Report Number: WUCS-99-17

1999-01-01

A Rapid Development of Dependable Applications in Ad Hoc Mobility

Amy L. Murphy

Advances in wireless communication and network computing technologies make possible new kinds of applications involving transient interactions among physical components that move across a wide range of spaces, from the confines of a room to the airspace across an ocean, and require no fixed networking infrastructure to communicate with one another. Such components may come together to form ad hoc networks for the purpose of exchanging information or in order to engage in cooperative task-oriented behaviors. Ad hoc networks are assembled, reshaped and taken apart as components move in and out of communication range; all interactions are transient; computations... **Read complete abstract on page 2.**

A Rapid Development of Dependable Applications in Ad Hoc Mobility

Amy L. Murphy

Complete Abstract:

Advances in wireless communication and network computing technologies make possible new kinds of applications involving transient interactions among physical components that move across a wide range of spaces, from the confines of a room to the airspace across an ocean, and require no fixed networking infrastructure to communicate with one another. Such components may come together to form ad hoc networks for the purpose of exchanging information or in order to engage in cooperative task-oriented behaviors. Ad hoc networks are assembled, reshaped and taken apart as components move in and out of communication range; all interactions are transient; computations become highly decoupled and rely on weak forms of data consistency; disconnections are frequent and unpredictable; and component behavior is sensitive to changes in location, context, quality of service, or administrative domain. Our objective is to develop an environment that will facilitate rapid development of dependable mobile applications executing over ad hoc networks. Our primary focus will be the development of coordination constructs that support transient interactions among components, specifically through the design of global virtual data structures. Operations and their effects on these data structures must be defined with respect to the current connectivity context. We intend to use formal modeling techniques to define these constructs and their operating constraints as well as providing the specification for implementing these structure. Part of this specification will involve the development of algorithms for the ad hoc environment such as leader election, termination detection, and transactions.

Rapid Development of Dependable Applications in Ad Hoc Mobility

Amy L. Murphy

WUCS-99-17

June 1, 1999

Department of Computer Science
Washington University
Campus Box 1045
One Brookings Drive
Saint Louis, MO 63130-4899

WASHINGTON UNIVERSITY

Department of Computer Science

Proposal for Dissertation Research in Partial Fulfillment
of the Requirements for the Degree of
Doctor of Science

Title: *Rapid Development of Dependable Applications in Ad Hoc Mobility*
Submitted by: Amy L. Murphy on: June 3, 1999
Advisor: Gruia-Catalin Roman

Abstract

Advances in wireless communication and network computing technologies make possible new kinds of applications involving transient interactions among physical components that move across a wide range of spaces, from the confines of a room to the airspace across an ocean, and require no fixed networking infrastructure to communicate with one another. Such components may come together to form ad hoc networks for the purpose of exchanging information or in order to engage in cooperative task-oriented behaviors. Ad hoc networks are assembled, reshaped and taken apart as components move in and out of communication range; all interactions are transient; computations become highly decoupled and rely on weak forms of data consistency; disconnections are frequent and unpredictable; and component behavior is sensitive to changes in location, context, quality of service, or administrative domain.

Our objective is to develop an environment that will facilitate rapid development of dependable mobile applications executing over ad hoc networks. Our primary focus will be the development of coordination constructs that support transient interactions among components, specifically through the design of global virtual data structures. Operations and their effects on these data structures must be defined with respect to the current connectivity context. We intend to use formal modeling techniques to define these constructs and their operating constraints as well as providing the specification for implementing these structures. Part of this specification will involve the development of algorithms for the ad hoc environment such as leader election, termination detection, and transactions.

1 Background

In the modern computing environment, distributed computing is often taken for granted. Connectivity among the computers of a localized group, as well as connectivity across the country enable many of the tasks which we view as essential. We are able to share ideas, share resources, and divide computational load with only minimal interference caused by the physical distance. Extensive research has been done in distributed computing to enable these tasks, from the development of distributed algorithms to remote object referencing systems, but much of this work relies on the stable physical configurations of the computing components and the stability and availability of communication.

