Towards Group Communication for Mobile Participants

[Extended Abstract]

Marc-Olivier Killijian — marco.killijian@cs.tcd.ie
Raymond Cunningham — rcnnnghm@cs.tcd.ie
René Meier — rene.meier@cs.tcd.ie
Laurent Mazare — laurent.mazare@polytechnique.org
Vinny Cahill — vinny.cahill@cs.tcd.ie

Distributed Systems Group
Department of Computer Science
Trinity College of Dublin
Dublin 2, Ireland

ABSTRACT

Group communication will undoubtedly be a useful paradigm for many applications of wireless networking in which reliability and timeliness are requirements. Moreover, location-awareness is clearly central to mobile applications such as traffic management and smart spaces. In this paper, we introduce our definition of *proximity groups* in which group membership depends on location and then discuss some requirements for a group membership management service suitable for proximity groups. We describe a novel approach to efficient coverage estimation, giving applications feedback on the proportion of the area of interest covered by a proximity group, and also discuss our approach to partition anticipation.

Keywords

group communication, location awareness, proximity, adhoc networks

1. INTRODUCTION

The widespread deployment and use of wireless data communications is generally recognised as being the next major advance in the information technology industry. In the long term, wireless data networks will represent a key enabling technology underlying the vision of *ubiquitous computing* [1]. In this vision, interconnected computers will be embedded in a wide range of appliances ranging in size from door locks to vehicle controllers, and will co-operate to perform tasks on behalf of their human users ranging from automatically opening doors to routing vehicles to their intended destinations in co-operation with other vehicles' controllers. Mobility, and hence wireless networking, is clearly central to this vision. We believe that, as is the case for fixed networks, group communication [2], [3] will be a useful paradigm for many such applications of wireless networking in which reliability and timeliness are important requirements. A major feature of wireless communications is the fact that participants can be mobile and hence that their location can have an impact on the information in which they are interested or that they can provide. Hence, we believe that any wireless group communication system should support location awareness. Moreover, knowledge of the location of participants can be exploited in the implementation of the group communication system itself.

In this paper, we consider the problem of group communication in a wireless network. Much of the previous work in this area deals with routing protocols for group communication based on multicast or geocast [4], [5]. In this paper, we concentrate on the definition and semantics of group membership for location-aware mobile participants. While some research has already been done on groups in which membership is based on location information [6], [7], our definition of *proximity groups* takes into account both location and functional aspects. When group membership depends on location, it becomes important to understand what proportion of the area of interest is within wireless network coverage. For this purpose we describe a novel approach to coverage estimation. Another important aspect of mobile computing is that partitions are very likely to occur. For this reason our membership layer also includes a new failure and partition anticipation scheme that can take into account movement of nodes, battery life, etc. This algorithm can be set up to be either optimistic or pessimistic and tries to anticipate partitions/failures in order to maintain consistent group views.

In section 2, we summarize different approaches to traditional group communication. In section 3, we outline our approach to group communication with mobile participants. We introduce the notion of *proximity group* taking into account aspects such as absolute or relative location and the function of the group. We then address some of the requirements of group membership management for proxim-

POMC '01 Newport, Rhode Island USA

ity groups, including location awareness, estimation of the coverage of the area covered by the group, and partition anticipation. In section 4, we compare our model to related work and, finally, section 5 provides a summary and some conclusions as well as pointers to future work.

2. GROUP COMMUNICATION OVERVIEW

Toolkits for group communication typically provide group membership management services as well as multicast protocols for reliable, ordered, and/or timely delivery of messages to the members of a group. Group membership management is primarily concerned with achieving consensus on the membership of a group. In the following sections we give a brief overview of the most important approaches to group membership management and group communication that are relevant to the remainder of the paper.

2.1 Group membership management

Group membership management in traditional group communication systems follows one of two distinct approaches. The first and most widely used approach is to assume that a group consists of a dynamically varying subset of a fixed number of group members. In this static approach, the maximum number of group members is fixed. In the second approach, there is no restriction on the maximum number of group members. Membership is dynamic with processes being created, carrying out some computation and/or communication, and then terminating. A well-known example of the use of static group membership management is in the CASD protocol suite [8]. Some systems that use the dynamic group membership management approach include ISIS [2], Horus [2], and Transis [9].

