

# Facilitating Match-Making Service in Ad hoc and Sensor Networks Using Pseudo Quorum

Ilknur Aydin Chien-Chung Shen  
Department of Computer and Information Sciences  
University of Delaware  
Newark, DE 19716  
{aydin, cshen}@cis.udel.edu

**Abstract—** In a match-making system, sources (producers) advertise generated data without any particular destination in mind. Destinations (consumers) are determined based on their interests (via subscriptions) in receiving the produced data. Advertisements and subscriptions are *matched* by the underlying network service. We propose to facilitate match-making capability in ad hoc and sensor networks by adapting the idea of quorum system. A quorum system is formed by organizing nodes into subsets called quorums, where every two quorums intersect and no quorum includes another quorum. To accommodate node mobility and network scale, we propose that producers and consumers systematically forward their advertisement and subscription messages to form ‘pseudo’ quorums, where they are matched at intersecting nodes. Simulation results shows that pseudo quorum based match-making system achieves a very high matching rate with much less messaging overhead as compared to that of event and query flooding.

## I. INTRODUCTION

A match-making system is different from traditional multi-cast and any-cast routing systems in the sense that routing is performed based on the data (contents) in the messages, rather than specialized address attached to the messages. Sources (producers) of data in the network generate messages and advertise them without any particular destination in mind. Destinations (consumers) are determined based on their interests (via subscriptions) in receiving the produced data. Advertisements from producers and subscriptions from consumers are effectively and efficiently *matched* by the underlying network routing system [1].

A match-making system is applicable to many different network types, application domains, and scenarios. For instance, in a battlefield, soldiers (producers) might advertise intelligence they have found out (*e.g.*, enemy camps, weapon depots, or bunkers around), while the commanders (consumers) state what they are interested in. A soldier may also become a consumer if he needs certain information (*e.g.*, any enemy tank less than 2 km away) or help (such as the closest medical equipment or medical care team), while other soldiers become the producers advertising intelligence and/or capability. Another applica-

tion scenario utilizes a sensor network to monitor hostile environments. For instance, deployed sensors are producers to advertise readings and events they have detected or observed, while some supervisory sensors act as consumers by asking various queries regarding the environment. The supervisory sensor nodes aggregate the information and inform the appropriate human supervisors (consumers) according to their interests. Our objective in this paper is to provide a match making system for both ad hoc networks and sensor networks, where nodes with limited power resources need to communicate without any pre-existing infrastructure. For ad hoc networks, node mobility causes dynamically changing topology. For the sensor networks, although nodes may not be mobile, the sheer number of nodes with limited resources poses unique challenges. Thus, a match-making mechanism for ad hoc and sensor networks should take all these stringent requirements into account and be *mobility-aware, adaptive, efficient, and scalable*.

We propose to *adapt* the notion of a *quorum system* to facilitate match-making capability in ad hoc and sensor networks. A quorum system is formed by organizing a collection of objects into subsets called *quorums*, where *every two* quorums intersect and no quorum includes another quorum. A formal mathematical definition is as follows. Given the finite set  $S = \{1, 2, \dots, n\}$  representing network nodes, a quorum system  $\{Q_1, Q_2, \dots, Q_m\}$  can be defined where each  $Q_i$  represents a quorum and  $Q_i \subset S$ , such that:

- Covering:  $\bigcup_{i=1}^m Q_i = S$
- Mutual intersection:  $Q_i \cap Q_j \neq \emptyset, 1 \leq i, j \leq m$  and  $i \neq j$
- Minimality:  $Q_i \not\subset Q_j, 1 \leq i, j \leq m$  and  $i \neq j$

A quorum-based match-making system works as follows. Nodes in the network are organized into quorums via a quorum construction algorithm, such as [2]. Each producer and consumer would create its own quorum and forward its advertisement and subscription messages only within the corresponding quorum. Since every two quorums intersect, producers and consumers will match at intersecting node(s).

