Application-oriented Self-organizing Hierarchical Clustering in Dynamic Networks

[Position Paper]

Andréa W. Richa[1], Katia Obraczka[2], and Arunabha Sen[3]

ABSTRACT

We are currently developping a suite of self-organizing clustering algorithms which will automatically adapt to network dynamics in response to application-specific requirements. As a starting point, we identify groups of dynamic network applications, or *application classes*, which will use a common family of clustering algorithms. Within an application class, the performance requirements of individual applications is translated into a common set of clustering constraints — these constraints determine which clustering algorithm(s) is (are) well suited for each application class. We present some initial ideas for developping clustering algorithms for a particular application class, which contains power-aware routing applications.

[1] Department of Computer Science and Engineering, Arizona State University, Tempe, AZ 85287–5406, aricha@asu.edu. This work was supported in part by NSF CAREER Award CCR–9985284.
[2] Department of Computer Engineering, University of California at Santa Cruz, Santa Cruz, CA 95064, katia@cse.ucsc.edu. This work was supported in part by NSF NGI grant number ANI-9813724.
[3]Department of Computer Science and Engineering, Arizona State University, Tempe, AZ 85287–5406, asen@asu.edu.

POMC 2001 Newport, Rhode Island USA

1. INTRODUCTION

In the near future, not only will networks grow two or more orders of magnitude in scale, but they will also become considerably more complex and heterogeneous. Using both wired as well as wireless communication technology, future networks will likely interconnect a wide range of devices from traditional desktops, laptops, and personal digital assistants (PDAs), to non-conventional devices such as home appliances, sensors, and actuators. We anticipate that a typical internetwork of the future will likely consist of several wired backbones, and a collection of fixed, mobile, and ad hoc stub networks. These large-scale, highly-heterogeneous networks will be far more challenging to engineer, manage, and control than today's networks.

To achieve scalability, distributed systems designers have made extensive use of hierarchical clustering. Internet routing is a good example: the Internet is decomposed into a collection of clusters, or *autonomous systems* (ASs). For inter-AS routing, ASs are abstracted as a collection of *border routers*. Once within an AS, packets are routed according to the specific intra-AS routing mechanism. Note that besides scalability, clustering hierarchies also address administrative decentralization and autonomy.

However, most existing hierarchical systems rely heavily on system administrators to manually configure and maintain their hierarchies. This includes maintaining cluster membership information at the nodes, electing clusterheads, and selecting gateway (or border) nodes. Clearly, manual cluster maintenance does not scale and fails to capture the dynamics of the underlying network.

We are currently developing a suite of self-organizing clustering algorithms which automatically adapt to network dynamics in response to application-specific requirements. One contribution of our work is to extend existing clustering algorithms developed by the network algorithms community with novel clustering techniques to address the performance requirements of various existing and future applications.

As a starting point, we introduce the concept of *application classes*: an *application class* defines a group of applications that use a common family of clustering algorithms. Within an application class, the performance requirements of individual applications are translated into a common set of clustering constraints. Let us take network routing as an example application domain. In "traditional" hierarchical shortest-path routing, the goal of the clustering algorithm is to produce shortest-path routes within clusters. However, if we consider routing in a network of thousands of power-anemic devices like sensors, the goals of the clustering algorithm will be totally different: it will try to maximize the network's overall lifetime by using energy-conserving clustering constraints. Thus, even though these are two instances of the routing problem, they belong to different application classes. Now consider yet another instance of the routing problem where the goal is to minimize router processing time. These two minimization problems — of minimizing energy consumption, and of minimizing processing time at network nodes — are equivalent and thus can be grouped into the same application class.

In Section 3, we identify some application classes and their clustering constraints. The set of clustering constraints associated with an application express features specific to the application, including characteristics of the application's underlying network environment (such as whether it is a stationary or a mobile network). These constraints determine which clustering algorithm(s) is well suited for each application class.

To run efficiently on a heterogeneous and dynamic network environment, the proposed algorithms will also adapt to a wide range of network conditions. For instance, in a stationary backbone network, the algorithms should be mainly concerned with adapting to faulty nodes and links whose costs change; if part of the network is composed of an ad hoc network, the algorithms will likely have to adapt to more frequent topology changes due to node mobility. This higher degree of adaptability will probably be achieved at the expense of the algorithms' performance guarantees.

We anticipate that this research will have significant impact as pieces together theoretical and practical research on self-organizing clustering algorithms. We believe that our work will be very useful to distributed application designers as it groups applications according to their clustering constraints and identifies clustering algorithms that are well suited to these application classes. Application developers will be able to either use the algorithms we develop directly, or extend/modify them to address the specific needs of their application. Our work is also novel in the sense that it targets a diverse set of applications.

The remainder of this paper is organized as follows. Section 2 introduces some basic notions and issues in clustering decomposition. In Section 3, we formalize the notion of an application class, identifying some of these classes; we also briefly describe the proposed approach for handling applications with similar constraints to power-aware routing. Section 5 reviews related work. Finally, in Section 6, we conclude with an outline of evaluation methods and future work.

