GEOGRAPHICALLY STATIC QUORUMS IN AD-HOC NETWORKS AND THEIR PERFORMANCE AS LOCATION SERVERS

Gareth Owen and Mo Adda
School of Computing, University of Portsmouth
Buckingham Building, Lion Terrace, Portsmouth
England, PO1 3HE
{gareth.owen, mo.adda}@port.ac.uk

Abstract
There are many essential applications for quorum systems in ad-hoc networks, such as that of location servers in large-scale networks. Existing research proposes many approaches to the problems, many of which are incomplete, cumbersome, or incur significant cost. We then examine the performance of the quorum we described in previous work as a location server and suggest improvements to the query mechanism and routing algorithm.

Keywords
Ad-hoc networks, quorum systems, self-organising systems.

1. INTRODUCTION

Ad-hoc networks are a means of networking computing devices together without requiring any setup or existing infrastructure. We define an ad hoc network as a graph of a set of nodes, *V*, and a set of communication paths, *E*, that vary through time as nodes move and fail.

$$G_t = (V_t, E_t) \qquad (1)$$

Multi-hop ad-hoc networks allow nodes to communicate with each other without being in transmission range by relaying their messages through intermediate nodes. Routing in small networks is usually achieved through an optimal broadcast mechanism; however, when the network becomes large (>200 nodes), this form of discovery becomes extremely expensive.

Routing in large-scale ad hoc networks can be achieved through using location information from a GPS device. Hou[1] describes Most-Forward-within-Radius (MFR) whereby nodes are aware of their geographical position (from GPS) and progressively route packets closer to the location of the destination. Before a node can route packets in this fashion, it will need to know an accurate location for the destination. Many routing protocols propose a function that maps a node's ID to its location, but whilst this is ideal when networks are static and such a function can be defined, it is not when nodes are mobile and networks span many kilometres. Location servers solve this problem by maintaining location information of a node that sends it frequent updates.

Several types of location server have been proposed to provide this information. Li[1] divides the area into grid-squares of different orders. Starting with the grid-square that the node currently occupies, an order-one square, an order-two square is the grid-square containing four of the order-one squares. Li proposes that a number of nodes in each order-n square should host the location information. Therefore, information about a node's location becomes less densely available the further one is from its location.

Giordano proposed[2] the most promising approach to location servers by allocating nodes within a specific radius of a geographical point to serve as a location server. Their work suggested modifying the radius depending upon the node density so that a minimum number of nodes participate; however, they did not address issues such as node mobility, fault tolerance and query success.

Finally, Stojmenovic proposed[3] that all nodes to the north and south in a fixed-width column assume the role of a location server. Location searches are performed by routing packets horizontally so that they will eventually intersect the vertical column. This approach is not feasible for large-scale ad hoc networks because of the large number of nodes. For a more complete review of approaches to location servers, the reader is referred to [4].

The approach shown in this paper is most similar to Giordano's work on home-regions but we further develop the quorum system we proposed in [5] by analysing its performance as a location server. Methods of querying the quorum and ways of mitigating the problem of incomplete quorum updates, along with a method to improve routing success in the face of out-dated information are investigated.

2. QUORUM SYSTEMS IN AD HOC NETWORKS

We adopt the quorum system that we described in our previous works [5, 6] whereby a self-organising approach is used to maintain a replicated state machine at a specific geographical point. The quorum consists of a master responsible for replication and a number of slaves who are able to adopt the master's role if it fails. All the components are able to move from node to node to maintain themselves as close as possible to the geographic point. In our work, we showed that not only is it possible for the quorum to stay within close proximity to the point, but also that the quorum survives exceptionally well yet incurring an overhead of 10 packets or less per minute.

In this paper, we describe how this quorum system can be used as a location server to support routing in large-scale ad-hoc networks. Our previous work said that at least one of the components must be within the minimum transmission range of the point so as to avoid a multi-hop search. If this were not the case then we would incur the extra overhead (C) of a search, which can be described given the average node degree (d) and the search radius in hops (n), of the order shown in equation 1.

$$C = \mathrm{O}(d^{n}) \tag{2}$$

The Section 2.1 describes how the quorum is queried with possible improvements and section 2.2 presents a method for improving routing success in the face of out-dated location information.

2.1 Query and update mechanisms

The mechanism to update the quorum is simple and based upon a zero knowledge approach to minimise overhead. When receiving a new update packet, a node simply updates itself if it holds a component and then rebroadcasts it. This way, each element of a quorum should receive a copy and the number of overhead packets will be equal to the number of components present ($q_{component}$) as shown in equation 3.

$$C_{update} = h + q_{components} \tag{3}$$

This approach does not guarantee a complete quorum update however due to its self-organising approach. Therefore, we examine the age of the data returned and propose a technique that we call Multi-query (MQuery) to try to reduce the likelihood of retrieving old data. MQuery requires every quorum query to attempt to query at least two quorum components and take the most up-to-date result. Simply, if a query is received first by a component, then the packet will be routed to the next closer node to the home location. Hopefully, this node will also be a participant in the quorum and therefore could have more correct data. In our results, we examine the effect this has on the age of data.