However, distributed quorum construction algorithms are neither scalable for sensor networks, nor feasible for mobile ad hoc networks. To address these issues, we propose a systematic way for producers and consumers to forward their advertisement and subscription messages, respectively, and the corresponding forwarding nodes form ‘pseudo’ quorums. Match-making takes place at the nodes where two pseudo quorums

Prepared through collaborative participation in the Communications and Networks Consortium sponsored by the U. S. Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement DAAD19-01-2-0011. The U. S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon.

intersect. Note that, for our purpose we are interested in satisfying only *mutual intersection*. This is why we call the quorums formed by producer and consumers in the network as ‘pseudo’ quorums. Although the pseudo quorum based match-making system may not be perfect given the sheer number of nodes and their mobility, it is scalable and can achieve very high matching rate with minimized overhead as evidenced by our simulations.

The remainder of the paper is organized as follows. The next section first details a generic approach to pseudo quorum system construction, and then describes three optimization schemes. In Section III, we will describe the performance metrics, simulation set up, and scenarios, as well as the analysis of simulation results. Section IV describes other related work, and Section V concludes the paper with future work.

## II. OUR APPROACH

The following notations will be used in the rest of the paper.

- *ADV packets*: Advertisement packets sent by the producers to advertise the information they are producing.
- *SUB packets*: Subscription packets sent by the consumers to query about the information they are interested in.
- *REPLY packets*: Returned by the match-making system to a producer (or a consumer) to inform the producer (or the consumer) about the corresponding consumer (or producer if REPLY packet is sent to a consumer).

The aim of our mechanism is to match the corresponding producers and consumers without flooding the ADV/SUB packets to the entire network. This motivates a systematic way to store and forward the ADV/SUB packets via a small subset of nodes, and when there is a match, REPLY packets are sent back to the producers or consumers. The systematic approach constructs a pseudo quorum system which ensures very high matching confidence of the producers and the consumers.

The key idea is as follows. If we can make each producer/consumer node spread its ADV/SUB packets in the shape of a *cross* (i.e., ‘+’) throughout the entire network coverage, then we can ensure that every two ‘+’ shaped pseudo quorums intersect at some intermediate nodes. For instance, in Figure 1(a), it is shown that the two ‘+’ shaped pseudo quorums of producer P and consumer C intersect. In reality, nodes do not necessarily align the lines of a cross. To address this issue, the proposed pseudo quorum match-making mechanism adopts certain heuristics so that a ‘+’ shape is approximated, at its best, by the ADV/SUB packets sent by each producer/consumer according to the current network topology. For instance, Figure 1(b) shows the approximation of a ‘+’ shaped pseudo quorum of producer P for the given network topology.

### A. The Generic Algorithm

Our algorithm is based on the geographic location of the nodes; thus, we assume that each node is equipped with a GPS receiver or other positioning mechanisms. Each node has the following data structures and parameters.

- *AdvTable and SubTable*: Tables to keep track of the ADV/SUB packets forwarded by the node. Each entry of the table has the tuple  $[node, infoType, nodePos, sendingTime, forwardingNode, hopCount]$ , where *node* is the producer or consumer of the infoType, *infoType* is the

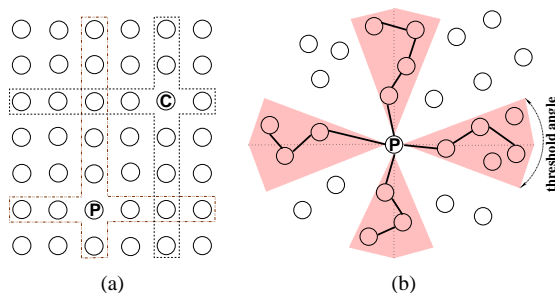


Fig. 1. (a) Nodes lying perfectly along the lines of grids allow perfectly formed ‘+’ shaped quorums of producer P and consumer C, and hence perfect intersection between the quorums. (b) Approximating the ‘+’ shaped quorum of producer P for the given network. The quorum is constructed out of the nodes lying in sectors along the four main directions as defined by the threshold angle.

data produced by the producer or queried by the consumer, *nodePos* is the position of the producer/consumer at sendingTime, *sendingTime* is the time producer/consumer sent the ADV/SUB packet, *forwardingNode* is the previous hop that forwarded the ADV/SUB packet (so that a *backward path* for the REPLY packets back to the producer/consumer can be set up), and *hopCount* is the number of hops of the path traveled so far by the ADV/SUB packet.