2. CLUSTERING

Wide-area, large scale internets are expected to grow two or more orders of magnitude in the near future. In such environments, hierarchical decomposition, or *clustering*, is a well-known technique to address scalability. Network nodes are grouped into *clusters* and need only keep state for nodes belonging to the same cluster.

A cluster consists of a subset of the nodes in the network and the links between them. In general, clusters form connected subnetworks. Usually, a cluster will have a *cluster head* (or *cluster representative*), responsible for cluster administration (including intra-cluster communication), and some *border nodes* for inter-cluster communication. A set of clusters whose union contains all the nodes in the network is called a *cover*. If no two clusters in a cover share a node, the clusters are said to be *disjoint*, or *non-overlapping*.

In some applications, it is required that clusters be non-overlapping. VLSI design is an application area that requires clusters at each level of the hierarchy to be disjoint [1]. Non-overlapping clusters incur less clustering overhead as nodes do not need to keep state about other clusters in the cover. On the other hand, cluster overlapping allows for a more robust cluster structure (e.g., see [7, 4, 43]). Also, it may be easier to find overlapping clusters which optimizes some given performance metric, since the problem of finding a disjoint partition on nodes while optimizing any two performance metrics has been shown to be NP-complete [46, 20].

Hierarchical clustering is a generalization of clustering, where a set of clusters form different levels of the hierarchy. The elements of a cluster at level i consist of a subset of the clusters at level $i-1$, level 0 being the lowest level of the hierarchy. If a node x needs to communicate with some node y, x will forward this communication request to the leader of its cluster(s) at the lowest level of the hierarchy. This request will be propagated up the hierarchy until a cluster which also spans node y is found. At that point, the communication request is pushed down the hierarchy until it reaches a cluster containing node y at the lowest level of the hierarchy: the leader of that cluster will then forward the request directly to node y.

Hierarchical clusters scale by partitioning state across different hierarchical levels. Hierarchical routing is a typical example: router clusters reduce the size of routing updates and router-maintained state. Also less frequent routing updates and lesser router-maintained state lead to more efficient routing protocols.

Self-organizing clusters will detect changes in the underlying network infrastructure and automatically adapt to them. For instance, if links get overly congested, servers may re-cluster so that they avoid congested paths when exchanging data. This results in more efficient communication without generating additional traffic on the already congested paths. Detecting and reacting to changes in the network topology and infrastructure will be done in a distributed, decentralized fashion. In addition to adapting to infrastructure changes, our approach to self-organizing clusters also addresses application-specific requirements. For example, if database updates are taking too long to propagate, participating replicas may re-cluster to improve convergence time. Alternatively, if replica consistency state grows too big, the self-adjusting clustering algorithm may decide to split the cluster in two, or even create another cluster hierarchical level.

Clusters also *preserve autonomy*, which is essential in the administratively decentralized networking environments of today and tomorrow. Clustering addresses administrative decentralization explicitly by using administrative boundaries as a clustering constraint. The effect of changes (triggered by administrative decisions) are limited to a well-defined set of clusters. For example, if a new replica joins a replicated directory service, the clustering algorithm will select the appropriate cluster for the new member. Rearrangements to accommodate the new member will be limited to that cluster's boundaries and will not affect neighboring clusters.

3. APPLICATION CLASSES

In this section, we identify several application classes, each of which consisting of applications with common clustering constraints. By no means we claim that the classes presented here are exhaustive: it is in fact one component of our ongoing research to identify and characterize application classes of interest to the network algorithms community. Our main goal is to identify classes whose self-organizing clustering algorithms are broad enough to meet the performance requirements of a number of existing and future applications. Thus, an application may belong to more than one class, which implies that different clustering algorithms could be applied to this application while still meeting, or closely approximating, its performance requirements (the guarantees on the performance bounds could vary for different clustering algorithms). An extreme example will be that of an application that has no performance requirements: this application belongs to any application class, since any clustering strategy will meet the (non-existent) performance requirements of this application.

We are particularly interested in application classes which contain some of the most relevant applications of self-organizing clustering techniques in dynamic network domains. Examples include applications in network routing, as well as data-centric and data-tracking applications. In fact, since clustering has extensive application in routing (and since we believe we can safely assume that the reader is familiar with this problem, understanding the major concepts behind it), many of the examples we examine throughout the paper are related to routing.

We present an initial list of application classes below. It is possible that in the future some of these application classes may be further decomposed into subclasses, or that some classes may be combined, depending on the clustering techniques developed.

I. Point-to-point minimum-cost communication in highly dynamic networks

In this class we group together applications in highly dynamic network scenarios whose main performance requirement is that pairs of nodes communicate over minimum cost paths (assuming that the cost of communication is a known function that monotonically increases with the number of edges and nodes traversed). "Traditional" shortest-path routing, as well as any other application which relies on shortest-path routes, belong to this class. Time-critical routing applications as well as some real-time applications also fall into this category.

