CAR protocol preliminary simulation results

Mirco Musolesi, Stephen Hailes, Cecilia Mascolo
Dept. of Computer Science University College London
Gower Street, London WC1E 6BT, United Kingdom
{m.musolesi|s.hailes|c.mascolo}@cs.ucl.ac.uk

Abstract

We are evaluating the Context-Aware Routing (CAR) protocol by using the OmNet++ discrete event simulator [6]. In order to obtain credible results and to test the peculiar characteristics of our protocol, it was also necessary for us to develop a new group mobility model, that will be presented in Section 2.

1 Description of the simulation

1.1 CAR Simulation

We simulated the CAR model using a utility function based on the evaluation of two attributes: (i) the change rate of connectivity and (ii) the probability of being located in the same cloud as the destination. We made the assumption that these factors have the same relevance, so assigned them the same weights in the evaluation of the overall utility (i.e., $w_i = 0.5$). Moreover, we also assumed that all the possible values in the range had the same importance (i.e, $a_{range_i}(x_i)$ = 1) and that the the values of attributes are always available during the simulation (i.e., $a_{availability_i}(x_i) = 1$).

The change rate of connectivity attribute is locally calculated by examining the percentage of a node's neighbors that have changed their connectivity status (connected to disconnected, or vice versa) between two instants. The co-location attribute measures the percentage of time that two hosts have been in reach. To calculate it, we periodically run a Kalman filtering process, assuming that the value is 1 if the host is currently in reach or 0 if not. Clearly, the resultant predicted values will be in the range $[0, 1]$ and they will directly express an estimation of the probability of being in reach of the host in the future.

We implemented a simplified version of the DSDV protocol [4] in order to simulate and test the synchronous delivery in connected portions of the network, as described in Section **??**.

Each host maintains a *routing and context information table* used for asynchronous and synchronous (DSDV) routing. Each entry of this table has the following structure:

```
(targetHostId, nextHopId, dist, bestHostId, delProb)
```

The first field is the recipient of the message, the second and the third are the typical values calculated in accordance with the DSDV specification, whereas the fourth is the identifier of the host with the best delivery probability, the value of which is stored in the last field. It is worth noting that all the autonomic mechanisms, such as the variable refresh period of routing tables, described previously, were implemented.

We also simulated flooding and the epidemic protocols in order to provide comparators for the performance of the CAR solution.

1.2 Flooding simulation

We elected to compare our approach with flooding. This decision may seem strange, since flooding only works in a fully connected environment. However, since communications patterns are random in the simulations, many messages will be passed between hosts that are in connected portions of the network, even when assessing the performance of the epidemic algorithm and of the CAR algorithm. In order to see the difference in delivery rates that result from the algorithms' ability to handle partial connectivity, it is therefore essential to compare against a synchronous protocol with optimum delivery ratio.

1.3 Epidemic routing simulation

The implementation of the epidemic protocol follows the description presented in [5]. The only assumption made by the authors is a periodic pair-wise connectivity, since the protocol relies on the transitive distribution of messages for delivery. When two hosts become neighbors (in other words, they are within each other's radio range), they determine which messages each possesses that the other does not, using summary vectors that index the list of messages stored at each node; they then exchange them. Each message is characterized by a unique message identifier and a hop count value; the latter determines the maximum number of possible exchanges of a message. Higher hop count values reduce the delivery latency, but, at the same time, increase the quantity of resources (memory, battery, bandwidth) consumed in this process. The epidemic approach represents the classic example of an asynchronous protocol and therefore provides the ideal comparator.

1.4 Simulation system parameters

We evaluated the performance of each protocol sending 100 messages with a simulation time equal to 300 seconds. The messages were sent after 40 seconds, in order to allow for the settling of initial routing table exchanges, and the intervals between each message were modeled as a Poisson process, with $\lambda = 5s^{-1}$, and the consequence that all messages are sent in about 20 seconds. The sender and receiver of each message are chosen randomly.

In the CAR simulation, each message has a field that is similar to a *time to live* value that is decreased each time that the message is transferred to another host (the initial value being 15). Moreover, in this case, we also introduced a *split horizon* mechanism to prevent messages from being retransmitted unnecessarily. The buffer for each node was set to 20 messages, unless otherwise specified. Table 1 summarizes the simulation parameters.

The one key aspect of the simulation not yet addressed is that of the mobility model. Clearly, the random way-point mobility model, which is used extensively in such studies largely for reasons of simplicity, does not accurately reflect human behaviour and renders prediction useless since movement is entirely random. Consequently, we devised a new group-based mobility model, which will be explored in detail in a later paper. This is presented briefly in the following section.