Introducing mobility to this distributed environment was first done with base stations similar to the cellular telephone network. This model is characterized by a fixed set of support stations and a fluid fringe of mobile units. Mobile IP laid the groundwork for integration of mobile components with the fixed network making basic communication to and from mobile units possible by utilizing the existing IP infrastructure. The resulting abstraction to the mobile user is an environment identical to the fixed network (with minimal startup configuration costs) where messages are sent and received as if the unit was stationary. To provide this abstraction, a Mobile IP implementation takes into consideration unit migration, interactions with existing protocols, security, etc. The complexity of the implementation is effectively hidden from the user, but the abstraction provided to the user is powerful enough to be extremely useful. Another aspect of mobility in the fixed network is the logical movement of code among servers. Logical mobility has been explored to reduce overall communication between a client and server by moving the computation closer to the data it accesses.

The next step in mobility research is the removal of the base stations, leaving only peer to peer communication based on component proximity. With this model, termed ad hoc mobility, the user is free from dependence on a fixed infrastructure, making it ideal for environments of small components with transient connectivity, disaster situations where infrastructure has been destroyed, or anywhere establishing a fixed network is economically infeasible such as a meeting room or conference setting. Freeing the user from the infrastructure also means losing the ability to rely on the existing properties of the infrastructure such as computational resources, storage resources, connectivity, etc. Therefore many of the issues that have been solved in the base station model of mobility must be solved again from the perspective of ad hoc mobility.

We take the stance that coordination among mobile components is a central concept to working in the ad hoc environment and coordination constructs and design techniques which allow mobile units to act locally and application designers to think globally about the aggregate behaviors will provide a meaningful level of abstraction to simplify the task of application development over ad hoc networks. Application code will be developed in a manner independent of the context in which a component might find itself during its lifetime by providing a highly decoupled style of programming that relies on transparent interactions embodied by the coordination semantics. Techniques will be available to reason formally about the global behavior resulting from a given pattern of interactions in a particular context and under a specific set of assumptions about the system and its components, yielding increased confidence in the final system.

2 Related Work

Much work has been done in the base station model of mobility, and although the focus of this proposal is ad hoc mobility, a majority of the related work is in the base station environment. Our intent is to reference the experiences in the base station environment when developing in the ad hoc environment taking advantage of the shared characteristics of mobility.

2.1 Mobile Applications

A fundamental characteristic of mobility is the presence of a new environment variable, namely location. Location-dependent services have been explored in the context of WWW browsers that resolve queries based on current location [45, 2]. Similarly, it has been argued that both software [42] and hardware [21] need to have the ability to change their behavior in a location-dependent (and more generally in a context-dependent) manner. The Odyssey system [32] addresses these concerns by separating the responsibilities of the application to define and respond to environmental changes from the responsibilities of the system to notify the application of given events. Although this work relies on the fixed network to provide the database of location dependent information, the ideas are applicable in the ad hoc environment when data is distributed among moving units. In this scenario, it is no longer absolute location that is important, but rather relative location and connectivity.

In a more data oriented approach, the idea of distributing a standard file system across mobile units has been addressed [41, 40, 43], raising issues such as relaxed consistency models, conflict resolution when components merge data spaces, and determining the consistency model used when reading data while maintaining the ability to reason about the system. Bayou [44], a replicated database system, provides application-specific integration policies so that updates to the dataspace that were written while disconnected can be integrated into a primary copy. Re-integration is accomplished sporadically on a pairwise basis, and the only guarantee offered is eventual consistency. We anticipate that in the ad hoc environment, replication and access strategies will be able to increase availability, but similarly integration policies must be defined and well understood in the environment where connectivity varies greatly. We believe that a consistent global view of the data must be available, but achieving strict consistency may require too much from the ad hoc environment with respect to connectivity assumptions.