In static networks these problems are easily solved using classical shortest-paths algorithms (with the appropriate costs assigned to the edges and nodes of the network). However, since the networks we target can be highly dynamic (e.g., highly mobile multi-hop ad hoc networks), clustering will be used in order to ensure that close to optimal (i.e., minimum cost) routes can be maintained. In highly mobile networks, simply maintaining routes (not necessarily of minimum cost) between all pairs of nodes becomes very costly due to the number of updates generated by frequent changes in the network topology. In these highly mobile scenarios, clustering is used in order to keep the updates local (within the relevant clusters in the hierarchy), unless the change in topology is significant enough to justify a global update of the routing tables (i.e., at some or all levels of the clustering hierarchy).

II. Local resource-preservation in dynamic networks

Applications in this class must minimize the utilization of some resource at each individual node, according to resource availability at the node. For example, one of the main goals of power-aware applications in energy-constrained networks is to keep as many network nodes alive for as long as possible. Energy-aware routing in power-anemic sensor networks (as described in Section 1) is a typical example.

Now, consider a routing application where the main requirement is that the amount of processing time each processor devotes to routing be minimized (also mentioned in Section 1). Processing time is inversely proportional to processing speed. One can think of processing time as being inversely proportional to the percentage of the energy resources used at a node. Hence, the constraints imposed by these two routing problems are in fact similar: minimizing routing processing time at the nodes will minimize the percentage of energy resources used at each node, keeping the nodes alive in the network for as long as possible and thus maximizing the network's overall lifetime.

Another type of clustering constraints which arise in some classical clustering applications is that the clusters be balanced, at each level of the hierarchy (see [46]). One important reason for requiring balanced clusters is to distribute the overhead associated with maintaining the cluster structure (i.e., maintaining cluster membership information, performing clustering operations, adapting to changes in network topology, etc.) evenly among clusterheads. Thus, it may be the case that applications which require balanced clusters belong in this class.

III. Balanced communication cost versus resource consumption in dynamic scenarios

In order to understand the characteristics of applications in this class, we first look at an example in the routing domain. In many routing applications it is desirable that the routing tables at the nodes be kept small, not only for space considerations at the nodes, but also for less frequent update generation, which is crucial in a dynamic scenario. The trade-off is that smaller routing tables in general imply more restricted routing information, which in turn results in non-optimal routes. For example, the Destination-Sequenced Distance-Vector (DSDV) routing protocol and the Dynamic Source Routing (DSR) protocol (see [50] for a survey on current ad hoc network routing protocols) are examples of two extreme cases of this trade-off in ad hoc network scenarios: DSDV maintains minimum-cost routes at the expense of possibly updating the routing table of every node in the network for every network topology change; DSR, on the other hand, does not maintain routing tables at the expense of computing a route from source to destination on demand every time a communication request takes place [1]. This type of trade-off will significantly shape the design of the self-organizing clustering structure for applications in this class.

Many other applications exhibit similar trade-offs. For example, consider the following data management application. Suppose we have a dynamic distributed database over a network, where there may be replicated copies of the same

[1] While DSR is considered a reactive routing protocol, DSDV falls in the category of proactive protocols.

database object at different nodes (the database is dynamic in the sense that copies of objects are continuously being created, deleted, or replicated at the nodes). The World Wide Web is a good example of such database over the Internet. We maintain object-locator tables at the nodes for locating copies of objects. While we would like to limit the size of these tables — in order to reduce the costs of updates every time a new copy of an object is inserted or deleted, or every time there is a topology change in the network — we would also like to ensure that a node's request for a given object will be satisfied with a copy of this object which is not "too far" from the requesting node (see [43] for a more descriptive information of this problem).

IV. Data-centric applications in highly dynamic networks

Applications in this class are data-oriented: they either disseminate or collect data provided by participating nodes. For example, broadcasting and multicasting applications in dynamic environments belong to this class. Another important application in this class is that of collecting information provided by a dynamic network of sensors (the network may be dynamic due to node mobility or to faulty nodes and/or links).

The data collection problem on a network of power-anemic micro-sensors belongs not only to this class, but also to class II, as previously discussed. A good clustering algorithm for this application probably needs to combine the algorithmic techniques developed for classes II and IV. If this is not the case, a new application class should be defined in order to accommodate applications whose set of clustering constraints is similar to those generated by the data-collection in power-anemic networks application.

At first, we anticipate that the constraints generated by data dissemination will be similar to those generated by data collection services. In case this turns out not to be true, we will break this class into two separate classes.

V. Visualization and control of large-scale dynamic networks applications

Many applications have as a performance goal the reduction of the visualization complexity of the underlying network. To properly observe and analyze the network, a self-adjusting hierarchical view of the network may be necessary. In this view, entities (nodes and edges of the network) are consolidated together and observed as a single metaentity (or cluster). If more detail is required, then the view of the metaentity can be expanded into its individual elements. Again, these elements may be comprised of sub-entities (sub-clusters) which in turn are also a collection of network metaentities. In addition to consolidating the view of many network elements, their performance metrics must also be consolidated (lumped) in a manner that can provide meaningful observation about the state of the metaentity. This observable consolidated state must provide sufficient information for an action decision to control the performance of the network. Note that, at all levels of the clustering hierarchy, the metaentities (clusters) must be disjoint.