2.2 Diffusion based group communication

In diffusion based group communication, messages to the group are probabilistically flooded to other members of the group. On receiving a message, a group member forwards the message in a similar manner. If a group member receives a previously seen message, it is discarded.

The CASD synchronous atomic broadcast protocol provides an example of diffusion based group communication. In this protocol, the maximum number of hops that a message may travel to reach any member of the group is bounded. However, the protocol assumes that the network will not partition due to failure and that the number of messages that can be lost during a single run of the protocol is also bounded.

The protocol works by a group member flooding each of its messages with certainty (probability = 1) to all other group members. Each message is time stamped before being transmitted and every member of the group delivers the message at a time given by the timestamp plus a constant Δ. Two or more messages with the same timestamp are delivered in order of their senders' identifiers. Δ depends on the network diameter and the latency in processing and transmitting a message.

The CASD protocol is a rather restricted example of a probabilistic protocol. Less restrictive protocols would reduce the probability from 1. These protocols are probabilistically reliable with the probability of a failed run of the protocol approaching zero as the number of participants increases.

2.3 Group communication based on message exchange

An alternative approach to group communication relies on the exchange of messages between the members of the group. One example of this approach is the 2 phase commit protocol. In this protocol, a coordinator initiates an initial round of messages to the participants. Each participant then decides which way to vote (commit or abort). A second round of messages occurs when the participants return their votes to the coordinator. Finally, the coordinator sends out a commit or abort message to the participants based on the collected votes.

Another protocol that uses message exchange to reach agreement on message ordering is the total ordering protocol developed during the course of the ISIS project, which is similar to 2 phase commit with the exception that it only requires a majority of group members to receive the message before it can be delivered.

3. LOCATION-AWARE GROUP COMMUNICATION FOR MOBILE PARTICIPANTS

Our goal is to provide mobile hosts in a wireless network with a suite of protocols for group communication. The wireless network can be either an infrastructure or ad-hoc network or even a hybrid of the two. As can be seen in the literature, traditional communication protocols such as medium access control, routing, etc. are inadequate in this context for various reasons [10], [11]. Location awareness can be used to overcome some of the problems raised by this kind of network. For example, several location-aware routing protocols [12] [13] have been designed that illustrate the advantages of making use of location information. Furthermore, location-awareness is, in our opinion, central to mobile applications such as traffic management or smart spaces. Therefore, we propose to make location awareness the basis for the definition of a new model of group communication for mobile participants. In this section, we firstly define this model and then we address some of the requirements for the design of a framework implementing the model.

3.1 Definition of proximity groups

At the heart of our approach to group membership management is the use of location for both functional and non-functional reasons.

- Firstly, in functional terms, it often makes sense to define a group in a mobile application in terms of a geographical area. We can easily imagine many cases where this would be interesting: in traffic management, for example, the area around a traffic-light could be used to define a group with cars in that vicinity becoming members of the group to receive notifications of changes to the state of the lights; in a similar way, we might want to define a group corresponding to the area around an ambulance in order to inform nearby cars to yield the right of way.

- Secondly, from the non-functional point of view, we can use location information to, for example, anticipate partitions and hence take preventative measures to ensure consistent group views when these partitions happen.

In classical group communications, a group is defined by its functional aspect, e.g. its name. Our notion of proximity group involves both location and functional aspects, i.e., to be able to apply for group membership, a node must firstly be located in the geographical area corresponding to the group and secondly be interested in the group. In the following paragraphs we discuss the various possibilities related to the location aspect of group membership.