2.2 Mobile Computing Algorithms

Below these applications lies a set of algorithms tailored to the complexities of the mobile environment, providing building blocks for complex application behaviors. Extensive research has been done in distributed computing, however Badrinath et al. [4] describe the pitfalls of directly applying these algorithms to the mobile environment. Mobility features such as changing connectivity, and reduction in computing resources at mobile units must all be considered in designing mobile distributed algorithms. Related work from Rutgers University [5] provides a two tier principle for structuring distributed algorithms to minimize computation and communications requirements, putting most of the burden on the fixed network. Following this approach, Acharya et al. [1] explored checkpointing mobile systems, making use of the fixed network for synthesizing global information as it is gathered.

As one of our primary goals is the development of design schemas and algorithms, we intend to leverage off the paradigms suggested by this work, but also take advantage of our own experiences in discovering useful algorithms for mobility which may be distinct from those in distributed computing. Examples of such algorithms were developed on the Ants project in the Artificial Intelligence Laboratory at MIT. This project uses robots for ordnance disposal [28], and develops algorithms which allow their system to achieve global properties (e.g., clustering) from local events (e.g., moving closer to a neighbor). Other work [10] suggests algorithms for arranging large numbers of unorganized components in hierarchies, emphasizing the ability to specialize the components as necessary for a particular task, but deemphasizing reconfiguration. Although we do not anticipate following an AI approach, we do expect to look into randomized strategies as well as ideas from self-stabilization.

2.3 Mobile Code

While our primary focus is the physical movement of components through space, logical mobility has many similar characteristics. The increasing popularity of the Java language and the Internet have aided in the interest in code mobility. Fuggetta, Picco and Vigna [13] provide an overview of many mobile code languages, explore application areas, and characterize paradigms of mobile code use. One of their contributions is a quantitative evaluation mechanism for determining the tradeoffs for a given application in multiple mobile code paradigms [6].

Several languages and toolkits exist to develop agent code with a variety of movement styles: Agent Tcl [16] (an interpreted approach providing primitives such as jump and fork), Rover [20] (designed to move code closer to clients, thereby reducing network traffic to a server), and μCODE [37] (a general object oriented Java toolkit which allows the programmer to specify whether class objects are carried by the agent or fetched on a need basis). Other systems, such as PageSpace [9] and Messengers [14, 15] focus on the coordination mechanisms of the mobile agents.

2.4 Protocol Layer

One of the underlying assumptions in building software for the mobile environment is the communication layer and associated protocols. Much work involves integrating mobile components into fixed infrastructures by relying on sets of connected base stations to which the mobile units are attached. Within this framework, Mobile IP [35, 36] has focused on expanding IP to accommodate mobility by allowing a laptop to move seamlessly from one LAN to another by specifying limited reconfiguration at the laptop and requiring no changes to the Internet infrastructure. The main idea is the use of a home agent to forward packets to the most up-to-date location of the mobile unit. Several approaches which do not rely on IP routing have been proposed to keep a path from a home agent to the mobile unit [24, 46]. Issues to consider include keeping this path minimal while incurring minimal database overhead. Our early work lies in this area and describes algorithms to provide guaranteed message delivery in a rapidly changing environment [29]. These algorithms use traditional distributed algorithms (such as distributed snapshot and diffusing computations) as their basis, and define a paradigm for translating the mobility model into traditional distributed computing by treating the mobile unit as a persistent message in the network.

In contrast to the home agent approach, information about the current location of the mobile unit can be distributed throughout the network and either queried for each packet sent [38] or used on a hop by hop basis [30] to direct a packet through the network. Another approach leverages off the location independence of multicast algorithms, to provide location independent addressing for mobile hosts [31]. The nature of the distribution of the location information is dependent on the multicast algorithm applied. Multicast to mobile groups is also being investigated in reliable arenas where a pre-defined set of hosts is known [3] as well as on top of Mobile IP [8] while leveraging off existing multicast schemes.