- *NTable*: Neighbor Table to keep track of the existence and most-up-to-date geographic positions of the neighbors. (Maintained by the periodic hello packets sent by every node.)
- *DrTable*: Detected Reply Table to keep track of the REPLY packets already forwarded by the node, to prevent flooding of REPLY packets back to the producers/consumers through the branches of the pseudo quorums.
- *Threshold Angle*: The angle given in degrees to limit the stray of the branches of the ‘+’ shaped pseudo quorums as well as the number of intermediate nodes forming the quorums. The bigger the threshold angle, the bigger the stray of the branches and the larger the number of nodes involved in the quorums.
- *InfoType*: ID of the information the producer is producing, or consumer is interested in. The node has InfoType only if it is a producer or consumer.
- *Update Interval*: At every update interval, a new ADV/SUB packet is created and sent by the producer/consumer (if nodes are mobile).
- *Update Lifetime*: Update lifetime is used to remove stale entries from Adv/Sub Tables.

The *generic* algorithm to construct and maintain ‘+’ shaped pseudo quorums is as follows.

#### 1) Bootstrapping of pseudo quorum construction process:

To start, a producer/consumer selects one neighbor from each of the four sectors along the main directions (east, west, north, and south) as defined by the threshold angle, prepares four different ADV/SUB packet by including the main direction the packet is to be sent through, and transmits packets to the selected neighbor in that direction. If there is node mobility, producer/consumer repeats the same process to *reinforce* the

branches of the pseudo quorums according to the current network topology for every update interval (Figure 1(b))

2) *Forwarding of ADV/SUB packets:* An intermediate node, upon receiving an ADV/SUB packet, first checks if it can store and forward the packet. The intermediate node can store and forward the packet only if it is within the sector along the main direction the packet is to be sent as defined by the threshold angle (Figure 1(b)).

The intermediate node also checks if it can satisfy any producers/consumers with the received packet. Let's say the intermediate node receives an ADV packet, it checks its SubTable to see if any of the consumers are looking for the InfoType included in the ADV packet. If so, it prepares a REPLY packet for each consumer<sup>1</sup>, includes the identity of the producer in the REPLY packets, and sends the packets along the corresponding backward paths to the consumers.

3) *Forwarding of REPLY packets:* If the intermediate node is the destination of a REPLY packet (*i.e.*, a producer/consumer itself), it stops forwarding the REPLY packet. Otherwise, the intermediate node forwards the REPLY packet to the next-hop in the backward path formed in the pseudo quorum construction process. Note that, in ad hoc networks node mobility may break the backward path. A 'geographic reply forwarding' mechanism is introduced to overcome this problem to be described in Section II-B below.

### B. Optimizations to the Generic Algorithm

To have better and more effective heuristics in approximating the '+' shaped quorums and to reduce the overhead, three optimizations to the generic algorithm are described in the following subsections.

1) *Selecting the furthest neighbor as the next-hop to forward an ADV/SUB packet.:* Each node A (either a producer/consumer or an intermediate node) selects the next-hop to be *the furthest neighbor* in the sector that is along the main direction the packet is supposed to be forwarded through as defined by the threshold angle. If there are more than one furthest neighbors having equal distance to node A, the neighbor closer to the main direction is selected. The selected furthest neighbor is included in the ADV/SUB packet, and then ADV/SUB is *broadcast*. Thus, while the packet is forwarded to the selected furthest neighbor, all the other overhearing neighbor nodes which are also in that sector can store the packet in their Adv or Sub Tables. This increases the chance of intersection between two pseudo quorums (because more nodes are storing the ADV/SUB packets), while decreasing the messaging overhead (because only the selected furthest node is responsible for forwarding the packet). (Figure 2(a))

2) *Geographic reply forwarding.:* This optimization is introduced to overcome the broken backward paths problem in the case of node mobility. Each intermediate node forwarding a REPLY packet to the next-hop in the backward path listens if the packet reaches the next-hop. If the REPLY packet can not reach the aimed next-hop, the intermediate node removes the aimed next-hop from its NTable and selects another neighbor,

<sup>1</sup>The intermediate node itself can be a consumer of the InfoType, in this case, no REPLY packet is prepared for it.

which is *geographically closest* to the currently-known position of the final destination node of the REPLY packet, to be the next-hop. The new next-hop receiving the geographically forwarded REPLY packet repeats the same procedure. There is a Time-to-Live value included in the REPLY packet, which is equal to the length of the remaining backward path to the final destination of the REPLY packet, to prevent the forwarding of the REPLY packet forever if the REPLY packet can not reach the final destination node. (Figure 2(b))