Applications in this class originate in diverse domains. For example, in the domain of text and organizational-knowledge analysis, it is often important to visualize discourse as a dynamic graph where nodes represent words and edges join together pairs of words which are "close" according to a technique called Centering Resonance Analysis (CRA). Another example arises in biology and chemistry, when the network in consideration represents molecules (nodes), which are moving in space, and how they interact with each other (there will be an edge in the network, possibly weighted, whenever two molecules are close in space). One of the goals of a visualization tool in this particular case is to present a clear view of the molecules mobility pattern. Computer network monitoring and management is another application that falls within this class. In particular, consider monitoring a network of mobile nodes (consisting for example of firefighters in a field operation, or soldiers in a battlefield). Visualization tools can be of tremendous help to network planners an administrators for the management and control of large-scale networks as it can illuminate network activity patterns in real time.

Existing techniques for reducing the visual complexity of large and dynamic network (see [17]) fail to provide a hierarchical view of the network. A major challenge in these applications is to be able to measure the "strength" of the relationship among the nodes, in order to define the clusters.

Other examples of application classes which we intend to further characterize and refine include:

- **Connectivity-oriented applications in dynamic networks**. Applications in this class have as their main goal to maintain a structure of minimum cost which would "represent" the connectivity properties of the underlying network (i.e., nodes are connected in the network if and only if they are connected in this structure). This structure could be used, for example, in applications which involve multicasting, broadcasting, or flooding.

  One idea is to use Steiner trees (or a variation thereof, such as group Steiner trees [21]) to represent the connectivity of each cluster. A major challenge is to maintain these trees in highly dynamic scenarios, since the Steiner tree problem in NP-hard even in a static networks.

- **Global resource-preservation in dynamic networks** Applications in this class, like applications in class II, aim at minimizing the utilization of some resource in the network. The difference is that in this class of applications we are concerned with the total utilization of the resource over the network, while in class II the goal is to minimize resource utilization at each particular node. In other words, here we would like to minimize the average resource consumption per node in the network, while in class II we aim at minimizing the maximum (possibly in terms of percent usage) consumption on the resource at any node. The classical minimum-cost maximum-flow network problem falls in this class.

4. POWER-AWARE ROUTING IN ENERGY-CONSTRAINED DYNAMIC NETWORKS

In this section, we present an overview of our ongoing approach to developping power-efficient self-organizing clustering algorithms for routing in an energy-constrained network. A complete outline of our proposed approach and

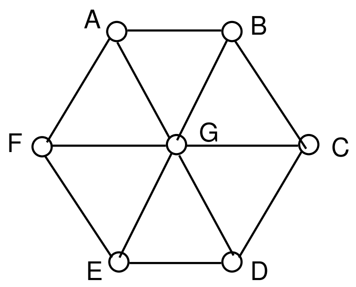

Figure 1:

initial progress can be found in [48]. An important contribution of the proposed algorithms is that they will capture changes in the energy resources of the underlying network and automatically adapt to them. The ability to detect and adapt to network dynamics is one of the main challenges when designing clustering protocols for highly-dynamic networks.

Power-aware routing is an example of a class III application (see Section 3). Therefore, the techniques developed for this particular application can be employed by other applications which share common clustering requirements and constraints. This is the case of the minimum processing time routing problem described in Section 1.

In order to illustrate the major problems to be overcome in applications which call for energy-efficient operations, we consider the network in Figure 1.

Suppose, that messages need to be transferred from A to D, B to E and C to F. Most routing algorithms will transfer data to the respective destinations using shortest-paths, all of which will use node G as an intermediate node. As a consequence, the energy at node G will be depleted very fast and soon the node will be "dead".

Although several routing protocols for mobile ad hoc networks have been proposed [50], most of them do not have energy preservation at the nodes as a stated objective or lack concrete energy preservation performance measures. Therefore, existing routing protocols will likely deplete energy very quickly. The protocols to be developed in this scenario need to include energy consumption as an important communication cost metric. Their objective is to keep as many of the network nodes "alive" for as long as possible.

For scalability, participating nodes have to self-organize into clusters. Besides being energy-efficient, the proposed self-organizing clustering algorithms will use power/energy as one of their clustering constraints. For instance, resulting clusters will likely employ more powerful devices to act as border nodes. These non-empty cluster intersections, or overlaps, ensure that information from one cluster make their way to other clusters. Power-aware inter- and intra-cluster routing protocols will provide the infrastructure for collecting and disseminating application data. Finally, the proposed protocols will adapt to changes in energy resources. Since border nodes will likely engage in long-haul communications more frequently, they can become energy-deficient; clusterheads are responsible for administering any intra- or inter-cluster communication. If the clustering algorithm detects energy-deficient clusterhead or border nodes, it will automatically re-organize the cluster to select other node(s) to replace the deficient one(s).