Although schemes that operate in the fixed network or in base station mobility can provide insight into the difficulties of reaching a mobile unit as it moves, the ad hoc environment provides further challenges because the mobile units themselves take on the role of routers in message delivery protocols. To maintain contact between the outermost agents, routing through intermediate nodes will be necessary. The IETF Mobile Ad Hoc Networking Working Group is focused on such protocols and most of the work concentrates on systems employing radio broadcast signals with omni-directional broadcast capabilities [11]. Johnson and Maltz [19, 7] proposed a discovery and source routing algorithm (Dynamic Source Routing), in which the source must first query for a path to the destination, and then encode this path in each packet. Other approaches distribute the routing information among the nodes, either grouping the units into clusters [23, 18], or treating the units independently (Temporarily-Ordered Routing Algorithm) [34, 33]. The Zone Routing Protocol (ZRP) [17] provides a hybrid approach between *reactive* routing schemes that search globally when a route is needed and *proactive* routing schemes that always have routes ready to send packets.

2.5 Current Technology

Many industry leaders are undertaking sizable efforts to provide mobile consumer products. For example, Lucent Technologies has developed the WaveLAN technology [26] to allow wireless connections among mobile units as well as connectivity to the fixed network through a wireless-to-wire bridge. Sun Microsystems' Jini [27] system promises networking anywhere, and overlaps with mobility in many aspects, including location dependent and dynamically changing object bindings. Alternately, a consortium of companies (including Ericsson, IBM, Intel, Nokia, and Toshiba) is supporting Bluetooth [12], a technology which intends to wirelessly link small groups of objects (on the order of ten) into ad hoc networks when they are in close proximity. Although Bluetooth is still in its specification phase, devices are anticipated within one year.

3 Objectives

The goal of this work is to enable the rapid development of dependable applications in the ad hoc mobile environment by developing a programming environment which provides a flexible level of abstraction to the programmer. This environment must balance the level of complexity with available expressive power. For example, location is a key element of mobility which can either be exposed to or hidden from the programmer. Our work will seek a balance and will expose locality in a meaningful way, but also allow it to remain entirely

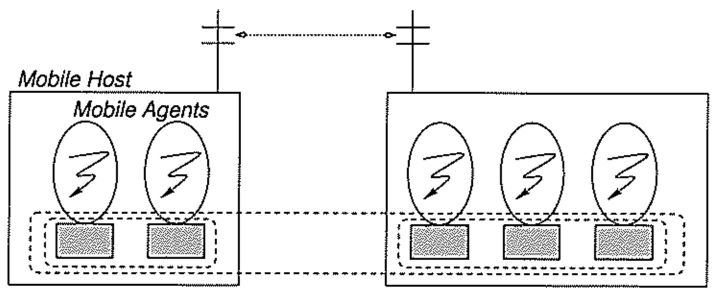

Figure 1: The goal of this work is to devise distributed data structures for the ad hoc coordination of both physically and logically mobile components. When components are co-located, the structures are shared, but when disconnected, operations are still possible and is manifested simply as a change in context for the operations.

hidden or abstracted, e.g., as a relative concept related to available connectivity or current host. The overall programming model must be easily understandable and natural to use in the ad hoc environment.

To address dependability, we intend to employ formal design techniques, specifically using Mobile UNITY, which at its core allows mobile applications to be specified and proven correct. By using Mobile UNITY to specify coordination constructs, as the programmer incorporates these constructs into applications, the constraints will be clearly defined, the behavior formally specified, and the integration of construct and application will be verifiable despite the inherent complexity and unpredictability of the environment. One tendency when using abstract specification models such as Mobile UNITY is to create models with large atomicity constraints which can lead to difficulties in the implementation phase. Therefore, we must take care that the models we define do not make such atomicity requirements from the environment. Instead, we will define models with minimal atomicity assumptions which are more implementable, however the guarantees of the model will be necessarily weaker. We expect that our specifications will rely heavily upon conditional properties such as patterns of movement, duration of connectivity, component density, etc. Exactly how these properties will be formally expressed is still to be determined and will contribute to the direction of this research.