3) *Detecting multiple REPLY packets.:* In the first optimization described above, there could be more nodes that can send a REPLY packet for the same query. Therefore, to prevent flooding of REPLY packets along the branches of the quorums, we introduce a REPLY detection mechanism, where interior nodes forward the *first* REPLY packet they receive for a query and ignore the other REPLY packets for the same query. (Figure 2(c))

## III. PERFORMANCE EVALUATION

The described pseudo quorum based match-making algorithm is evaluated by using two performance metrics:

- *Total Number of Transmissions (TNT)* – the total number of ADV and SUB packets transmitted to construct the pseudo quorums for all the consumer and producers in the system *plus* the total number of REPLY packets transmitted back to producers/consumers during the simulation. We also compare our TNT results with those of event-flooding. In event-flooding<sup>2</sup>, a producer floods its advertisements to the entire network, and hence message overhead of a consumer's querying the producer is negligible. Consequently, if there are  $N$  nodes and one producer in the network, the total number of control packets transmitted is  $N * 1$  in event-flooding.
- *Success Rate (SR)* – how successful the algorithm is in matching the producers and consumers under the given network characteristics (network mobility and density). For event-flooding if there is no node failure and partitioning, consumers can always find producers with Success Rate of 100%.

### A. Simulation Setup

We conducted extensive simulations using GloMoSim [3] to evaluate the performance of our algorithm under different network characteristics for both ad hoc and sensor networks.

1) *Ad hoc Networks:* We evaluate the performance of our algorithm for three different ad hoc network configurations with 400 nodes: *dense ad hoc network*, *sparse ad hoc network*, and *sparse & faster-moving ad hoc network*. We used IEEE 802.11 in the MAC layer with radio range of 250m.

For the dense network, the terrain size is 1400m X 1400m, and nodes are distributed uniformly 70m apart. For the sparse network, the terrain size is 2000m X 2000m, and nodes are 100m apart. For the mobile case, the mobility pattern is RANDOM-WAYPOINT<sup>3</sup>, where the speed of nodes ranges between 1 m/sec and 4 m/sec, and the pause time is either 0 sec or

<sup>2</sup>Query flooding is the opposite of event flooding, where a consumer floods its subscriptions to the entire network.

<sup>3</sup>A node randomly selects a destination, moves in the direction of the destination in a speed uniformly chosen between a given min. and max. speed, and after reaching the destination, stays there for a given pause time period.

