Routing Group Formation in Ambient Networks

Abigail Surtees*, Ramón Agüero†, Jari Tenhunen‡, Michele Rossi§ and Daniel Hollos¶
*Siemens/Roke Manor Research Ltd, UK, abigail.surtees@roke.co.uk
†University of Cantabria, Spain, ramon@tlmat.unican.es
‡VTT Technical Research Centre of Finland, jari.tenhunen@vtt.fi
§Consorzio Ferrara Ricerche (CFR), Italy, mrossi@ing.unife.it
¶Technische Universität Berlin (TUB), Germany, hollos@tkn.tu-berlin.de

***Abstract*— Mobile nodes are generally treated individually and in an uncoordinated fashion, even when nodes are, in fact, moving as a group. However, the ability to recognize that the nodes are travelling together enables a number of important optimizations to routing and mobility management mechanisms, in particular, aggregation and delegation of mobility management and use of ad-hoc routing protocols. The key challenges to providing these optimizations are recognizing the presence of a group of nodes, assessing the stability of the group, and deciding what optimizations are appropriate. Developing solutions to these challenges has been a key focus of the mobility work within the Ambient Networks project and this paper discusses these challenges and presents the solutions developed.**

I. Introduction

Ambient Networks [1] is a collaborative project within the European Sixth Framework Programme, targeted at investigating the networking aspects of mobile systems beyond today's 3rd generation standards. The concept of an Ambient Network embraces the heterogeneity arising from different network control technologies so that it appears homogeneous to the potential users of the network services. The vision is to allow network composition - the agreement for co-operation between networks on demand, transparently and without the need of pre-configuration or offline negotiation between network operators [2], [3]. Mobility mechanisms and the possibility to make use of optimizations when appropriate are particularly important. Where nodes are moving together in a group, there is the potential for the following optimizations:

- Mobility management optimizations - as the devices in the group are all moving together, they will be handing over at the same time. Consequently, it could be beneficial if the mobility management for all the nodes were to be aggregated, thus reducing the amount of signalling needed to handover.
- Routing optimizations - due to the fact that the devices are in close physical proximity, it is possible to provide more efficient routes for traffic between nodes in the group. These optimizations can be divided into two classes:
  - Local routing: e.g. using an ad-hoc routing protocol between nodes becomes possible because they are in close proximity.
  - Routing in the network: e.g. performing optimizations in the network that take advantage of the fact that certain nodes are in close physical proximity.

The following two concepts are defined:

- A (physical) cluster is a group of nodes that are physically near to each other, are likely to stay near to each other, and are able to communicate.
- A routing group (RG) may be formed by a subset of the nodes within the (physical) cluster. Formation of a routing group requires exchange of information and agreement between the nodes involved.

The distinction between a (physical) cluster and a RG is that the nodes in the cluster are not aware of group membership, whilst the nodes in the RG are aware of group membership. The concept of RG allows more possibilities for routing and mobility optimizations. Conceptually, a RG represents a specific Ambient Network scenario where particular mobility and routing optimizations can be applied.

In order to make use of RGs and the associated optimizations, it must be possible to form a RG. The formation and maintenance of RGs bring significant challenges. Discussion of these challenges and presentation of the proposed solutions to them form the rest of this paper which is structured as follows: Section II briefly reviews earlier work that has been done on clustering algorithms and discusses the challenges associated with forming a RG and an overall approach to so-doing. Section III presents three proposals for RG formation algorithms. The first is a mechanism for initially triggering the formation process. The second and third are traditional in style but consider more input information than the algorithms mentioned in Section II. Finally, Section IV concludes the paper, introducing future work that will be carried out.

II. Routing Group Formation

A. Challenges

The RG concept entails some interesting challenges, which are not completely addressed by existing clustering algorithms. As, by definition, there must be agreement between nodes to be a RG, there must be some way for the agreements, and therefore RGs, to be formed. The concept of network clustering has been studied for many years [4], especially in conjunction with ad hoc routing protocols e.g. CBRP [5]. The idea of using patterns of node mobility as clustering criteria is, however, relatively new [6]. One of the presented algorithms is partly based on the concept of associativity (based on stability assessments) [7], which has also been applied to clustering [8], [9].

