HRAN: Heat Routing Protocol for Ad-Hoc Networks

João Trindade[ψ], Teresa Vazão[Ψ]

INESC-ID / Department of Computer Science and Engineering, Instituto Superior Técnico
Av. Prof. Dr. Cavaco Silva, 2744-016 Porto Salvo, Portugal;
Email: [ψ] jtrindade@tagus.inesc-id.pt [Ψ] tvazao@tagus.inesc-id.pt

***Abstract*—In this paper we propose a new routing protocol for large scale Mobile Adhoc Networks, named HRAN. The protocol supports large networks, requires few resources from devices and lowers the control message overhead associated with discovering new routes. It works by simulating the paradigm of heat trails in a physical environment through the use of bloom filters. In our protocol each node emits a certain quantity of heat and as the node moves a heat trail is formed. After a node leaves a location the heat from surrounding nodes slowly dissipates. This heat information is then used to guide routing queries from the source to the destination.**

***Index Terms*—Routing Protocol, Manet, Bloom Filter**

I. Introduction

In recent years the proliferation of wireless devices lead to emergence of Mobile Ad-hoc NETworks (MANET). These type of networks are becoming more ubiquitous in every day life and are used by an increasing number of people in different application environments. Some examples include the use on emergency and rescue situations where it is essential to provide network connectivity amongst participating actors, special when there is no infra-structured network available or operating in the proper conditions. They can also be used to provide communication to moving entities, as for example are cars, trains, pedestrians or even nodes equipped with sensing devices.

MANETs have specific characteristics which are not found in other types of network. The more relevant are the limited lifetime of the node's batteries and the frequent variation of both network topology and signal propagation. Hence, MANET networks pose some novel research problems which are being addressed by the scientific community [1] [2]. From the different MANET components that are currently under research, one of the most challenging is routing because traditional protocols have not been designed to cope with frequent and unpredictable topology variations. There is a wide variety of routing protocols which were developed to respond to these problems. The Destination-Sequence Distance Vector (DSDV) or the Optimized Link State Routing protocol (OLSR) are a direct evolution of routing protocols that were designed for wired networks, whilst others, as the Zone Routing Protocol (ZRP) or the Fisheye Routing Protocols (FSR), were designed using new concepts specifically suited for MANET environments. Nevertheless, as shown in several studies [3] [4] no protocol is able to provide consistent good results in different scenarios.

In this work we propose a new routing protocol named Heat Routing for Ad-hoc Networks (HRAN) suited for MANET scenarios. HRAN is scalable to large networks, in which the nodes have limited resources (memory, processing power and bandwidth), as its execution relies on few control messages each one having a small size. HRAN differs from other routing protocols developed for MANETs as it seizes the small size and versatility of bloom filters to store and spread topology information. In order to lower the routing overhead, HRAN nodes do not explicitly rebroadcast topology information messages coming from other nodes. Instead, this information is merged by each node with its own topology data and only the result is shared. Also to reduce the number of messages sent by a node, whenever a route to a destination is requested, HRAN does not flood the network but uses the information present in the node's bloom filters to discover what nodes are potentially useful for the route. This process effectively guides the query to the destination reducing unnecessary message transmissions.

The remainder of this article is structured as follows: Section 2 presents the details of the proposed new routing protocol. Section 3 presents tests regarding the proposal. Finally, section 4 draws the conclusions and lists future work.

II. The Heat Routing Procol for Ad-hoc Networks

HRAN mimics the paradigm of heat trails in a physical environment mapping it to a network topology. In our protocol, each node emits a heat signal to the group of nodes localized in its perimeter. This signal is used to spread the knowledge that each contains regarding the network topology: one-hop neighbours are hotter than two-hop neighbours and unknown neighbours do not have any heat gradient.

This paradigm is depicted on Figure 1, where the various gradients of heat information regarding solely node D are represented by the shades of grey without stripes. When a route is established, there is a creation of a heat tunnel that indicates that the nodes know how to reach each other. The heat tunnel between nodes A and D is represented by the grey fields with stripes: darker grey corresponds to the route found and lighter grey to near possible routes.

Whenever a route to an unknown destination is requested, a random query is launched until an heat trail is found. This is the case of node B that needs to send traffic to node D, but does not have any heat information representing a possible route to it. Thus, node B node launches a random query and this query continues until it finds a section which has heat information of node D.