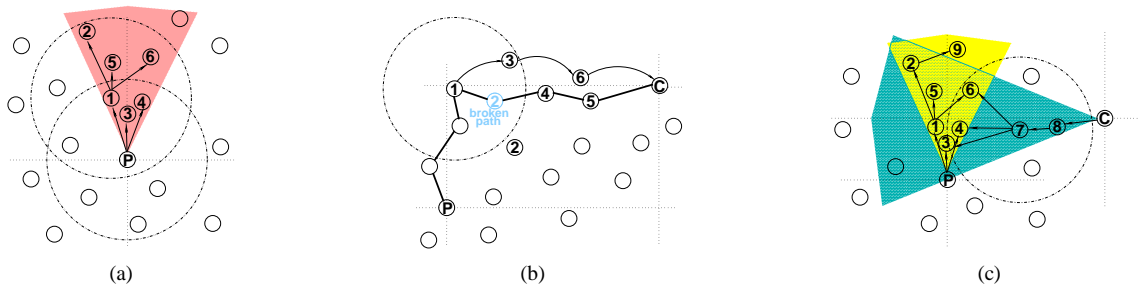


Fig. 2. Optimizations to the Generic Algorithm (For illustration purpose, only the north branch of producer P’s and the west branch of consumer C’s pseudo quorums are shown.) (a) *Selecting the furthest neighbor to forward an ADV/SUB packet.* The furthest neighbor of P is 1 in the north direction. Thus, P sends ADV packet to 1 which forwards the ADV packet to 2, since 2 is the furthest neighbor of 1 in the north direction. Nodes 3, 4, 5, and 6, located within the transmission range of the north sector, overhear and store the ADV packet. (b) *Geographic reply forwarding.* Let’s say 1 first gets the ADV packet of P and then SUB packet of C from 2. After forwarding the SUB packet, 2 moves out of the communication range of 1. 1 prepares a REPLY packet and sends back to 2. Since 2 has moved away, the backward path to C is broken. After 1 detects the broken path, 1 sends a new REPLY packet to 3, which is the closest neighbor of 1 to C. 3 forwards the REPLY packet to 6, which is the closest neighbor of 3 to C. Finally, 6 forwards REPLY to C. (c) *Detecting multiple REPLY packets.* Let’s say 3, 4, and 6 receive the ADV packet from P first. Later, 7 receives the SUB packet from 8 and forwards the SUB packet to 3 because 3 is 7’s furthest neighbor in the west sector of C. Meanwhile, 4 and 6, which are also in the west sector of C, overhear and store the SUB packet. Hence, all three nodes, 3, 4, and 6, send REPLY packets back to 7. But 7 forwards only the first REPLY packet coming from any of 3, 4, and 6 to 8. Otherwise, there will be unnecessary transmissions of the REPLY packets along the west branch of C’s quorum.

15 sec. In this way, both the dense and sparse networks can be thought as imitations of soldiers walking or running in a battlefield. In the sparse & faster-moving network, nodes are 100m apart while node speed ranges between 10-15 m/sec (similar to cars driving in city traffic or tanks maneuvering in a battlefield).

The simulation time is 60 sec. Update interval for ADV/SUB packets is 15 sec, and update lifetime for the stored ADV/SUB packets in intermediate nodes is 30 sec.<sup>4</sup> We randomly select one producer-consumer pair for each simulation. When the pair is matched, both the producer and consumer stop reinforcement of its quorum. If an intermediate node, upon receiving an ADV/SUB packet, is the consumer/producer of the InfoType included in the packet, the intermediate node stops the forwarding process. Therefore, we measure how many control packets is needed in total to match this one pair in 60 sec. Both the producer and consumer start sending packets at random times within the first 15 sec of the simulation. Moreover, in every 1 sec, hello messages are sent by each node to maintain the neighbor table.

2) *Sensor Networks:* We evaluate sensor networks of 3600 nodes and 4900 nodes. The terrain size is 150m X 150m and the radio range of each sensor node is 5m. Therefore, the node density of 4900-node sensor network is more than that of 3600-node sensor network. We used IEEE 802.11 in the MAC layer.

The simulation time is 120 sec for each sensor network. Again there is only one randomly selected producer-consumer pair. An intermediate node stops the forwarding of an ADV/SUB packet if it is the consumer/producer. Since there is no node mobility, there is only one hello packet broadcasted to neighbors by each node in the beginning of the simulation. The producer and consumer wait for 35 sec to let sensor nodes learn about their neighbors. Then the producer and the consumer send one ADV and one SUB packet at a random time within the following 15 sec (no periodic reinforcement of quorums).

For each configuration of the ad hoc and sensor networks,

<sup>4</sup>In our simulations for ad hoc networks, periodic updates are done even for no mobility cases.

five simulations are executed and the average of TNT and SR are obtained as the results and shown on the Figures 3, 4, and 5.

## B. Results and Analysis

In this section, we first discuss our results for the ad hoc networks and then for the sensor networks.

We classify our results for the ad hoc networks in two ways. The first way, as shown in Figure 3, depicts the effects of mobility on ad hoc network types.

For dense ad hoc network, node density and radio range of each node are such that each node may reach up to 3-hop away neighbors on the average in each direction. Since there are 20 X 20 nodes in the network, the number of nodes forwarding the ADV/SUB packets to form a ‘+’ shaped pseudo quorum is  $\lceil 20/3 \rceil * 2 = 14$  nodes on the average. This implies that the producer and consumer quorums may be formed by forwarding  $14 * 2 = 28$  messages. We should also count the transmissions of the REPLY packet forwarded back by the intermediate nodes, along the two branches of either of the quorums, which adds  $\lceil 20/3 \rceil * 2 = 14$  more extra control packets. As a result, for the dense ad hoc network, we expect TNT to be around 28 packets (to form the quorums of the producer and consumer) + 14 packets (to sent REPLY packets back) = 42 packets, if the producer and consumer can match in the first update interval.