To define a proximity group, we firstly have to define the area that it covers as a geometric shape with associated coordinates. Any kind of shape can be used, i.e., it need not necessarily be a circle or a square but can be arbitrarily complex. We can obviously imagine using 2 or 3 dimensions, but it is also possible to include time in the definition of the area, e.g., "the area around the position at which the ambulance was located at 2.00pm". To define the coordinates of the area, we associate a reference point with the shape. We distinguish two cases: either the group is absolute, i.e. geographically fixed, or it is relative to a moving point, its so-called navel. In the absolute case, the reference point is attached to a fixed point in space. In the relative case, it is attached to the navel, i.e. an identified node.

Figure 1 illustrates this notion of an area. The first shape S on the left is associated with a reference point R. This reference point is relative to the shape. The definition of the area is not complete since R has not been attached to a point (possibly moving) in space. For the second shape, R has been attached to the point (0,0), making the area absolute. The reference point of the third shape has been attached to a node M that represents the navel of this relative proximity group. Table 1 gives some example group areas for two different application domains.

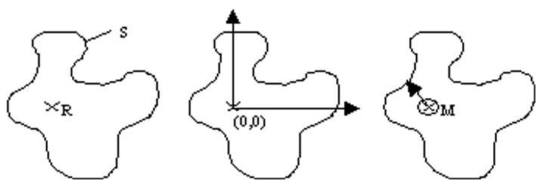

Figure 1: Area definition

In typical group communication, a group is roughly defined by a topic (or a name) and nodes can join this group if they are interested in its topic. We believe that this is also necessary for proximity groups because a node in the area of the group may or may not be interested in joining the group. We then add the functional aspect to the previous definition of a proximity group by associating a name with each group:

Group Membership	
Partition/Failure Anticipation	
Coverage Awareness	Routing/Geocasting
Connectivity Awareness	
Location Awareness	

Table 2: Summary of Requirements

Definition 1. A Proximity group G is completely defined by the shape, the reference point, the navel and the name:

$$G = \{Shape, ReferencePoint, Navel, Name\}$$

3.2 Membership management for proximity groups

Our goal is to define a group membership management layer suitable for proximity groups. Since in this model, location is intrinsic to group membership, it is important to be able to provide applications with at least an estimate of the probability of there being one or more nodes, which while currently in the area of interest, are disconnected from the group, typically because of lack of network coverage. To address this issue we provide a coverage estimation tool based on a novel algorithm described in section 3.2.2. below that uses knowledge of the connectivity graph of the network. As we describe later, coverage estimation can also be used to select the appropriate approach to our group communication. Our membership management protocols should also be failure aware and anticipate partitions, which are very likely to occur in the kind of networks that we are considering. This has lead us to the definition of a partition anticipation tool. Given these tools we would be in a position to define appropriate routing and geocasting protocols to be used by the group membership management layer. Table 2 summarizes the resulting architecture. We elaborate on the most important components below.

3.2.1 Location information distribution protocol

In our model, the group membership management layer must be able to determine how well the group members collectively cover the area of interest. This is a very important question since we are dealing with mobile participants: if a sub-part of the area is not covered, potential group members located in this sub-area may be unknown to other group members. Knowledge about the coverage of the area by the network participants is distributed: to evaluate the coverage of the area, one would need to know each node's location and at least an estimate of its own coverage. We propose an algorithm to share the connectivity information and another algorithm to evaluate the coverage of the area using this information; the precision of this latter algorithm is then discussed.

Wireless MAC protocols like the point coordination function of IEEE 802.11 [14] and routing protocols like AODV [15], often require nodes to periodically send beacon mes-

Domain	Absolute Proximity	Relative Proximity
Traffic Management	*Traffic light*: a traffic light informs nearby cars of its status. The shape is a circle; the reference point is the centre of the circle and is attached to the geographical coordinates of the traffic light.	*Ambulance*: an ambulance on call informs nearby cars to yield the right of way. The shape is a square, the reference point its centre and the navel is the ambulance itself.
Smart Spaces	*Resource access*: to use a printer, nearby people must reserve it using the printer proximity group. The system administrator defines the shape according to the available printers and offices; the reference point is at the printer and attached to the printer's coordinates.	*Centralised tour guide*: in a museum a group of tourists wear headsets and are remotely guided by an automatic guide. The area surrounds the group of tourists, while the navel is attached to one of them.