The RG formation objective is to assess whether there is a set of relatively stable nodes, and if so, both whether they should form a RG and which nodes should be part of it. Whereas, the basic objective of almost all existing proposals is to partition (in an efficient way) a fixed group of nodes into clusters and expect all nodes to end up in one cluster or another, this is not the case in the RG scenario.

Another drawback of existing solutions is that there is normally a limit on either the radius of the cluster (the maximum number of hops) or the number of components. The RG concept goes beyond this view, as there are not, theoretically, any bounds. Nonetheless, it is clear that there will indeed exist a limit, given by the costs associated with the formation and maintenance of a RG with larger radius.

Existing solutions do not consider Multi-RAT (*Radio Access Technologies*) scenarios, as they assume that some means of local communication always exists. In RG, however, the scope is broader, as two nodes which are not able to communicate directly could belong to the same RG, as long as there is another node that forwards traffic from one to the other. Considering these scenarios, brings challenges, both in terms of dealing with multiple RATs e.g. in acquiring locators and the general challenges of the *multi-hop multi-homed* scenario, such as synchronization of nodes and avoiding routing loops.

Another important difference comes from the membership decisions. Traditional clustering algorithms do not pay attention to these aspects. Within a RG, though, there are various ways to decide upon the acceptance of a new member, including a centralized approach (i.e. there is one node which actually takes the decision) and a distributed approach (the decision is taken jointly by a number of nodes).

One of the most important differences between existing solutions and the RG algorithm is the number of inputs that are considered by the algorithm. The RG algorithm would be flexible both in the number and range of inputs and in the ability to be extended. For example, the following could be used to assess stability or willingness of a node to take an extra role in the RG, such as gateway or *cluster-head*:

- Number of neighbors - this is probably the most relevant parameter for assessing stability.
- Link quality - either using error rates (FER, PER) or signal strength (SNR), it can be used to refine stability measurements.
- Indication of level of mobility - if available, it can be used to improve stability measurement.
- Battery levels - it may be used to assess nodes' willingness to take on an extra role e.g. gateway or *cluster-head*, the more the battery level, the more appropriate for such role.
- Traffic load - it could be an interesting parameter to look at when determining the most appropriate gateway.

Many of the algorithms use an exchange of *HELLO* messages in order to discover the information required to form the clusters. However, even a simple message exchange can use up large amounts of bandwidth, which, particularly in the wireless environment, may be limited. Therefore, the cost of attempting to form a RG, as well as maintaining it must be taken into account.

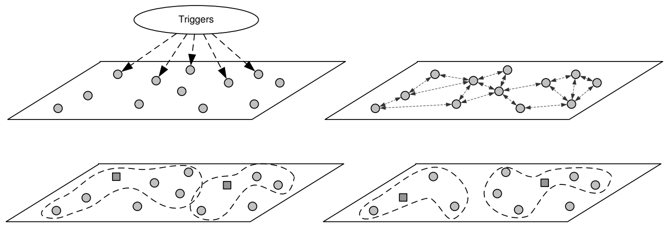

Fig. 1. Formation Stages

B. Formation Stages

As a result of the analysis, RG formation is defined to have four stages, as shown in Fig. 1:

1) Stage 1 is where there is some form of trigger (internal or external (see, for example, Section III-A)) to the node that it is potentially worth attempting to form a RG.
2) During stage 2, a basic exchange of information takes place between nodes, to allow them to establish which among them are suited to being part of a RG. Part of this process may involve exchanging information to establish the control of the RG (either electing one or more *cluster-heads* or fully distributed control).
3) Alternatively or additionally establishment of control may be part of the stage 3 in which extra information can be exchanged and the membership of the RG can be refined. For example, QoS and security requirements may be taken into account during this stage.
4) Stage 4 is that of maintaining the RG once it has been established. In particular, this may relate to new nodes joining or existing member nodes leaving, however, it is not limited to these processes.

