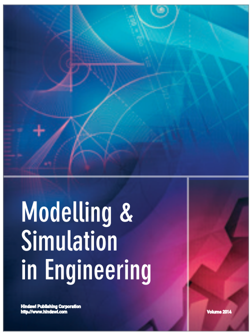
Journal of
Robotics

Journal of
Electrical and Computer Engineering

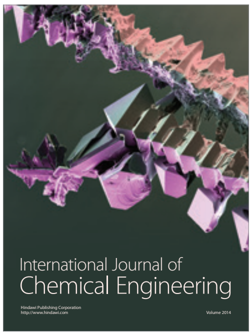

Advances in
OptoElectronics

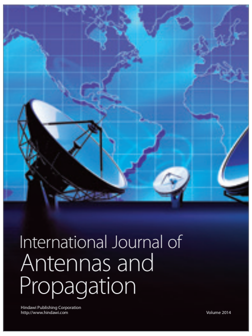
VLSI Design

International Journal of
Navigation and Observation

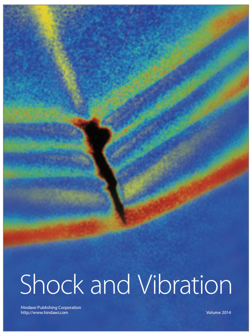

Modelling & Simulation in Engineering

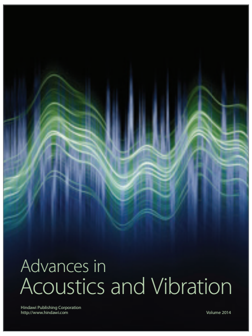
International Journal of
Aerospace Engineering


International Journal of
Chemical Engineering


International Journal of
Antennas and Propagation

Active and Passive Electronic Components


Shock and Vibration


Advances in
Acoustics and Vibration