Table 1: Examples of group areas

sages in order to make their neighbours aware of their presence. In a similar way, we use location-stamped beacons. Each node keeps a map of its knowledge of the location and connectivity of other nodes, which is represented as a graph as shown in figure 2. This graph is regularly updated when the node receives a beacon and is also regularly sent to the node's neighbours in its beacons. In this way, a node's knowledge increases over time:

- firstly, the node knows its own location (level 0),
- secondly, when the node receives its neighbours' beacons, including their locations, it knows about its one hop connectivity (level 1),
- then, the node receives beacons including its neighbours knowledge (level L-1) and updates its own graph with this information (level L).

If beacons are sent each τ time units, level L information is (L * τ) old. Because high-level knowledge is older and because it is not desirable to have the knowledge of the whole network, the maximum level of knowledge is bounded. This bound, Lmax, is determined dynamically according to the size of the group area and the density of the network.

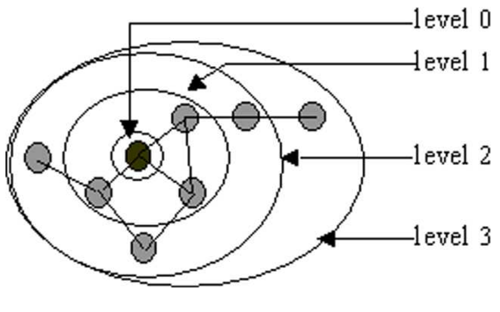

Figure 2: Location knowledge levels

This algorithm is pro-active and enables a node to know the other nodes physically present within the area if Lmax is sufficiently large. When Lmax is not large enough, or the coverage obtained is not sufficient, a reactive protocol is used to collect location information further than Lmax hops.

In addition to the location of the node, the beacon may also include other useful information about the sending node's radio coverage, its battery life, etc. At any time after $(Lmax.\tau)$ time units, a node knows the location and coverage of its Lmax hops neighbours and is then able to estimate the coverage of the area by this set of nodes (see 3.2.2 below). It should however be noted that at time t, every node in the network has a different view of the connectivity since its level 1 information is τ time units old, its level n is $(n.\tau)$ time units old, etc. If this connectivity information is to be used for some protocol where consensus is necessary, one would consider that at time t each node knows the exact connectivity graph that we had at time $t - (Lmax.\tau)$ [1].

3.2.2 *Evaluation of the coverage of an area*

Evaluating the percentage of the area that is covered by a set of nodes can be a very complex calculation. Actually, the complexity depends on the shape of the area defining the group and on the number and shapes of the coverage areas. For instance, calculating the area covered by two overlapping circles, representing the transmission ranges of two nearby nodes, necessitates an integral [16] and the complexity increases with the number of circles. To circumvent this problem we propose to estimate coverage using a number of sample points, randomly generated, and to check whether or not these points are in an area that is covered by any of the nodes. Table 3 presents this algorithm for circular shapes but it can be implemented for any kind of shape provided that one can give the specific function *inside()* that determines if a point is inside or outside the area covered by the particular shape.

3.2.3 *Precision of the coverage estimation*

The precision of this evaluation algorithm can be considered as a direct result of the Weak Law of Large Numbers, which states that the sample mean of a sufficiently large number of independently identically distributed random variables can be made arbitrarily close to the true mean with high probability [17].

[1] This is actually true if no message is lost during the location information distribution protocol. A stronger consensus for the coverage estimate could be achieved in conjunction with the group membership management. This estimate can then be seen as part of the group view.

```
float coverage(Area A,int N, ListOfCircles Clist) {
   for (int I=1 to N) {
      point p = new random point in Area A;
      boolean found = false;
      circle C ;
      while (not found and C=Clist.next()!=null) {
         found=inside(C,p);
      }
      if (found) nb_points_found++;
   }
   return (nb_points_found/N);
}
bool inside(circle C, point p) {
   return (sqrt(pow(C.x-p.x,2)+pow(C.y-p.y,2))
         < C.radius);
}
```

Table 3: Coverage estimation algorithm

In the context of the estimation algorithm, we consider the coverage function as generating a sequence of N independent and identically distributed Bernoulli trials, X_i with the inside function returning 1 if the generated point is within the list of circles and 0 otherwise.