In following the same analysis, TNT could be expected to be around 60 packets for both sparse ad hoc network and sparse & faster-moving ad hoc network, where each node can reach up to 2-hop away neighbors on the average. TNTs will be bigger than the expected numbers if the producer and consumer do not meet in the first update interval (due to mobility or a small threshold angle) and need to send ADV/SUB packets more than once within 60 sec. TNTs may be smaller than the expected number for the following reasons. One reason is attributed to the relative orientation of the producer and consumer. For instance, the producer-consumer pair is located such that while the producer’s quorum is being constructed, the consumer itself is selected to be one of the forwarding nodes, so that there is no need for consumer to construct its own quorum. Another reason involves the positions of the producer/consumer relative to the

network terrain, such that some branches of the quorums will not be extended (like in the case where the producer/consumer is closer or on a corner of the network terrain). Moreover, if the threshold angle is very small or the network is very sparse, it may not be possible to extend the branches of the quorums, because no neighbor node can be found to forward the ADV/SUB packets. Furthermore, it is possible that a neighbor node selected to forward an ADV/SUB packet does not receive the packet due to collisions<sup>5</sup> with the hello packets and so that again the branches of the quorum in that direction would stay short. As a result, all those cases, where branches of the quorums can not be extended, lead to a reduction in the number of ADV/SUB packets transmitted, but meanwhile reduces the chance that the consumer and producer will meet (*i.e.*, reduction in SR) as well.

As the threshold angle increases, the branches of the quorums can be extended better and the number of nodes in the intersection of the quorums increases. Hence, the number of intermediate nodes sending REPLY packets increases as well. There is a trend in that TNT as well as SR increase as threshold angle increases for all three ad hoc network types (Figure 3). However, there is an exception where SR of the sparse & faster-moving ad hoc network does not increase consistently. This might be due to nodes moving very fast (see the SR graph in Figure 3(c)). To increase SR for this case, a longer lifetime period for the storage of the ADV/SUB packets in the intermediate nodes could help because with the help of geographic reply forwarding mechanism, a reply could still be sent by the intermediate nodes.

Moreover, around degree 15 for the threshold angle behaves as a critical angle for both dense and sparse networks (see Figure 3(a) for the peaks in TNT at 15 and and 3(b), (c) for the peaks around 15 and 17.5 degree). At degree 15, it might be critical that the branches of quorums can either be extended very-well or not at all depending on the the randomness in the positions of nodes for a particular simulation. Therefore, producer-consumer can either find each other in the first update interval, or never find each other so that they keep sending ADV/SUB packets.

The other way of classification, as shown in Figure 4, depicts the effects of network characteristics on the performance of our algorithm. As expected, the dense network provides a better SR than the sparse network even for lower threshold angles in the no mobility case (Figure 4(a)). Furthermore, a more dynamic network has a lower SR than a less dynamic network (Figure 4(b) and Figure 4(c)).

The results for the sensor networks are shown in Figure 5. The percentages on the graphs show SR for each threshold angle. In the 3600-node network, node density is such that sensor nodes can reach 1 or 2 hop away neighbors. Node density of the 4900-node network is higher, where nodes can reach 2 or 3 hop away neighbors. Therefore, as expected, the SR of the 4900-node network is higher even for smaller threshold angles than those of the 3600-node network. Note that TNT never exceeds 141 packets for both sensor networks, where the total number of nodes is stated with thousands. Meanwhile, we achieve 100% confidence in matching the consumer and pro-

<sup>5</sup>The broadcast mode of IEEE 802.11 does not use collision avoidance nor re-transmission mechanism.