Through the development of an environment to aid in the rapid development of dependable applications we will gain insight into the nature of ad hoc mobility, define problems and reasonable solution approaches, and develop a meaningful set of constraints within which to work to provide a flexible, powerful tool to aid the application programmer.

- **Global Virtual Data Structures.** By treating the issues of the ad hoc environment fundamentally as a coordination problem, a shared data structure arises as a natural solution to managing complexity. However, the concept of a single, global, persistent, shared data structure is not feasible in the mobile environment. Therefore we must develop ways to split the data structure among the mobile components while maintaining the illusion of a global data structure from the application perspective. The approach we take is to *virtually* maintain a global data structure while physically distributing the data to the mobile components. The connectivity of the network defines which portion of the data structure is available to the application, but we propose that the set of available operations remain constant. The execution of the operations themselves is dependent on the current context in which they are issued rather than the entire global context which may or may not be available due to connectivity. An additional aspect to this work is the integration of logical mobile components, or mobile agents, which reside on physically mobile units. The approach to maintaining the data structure and defining the context of the operations must be extended to handle this new tier of mobility. One possibility as shown in Figure 1, is to distribute the data among the mobile agents instead of the physical components, define complete connectivity of agents resident on the same physically mobile unit, and share the data structure at the physical level when physical connectivity is available. We expect that this duality between physical and logical mobility will prove to be valuable when devising solution strategies for applications in the ad hoc environment.

- **Programming Model.** For each data structure developed, we must define the programming model, provide formal semantics, and fully specify the implementation. The programming model will address what primitives are available to the application programmer, what view of the virtual global data structure is available to the programmer with respect to connectivity, and the way of thinking necessary to effectively use the data structure. The formal semantics will be provided using Mobile UNITY, and the accompanying proof logic will be available to reason about both the correctness of the implementation of the data structures as well as the correct composition of the data structures with user applications. This will lend dependability as well as provide a formal model for thinking about application development in the ad hoc environment. Implementation of these data structures will provide a key to the specific directions of our research. First of all, in order to contribute to our goal of rapid application development, these data structures must be implementable. Algorithms and protocols must be developed to support the data structures in the presence of changing connectivity. Performance bottlenecks must be identified to provide refinements to the data structures, increasing performance and further addressing issues such as weak consistency and conditional properties.

- **Evaluation.** Finally, we must address the applicability of these data structures and their ability to facilitate rapid application development. We must evaluate the overall usefulness of the programming model and interface from a usability standpoint. Because we anticipate the design of multiple global data structures, another task will be clarifying the benefits of these structures, comparing their performance under different constraints, as well as showing that we cover a breadth of application scenarios for the ad hoc environment.

4 Technical Challenges

In a sense, the ad hoc environment is mobility taken to the extreme. As discussed in Section 2.2, one of the common techniques for moving standard distributed algorithms to the base station mobility network is to push most of the computation onto the fixed structure to reduce the burden on the wireless communication links as well as reducing the computational burden on the mobile units. Additionally, the known reliability and routing of the fixed network can be exploited to carry out any state distribution necessary to the algorithm. However, in the ad hoc network, no such fixed network exists, the computation of the individual components is limited, and the only available bandwidth is wireless. Therefore any algorithms running in this environment must consider the limitations of the units, and be aware of communication overhead.

Another fundamental issue is the constantly changing connectivity of components. If the available connectivity among mobile components is modeled as a graph where the nodes are the mobile units and the channels are the available communication links, then the ad hoc mobility model is one where this graph is constantly changing. Although this could be viewed as similar to a fully connected graph of components with faults on certain communication channels, the typical assumption of that model is that a fault is temporary. In ad hoc mobility, however, the disconnections are not faults, but events caused by the mobile units themselves either intentionally to reserve power, or unintentionally by moving out of range. Additionally, the network might be arbitrarily partitioned and some nodes might permanently disconnect, changing the participants in the network. The changing view of a global virtual data structure must be defined with respect to changes of connectivity. The integrity of the data structure must be maintained in the presence of disconnections.