Let p be the probability of a point being within at least one circle. We define the sample mean S_N to be:

$$S_N = \frac{X_1 + X_2 + \ldots + X_N}{N}$$

Then

$$E[S_N] = p, Var[S_N] = p(1-p)$$

By the Weak Law of Large Numbers, for any $\varepsilon > 0$,

$$P(|S_N - p|) > \varepsilon) < \frac{p.(1-p)}{N.\varepsilon^2}$$

For example, if $\varepsilon = \frac{1}{10}$ and n = 1000, we obtain

$$P(|S_{1000} - p|) > \frac{1}{10}) < \frac{\frac{1}{4}}{1000 * \frac{1}{10}^2} = 0.025$$

Since the value of

$$p(1-p) \leq \frac{1}{4}$$

In words, by choosing 1000 random points, the probability that our estimate of the coverage area is wrong by more than 10% is no larger than 0.025. Table 4 illustrates some values for the accuracy of the coverage area estimate and some probabilities which give corresponding values for N.

The final value in the table states that if we want our estimate of the coverage area to be correct to within 1% of the actual area with a probability of 0.9999 then we must choose 25 million sample points. This last value seems very large, fortunately it can be reduced due to the Central Limit Theorem. By the central limit theorem, if N is large then

accuracy/probability	0.95	0.99	0.9999
10%	500	2,500	250,000
5%	2,000	10,000	1,000,000
1%	50,000	250,000	25,000,000

Table 4: Sample values of N

S_N can be treated as if it follows a normal distribution. By the symmetry of the normal distribution, we have

$$P(|S_N - p| > \varepsilon) \approx 2 * P(S_N - p > \varepsilon)$$

Again by taking the largest possible variance of S_N – p to be $\frac{1}{4}.N$, we use the normal approximation

$$P(S_N - p > \varepsilon) \leq 1 - \Phi(z)$$

where

$$\Phi(z) = 2.\varepsilon.\sqrt{N}$$

and $\Phi(z)$ is the normal distribution function.

Now consider the problem of choosing N given the accuracy of our estimate of the coverage area to be within 1% of the actual coverage area with probability at least 0.9999. Then

$$P(|S_N - p| > \varepsilon) \approx 2 - 2\Phi(2.\frac{1}{100}.\sqrt{N})$$

And

$$2 - 2\Phi(2.\frac{1}{100}.\sqrt{N}) \leq \frac{1}{10000}$$

Rearranging this inequality we get,

$$\Phi(2.\frac{1}{100}.\sqrt{N}) \geq 0.99995$$

From the normal distribution tables, we see that

$$\Phi(3.8906) = 0.99995$$

Then

$$2.\frac{1}{100}.\sqrt{N} \geq 3.8906$$

and

$$N \geq 37,842$$

Thus, if we require our estimate of the coverage area of a set of nodes to be within 1% of the actual coverage area with probability 0.9999 then we require at least 37,842 sample points to be chosen. Table 5 illustrates a table corresponding to Table 4 using this revised calculation for N.

accuracy/probability	0.95	0.99	0.9999
10%	96	166	378
5%	384	663	1,514
1%	9,604	16,587	37,842

Table 5: Improved values of N

3.2.4 *Performance evaluation of the coverage estimation algorithm*

The time it takes to carry out the coverage estimation depends on a number of parameters:

- the probability/accuracy pair desired, i.e. the number of samples to be genrated and tested, and
- the number of shapes to be tested, i.e. the number of nodes covering the area,
- the algorithm used for the random number generation[2].
- the complexity of the *inside()* function[3],

We now analyse the effects of the three first parameters. We use a simple circle for the coverage area of each node, and then the *inside()* function is as given in Table 3. The most important parameter is obviously the desired accuracy of the coverage estimation, it greatly influences the cost of the algorithm since it determines the number of samples to be used in the computation. We saw for instance that for a probability of 99.99% and an accuracy of 1%, the total number of samples must be 37,842. Figure 3 shows the time spent by the algorithm for estimating the coverage of 1000 nodes (the range of a node being 50 meters) over a circular area of radius 1000 meters [4]. From this experiment, we can see that the time spent to estimate the coverage of the area is directly proportionnal to the number of samples and a rough estimate of the cost of generating and computing each sample is 76 microseconds.