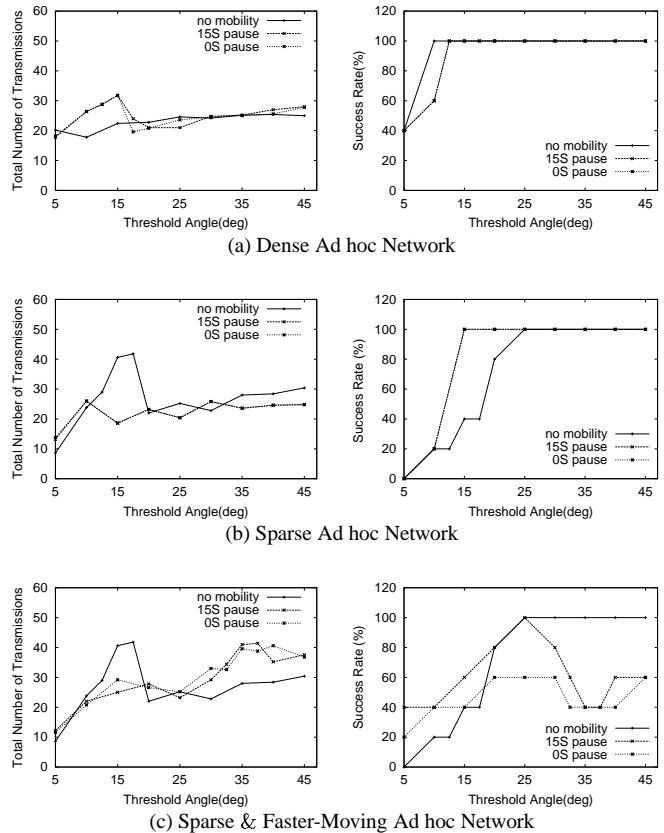


Fig. 3. Effects of node mobility and threshold angle on TNT and SR for each ad hoc network type. (a) Dense ad hoc network: nodes are uniformly distributed 70m apart with speed of 1-4 m/sec. (b) Sparse ad hoc network: nodes are uniformly distributed 100m apart with speed of 1-4 m/sec. (c) Sparse & faster-moving ad hoc network: nodes are uniformly distributed 100m apart with speed of 10-15 m/sec.

ducer when the threshold angle is large enough. In contrast, in the event-flooding, the total number of transmissions would be 3600 and 4900 packets for the 3600-node and 4900-node network, respectively. For a given threshold angle, if the producer and consumer can *not* meet, it is enough just to increase the threshold angle and re-run the algorithm. In this ways, TNT could still be reduced drastically as opposed to that of event-flooding.

#### IV. RELATED WORK

The Uniform Quorum System (UQS) was described in [4] to facilitate mobility management in ad hoc networks, where a location database system is constructed to store the location information of each node at all the backbone nodes belonging to the same quorum. The cost analysis of the UQS was also performed to investigate the trade-off between the system reliability and the cost of location updates in the UQS scheme.

The Content Based Multicast (CBM) for ad hoc networks was described in [5], where the dissemination of information is performed by taking the type and velocity of the threads/resources into account to determine the region the information should be spread. The CBM scheme works well for its presented application scenarios, while our heuristics developed for the pseudo quorum system to direct message flows are

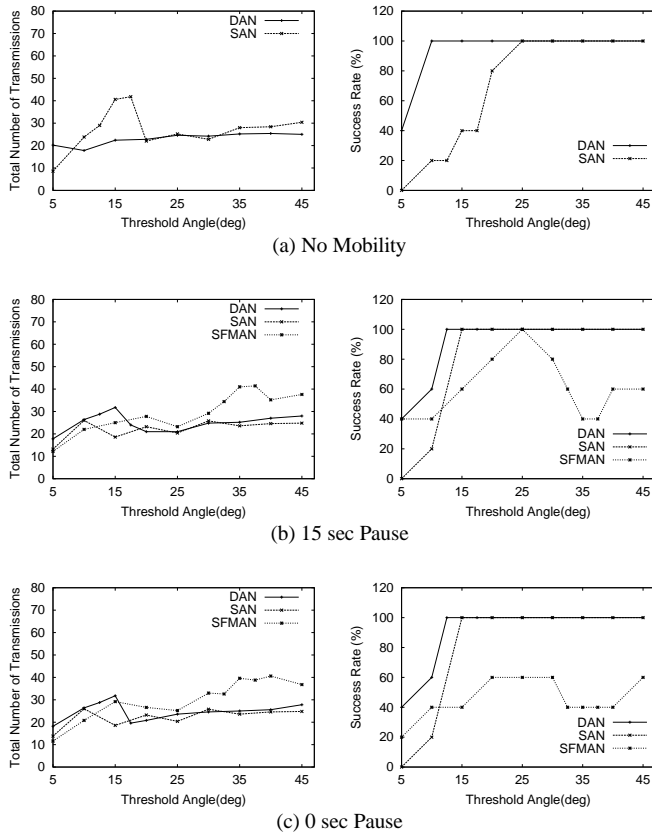


Fig. 4. Effects of ad hoc network characteristics on TNT and SR. DAN (Dense Ad hoc Network): nodes are uniformly distributed 70m apart with speed of 1-4 m/sec. SAN (Sparse Ad hoc Network): nodes are uniformly distributed 100m apart with speed of 1-4 m/sec. SFMAN (Sparse & Faster-Moving Ad hoc Network): nodes are uniformly distributed 100m apart with node speed of 10-15 m/sec.