We are currently investigating self-organizing clustering algorithms and the clustering constraints that will drive these algorithms in energy-constrained environments consisting of a large number of energy-depleted nodes. Clustering constraints capture network dynamics, including membership changes and current energy levels. Examples of (often competing) clustering constraints that are directly related to power consumption include *network proximity* — which in this context is defined as the *energy cost of communication* between two nodes — *cluster size*, and *administrative overhead imposed on clusterheads and gateway nodes.* For example, the size of a cluster determines the size of the routing tables to be kept at the nodes for intra-cluster communication: the larger the cluster, the larger the intra-cluster routing tables at the cluster members. This implies more frequent updates to these tables which, along with larger storage space and more processing, result in more energy consumption at the cluster nodes. On the other hand, if the size of the clusters is small, then the amount of inter-cluster communication increases, overloading the clusterheads.

Designing self-organizing hierarchical clustering algorithms in highly dynamic environments in response to the performance requirements of point-to-point routing applications is a challenging problem on its own, as is the problem of designing energy-efficient clustering techniques in less dynamic networks. Therefore, we focus on each of these problems separately at first, in order to understand the major issues to be overcome in the combined problem. We will then combine the resulting algorithms, in order to design energy-efficient self-organizing clustering protocols for networks of energy-depleted nodes. Below, we briefly describe how we address several of these issues.

When focusing on the issue of designing *self-organizing* clustering algorithms, we intend to adapt and extend some of the ideas in Awerbuch and Peleg's clustering algorithm (AP) presented in [7]. The AP algorithm, which was designed for static network scenarios, is a robust algorithm that benefits from redundancy in order to guarantee the efficiency of its clustering operations. Also, we are investigating whether we could use the main ideas in the TORA algorithm [40]. The TORA algorithm seems to adapt well to changes in the network topology, but is based on a flat, non-hierarchical view of the network. The work on adaptive clustering algorithms by Lin and Gerla in [36] and by Gerla and Tsai in [22] is also relevant. For energy-constrained environments, the major drawback of these approaches is that nodes selected as cluster leaders may remain so until significant changes in network topology occur. This of course may lead to fast energy depletion at those nodes.

Now we briefly describe our proposed approach for addressing *energy-efficiency.* Having several clusterheads in a cluster would share the cluster's administrative tasks among the (various) clusterheads, without overloading any one of them. The same could be true of the border gateway nodes: a larger set of border nodes could evenly distribute the inter-cluster routing responsibilities among themselves. As a starting point, we will explore some of the ideas developed by Plaxton et al. in [43] and by Rajaraman et al. in [45] in order to devise clustering strategies that distribute the clustering administration and communication tasks in an even fashion among the members of the cluster, at all levels of the clustering hierarchy. In both works, hierarchical clustering strategies are considered for the problem of maintaining a distributed database with replicated objects. They present clustering schemes in this context, which evenly distribute the cluster administrative tasks, including cluster membership information, among the nodes of a cluster.

Depending on the results obtained, we intend to combine the adapted versions of the protocol presented in [4] with the protocol in [43], in order to obtain a clustering protocol which is both self-organizing and energy-efficient. Preliminary work has been done from the algorithmic point of view in [45], but no conclusive result for the scenario in question has been found yet.

We expect that the results obtained when studying self-organizing clustering techniques for routing in highly-dynamic energy-depleted environments will provide us insight into self-organizing clustering techniques applicable to other application classes.

5. RELATED WORK

Clustering has been used extensively in a wide range of areas within computer science and engineering, including networks, software engineering, visualization, library systems, and VLSI design. Just as the application domains for clustering are diverse, so are the criteria for cluster formation. Even if we leave aside the numerous clustering techniques developed for statistical applications and concentrate on clustering techniques for graphs, we find that in VLSI design domain alone, over three hundred papers were published between 1970 to 1995 on generation of nonoverlapping clusters [1]. Therefore, instead of attempting to present an exhaustive list, we try to identify a representative set of related work.

Although the topic of graph clustering has been extensively studied, there has been no systematic attempt to classify different applications into classes, identify class-specific clustering constraints, and develop clustering algorithms for each application class. Ramamoorthy et al. [46] were one of the only to study a general framework for graph clustering satisfying multiple objectives. They also noted that the disjoint (nonoverlapping) clustering problem with the objective of minimizing any two common clustering constraints is NP-complete. They were, to the best of our knowledge, the first to formalize some preliminary ideas on designing clustering techniques for subsets of common types of clustering constraints. However, they followed a "clustering-oriented" approach in their work — rather than the "application-oriented" approach we propose. In other words, they first generate the set of clustering constraints for which they could develop (possibly approximate) clustering algorithms, and then they will find applications which could use their algorithms to some extent. In this project, we propose to start from the applications end, deriving from the applications performance requirements the sets of clustering constraints we intend to address. Also, Ramamoorthy et al. focussed only on nonoverlapping clusters. In some applications, it is necessary, or advantageous, to have overlapping clusters, as discussed earlier.

The problem of finding nonoverlapping clusters where the objective is to develop only two clusters with minimum number of inter-cluster links [25, 19], or minimum *ratio cut* (see [35, 26, 34, 51]) has drawn considerable attention from researchers. In the VLSI domain this is known as the *two-way partition problem*. Kernighan et al. [25], and Fiduccia et al. [19] developed effective heuristics for this problem, which are still extensively used.