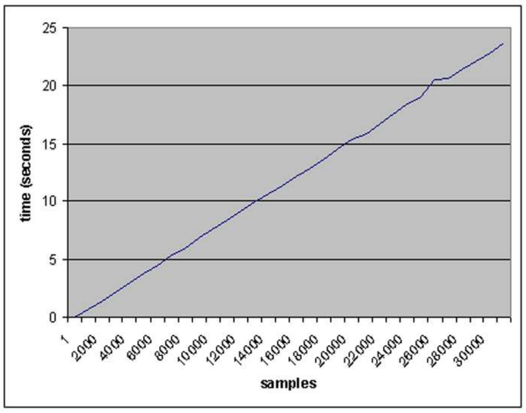

Figure 3: Time of coverage estimation for 1000 nodes

[2]The random number generator used influences both the performance and the accuracy of the algorithm. Good random generators are expensive to run but provide better results.

[3]This parameter depends on the precision of the information concerning the coverage of a node. In practice the shape used will often be a circle and then, the *inside()* function be rather simple (c.f. Table 3).

[4]This experiment has been carried out on a Pentium II 650 Mhz running RedHat Linux 7.1.

The second most important parameter is the number of nodes covering the area. This parameter is influenced by both the size of the area and the density of the network in that area. Figure 4 shows the time spent by the algorithm using 37,842 samples for a total number of nodes varying from 1 to 1000. This simulation shows that the computational overhead is quite reasonable. It may seem proportional to the number of nodes but as shown on Figure 5, the time of the coverage estimation divided by the number of nodes decreases with the number of nodes. This is due to the fact that a sample that has been found in a circle doesn't need to be checked against other circles.

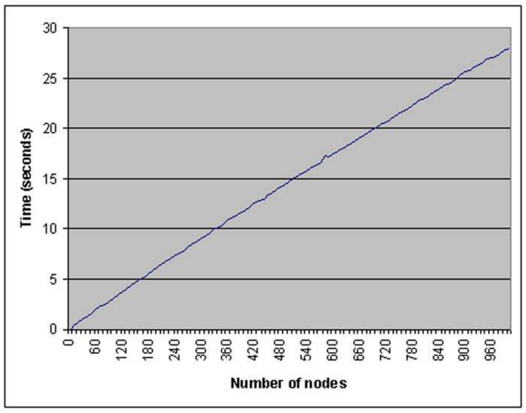

Figure 4: Time of coverage estimation for 37,842 samples

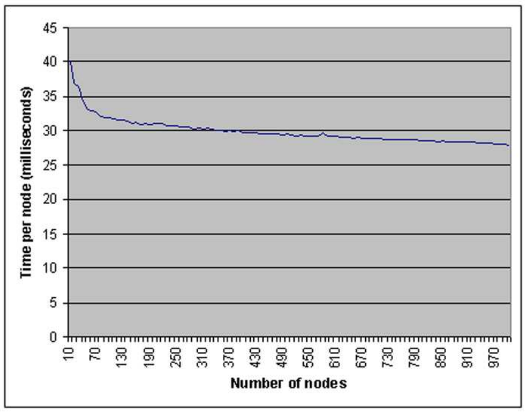

Figure 5: Time of coverage estimation divided by the number of nodes for 37,842 samples

Finally, particular attention must be given to the algorithm used for the random number generation. The quality of the generator greatly influences the validity of the algorithm; the proof we gave above is based on completely independent samples. A lot of pseudo-random number generators exist, for instance we used both the system provided number generator and a "minimal" generator of Park and Miller with Bays-Durham shuffles [20].