III. Proposed Solutions

This section will describe three different solutions proposed to create and maintain RGs. The first one (Section III-A) exploits the triggers provided by external entities to assess when it is worth forming a RG. The other two solutions follow a more traditional approach, where the nodes cooperate to create and maintain RGs in a distributed manner. It should be observed that, even though the three approaches can be used as alternatives, the first solution could also be exploited in conjunction with either of the latter two algorithms, i.e., to actually measure the worthiness of forming a RG and subsequently trigger a distributed RG formation and maintenance stage. The final section discusses further developments required for stage 3 of RG formation.

A. Cell-Based Triggering

The main goal of this procedure (which is a stage 1 solution) is to use information provided by external entities to assess whether the creation of a RG is feasible and useful. This decision includes determining whether the benefits provided by the RG outweigh the costs of creating and maintaining it. This is in contrast to traditional clustering mechanisms, where nodes execute the expensive discovery procedures without knowing whether it is worth doing so or not. The mechanisms are expensive due to the fact that, in order to assess link states, traditional clustering algorithms perform periodic *HELLO* message exchange between the nodes. This results in considerable overhead for the following reasons: (i) the receiver circuits must be kept switched on to receive such messages, which consumes power, (ii) receiver circuits must scan the available frequencies periodically, while many devices can not perform user data exchange during the frequency scan procedure, and (iii) the *HELLO* messages must be transmitted periodically, consuming power as well.

Consequently, a novel method that reduces this overhead by running the traditional style algorithms only when there is some degree of confidence that a RG will be formed and is useful is presented. The fixed infrastructure is used to recognize that a group of nodes is moving together and assess whether there may be possible benefits to the nodes if a RG is formed (addressing point (iii) above). As the infrastructure does not have all the information that the nodes do, the assessment will be relatively general but should nonetheless prevent occurrences where the nodes run a traditional style algorithm, when it was obvious to the network that there aren't sufficiently stable nodes and so RG formation fails. This method reduces the number of frequencies on which nodes need to periodically scan (addressing point (ii)) because it only expects messages from the network, rather than *any* neighboring node.

The basic idea uses the fact that if some nodes are moving together they will be associated with the same sequence of Access Points (AP)[1]. In Fig. 2 all three nodes are communicating through an AP, although direct communication may be possible. By analyzing the association tables from AP1 and AP2, it can be noted that the nodes A, B, C were first all associated with AP1 and then with AP2, therefore, it can be deduced that it is likely that they are moving together. AP1 and AP2 can exchange the relevant information to make this deduction and then AP2 can send triggers to the nodes to indicate that it would be worth attempting to create a RG using one of the algorithms proposed in the subsequent sections. For greater certainty that nodes are travelling together, the information could be exchanged between more than two APs.

The algorithm for the above method is as follows. For simplicity, assume only two APs are involved and that nodes handover from AP1 to AP2. As each node hands over the following two steps will be executed:

1) AP2 identifies the set of nodes S_1 that is currently associated with it, that handed over from AP1. If S_1 is not empty, execute step two
2) AP2 sends S_1 to AP1; AP1 identifies the set of nodes $S_2 \subset S_1$, where all nodes in S_2 were simultaneously associated with AP1 for a given (arbitrary) period of time

The nodes in S_2 (if any) are possibly moving together and can be sent triggers that it may be worth forming a RG. Note that

[1]The acronym *AP* is used for the cell representative of any cell-based systems, regardless of the access technology in use.

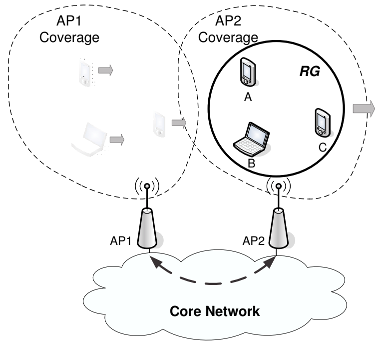

Fig. 2. Cell-based triggering

in order to increase the recognition confidence the nodes may need to pass several APs together to get triggered.

As can be seen, the cell-based approach does not aim at replacing a traditional approach, but at complementing it, as it determines whether or not it is worth initiating the corresponding RG operations.

B. Associativity Based Algorithm

This algorithm (a stage 2 algorithm) assesses nodes' stabilities with an associativity-based method. The stability metric is obtained from *associativity ticks* and link quality, using either *packet error rate* (PER) or *signal-to-noise ratio* (SNR) as parameters. The algorithm's operation can itself be divided into three phases: *neighbor discovery*, *cluster-head selection*, and *maintenance phase* (Fig. 3).