Additionally, the set of components in the system may be unknown in advance. Therefore any algorithms must first address current neighbor discovery, and possibly deal with the existence of nodes they will never be in direct contact with, but may be transitively connected through contact with another unit.

Another challenge with respect to disconnection is intentionality. When a user intentionally disconnects from the network for any reason, it is possible to use this advance knowledge to put the system into a consistent state before the actual disconnection occurs. We term this announced disconnection, because a user announces the intent to disconnect, and then waits for acknowledgment before actually disconnecting. However, in a real network, unintentional, or unannounced disconnection must be taken into consideration. For example, when a user suddenly moves into a region where the radio transmission is lost, not only is the user isolated from the rest of the network, but the status of any ongoing transmissions is unknown. Neither the party that was disconnected, nor those it was disconnected from know the status of the messages that were in transit. The problem is identical to that of distributed consensus. In practice, the impossibility of distributed consensus is addressed by using protocols such as two-phase commit, however these protocols rely on the fault model that a dropped link will eventually be restored and any inconsistent state can be reconciled at that time. In the model of ad hoc networks as a graph with faulty channels, this is not the case because a mobile component can fail permanently or for a long period, leaving an inconsistent, irreconcilable state. In addition to the previous constraint of data integrity in the presence of disconnection, the possibility of unannounced disconnection further brings up the ability to restore the data structure when connectivity is re-established, and the availability of data when a mobile unit is isolated after abrupt disconnection.

This leads to the more general notion of transactions in the ad hoc environment. In traditional networks, transactions should involve as few nodes as possible in order to maximize data or processing parallelism. In the mobile environment, the changing connectivity provides increased motivation to limit participants because as the connectivity changes, the transaction may no longer be able to commit. New strategies must be developed for deciding when a transaction should commit. For example, it might be meaningful for a transaction to commit only if there are no connectivity changes between the initiation and completion of the transaction. Overall, the issue of maintaining consistency among ad hoc components becomes more difficult to maintain. Transactions are a powerful tool to maintain a consistent data structure, but the scope of these transactions must be limited to increase performance of the system as a whole.

5 Research Strategy

We view the mobile system as a set of mobile units which move through space and communicate as conditions permit. At the programming level, we need to ask what constructs are best suited for building code for mobile applications. At the design level, we must determine what abstract computational model is most likely to simplify (informal) understanding of and (formal) reasoning about the overall system behavior. Rapid development comes with commensurate increases in expressive power, i.e., with a rise in the level of abstraction. Message passing and even remote method invocation are too low level. In the mobile context, matters are made worse by the fact that the open nature of the systems requires one to discover first who is part of the network at any given moment before accessing a resource on a different unit.

5.1 Global Virtual Data Structures

In the field of parallel programming tuple space communication à la Linda provides a good example of how coordination can simplify the programming task. Tuple space coordination facilitates temporal and spatial decoupling among parallel programs. By limiting the power of the tuple space access primitives efficient implementation is achieved as well. The reason tuple space coordination simplifies programming is that it presents to the programmer the appearance of a persistent global data structure which can be readily understood and operated on: a set of tuples accessed by content. We want to make this level of abstraction available for mobile applications using similar techniques. The absence of a fixed network that can store the tuple space in a persistent manner makes the task much more difficult. Our initial work in this area has lead to a model for Linda in a Mobile Environment, LIME, which distributes the global tuple space among the mobile units and limits access to the confines of each ad hoc network. For a programmer, mobility is perceived simply as an independently evolving host tuple space, i.e., a continuously changing context. When the mobile components are co-located, the tuple spaces are transiently shared and all tuple space accesses, including pattern matching for reading and removing data, are done on the now shared data space (see Figure 2). Additional primitives with extensions for location are provided to address specific tuple spaces, however the presence of the specified tuple space is dependent on connectivity. In addition to these primitives, LIME provides a mechanism to register for tuple space events, extending Linda to provide a "push" model in addition to the traditional "pull" model. The physical distribution of the tuple spaces also reduces the amount of coupling that should be expected among the components of the distributed system. Therefore we provide an additional primitive with reduced atomicity guarantees to allow reactions to events among hosts, but provide a stronger primitive for reactions within a host (among logical agents). The specification of LIME is given using the Mobile UNITY [39] semantics, allowing the formal verification of applications and algorithms built on top of LIME.