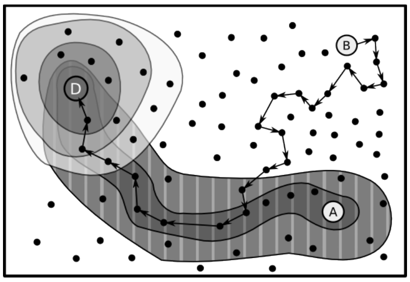


Figure 1: Concept of HRAN

At this stage the random query is replaced by a directed query

All heat trails are represented by bloom filters, having gradients representing the distance in number of hops to a node. These gradients disappear with time if not refreshed periodically. Also, as bloom filters are merged between neighbour nodes, topology information is spread with an extremely small overhead. Notice that the size of the used bloom filters and transmitted control messages never changes, even as networks scale regarding the number of nodes.

The Heat Route for Ad-hoc networks (HRAN) protocol execution is divided into three stages: heat overlay construction, route discovery and route maintenance. In following paragraphs each one of these stages will be detailed, but first an explanation regarding a new type of bloom filters necessary for HRAN is presented.

A. Time Aware Bloom Filter

The HRAN routing protocol requires the extension of the concept of bloom filter in order to check if a set of bits present in the structure was updated in a predetermined amount of time. This extension enables nodes to gradually lose heat information of its neighbours if this information was not updated during a predefined period of time. In order to achieve this functionality, we propose a new type of bloom filter, the Time Aware Bloom (TAB) filter. This new structure maintains the property which states that false positives may occur but enables the removal of items which have not been updated during a predefined time period.

The TAB filter is a tuple of two bloom filters, with size n and a group of k hash functions. These structures are named the *sticky* and *plain* filter. The *sticky* filter is used to identify whether a bit was updated in the last δ time interval or not, whilst the *plain* bloom filter contains the bits present in the current plain filter plus the bits set to one in the previous δ time interval.

To add a new element in the TAB filter, the algorithm performs the default bloom filter operation on the *plain* and *sticky* filters. This consists on hashing the inserted element with the k hash functions group in order to get the corresponding array positions. The bit array positions are then set to 1 on both filters. To search if an element is in the TAB filter, a query is made only on the *plain* bloom filter. Once again this operation is similar to the default bloom filter lookup function. The element value is hashed by the group of hash functions and the respective bits are checked. If all bits are set to value 1 the TAB filter returns a positive value indicating the element is present in the structure.

Finally, at regular δ time intervals, the *plain* filter is erased and its content is then overwritten by the *sticky* bloom filter bit array. After this operation is completed all the *sticky* filter positions are reset to 0 value in order to indicate a new δ time interval.

B. Overlay Construction

During the overlay construction stage each node spreads its topology information to the neighbour nodes, using a pro-active approach. In order to achieve this the protocol uses an array of TAB filters, with size N, called E vector. N is a static configuration parameter that represents the level of heat gradients of the heat overlay. Node identifications present in E vector position's closer to 1 are "hotter" than those closer to index N.

The construction of the heat overlay, mapping the network topology, is created by each node sending HELLO messages at regular time intervals. Each one of these messages contains a vector with $N-1$ bloom filters because a node does not transmit the bloom filter for colder gradient it knows (index N). The bloom filters contained in the HELLO message are a copy of the *plain* filters present at the TAB filters at the sending node's E vector from position 1 to position $N-1$.

When a node receives a HELLO message a bitwise OR operation between the bloom filters present in the messages and its E vector is performed. The iterator in the E vector starts at position 2 while the index that iterates over the array in the message starts at position 1.

C. Route Discovery

The second stage of the HRAN routing protocol is named route discovery. This stage is responsible for discovering a valid route from the source to the desired destination.

When a source node requests a route to an unknown destination, it initiates a predetermined number of *random walk request* queries (RwREQ) which travel through the network. Upon receiving a RwREQ, the node analyses its E vector and checks if it contains the destination node's identification in any one of the *layers*. If this is not the case, the random walk continues to a next neighbour node chosen randomly. If a match is found, the random walk ends and a directed walk is started by sending a *follow heat* (FoHEAT) message. FoHEAT messages are only forwarded by nodes that also have the destination node in any of the E vector positions.

The route discovery mechanism enables the query to rapidly reach the destination with few messages exchanged because only nodes with information of the destination node's heat rebroadcast the query. Nodes do not send twice the same message, as each FoHEAT message contains a unique identifier composed by the route requester id and a sequence number. This way a FoHEAT message is not transmitted if a node has already delivered a message with the same identifier.