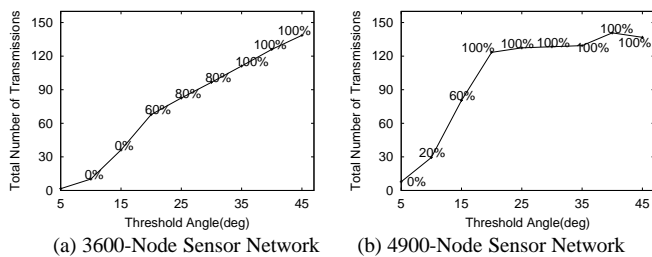


Fig. 5. TNT and SR vs. Threshold Angle for the Sensor Networks

applicable to more general application scenarios.

Rumor Routing (RR) was introduced in [6] for sensor networks where paths leading to events occurring within the network terrain are created with the help of agents (long lived packets). Each sensor node witnessing an event can create a corresponding agent to be forwarded along a straight path in a randomly selected direction to spread the event, depending on a probability. Queries are also forwarded in the same fashion (*i.e.*, to straighten the paths in a randomly selected direction), until the queries meet the paths leading to the corresponding events. RR dictates that undelivered queries are flooded to the network to guarantee eventual delivery, which in the meantime increases the average number of transmissions per query.

Given the same configuration, the pseudo quorum scheme may demonstrate better performance due to (1) a more deterministic approach (*i.e.*, neither advertisements nor subscriptions is sent in random directions (cross shaped pseudo quorums are formed by sending ADV/SUB packets in east, west, north, and south directions) and better heuristics (three optimization mechanisms are added to the generic algorithm) to forward the ADV and SUB packets, which increase the possibility to intersect, and (2) no flooding of ADV/SUB packets. In case that the producer and consumer do not match, it would help just to increase the threshold angle to fatten the '+' shaped quorums.

## V. CONCLUSION AND FUTURE WORK

Facilitating match-making capability in ad hoc and sensor networks poses unique challenges: (1) producer(s) and consumer(s) finding each other without relying on any directory service nor flooding the network, (2) dynamic topology caused by node mobility (adaptability for ad hoc networks), and (3) the sheer number of nodes (scalability for sensor networks). In this paper, we proposed to facilitate the match-making capability in ad hoc and sensor networks by adapting the idea of quorum system. The producer and the consumer systematically forward the ADV and the SUB packets to form '+' shaped pseudo quorums, where matches may occur at the intersecting nodes. The pseudo quorum approach together with optimizations adapts to dynamic topology caused by node mobility in ad hoc networks and scales well to accommodate the sheer number of nodes in sensor networks. As demonstrated by simulation results, its performance is much more efficient than that of event and query flooding.

For future work, we plan to further investigate the interaction among mobility pattern, threshold angle, update interval, and update lifetime, and to quantify their interrelationships.

The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U. S. Government.

## REFERENCES

- [1] A. Carzaniga, D. S. Rosenblum, and A. L. Wolf, "Content-based Addressing and Routing: A General Model and its Application," Tech. Rep. CU-CS-902-00, Department of Computer Science, University of Colorado, January 2000.
- [2] W. K. Ng and C. V. Ravishankar, "Coterie Templates: A New Quorum Construction Method," in the *15th International Conference on Distributed Computing Systems (ICDCS'95)*, (Vancouver, British Columbia, Canada), 1995.
- [3] <http://pcl.cs.ucla.edu/projects/glomosisim/>.
- [4] Z. J. Haas and B. Liang, "Ad Hoc Mobility Management with Uniform Quorum Systems," *IEEE/ACM Transactions on Networking* 7, pp. 228–240, April 1999.
- [5] H. Zhou and S. Singh, "Content based multicast (CBM) in ad hoc networks," in the *ACM/IEEE Workshop on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, p. 5160, (Boston, MA), August 2000.
- [6] D. Braginsky and D. Estrin, "Rumor Routing Algorithm for Sensor Networks." <http://citeseer.nj.nec.com/462331.html>.