We intend to continue our work on LIME, as well as explore additional data structures. For each of these structures, the fundamental issues that need to be addressed as part of the evaluation are similar:

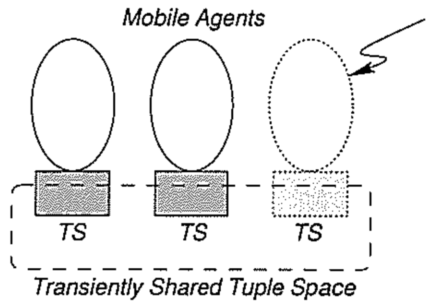

Figure 2: Creating the illusion of a globally shared tuple space in LIME. When units are within range, their tuple spaces are shared to form a federated tuple space. When disconnection occurs, tuples are distributed among the agents. Each agent perceives only local access to its own tuple space, but the content of the tuple space changes with connection and disconnection.

Does the data structure match the basic needs of the underlying application? Is there a natural and useful partitioning of the data structure across units and ad hoc networks? How is the data structure perceived by the individual units as changes in connectivity occur? In the case of a set (of tuples) there is little or no structure and the partition is an elegant mathematical concept. A tree, as in Figure 3, could also be partitioned among units with the nodes where a cut occurs being replicated. A global naming convention would allow communicating units to determine the relation between the tree fragments they carry and make contents and structural changes (e.g., swapping subtrees) as long as no disconnected units are affected. In principle, certain operations (e.g., adding a leaf node) could be issued at any time with their evaluation being delayed until such time that the affected units are within range. Attempts to access nodes on disconnected units may result in blocking the respective agent. One possibility is to let the data stored at each node in the tree be a set of tuples and and provide built-in rules to force data to migrate up and down the hierarchy in a transparent manner as units come in contact with each other. The generalization to a directed graph is straightforward and can overcome the problems caused by the possible loss of one of the units.

5.2 Integration with Logical Mobility

The LIME system integrates the research areas of physical and logical mobility, using the same abstraction for coordination of components. One of the uses for logical mobility which has been shown to be advantageous is semantic compression, reducing the overall bandwidth required of a network to complete a computation. Additionally, agents provide a mechanism which allows the initiator of the agent not to remain connected for the duration of the computation. These approaches seem easily tailored to the ad hoc environment where bandwidth is limited and connection varies. Therefore our work will not focus solely on the physical mobility, and will not only take advantage of the results from the logical mobility research community, but will provide new results for coordinating logical components.

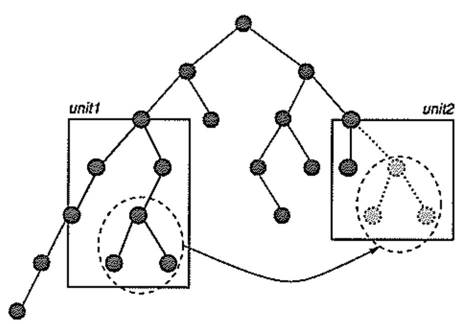

Figure 3: A hierarchical data structure where units in range agree to transfer a subtree which is under their jurisdiction even though parts of the global structure remain hidden. Moving a subtree distributes data to a different location to satisfy changing access patterns.