3.2.5 Partition anticipation

In our model, partitions can be due to the failure of either a node or the failure of a link between two nodes. Various reasons can cause a node to fail such as an operating system or hardware crash, or simply a drained battery. A link can fail because of the movement of the nodes or because of other environmental conditions such as obstacles or interference. We aim at anticipating partitions in order to keep consistent views of the group membership. As we use a probabilistic approach, we distinguish two different cases: either the probability of a partition is beyond a given threshold T and then we use an optimistic algorithm or it is below T and thus we run in a pessimistic mode. Handling a partition is obviously easier when it was anticipated (when we were in the pessimistic mode), but we also provide some kind of recovery procedure when it was not anticipated (optimistic mode).

The probability of a partition occurring is given by partition anticipators that are a combination of failure anticipators, movement planners and environment evaluators.

Failure anticipators are responsible for suspecting nodes of crashing, having a low battery level, entering a power saving mode, etc. In our model, nodes are fail-silent, fail-still and can recover. The metric used to evaluate the probability of a node failing involves both local information (battery life) and distributed information (crash, suspicious neighbours). Each node has a failure anticipator that evaluates a list of other nodes such as its neighbours and eventually its most accessed partners. Some of the information necessary for failure anticipation is provided in the beacons used by the location service described above, e.g., the battery level.

Movement planners are based on [7], they use the notion of "safe distance" to determine the probability of a node failing because of movement. Roughly, if two nodes are not within a safe distance, the link between them is considered to have failed and, if this link represents the only connection between two sets of nodes, a partition can occur. Movement planners obviously rely on the connectivity graph built by the location service described above. Additionally, the movement planners can eventually use knowledge of the direction of the nodes as well as their velocity to evaluate the link.

Environment evaluators can be used to share knowledge about some environmental conditions that could potentially disturb communications. For instance, a node that is aware of the presence of an obstacle in some area or about a truck that causes radio interferences can tell the other members the location (eventually direction and velocity) of the obstacle that may cause a partition.

4. RELATED WORK

In [6], a simple architecture for group communication in mobile systems is proposed. The key idea is to create a group of all the nodes that are within a given distance D from the group creator *gc*. Using our definition of a group it is similar to : G={*circle C of radius D, center of C, gc, name*}.

Their model does not consider disconnection or partition within a group. The proposed architecture is composed of two different layers. The proximity layer consists of a protocol that uses the underlying MAC sublayer to find all nodes that are within a given distance from the mobile host. It uses flooding for the discovery phase and convergecast for the replies. The group membership layer uses a three-round protocol that (1) proposes to the nodes discovered by the proximity layer to become members of the group, (2) allows them to reply and then (3) confirms their membership. This solution suffers from a number of drawbacks. Firstly, the first phase uses pure flooding to discover the nodes located in the area, and therefore does not scale well. For instance, in a traffic management scenario, every single node of the network will receive and repeat every flood message. A second drawback is the restrictive definition of a group and the fact that a node can be involved in only a single group.

In [7], as explained above, the proximity group is defined by the notion of safe distance. Each node has to be within this safe distance of its nearest neighbour to be considered as a member of the group. This notion of a group is very restrictive and does not cope well with the common understanding of group communication but is nevertheless particularly interesting for implementing partition awareness and anticipation.

5. SUMMARY, CONCLUSIONS AND FUTURE WORK

In this paper, we described a model of group membership for location-aware mobile participants that is at the heart of a new group communication toolkit for wireless networks that we are developing. In this model, eligibility for group membership depends on the location of the potential member and, in particular, each group is associated with a static or a mobile area of interest within which its members should be located. This model is aimed primarily at applications in the traffic management [18], smart space [19] and augmented reality domains. We also described some of the considerations underlying our approach to group membership management that exploits location information to achieve coverage estimation and partition anticipation. We are currently developing a suite of multicast protocols providing different ordering, reliability and timeliness guarantees based on this membership substrate.