1) Neighbor Discovery: It is assumed that if the neighboring nodes are reachable and stable, they reside nearby. For each of its neighbors, every node maintains a stability value that is updated periodically. In this sense, all nodes transmit *HELLO* messages (every T_{H} seconds); then, every time such a message is received by a node, it increases the stability value of the neighbor that sent the message. Correspondingly, if no *HELLO* message is received within a defined time-frame, the stability value is decreased. *HELLO* messages are not forwarded, so they are only heard by the sender's first neighbors (those which are directly reachable from the node or within its coverage area). When the stability value of a node exceeds a certain threshold, the node is considered stable.

The node stability values are adjusted using the link quality parameters. When bad quality reception (erroneous/lost packets or low SNR) is detected, the stability value associated with the sender node is decreased, meaning that additional successful *HELLO* messages must be received until the sender will be considered stable again.

2) Cluster-head Election: The algorithm supports the presence of *cluster-heads* that are elected using an easily extensible mechanism. The main selection criteria are the degree of connectivity and the stability of the links to neighboring nodes, as *cluster-heads* are likely to have a significant role within the communications occurring inside the RG. Nonetheless, the design is flexible enough to be able to add other context

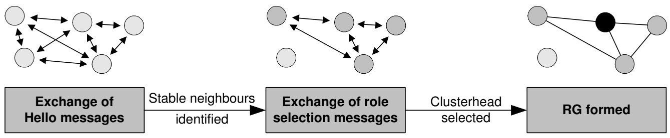

Fig. 3. Overview of the associativity based algorithm

information (such as battery level) that might have an effect on a node's stability and suitability to be a cluster-head.

3) Maintenance Phase: In the maintenance phase new nodes may join the RG, former members may leave, or two RGs may merge into one. Since periodic transmission of *HELLO* messages does not stop after the RG is formed, it is easy to detect nodes that have left the RG. Correspondingly, new nodes are detected when they start to send *HELLO* messages, and they are able to join the RG after their stability towards one of the RG members is enough. Similarly, two RG may merge whenever a node in one of them establishes a stable link with a node in the other. The decision of whether to accept a new member, or to merge two existing RG could be taken by a central entity (e.g. by the *cluster-head*, if it exists at all) or in a distributed manner.

C. Candidate set algorithm 2: PAW: Propagation of Aggregation Willingness

This algorithm is also a solution for stage 2 of RG formation. Its basic idea is that the whole RG procedure is based on a "distributed collective willingness" to form a RG. The algorithm called Propagation of Aggregation Willingness (PAW) has two versions. In the first one (PAW-D, "distributed"), nodes aggregate without the need to elect a leader. In the second (PAW-L, "leader-based"), coordinating entities ("*cluster-heads*" or leaders) are elected within the RG and used to evaluate the RGs. This algorithm uses inputs such as number of neighbors and node energy.

In the following, the neighbor set will be considered as a vector S containing the identifiers of all the detected neighboring nodes, i.e., of the nodes whose messages are received with a minimum quality. A node i is a neighbor of another given node j if both $i \rightarrow j$ and $j \rightarrow i$ are verified. To monitor this set, it is sufficient that each node periodically (not necessarily at regular intervals) broadcasts *HELLO* messages.

1) Distributed version of PAW (PAW-D): In this version of the algorithm, these metrics are sent to the one hop neighbors in an appropriate packet (*HELLO*) which also includes a parameter indicating the willingness of the node to participate to a RG. This willingness is at first determined by local measurements, i.e., stability of the neighbor set. However, this metric evolves according to the exchange of *HELLO* packets with neighbors, i.e., the decision to create a RG or to participate to an existing RG can be either strengthened or weakened by similar decisions performed by other neighbor nodes. To sum up, stability metrics are gathered at each node and dynamically transmitted in order to reach consensus about the need for RG formation among nodes. The algorithm steps are as follows:

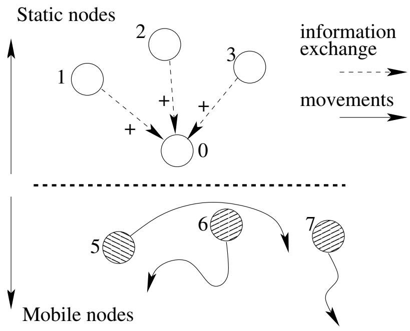

Fig. 4. Reinforcement of RG formation decision

1) Each node computes its neighbor stability based on measurements of its neighbor set. This stability value is a function of the "variability" of its neighbor set, therefore accounting for arrivals/departures and nodes that remain constant (loyal) over the time. The neighbor stability corresponds to the (local) willingness of a given node to form a RG.
2) After having computed its willingness, each node transmits a *HELLO* packet to its neighbors, which contains its willingness (stability) value and the list of loyal follower nodes, i.e. the nodes that have been stable over a sufficiently long period of time.
3) Each node receiving the *HELLO* checks for its own inclusion in the loyal follower list therein and, in the positive case, adds the willingness value contained in the packet to its own local willingness (stability). Thereby, the willingness of a node to join a RG is reinforced.
4) Each node joins the RG if the accumulated willingness value exceeds a given threshold thr.

In the example network shown in Fig. 4, there are two different node behaviors. In the upper part of the network nodes are stationary, while nodes below the sketched line are mobile. Node 0 in the figure has a neighbor stability value which depends on its neighbor set $S_0 = (1, 2, 3, 5, 6, 7)$. Nodes $(1, 2, 3)$ are static and, therefore, positively contribute to the stability metric, while the same metric is reduced by nodes $(5, 6, 7)$, that are on the move. In such a case, node 0 may be undecided about the need for RG formation, i.e., to join the existing RG (node 1, 2 and 3). In such a case, its stability metric is below the RG activation threshold (thr). However, thanks to the reinforcements ($+$) coming from nodes $(1, 2, 3)$ node 0 is able to refine (increase) its own belief that it is a stationary node and, even more importantly, to have bidirectionally stable links to nodes $(1, 2, 3)$. In this case, the reinforcement procedure makes it possible for node 0 to join the RG by "filtering out" the oscillatory effect on its stability metric which can be present due to the surrounding mobile nodes.

2) Leader-based version of PAW (PAW-L): As in PAW-D, each node computes its own stability metric. However, differently from PAW-D, every node is also characterized by a weight (W) which is obtained as a combination of several factors including, but not limited to, node energy, mobility status (if available), and number of neighbors. W is intended to evaluate the suitability of a given node to act as a coordinating entity. After this, every node decides whether it should become a coordinating entity and such a decision is based on a given threshold on its weight $W(thr)$. Every coordinating entity periodically sends *HELLO* packets including the set of loyal nodes, i.e., the most stable in its neighborhood, W and its willingness to form a RG. Normal nodes, i.e., those nodes with a weight below $W(thr)$ are not allowed to send *HELLO* packets and thereby to propagate their stability. Every node receiving a *HELLO* checks for its own inclusion in the loyal nodes list contained therein and, in the positive case, it also checks whether the sending node is a stable neighbor. Finally, if both these checks are passed, the node adds the sending node to its *candidate leader list*. When a node $i \in \mathcal{N}$ receives a *HELLO* message with weight W_j from node $j \in \mathcal{N}$ it performs the following check:

1) If the node is not a member of any RG, then it joins node j, thereby becoming a member of the RG leaded by j.
2) Otherwise, the node checks for the node (i_{max}) in its proposed candidate leader list with the highest weight ($W_{i_{max}}$), i.e., the node's current leader, and, if $W_j > W_{i_{max}} + h$, it leaves i_{max} and joins j where h is an opportune threshold. This threshold mechanism is similar to that in [10] and is used to avoid oscillatory behaviors.

In addition to the two steps above, it is sensible to use link qualities $Q_{j \to i}$ and stability measurements $S_{j \to i}$ to refine node weights. That is, W_j above may be scaled as $W'_j = f(Q_{j \to i}, S_{j \to i}, W_j)$, where $f(\cdot)$ is a generic function, whereas Q and S are the quality and the stability metrics associated with link $j \to i$, respectively. In this case, the above check is performed using W'_j instead of W_j.