Once the RwREQ query reaches the destination node, the route is given by the list of nodes present in the path log of the random walk. The destination then sends a *route reply* (RoREP) to the source using the inverted route discovered. If the source node, after sending its RwREQ message receives no answer for a predefined amount of time, then it falls backs to a reactive source routing protocol (like DSR). This backup procedure is performed in order to ensure the creation of heat tunnels which following queries will be able to use for reaching the same destination.

D. Route Maintenance

Route maintenance stage includes three different tasks: heat tunnel creation, route repair and route improvement. The first one aims at creating and maintaining an heat tunnel when a route is being used to transfer data packets. This process helps future route discoveries to quickly find the destination node and supports the route repair stage. Finally, route improvement allows the HRAN protocol to iteratively improve a route as it is being used.

When a data packet is routed by an intermediate node, a heat tunnel is created or updated with the identifier of the destination. This is accomplished by inserting the destination nodes unique identifier in the respective E vector of the intermediate nodes. This operation maintains heat tunnels information on the TAB filters for routes which are actively being used. Inactive routes dissipate with time as their respective heat tunnels information is not updated on the TAB filters.

When a route failure is detected by the origin endpoint of a communication, the previously created heat tunnel is used to discover a new route. A *route repair* (RoREPAIR) message is broadcasted until it reaches the i-hop neighbours (usually, $i = 2$). Every node that receives this message and has in its E vector positions $0 \leq i$ rebroadcasts the message. When the destination receives a RoREPAIR message, it sends a RoREP message to the source using the inverted route discovered. If the source does not receive a RoREP message during a predetermined time period, it performs the same operation, increasing the value of i. This is repeated until i value surpasses the value of E array positions $(N-1)$, at which time the source node considers the route to be irreparable and falls back to the normal route discovery mechanism to find the new route. Through this procedure, route repairs are achieved by incrementally flooding the various levels of the heat tunnel.

Due to the random process of route discovery, in the HRAN protocol it is not guaranteed that discovered routes are the best paths in terms of number of hops. To improve the routes HRAN uses a process which enables the iterative and gradual approximation of the discovered routes to the optimal ones. Also, as nodes move there is the possibility that a potential better route between the source and the destination is created.

HRAN's route improvement process uses the heat tunnel described previously to discover new and faster routes. When a route is being used, HRAN keeps a counter of the number of packets sent by the source node and when a threshold value is reached the source node sends a *route improvement* (RoIMP) message containing the destination ID. Each node which receives this message and has in its E vector the destination, rebroadcasts it. This process happens recursively until the destination node receives the RoIMP message. When this happens a RoREP message is transmitted over the inverse path. The source node uses the path reported by RoREP as the new route. As FoHEAT messages, RoIMP messages contain an unique identifier in order to prevent messages being sent twice by the same node. By using this mechanism RoIMP messages are only broadcasted inside the heat tunnel. Route improvement can therefore occur using a limited number of transmitted messages and only impacting nodes which are close to the path between the source and the destination.

III. Simulation Results

To validate HRAN protocol we used a simple and regular network based on a grid topology, in which the coverage area of each node encompasses the nodes of the diagonal. Three different networks with varying sizes were used: small network of 3x3 nodes; medium network of 6x6 nodes and large network with 10x10 nodes.

A. Amount of resources

HRAN was designed with the goal of offering a good performance in large scale networks by using a small amount of resources. The first simulations evaluates if this goal was achieved. We measured the number of messages considering a different number of route discovery processes occurring in the larger network, each one of them starting on a different node and all of them targeting the same destination. The expected minimum number of messages per node is achieved when each node in the path transmits one RoREQ and receives one RoREP for each one of the hops in the optimal path. Considering that each one of the 99 nodes have an active session to the chosen destination, the expected average minimum number of hop nodes of a route query is approximately 12.42. The results are shown on Figure 2 being the red line representative of the 12.42 threshold.

According to this figure, apart from the first route discovery which is not able to seize the information in any heat tunnels, HRAN reduces the number of control messages per request as the number of requests increases.

B. Route discovery time

In HRAN the way routes are discovered depends on the existence of heat tunnels located somewhere between the source and the destination nodes. The three type of strategies used to discover the routes are evaluated in the next tests. The results in Table I show the time necessary to establish a new route in these three conditions.

