Holmes: Probabilistic Object Location in MANET

Irfan Sheriff, Prashanth Acharya, Ashwin Sampath, Elizabeth Belding-Royer, Ben Zhao
Department of Computer Science
University of California at Santa Barbara
{isheriff, acharya, ashwins, ebelding, ravenben}@cs.ucsb.edu

Internet protocols have moved from host centricity, which involves looking for a specific host, to data centricity, which involves looking for specific data. Data location in MANET is incipient. The unstructured and dynamic nature of MANETs makes object location a difficult problem. Structured object location techniques for the Internet, such as Tapestry, Pastry, etc., do not apply well to ad hoc networks because the underlying routing layer is unstable and the protocols are too heavyweight for a bandwidth contrained environment. Most techniques proposed for MANET are broadcast based techniques [1] [2].

Tchakarov et.al.[3] proposed a technique in which the server publishes the object information at nodes in four geographic directions. A search for an object is initiated in all four directions and the intersection with a node containing published information locates the object. The drawback of this approach is that it requires frequent publishing of every object which leads to significant traffic overhead with increase in scale of objects. Another drawback is the assumption of GPS information at each node.

In this poster, we describe *Holmes*, a lightweight location aware probabilistic approach to find data without the use of any location aided devices such as GPS. Object information at nodes is summarized into a single string using the bloom filter technique and piggybacked on route control messages. The protocol binds the object discovery operation to route discovery. Each operation aids in building distributed information. *Holmes* allows a node to efficiently determine whether the desired object is stored on a node with known routing information. The idea is that if the object resides on a node for which there is no route, then a route discovery broadcast must be performed. In this situation, the object query imposes no additional cost. The bloomstring information at each node helps to predict the existence of an object with a certain probability.

Bloom filter[4] is a technique of summarizing information using hash functions in a smaller, fixed-size string. Object lookup can be performed with a certain probability of success. The size of the bloomstring m and the expected probability of success k determines the number of objects that can be hashed. We apply the bloom filter technique to summarize the object information at a given node. This information is piggybacked on the route control messages and updated in the routing table of the nodes in the network. The bloomstring is associated with the IP address of the node. No explicit packets are needed to publish object information in the network.

Object lookup is done as follows. The given object O is hashed into all the bloomstrings in the routing table and a candidate list is compiled, representing the set of nodes that might contain the object. The hop count is used to select a set of nodes from the candidate list. These nodes are contacted through unicast packets to check the presence of the object. If the requests fail at all the nodes, we resort to contacting the next set of nodes from the candidate list or doing a broadcast based lookup, in the worst case.

The idea of piggybacking object information on routing protocols is not new. However, bloom filter based summarization of objects reduces the overhead greatly in a bandwidth constrained environment. Object replication in the network necessitates only a small probability of success per node because a set of servers is contacted through simultaneous unicast packets.

Simulations were performed using Qualnet for scenarios with varying network size, objects per node and replication per object. It is observed that an additional 256 bits per routing control packet reduces the overhead traffic by 50% over broadcast based lookup for 100 nodes, 50 objects per node and replication factor of 5. Replication of objects in the network reduces the overhead further. This is due to the increase in the number of candidate nodes per look up and hence an increase in the probability of success.

REFERENCES

[1] W. Ma, B. Wu, W. Zhang, and L. Cheng, "Implementation of a lightweight service advertisement and discovery protocol for mobile ad hoc networks," in *Proceedings of GLOBECOM*, San Francisco, CA, December 2003.

[2] S. K. Goel, M. Singh, and D. Xu, "Efficient peer-to-peer data dissemination in mobile ad-hoc networks," in *Proceedings of ICPPW*, Vancouver, Canada, August 2002.

[3] J. B. Tchakarov and N. H. Vaidya, "Efficient content location in wireless ad hoc networks," in *Proceedings of MDM*, Berkeley, CA, January 2004.

[4] B. H. Bloom, "Space/time trade-offs in hash coding with allowable errors," *j-CACM*, vol. 13, no. 7, pp. 422–426, July 1970.