D. RG Membership Refinements

In stages 1 and 2 moving clusters are recognized based on the mobility pattern and some simple parameters of nodes keeping the communication overhead as low as possible. However, there are other aspects to be taken into account when making the final RG membership decisions. For example, a node may participate in the RG only if a specific service (e.g. a path to node B exists) is offered by the RG; or taking into account security issues (node B is allowed to relay for node A). Such information will be exchanged only during stage 3 of formation. Detailed design of the algorithms for this stage has not yet been done, however, the solutions presented have been designed in an extensible way.

These refinements also relate to other aspects of Ambient Networks, for example, security and composition [11].

IV. Conclusion and Future Work

This paper has discussed the challenges introduced when attempting to form routing groups and proposed three methods for forming routing groups. Two are similar to traditional clustering algorithms, however, they take into consideration the need to recognize that a group of nodes is moving together. These include novel ideas like the propagation of willingness approach to reach distributed consensus among nodes about the cluster memberships and stability assessments based on associativity. The third solution, makes use of the fixed infrastructure to assess whether nodes are travelling together, and if so, trigger them to begin one of the traditional style formation algorithms. This allows the nodes to reduce the formation overhead by not attempting to form a routing group when the isn't a sufficiently stable set of nodes.

On-going simulation work is being done to assert that the algorithms do lead to the expected formation of routing groups. Further work will be done in this area and also in the refinement of the algorithms.

Acknowledgements

This document has been produced in the context of the Ambient Networks Project. The Ambient Networks Project is part of the European Community's Sixth Framework Program for research and is as such funded by the European Commission. All information in this document is provided "as is" and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability. For the avoidance of all doubts, the European Commission has no liability in respect of this document, which is merely representing the authors view. The authors thank people involved in Work Package 4, in particular, those involved in Task 3.

References

[1] "IST Ambient Networks," www.ambient-networks.org.
[2] N. Niebert, H. Flinck, R. Hancock, H. Karl, and C. Prehofer, "Ambient Networks - Research for Communication Networks Beyond 3G," in *13th IST Mobile and Wireless Communications*, June 2004.
[3] P. Eardley, R. Hancock, and E. Hepworth, "Ambient Internetworking: An Architecture for Extending 3rd Generation Mobile Networks," in *5th IEE International Conference on 3G Mobile Communication Technologies*, Oct. 2004.
[4] D. J. Baker and A. Ephremides, "The architectural organization of a mobile radio network via a distributed algorithm," *IEEE Transactions on Communications*, vol. COM-29, no. 11, pp. 1694–1701, Nov. 1981.
[5] M. Jiang, J. Lie, and Y. Tay, "Cluster Based Routing Protocol (CBRP)," July 1999, IETF Internet-Draft, http://www.comp.nus.edu.sg/∼tayyc/cbrp/draft-ietf-manet-cbrp-spec-01.txt, work in progress.
[6] I. I. Er and W. K. Seah, "Mobility-based d-hop clustering algorithm for mobile ad hoc networks," in *IEEE Wireless Communications and Networking Conference*, Atlanta, Georgia, USA, Mar. 2004.
[7] C.-K. Toh, "Associativity-Based Routing For Ad-Hoc Mobile Networks," *Kluwer Wireless Personal Communications*, vol. 4, no. 2, pp. 103–139, Mar. 1997.
[8] A. Ramalingam, S. Subramani, and K. Perumalsamy, "Associativity based cluster formation and cluster management in ad hoc networks," in *9th International Conference on High Performance Computing*, December 18-21, 2002, Bangalore, India, Dec. 2002.
[9] Y. Choi and D. Park, "Associativity Based Clustering and Query Stride for On-demand Routing Protocols in Ad Hoc Networks," *Journal of Communications and Networks (KICS)*, vol. 4, no. 1, Mar. 2002.
[10] S. Basagni, "Distributed and Mobility-Adaptive Clustering for Multimedia Support in Multi-Hop Wireless Networks," in *In Proceedings of IEEE VTC Fall 1999*, Amsterdam, The Netherlands, 1999.
[11] Campos et al, "Scenarios for Network Composition in Ambient Networks: a new paradigm for Internetworking," June 2004, Wireless World Research Forum.