Controlling Route Discovery for Efficient Routing in Resource-constrained Sensor Networks

Abhik Banerjee, Juki Wirawan Tantra, Chuan Heng Foh, Chai Kiat Yeo, and Bu Sung Lee
Centre for Multimedia and Network Technology
School of Computer Engineering
Nanyang Technological University
Singapore
Email: abhi0018@ntu.edu.sg, jw.tantra@ieee.org, aschfoh@ntu.edu.sg, asckyeo@ntu.edu.sg, ebslee@ntu.edu.sg

***Abstract*— Existing ad-hoc network routing strategies base their operations on flooding route requests throughout the network and choosing the shortest path thereafter. However, this typically results in a large number of unnecessary transmissions, which could be expensive for resource-constrained nodes such as those in a sensor network. In this paper, we propose a new mechanism HopAlert which optimizes route establishment and packet routing by limiting the number of nodes taking part in the route discovery process while achieving a low number of hops establishment. Using analysis and simulations, we show that this results in more routes with shorter hop counts than a reactive flooding protocol such as AODV while achieving higher savings.**

***Index Terms*— Sensor Networks, Routing, Broadcasting**

I. Introduction

Routing techniques currently in use in ad hoc networks try to obtain the shortest path between the source and the destination in terms of the number of hops. In order to decide on a route, a source engages in a route discovery mechanism which involves flooding route request messages across the network before the route to the destination is obtained.

However, as has been pointed out in [1], the drawbacks of broadcasting make it inefficient for implementation in large networks. As all nodes involved in route discovery rebroadcast the messages, a majority of the packets transmitted are redundant. A large number of transmissions result in additional contention overheads and increase in collision frequency. Alternative broadcast schemes seek to limit the number of messages being transmitted by allowing nodes to decide whether or not to forward. Williams and Camp categorized and compared several such schemes in [2].

Previous research work by Foh et al. in [4] showed that successive hop lengths for reactive routing protocols tend to be longer than the previous. Hence, as the number of hops increases, the successive forwarding nodes are more likely to be near the periphery of the transmission range of the broadcasting node. Based on this observation, we propose a mechanism in this paper which reduces the number of contending nodes while optimizing the number of hops. We propose the routing protocol HopAlert which improves the likelihood of nodes near the periphery to act as forwarding nodes. We show that this results in a higher percentage of routes with smaller hops and achieves greater savings.

II. HopAlert algorithm for sensor networks

For the HopAlert algorithm, we introduce a new packet type called AODV_RREQ_ALERT, which is meant to alert nodes to refrain from immediate broadcast, hence the name. A source initiates route discovery in the same manner as in AODV. However, after it transmits an AODV_RREQ, it also sends an additional AODV_RREQ_ALERT packet. This is sent at the MAC layer after a Short Interframe Space (SIFS) duration following the transmission of the AODV_RREQ and is transmitted at a lower transmission power to cover fewer nodes which are closer to the source. We refer to the coverage area of the alert packet as *Alert Range (AR)* and that of the route request as *Request Range (RR)*. Intuitively, $AR \leq RR$ with $AR = RR$ in pure AODV.

As in AODV, in the absence of any existing route, each node would broadcast the route request. However, in HopAlert, any node within the AR, on receiving the AODV_RREQ_ALERT, would defer broadcasting for a duration large enough for it to hear a rebroadcast from any other node within its range. This duration is predetermined in the implementation of the algorithm and kept constant across all nodes. If, during this duration, it receives another copy of the same AODV_RREQ, it drops its own copy of the packet. This would imply that there are nodes located farther away from itself which can act as the forwarding node. On the other hand, if there are no nodes located in the region between AR and RR, there would be no broadcasts of the same request during the waiting period. In this case, the nodes within AR would broadcast their own copies of AODV_RREQ once the wait duration times out. Any subsequent node which rebroadcasts the route request also sends an alert message in the same manner. Thus, for each hop, the forwarding node is likely to be towards the periphery of the RR. Therefore the number of hops is minimized.

The operation of a node within AR could be understood if we observe the operations at the MAC layer. Once a node receives a AODV_RREQ from a preceding node, it would enqueue it for rebroadcasting in the absence of an existing route. Thereafter, it has to sense that the channel is free for the duration of a DIFS before it attempts to transmit. As the AODV_RREQ_ALERT is transmitted after a SIFS duration, it will be received by each node within the AR before each forwarding their own copies of AODV_RREQ. Hence the

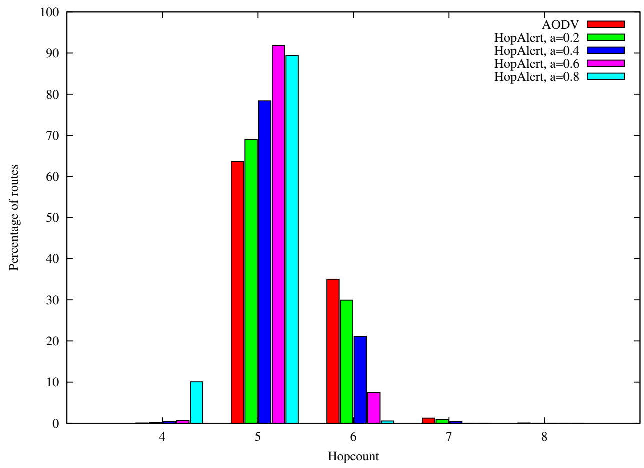

Fig. 1. Comparison of the percentage of routes generated with different hop counts for AODV to that of HopAlert with different values of a for a 1-D network with uniform placement of nodes.