RECVUPDATEPACKET(p)
$c \leftarrow$ Component(p_{nodeID})
if not exist(c) **or** seenBefore(p) **then** ignore
if $p_{revision} \leq c_{revision}$ **then** ignore
$c_{data} \leftarrow p_{data}$
broadcast(p)

Quorum systems are traditionally analysed based upon their performance in terms of load[7], fault-tolerance [8] and failure probability [9]. Therefore, we measure the survivability or failure probability of the quorum over 30-minutes and the overhead in packets per minute of one quorum.

In addition to analysing the performance as a quorum, we need to look at the ability of this approach to act as a location server for large-scale ad hoc networks. The metrics we will measure are update cost, query success, age of results and routing success.

2.2 DSR-aided MFR to improve routing success

It is possible that the location information received could be out-of-date and so we analyse the effect this has on a routing and ways to improve it. For comparison, we use the MFR approach used in RoutePacket(*p*) and compare it with the technique proposed here. It is worth noting that several other techniques have been examined to improve routing success (recursive search[10], GRA[11], and a planar sub-graph[12]); however, these do not examine success in the face of incorrect location information. Here, we propose a technique to handle not only small voids but also out-of-date location information.

We also propose that if a packet reaches a node in which the MFR algorithm is unable to recover, then the node initiates a DSR[13] search (with limited TTL) to find a node that is closer. Upon finding one, the packet is source routed there and then resumes using MFR to continue to the destination. If the DSR discovered path becomes invalid before the packet is routed completely, then the node will reinitiate discovery. This is described in the DsrMfrRoutePacket function below.

```
DSRMFRROUTEPACKET(p)
if not routePacket(p) then
    if routes.exist(p_location) then
        use existing route
    else
        initiateDSRdiscovery(p_location, p_dest)
        queuePacket(p)
    endif
```

A density of 200 nodes per square kilometre with 140.5m TX range almost guarantees connectivity. Therefore, we set the TTL of the DSR search to three hops as we expect that routing failures will be due to a locally poor node arrangement rather than large voids. The TTL could be varied depending on the node density but we do not examine this here.

3. METHODOLOGY

Simulation is widely accepted as a means for analysing ad hoc networking protocols due to the mathematical complexity of the scenario. Here we use the GloMoSim (v2.03) simulator[14] that provides models of all the layers experienced in a real experiment. We configure the simulator with the parameters shown in Figure 1 to represent that of a kilometre-squaed section of a large-scale ad-hoc network.

Figure 1: Simulation parameters

Parameter	Value	Parameter	Value
Terrain	1000x1000	**Number of nodes**	200
Propagation Model	Two-ray	**Number of quorums**	50
Tx/Rx Range	140.5m	***threshold*** **(number of slaves)**	5
Mobility Model	Random Waypoint v_{min} = 0.1m/s v_{max} = 3.0m/s	**Beacon rate**	Every 5 seconds + jitter
Simulation-time	30 minutes	**Manage Component call rate**	Every 6 seconds + jitter

4. RESULTS

Figure 2 shows the number of queries returned by the quorum that were out-of-date due to an incomplete update of all components. We then also show how this can be improved by querying at least two components and taking the most up-to-date result. This simple technique halves the number of old-results returned.

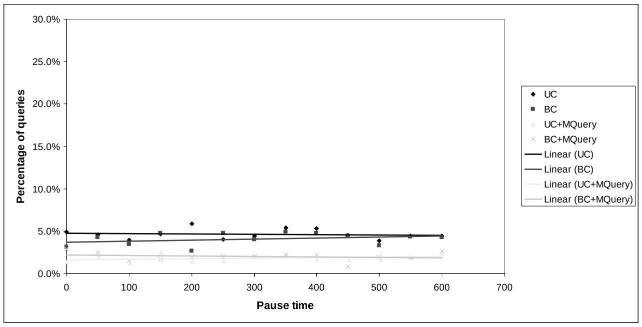

Figure 2: Out-of-date results returned

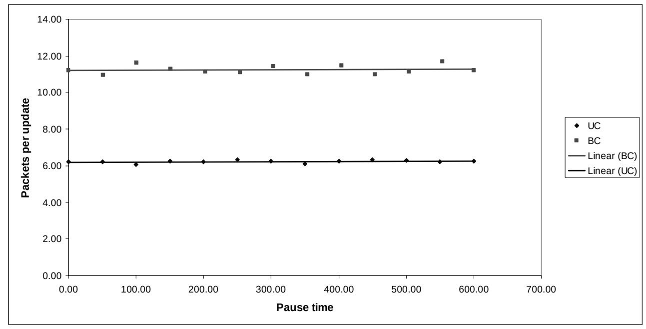

Figure 3: Update cost (packets)