Clustering techniques have also been widely used in the telecommunication domain, where the size of the network is usually large. *Hierarchical routing* was used by DARPA's Survivable Packet Radio Network (SURAN) project in the early 1980s [33]. More recently many other researchers have considered hierarchical routing in large dynamically changing networks [2, 42, 27, 29, 40, 41]. Cluster formation using different objectives has been considered in [3, 24, 31]. Basagni in [14, 15] considers distributed algorithms for 2-hop clusters in ad-hoc networks. Various aspects of clustering in radio network design have been examined in [11, 9, 12, 10].

Efficient clustering techniques, which rely on significant cluster overlap in order to guarantee communication performance bounds were developed by Awerbuch et al. in [7, 5]. Their work has been extensively used (e.g., [5, 6, 13, 8]) in designing efficient algorithms in the context of several network applications, including routing. Their techniques apply to a static network scenario.

Plaxton et al. [43] and Rajaraman et al. [45] focus on the issues that relate to data tracking alone, proving efficient performance bounds guarantees in more general network cost models. The data tracking problem is strongly related to routing [45].

Self-organizing communication networks have been examined in [49, 44]. In the context of mobile networks, Gerla et al. in [22] considered two distributed clustering algorithms for multihop packet radio networks for mobile information systems. Their *lowest-id* algorithm (originally proposed by Ephremides [18]), uses the lowest-id node in a neighborhood as the clusterhead. In their other algorithm (a modified version of an algorithm due to Parekh [39]), the highest degree node in a neighborhood is chosen as the clusterhead.

The work in [36] describes a self-organizing multihop mobile radio network where nodes are grouped into non-overlapping clusters. Clusters are independent and they are dynamically reconfigured as the nodes move. The main advantage of the clustering algorithm proposed in [36] is that clusters generated by this algorithm are much more stable in comparison with the ones produced by the algorithm in [22], which leads to less updates to the clustering structure. Clustering with power control and routing in a fading channel environment is considered in [32] and [16] respectively.

The work by Kulik et al. [30] proposes aplication-level energy-efficient routing protocols to disseminate data in sensor networks. Through high-level data descriptors, or *meta-data*, nodes negotiate to try to eliminate redundant data transmission.

Ramanathan and Steenstrup [47] considered a hierarchically organized, multihop mobile wireless network that meets quality-of-service requirements. More specifically, they developed a clustering procedure for defining a virtual, hierarchical control structure to be superimposed on a large network of mobile switches. In their clustering algorithm, the clusters have bounded size and the notions of *cluster splitting* and *cluster merging* are used to deal with the dynamic nature of the network topology. In [28] the same group at BBN presents a heuristic technique for the formation of non-overlapping clusters for large self-structuring networks.

According to the authors, the novelty of their scheme is that the clusters generated by the algorithm are connected, bounded in size, and actively try to minimize the cost of inter-cluster links. It may be noted that the research objective of our proposal is markedly different from the one addressed by Ramanathan and Steenstrup. In their research, Ramanathan and Steenstrup consider general hierarchical clustering issues that arise in multihop mobile wireless networks, without targeting any specific applications. In our proposed research, we will consider many different applications, classify them into different application classes, develop the clustering constraints for each of the classes and develop novel techniques to solve those clustering problems.

McDonald et al. in [37] developed a novel framework for cluster formation in a dynamic fashion for nodes in a mobile ad-hoc network. The cluster formed by their family of algorithms, called $(\alpha, t)-Clusters$, can provide a probabilistic guarantee on the availability of paths between specified nodes. Krishna et al. [29] also propose a cluster-based approach for routing in dynamic networks.

Directed diffusion [23] is a novel communication paradigm for sensor networks. It targets data-centric applications whose focus is to gather (and possibly process) relevant data generated by the sensor network. Because it operates in power-constrained environments, directed diffusion must be power-aware. Their approach to power conservation is to use data aggregation techniques which allow the sensor network as a whole to reduce the amound of data being transmitted.

6. EVALUATION AND FUTURE WORK

We intend to evaluate our algorithms through extensive use of simulations. Simulations will help us evaluate how the proposed algorithms perform in practice when subject to a wide range of network conditions and workloads. Besides algorithmic efficiency issues, we are particularly interested in how the algorithms respond to changes in network topology and load to meet specified application requirements.

One of the simulation platforms we are considering is the `ns` [38] network simulator, which has been extensively used and validated by the experimental networking community and has accumulated substantial common knowledge and contributed modules.

Guided by the preliminary evaluation results of the proposed clustering algorithms through simulations, we will then use formal methods in order to formally analyze the correctness and performance of these algorithms, in particular of the ones that showed good performance in the simulation experiments.

The next step is to prototype our algorithms and evaluate them through live experiments. In particular, we propose to prototype networks of energy-constrained micro-sensing devices, and to implement a testbed for evaluating clustering algorithms for power-aware applications on these networks.

7. REFERENCES

[1] C. L. Alpert and A. Kahng Netlist Partitioning. *Integration: the VLSI journal*, vol. 19, pages 1–81, 1995.

[2] I. F. Akyildiz, W. Yen, B. Yener. A new hierarchical routing protocol for dynamic multihop wireless networks. *Proceedings of IEEE INFOCOM'97*, Kyoto, 1997.