6. REFERENCES

[1] M. Weiser, "Some Computer Science Issues in Ubiquitous Computing," *Communications of the ACM*, vol. 7, pp. 74-83, 1993.

[2] K. P. Birman, *Building Secure and Reliable Network Applications*: Prentice Hall Professional Reference Series and Manning Publishing Company, 1997.

[3] F. Cristian, "Synchronous and Asynchronous Group Communication," *Communications of the ACM*, vol. 39, 1996.

[4] E. M. Royer and C. E. Perkins, "Multicast Operation of an Ad Hoc On-demand Distance Vector Routing Protocol," presented at MobiCom, Seattle, WA, 1999.

[5] Y.-B. Ko and N. H. Vaidya, "GeoTORA: A Protocol

for Geocasting in Mobile Ad Hoc Networks," presented at 8th International Conference on Network Protocols (ICNP), Osaka, Japan, 2000.

[6] R. Prakash and R. Baldoni, "Architecture for Group Communication in Mobile Systems," presented at Symposium on Reliable Distributed Systems, West-Lafayette (IN), USA, 1998.

[7] G.-C. Roman, Q. Huang, and A. Hazemi, "On maintaining Group Membership Data in Ad Hoc Networks," Washington University, St Louis, Technical Report wucs-00-26, April 16, 2000 2000.

[8] F. Cristian, H. Aghili, R. Strong, and D. Dolev, "Fault-Tolerant Atomic Broadcast Protocols," presented at 15th International Conference on Fault-Tolerant Computing (FTCS), Ann Arbor, Michigan, USA, 1985.

[9] Y. Amir, D. Dolev, S. Kramer, and D. Malki, "Transis: a Communication Subsystem for High Availability," presented at 22nd International Symposium on Fault Tolerant Computing (FTCS-22), Boston, Massachusetts, USA, 1992.

[10] J. Broch, D. Maltz, D. Johnson, Y.-C. Hu, and J. Jetcheva, "A performance comparison of multi-hop wireless ad-hoc network routing protocols," presented at The Fourth Annual International Conference on Mobile Computing and Networking, MobiCom 2000, Dallas, Texas, US, 1998.

[11] A. Chandra, V. Gumalla, and J. O. Limb, "Wireless Medium Access Control Protocols," in *IEEE Communications Surveys & Tutorials*, 2000.

[12] Y.-B. Ko and N. H. Vaidya, "Location-aided routing (LAR) in mobile ad hoc networks," presented at ACM/IEEE Int. Conf. on Mobile Computing and Networking (MobiCom'98), 1998.

[13] B. Karp and H. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," presented at 6th Annual Int. Conf. on Mobile Computing and Networking (MobiCom), Boston, MA, USA, 2000.

[14] IEEE, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," in *IEEE Std 802.11-1997*. New York: The Institute of Electrical and Electronic Engineers, 1997.

[15] C. E. Perkins, E. Royer, and S. R. Das, "Ad Hoc On Demand Distance Vector (AODV) algorithm," presented at 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'99), New Orleans, Louisiana, USA, 1999.

[16] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu, "The Broadcast Storm Problem in a Mobile Ad Hoc Network," presented at ACM/IEEE Mobicom, 1999.

[17] S. M. Ross, *Introduction to Probability and Statistics for Engineers and Scientists*, 2nd ed: John Wiley & Sons, 1999.

[18] R. Cunningham and V. Cahill, "System Support for Smart Cars: Requirements and Research Directions," presented at 9th ACM SIGOPS European Workshop, Kolding, Denmark, 2000.

[19] P. Nixon, S. Dobson, and G. Lacey, "Smart Environments: Some challenges for the computer community," presented at 1st International Workshop on Managing Interactions in Smart Environments, Dublin, Ireland, 1999.

[20] William H. Press, B. P. Flannery, S. A. Teukolsky and W. T. Vetterling, "Numerical Recipes in C : The Art of Scientific Computing", Cambridge University Press, 1986.