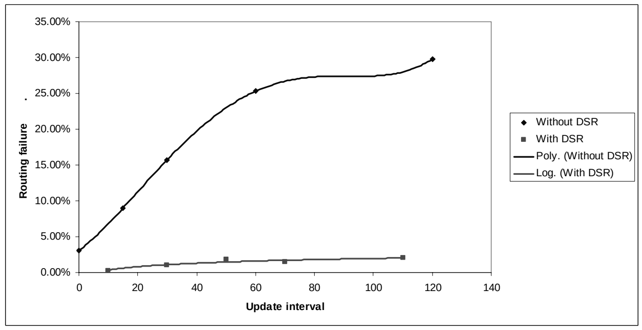

Figure 4: DSR-aided geodesic routing

The overhead incurred to fully update the quorum depends on the number of components present. Figure 3 clearly shows the unicast algorithm maintains the lower number of components while the broadcast version maintains almost twice as many.

Figure 4 shows the results obtained when the information from the quorum is out-of-date by varying degrees. With update intervals of two minutes, almost a third of packets do not reach their destination. However, when we use the DSR-aided MFR routing algorithm, less than 3% of packets fail to reach their destination.

5. CONCLUSION

Existing algorithms are not comparable with the approaches we have presented here, as they do not address node mobility and failure that are the key motivation to our work. Therefore, we present our results without comparison with them.

We found that approximately 4% of queries returned out of date results, but that this is halved by querying at least two components. Further improvements could be made if elements of the quorum communicate with each other, or revision information is added to the beacon packet.

We showed that by aiding the routing algorithm with DSR when the packet was unable to make any further progress forward, with a search radius of just three hops, increased the routing success by up to a factor of 10 even when the location information is up to two minutes old.

We demonstrated the performance of the quorum for obtaining location information and measured the percentage of out-of-date data returned. Location information does not necessarily have to be up-to-date as long as the node has not moved too far and the routing algorithm is able to recover. One way to improve geodesic routing algorithms to cope with incorrect information would be to aid the algorithm with a limited DSR search when the node is not at the given location. Undertaking this simple improvement increased the routing success by up to a factor of ten; however, this will of course incur a higher routing overhead.

References

[1] T.-C. Hou and V. Li, "Transmission range control in multihop packet radio networks," *IEEE Transactions on Communications*, vol. 34, pp. 38-44, 1986.

[2] S. Giordano and M. Hamdi, "Mobility Management: The Virtual Home Region," EPFL, Lausanne, Switzerland, 1999.
[3] I. Stojmenovic, "A scalable quorum based location update scheme for routing in ad hoc wireless networks," University of Ottawa, 1999.
[4] T. Camp, J. Boleng, and L. Wilcox, "Location Information Services in Mobile Ad Hoc Networks," presented at IEEE International Conference on Communications, 2002.
[5] G. H. Owen and M. Adda, "Quorum based geographically static data storage in ad-hoc networks," presented at International Network Conference, Plymouth, UK, 2006.
[6] G. H. Owen and M. Adda, "Feasibility of geographically static data storage in ad-hoc networks," presented at 2nd International Conference on Telecommunications and Computer Networks, Portsmouth, 2005.
[7] M. Naor and A. Wool, "The load, capacity and availability of quorum systems.," *SIAM Journal of Computing*, vol. 27, pp. 423-447, 1998.
[8] D. Barbara and H. Garcia-Molina, "The vulnerability of vote assignments," *ACM Transactions on Computer Systems*, vol. 4, pp. 187-213, 1986.
[9] D. Peleg and A. Wool, "The availability of quorum systems," *Information and Computation*, vol. 123, pp. 210-233, 1995.
[10] G. Finn, "Routing and addressing problems in large metropolitan-scale internetworks," USC/Information Sciences Institute, 1987.
[11] R. Jain, A. Puri, and R. Sengupta, "Geographical Routing Using Partial Information for Wireless Ad Hoc Networks," *IEEE Personal Communications*, vol. 8, pp. 48-57, 2001.
[12] B. Karp and H. T. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks," presented at Mobile computing and networking; MobiCom 2000, Boston, MA, 2000.
[13] S. Johnson, *Emergence: The Connected Lives of Ants, Brains, Cities and Software*: Scribner, 2001.
[14] X. Zeng, R. Bagrodia, and M. Gerla, "GloMoSim: a library for parallel simulation of large-scale wireless networks," presented at 12th Workshop of Parallel and Distributed Simulations - PADS '98, 1998.