[3] A. D. Amis, R. Prakash, T. H. P. Vuong Max–Min D-cluster formation in wireless ad hoc networks *Proceedings of IEEE INFOCOM'2000*, Tel Aviv, 2000. (The paper can be downloaded from the Ad-hoc network session of the Technical Program of the INFOCOM'2000: http://www.cse.ucsc.edu/~rom/infocom2000/)

[4] B. Awerbuch, B. Berger, L. Cowen, and D. Peleg. Fast distributed network decompositions and covers. *Journal of Parallel and Distributed Computing*, 39(2), pages 105–114, December 1996.

[5] B. Awerbuch, B. Berger, L. Cowen, and D. Peleg. Near linear time construction of sparse neighborhood covers. *SIAM Journal of Computing*, vol. 28, no. 1, pages 263–277, 1998.

[6] B. Awerbuch, Y. Bartal, and A. Fiat. Distributed paging for general networks. In *Proceedings of the Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 574–583, January 1996.

[7] B. Awerbuch and D. Peleg. Sparse partitions. In *Proceedings of the Thirty-First Annual IEEE Symposium on Foundations of Computer Science*, pages 503–513, October 1990.

[8] B. Awerbuch, and D. Peleg. Routing with polynomial communication-space tradeoff. *SIAM Journal on Discrete Math.*, 5, pages 151–162, 1992.

[9] D. J. Baker. Distributed control of broadcast radio networks with changing topologies. *Proceedings of IEEE INFOCOM'83*, pages 49–55, 1983.

[10] D. J. Baker, A. Ephremides, and J. A. Flynn. The design and simulation of a mobile radio network with distributed control. *IEEE JSAC*, SAC-2(1), pages 226–237, January 1984.

[11] D. J. Baker and A. Ephremides. The architectural organization of a mobile radio network via a distributed algorithm. *IEEE Transactions on Communications*, 29(11), pages 1694–1701, 1981.

[12] D. J. Baker, J. Wieselthier, and A. Ephremides. A distributed algorithm for scheduling the activation of links in a self-organizing, mobile, packet-radio network. In *Proceedings of IEEE ICC'82*, pages 2F.6.1–2F.6.5, 1982.

[13] Y. Bartal, A. Fiat, and Y. Rabani. Competitive algorithms for distributed data management. In *Proceedings of the 24th Annual ACM Symposium on Theory of Computing*, pages 39–50, May 1992.

[14] S. Basagni. Distributed and mobility-adaptive clustering for multimedia support in multi-hop wireless networks. *Proceedings of the IEEE Vehicular Technology Conference (VTC)*, Amsterdam, The Netherlands, pages 19–22, September 1999.

[15] S. Basagni Distributed clustering for ad-hoc networks. *Proceedings of the 1999 International Symposium on Parallel Architectures, Algorithms and Networks (I-SPAN'99)*, IEEE Computer Society,Australia, pages 310–315, June 1999.

[16] C. C. Chiang, H. K. Wu, W. Liu and M. Gerla. Routing in clustered multihop, mobile wireless networks with fading channels. *The IEEE Singapore International Conference on Networks*, pages 197-211, 1997.

[17] J. Edachary, A. Sen and F. Brandenburg Graph

clustering using distance-k cliques. *Proceedings of the International Graph Drawing Symposium*, pages 98-106, September 1999.

[18] A. Ephremides, J. Wieselthier and D. J. Baker, . A design concept for reliable mobile radio networks with frequency hopping signaling. *Proceedings of IEEE*, 75(1), pages 56–73, 1987.

[19] C. M. Fiduccia and R. M. Matheyses A linear time heuristic for improving network partitions *Proceedings of the Design Automation Conference*, pages 175–181, 1982.

[20] M. R. Garey, and D. S. Johnson. Computers and Intractability: A guide to the theory of NP-Completeness. W. H. Freeman and Company, New York, 1979.

[21] N. Garg, G. Konjevod, and R. Ravi. A polylogarithmic approximation algorithm for the group Steiner tree problem. *Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 253–259, 1998. Invited for publication in a special issue of Journal of Algorithms.

[22] M. Gerla and J. T. C. Tsai. Multicluster mobile multimedia radio networks. *ACM-Baltzer Journal of Wireless Networks*, vol. 1, no. 3, pages 255–265, 1995.

[23] C. Intanagonwiwat, R. Govindan and D. Estrin. Directed Diffusion: A scalable and robust communication paradigm for sensor networks. http://netweb.usc.edu/estrin/papers/diffusion-mobicom-submitted.ps.

[24] R.A. Jarvis and E.A. Patrick. Clustering using a similarity measure based on shared near neighbors. *IEEE Transactions on Computers*, C-22(11), pages 1025–1034, November 1973.

[25] B. W. Kernighan and S. Lin An efficient procedure for partitioning graphs. *Bell Systems Technical Journal*, 49, pages 291–307, 1970.

[26] P. Klein, S. Plotkin, C. Stein and E. Tardos. Faster approximation algorithms for the unit capacity concurrent flow problem with applications to routing and finding sparse cuts. *SIAM Journal of Computing*, 23, pages 466–487, 1994.