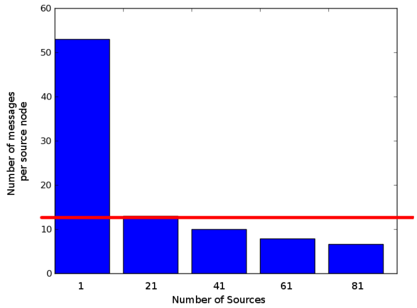


Figure 2: Amount of control packets

Network	No heat tunnel	Source inside heat tunnel	Source outside heat tunnel
Small	0.3s	0.3s	0.3s
Medium	3s	1.2s	1.3s
Large	4.8s	1.9s	2.2s

TABLE I: Route discovery time

When the first source triggers the route discovery there is no heat tunnel created and so a random query was launched, leading to the worst discovery time observed. When the second source repeats a similar process, an heat tunnel already exists in its way to the destination, hence only directed walks were used, reducing significantly the time needed to discover the route. Finally, when the third route discovery was triggered the source is outside the heat tunnel; it needs to use a random walk to reach the tunnel followed by a directed walk inside the tunnel to reach the destination.

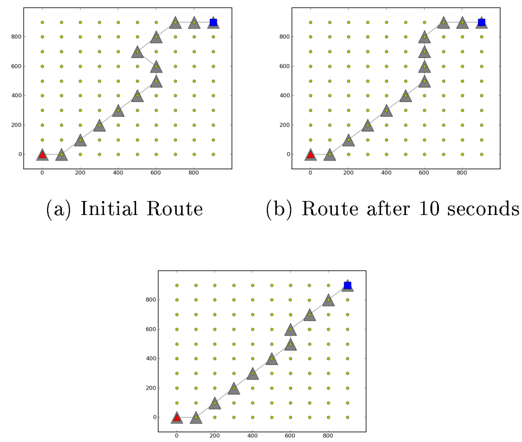
C. *Number of hops per route*

HRAN does not use either a location service to find out the node's position or floods the network to discover the destination node. Instead, it uses random queries that would probably lead to longer paths which are shortened by the route improvement. This test assess this mechanism. A single route discover process is triggered and the routes used were evaluated in different time instants. Figure 3 exemplifies the effect of the route improvement mechanism in a single simulation and Figure 4 characterizes the paths of the different networks. Their analysis shows that routes tend to the shortest path as time goes by, and the convergence time is kept under acceptable values, being smaller than the expected duration of an user session.

IV. Conclusion

This paper proposes a novel routing protocol designed for large scale MANET that relies on the concept of heat tunnels to discover and maintain active routes. The HRAN protocol spreads topology information across nodes by using bloom filters. These structures are small in size and capable of storing a large number of binary information. As the proposed algorithm requires the removal of values from these structures if they are not updated during a certain time, the time aware bloom (TAB) filter was developed. Simulation results demonstrated a case scenario where HRAN considerably reduced the number of routing messages necessary to find a route, whilst keeping the time to discover new routes acceptable and the number of hops almost optimum.

As future work we intend to dynamically adjust parameters to improve the random walk performance without introducing significant complexity in the protocol. Also new mechanisms which enable a trade-off between the number transmitted messages inside a heat tunnel and route discovery effectiveness will be studied.



(a) Initial Route (b) Route after 10 seconds

(c) Route after 40 seconds

Figure 3: Route improvement

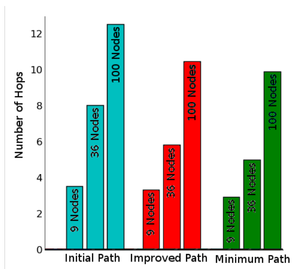


Figure 4: Number hops per route

References

[1] Nurul I. Sarkar and Wilford G. Lol, "A study of manet routing protocols: Joint node density, packet length and mobility," in *Computers and Communications (ISCC), 2010 IEEE Symposium on*, June 2010, pp. 515 –520.

[2] J. Kumar and R. Rajesh, "Performance analysis of manet routing protocols in different mobility models," *IJCSNS International Journal of Computer Science and Network Security*, vol. 9, no. 2, pp. 22–29, Feb. 2009.

[3] Kwan wu Chin, John Judge, Aidan Williams, and Roger Kermode, "Implementation experience with manet routing protocols," *ACM SIGCOMM Computer Communications Review*, vol. 32, pp. 49–59, 2002.

[4] S. Mohseni, R. Hassan, A. Patel, and R. Razali, "Comparative review study of reactive and proactive routing protocols in manets," in *Digital Ecosystems and Technologies (DEST), 2010 4th IEEE International Conference on*, April 2010, pp. 304 –309.