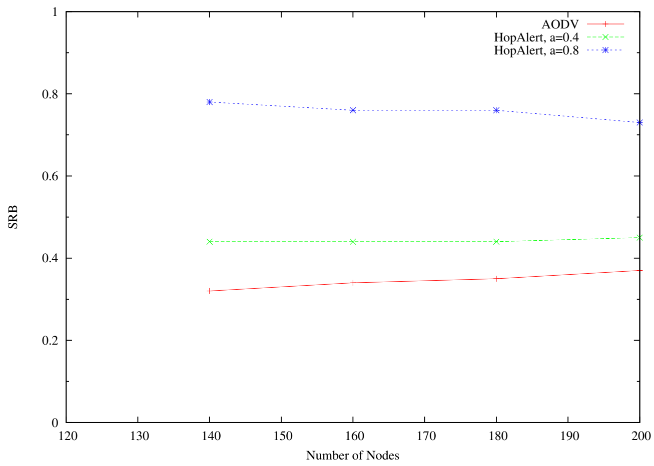

Fig. 2. Comparison of SRB for HopAlert with different values of a to that of AODV

nodes will refrain from doing so.

III. Simulation Results

We notice here that in high density networks, HopAlert is likely to result in much fewer nodes taking part in route discovery as compared to AODV. As each node is likely to have more neighbours, nodes which receive the alert message are likely to overhear forwarded route requests before their wait durations time out. Thus, only nodes which do not hear the alert take part in rebroadcasting resulting in an implicit lower bound on the length of each hop which in turn leads to lower hop counts.

To evaluate the performance of HopAlert algorithm, we compare it to AODV on the basis of the following performance metrics:

- *Hop Count*: We compare the percentage of routes with different hop counts to identify the strategy which yields more routes with fewer hops.
- *Saved ReBroadcast (SRB)*: As in [1], this is obtained as $\frac{(r-t)}{r}$, where r is the number of nodes receiving the broadcast message while t is the number of them who actually transmit it. Thus, this gives an estimate of the savings achieved.

We simulate both HopAlert as well as AODV on a 1-D network using the ns2 [6] network simulator. The normal transmission range (RR) over which AODV_RREQ is transmitted is set at 100m. The simulation is carried out by varying the AR. Due to lack of space, we only present the results for uniform placement of nodes. Random node placement gives similar performance. In case of uniform distribution, nodes are spaced 2m apart from each other. Thus, there are 50 nodes within the transmission range of any node in any particular direction. We consider a constant bit rate (CBR) packet stream with packet size of 256 bytes sent at intervals of 15 seconds. In the figures, AR is denoted by a normalized variable a such that $a = 1$ implies $AR = RR$.

The results for uniform random distribution of nodes is shown in Fig. 1. The comparison is done for a set of 200 nodes with the source and destination being the first and last nodes respectively. Hence, when nodes are distributed at equal distance, the route with minimum hop count would have 4 hops. As we can see, higher values of a result in a higher percentage of routes with shorter hop counts.

Fig. 2 compares the SRB ratio for HopAlert with $a = 0.8$ and $a = 0.4$ to that of AODV. As expected, the amount of savings is proportional to the value of a. Hence, higher values of a can result in higher savings which is desirable for a resource-constrained sensor network.

IV. Conclusion

In this paper, we propose a routing algorithm for sensor networks with resource-constrained nodes. The proposed algorithm HopAlert actively tries to reduce the number of nodes taking part in the route discovery phase by making use of an alert message which restricts some nodes from participating in route request forwarding. We show in our simulations that this helps to obtain paths with shorter hop counts while using up fewer transmissions.

Given the benefits observed from HopAlert, we would like to examine in detail its performance in different topologies and node distributions as part of our future work.

References

[1] S-Y. Ni, Y-C. Tseng, Y-S. Chen and J-P. Sheu, "The broadcast storm problem in a mobile ad hoc network," in *Proc. 5th Annu. Int. Conf. on Mobile Computing and Networking (MobiCom)*, 1999, pp. 151-162.

[2] B. Williams and T. Camp, "Comparison of Broadcasting Techniques for Mobile Ad Hoc Networks," in *Proc. 3rd ACM Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, 2002, pp. 194-205.

[3] Z. J. Haas, J. Y. Halpern and L. Li, "Gossip-Based Ad Hoc Routing," in *IEEE/ACM Trans. on Networking*, vol. 14, no. 3, Jun. 2006.

[4] C. H. Foh, J.W. Tantra, C. Jianfei, C. T. Lau and C. P. Fu, "Modeling Hop Length Distributions for Reactive Routing Protocols in One Dimensional MANETs," in *Proc. IEEE Int. Conf. on Communications (ICC)*, Jun. 2007, pp. 3882-3886.

[5] C. Perkins, E. Belding-Royer, S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing," Jul. 2003. Available: http://www.ietf.org/rfc/rfc3561.txt

[6] The UCB/LBNL/VINT Network Simulator-ns (Version 2). VINT Project. [Online]. Available: http://www.isi.edu/nsnam/ns