[27] Y. B. Ko and N. H. Vaidya Location-aided routing (LAR) in mobile ad-hoc networks. *Proceedings of MOBICOM'98*, pages 66–75, October 1998.

[28] R. Krishnan, R. Ramanathan and M. Steenstrup Optimization algorithms for large self-structuring networks. *Proceedings of IEEE INFOCOM'99*, pages 71–78, April 1999.

[29] P. Krishna N. H. Vaidya, M. Chatterjee, D. K. Pradhan A cluster-based approach for routing in dynamic networks. *ACM SIGCOMM Computer Communication Review*, pages 49–65, April 1997.

[30] J. Kulik, W. Rabiner, H. Balakrishnan. Adaptive Protocols for Information Dissemination in Wireless Sensor Networks. *Proceedings of the MOBICOM'99*, Seattle, 1999.

[31] K.S. Kumar. Self-configuring hierarchy for inter-domain protocol independent multicast. Available from http://www.isi.edu/˜kkumar/hpim/.

[32] T. J. Kwon and M. Gerla. Clustering with Power Control. *Proceedings of IEEE MILCOM'99*, Atlantic City, NJ, Nov. 1999.

[33] G. S. Lauer. Hierarchical routing design for SURAN. *Proceedings of IEEE ICC'86*, pages 93–102, 1986.

[34] T. Leighton, F. Makedon, S. Plotkin, C. Stein, E. Tardos and S. Tragoudas. Fast approximation algorithms for multicommodity flow problems. *Journal of Computer and System Science*, 50, pages 228–243, 1995.

[35] T. Leighton and S. Rao. An approximate max-flow min-cut theorem for uniform multicommodity flow problems with applications to approximation algorithms. *Proc. of IEEE Symposium on Foundations of Computer Science*, pages 422–431, 1988.

[36] C. R. Lin and M. Gerla. Adaptive clustering for mobile wireless networks. *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 7, pages 1265–1275, 1997.

[37] A. B. McDonald and T. Znati. A mobility-based framework for adaptive clustering in wireless ad-hoc networks *IEEE Journal on Selected Areas in Communication*, vol. 17, no. 8, August 1999.

[38] S. McCanne. ns - LBNL Network Simulator. Available from http://www-nrg.ee.lbl.gov/ns/

[39] A. K. Parekh Selecting routers in ad-hoc wireless networks. *Proceedings of ITS*, 1994.

[40] V. D. Park and M. S. Corson A highly adaptive distributed routing algorithm for mobile wireless networks. *Proceedings of IEEE INFOCOM'97*, pages 1405-1413, Japan, 1997.

[41] C. E. Perkins and P. Bhagwat. Highly dynamic destination sequenced distance vector routing (DSDV)for mobile computers. *Proceedings of ACM SIGCOMM'94*, pages 234–244, 1994.

[42] M. R. Pearlman and Z. J. Haas Determining the optimal configuration for the Zone Routing Protocol. *IEEE JSAC* pages 1395-1414, August 1999.

[43] C. G. Plaxton, R. Rajaraman, and A. W. Richa. Accessing nearby copies of replicated objects in a distributed environment. *Theory of Computing Systems*, 32:241–180, 1999. A preliminary version of this paper appeared in *Proceedings of the 9th Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA)*, pages 311–320, June 1997.

[44] M. J. Post, A. S. Kershenbaum, and P. E. Sarachik. A distributed evolutionary algorithm for reorganizing network communications. *Proceedings of IEEE MILCOM'85*, 1985.

[45] R. Rajaraman, A. W. Richa, B. Vöcking, and G. Vuppuluri. Near-optimal data-tracking schemes in a distributed environment. *Proceedings of 13th Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA)*, July 2001.

[46] C.V. Ramamoorthy, J. Srivastava, and W. Tsai. Clustering techniques for large distributed systems. *Proceedings of the IEEE Infocom 1986*, pages 395–403, April 1986.

[47] R. Ramanathan and M. Steenstrup. Hierarchically-organized, multihop mobile wireless for quality-of-service support. *Mobile Networks and Applications*, 3, pages 101–119, 1998.

[48] A. W. Richa,K. Obraczka, and A. Sen. Power-aware Self-Organizing Hierarchical Clustering in Dynamic Networks. *Technical Report ASU*, Arizona State

University, Tempe, AZ, August 2001.

[49] T.G. Robertazzi and P.E. Sarachik. Self-organizing communication networks. *IEEE Communications Magazine*, 24(1), pages 28–33, January 1986.

[50] E. M. Royer and C. K. Toh. A review of current routing protocols for ad hoc mobile wireless networks. *IEEE Personal Communications*, pages 46–55, April 1999.

[51] Y. C. Wei and C. K. Cheng. Ratio cut partitioning for hierarchical design. *IEEE Trans. on CAD*, 40(7), pages 911–921, 1991.

[52] R. Wolski, N. T. Spring and J. Hayes. The network weather service: A distributed resource performance forecasting service for metacomputing. *http://nws.npaci.edu/NWS/*