Table 1: Simulation parameters

Number of hosts	16/24/32
Simulation area	1 Km x 1 Km
Propagation model	free space
Antenna type	omnidirectional
Transmission range (radius)	200 m
Mobility model	clustered random way point
Number of clouds	4
Cloud area	200 m x 200 m
Node speed	1-3 m/s (randomly generated)
Cloud speed	1-2 m/s (randomly generated)
Number of messages sent	100
Max number of hops	15
Message buffer size	10 to 100
Routing table size	20 entries
Max distance	15

2 Mobility model

Mobility models that assume that individuals move independently of one another in random ways are unrealistic in terms of the deployment scenarios for ad hoc networks that are most commonly expounded. For example, on a battlefield, it would be indicative of a very troubled army if each soldier were to move randomly with respect to all others. Thus, we have extended the random-way point model [1] with a form of hierarchical clustering that better reflects the ways in which collections of people are structured at an organizational level and, consequently, the ways in which they move. This model has been instantiated in a simple way for these experiments, and, as used here, is somewhat akin to those in [3, 2]. Thus, we introduce the concept of a collection of nodes, which has its own motion overlaid on a form of random motion within the cloud.

By parameterizing this model differently, we can represent different archetypes: for example, one would expect to use different parameters for an academic who spends her life traveling between home and the university, interacting with a very closed set of people, as opposed to a salesman who travels much more extensively and interacts less discriminatingly.

A host that belongs to a cloud moves inside it towards a goal (i.e, a point randomly chosen in the cloud space) using the standard random way-point model. When a host reaches a goal, it also implicitly reaches a decision point about whether to remain within the cloud, and, if leaving, to where it should go. Each of these decisions is taken by generating a random number and comparing it to a threshold (which is a parameter of the model). It is worth noting that clouds also move towards randomly chosen goals in the simulation space.

In the remainder of this section, we will discuss the details of the simulation of CAR.

50% of the hosts are initially placed randomly in a cloud, whereas the others are positioned randomly in the simulation area. Each cloud is defined using a squared area with a side length of 200 m. In other words, we randomly select the point $(minX, minY)$ that, together with the length of the side, defines the cloud area. For these simulations, there is only a single level of cloud.

Every host is characterized by two values, P_{escape}, indicating the probability of escaping from the current cloud, and $P_{escapeCloud}$ describing the probability of choosing a new goal in the space between clouds.

Each cloud moves with a random speed (with a value in the range 1-2 m/s); moreover, each host moves with a randomly generated different speed (with a value in the range 1-3 m/s). It is worth noting that the movement of a host is the result of the composition of these speeds.

In our simulation, the positions of all the hosts and clouds are updated every second. When a cloud reaches its goal, a new goal is chosen in the simulation space. When a host reaches its goal, a threshold probability $P_{escapeThreshold}$ is generated randomly (its range is clearly $[0, 1]$). If its P_{escape} is greater than $P_{escapeThreshold}$ the new goal is chosen outside the current cloud, else inside. If outside, we randomly generate $P_{escapeCloudThreshold}$ and compare it to $P_{escapeCloud}$ to determine whether or not the goal should be chosen in some other cloud or in the open space between clouds. For those hosts that are already outside a cloud, the choice of a new goal is done in an analogous way.

3 Analysis of results

In this subsection we will analyze the results of our simulations, comparing the performance of CAR with the flooding and epidemic protocols. We will discuss the variation of some performance indicators as functions dependent on the density of hosts (i.e., the number of the hosts in the simulation area) and the size of the buffers used to store messages in both the epidemic and CAR.

In Figure 1, there is a comparison between the delivery ratios of the three protocols in each of three different scenarios (with 16, 24 and 32 hosts). In all cases, the number of messages that may coexist within a node's buffer is unconstrained.

CAR achieves a performance between that of flooding and epidemic routing, as expected. Flooding suffers from the inability to deliver messages to recipients that are in other clouds when the messages are sent but is here simply as a comparator to demonstrate the numbers of messages being delivered that cannot be delivered directly, because the recipient is in a cloud different from the cloud of the sender. The epidemic protocol can be considered optimal in terms of delivery ratio, simply because each message is propagated to all accessible hosts, all of which have buffers large enough to hold it. In CAR, we have chosen to operate under the most stringent conditions: there is only ever a single copy of each message, which represents the worst case for this protocol. Clearly, it would be possible to trade off a small amount of intelligent replication (to improve the delivery ratio) against an increase in overhead.

The dependency of the delivery ratios on the buffer size is similar for all the protocols (see in Figure 2 the results for the 32 hosts scenario). Both of these demonstrate a substantial degradation of their performance as buffer size decreases; however, this phenomenon is more evident in the epidemic approach as a result of the degree of replication of messages.

Figure 3 is interesting because there are two competing effects at work for the epidemic protocol. When the buffer size is small, there is a high probability that messages will be eliminated due to overflow, as discussed above. Consequently, the number of messages exchanged is also low. At the other end of the scale, as the buffer size increases to a point where it can accommodate all the messages in the system, there is no repeated exchange of messages, so the number is also low. In the middle of the range, however, the buffer size is insufficient to hold all messages and there is a cycle in which messages are eliminated by buffer overflow and then reinstated by other nodes, resulting in very high overhead. In the case of CAR, it is worth noting that the overhead in terms of the number of messages exchanged is more or less constant, regardless of buffer size, demonstrating its *scalability*. CAR will always be the limiting case for performance under this metric because it only creates a single copy of each message. Thus, even at the point where buffer size becomes effectively infinite, the epidemic protocol will necessarily exchange more messages than ours, simply as a result of the replication.

Figure 4 shows the distribution of the number of messages with respect to their delivery latency in the 32 hosts scenario. It is possible to observe that a proportion of the messages are delivered more or less immediately, since the recipients are in the same cloud as the sender. Another interesting comparison is showed in Figure 5: the distributions of the delivery latency in the case of different node densities are very similar.

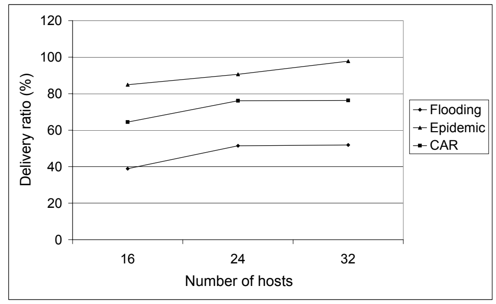

Figure 1: Delivery ratio vs population density

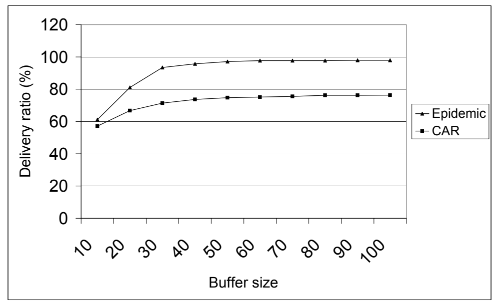

Figure 2: Delivery ratio vs buffer size

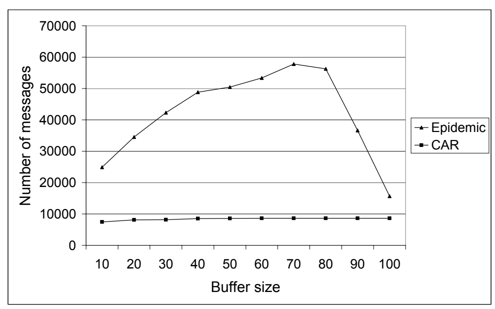

Figure 3: Number of messages vs buffer size

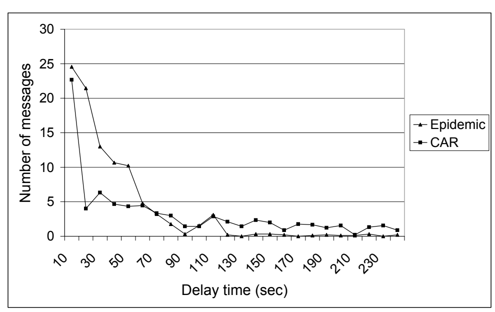

Figure 4: Average delay vs population density

References

[1] Tracy Camp, Jeff Boleng, and Vanessa Davies. A survey of mobility models for ad hoc network research. *Wireless Communication and Mobile Computing Special Issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, 2(5):483–502, 2002.

[2] Juan Carlos Cano, Pietro Manzoni, and Miguel Sanchez. Evaluating the impact of group mobility on the performance of mobile ad hoc networks. In *Proceedings of the 2004 IEEE International Conference on Communications (ICC 2004)*, 2004. To appear.

[3] Xiaoyan Hong, Mario Gerla, Guangyu Pei, and Ching-Chuan Chiang. A group mobility model for ad hoc networks. In *Proceedings of the 2nd*

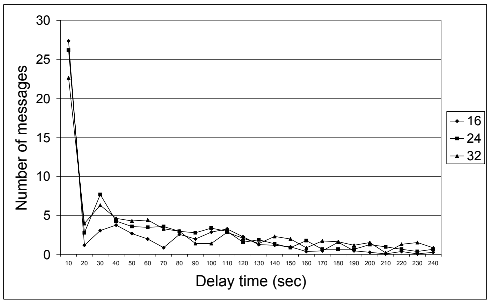

Figure 5: Average delay vs population density

International Workshop on Modeling Analysis and Simulation of Wireless and Mobile Systems, pages 53–60, 1999.

[4] Charles E. Perkins and Pravin Bhagwat. Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers. In *Proceedings of thr Conference on Communications Architecture, Protocols and Applications (SIGCOMM 94)*, August 1994.

[5] Amin Vahdat and David Becker. Epidemic routing for partially connected ad hoc networks. Technical Report CS-2000-06, Department of Computer Science, Duke University, 2000.

[6] Andras Vargas. The OMNeT++ discrete event simulation system. In *Proceedings of the European Simulation Multiconference (ESM'2001)*, Prague, 2001.