5.3 Coordination protocols

While the theoretical development of the data structures themselves is a research goal, the actual implementation will provide a measure for the success. The implementability of these coordination constructs is vital, however as the level of abstraction provided to the programmer is increased, the details necessary to complete the implementation similarly increase. Ensuring fairness and atomicity in the presence of disconnections is possible only under certain fortuitous assumptions about the behavior of the individual units, e.g., eventual reconnection, a sufficiently long connection time, or announced intent to disconnect. Actually, in the absence of proper assumptions most protocols, mobile or otherwise, make no guarantees. For this reason, proofs involving mobility tend to be conditional in nature. Some of the coordination actions we are contemplating at this time suggest the need for an underlying transaction system. We plan to investigate this option but we expect that more often than not we will need to be satisfied with weaker forms of consistency among the units' views of the global data structure, even when they are actually connected. As such we expect to propose implementation protocols that rely on the notions of stability and eventual convergence rather than atomic transition.

5.4 Algorithm development

In addition to these protocols, we anticipate that the implementation of the data structures will require algorithms not specific to the data structures themselves, but rather general to the ad hoc environment. Just as distributed computing has developed a rich set of algorithms for solving recurring issues, developing a similar set of algorithms tailored to ad hoc mobility is necessary. Although research has been done in fixed base station mobility [25, 1] by placing the majority of the processing at the fixed stations, this schema is not appropriate where each network component is itself mobile. Therefore algorithms analogous to established distributed computing algorithms, such as termination detection, leader election, and stable property detection must be revised with ad hoc mobility in mind. Not only do the algorithms themselves need to be redefined, but also weaker semantics must be formally defined to allow for reasoning about the performance of these algorithms.

The ad hoc environment also demands that certain classes of algorithms be revisited from the perspective of mobile computing. For example, randomized algorithms may provide a convenient mechanism for describing probabilistic guarantees. Also, as non-determinism proved to be a useful model in defining parallelism, randomness may prove to be a useful tool for modeling arbitrary movement of mobile units. Self-stabilization algorithms in distributed computing provide guarantees conditional on the relative stability of the network. Such guarantees are directly translatable to the mobile environment when considering automatic reconfiguration after mobile unit movement. Finally, epidemic algorithms may provide insight into mechanisms to spread information from one mobile unit to another as connections change.

The LIME system employs the basic data structure of a tuple space and defines the movement of data with respect to the location of a tuple inside a tuple space. This is one possible coordination construct designed to transfer data transparently with respect to a predefined policy. Other such policies and data structures can be developed for other application arenas. For example, data replication policies play an important role in many mobile settings due to the increase in availability, performance, and dependability. Weak consistency models have been discussed in the area of fixed networks and file systems [22], however the ad hoc environment remains unexplored. Although the tuple space provides general data storage, other structures, such as trees, may be more intuitive.

Other basic properties of the ad hoc environment can be exploited when developing standard algorithms. Properties such as position in space and direction of movement lead to optimization problems such as dynamic route minimization in connected systems, boundary detection in dense communities, and connectivity preservation in autonomous systems. One possible approach is the decomposition of a problem into subproblems based on connectivity. For example, identifying regions of high connectivity, then linking these regions with mechanisms with lower arity of connectivity, and different communication overheads. Although this decomposition seems to be a natural first step in mobile settings, there are many details to be worked out. For example, in order to not restrict mobile unit movement from one region to another, how can the algorithm be adapted to allow such movement? Is it reasonable to assume that such movement will be infrequent, and if so, what overhead are we willing to allow a reconfiguration protocol to consume? The issues of orphan detection also become significant when loss of units is expected as in robot systems with multiple redundant units and a built-in expected loss.

Another location dependent property of ad hoc mobility which could be exploited is the notion of network density. What are the properties of a sparse network where components are loosely connected and network partitioning is frequent? How do these properties differ from those of a dense network where it is possible that components are always transiently connected? Are the packet transmission collisions of a dense network a significant factor in the design of algorithms in this setting?

In each of these cases, solutions that favor early decisions by a subset of participants, provide stable or monotonic metrics, and can tolerate extended lack of involvement on the part of many units must be explored, formally defined, proven, and tested. In general, when developing these algorithms we must address which patterns of mobility are meaningful, what assumptions we can make about the network given these patterns, how can we exploit the patterns themselves, and what algorithms are actually meaningful and necessary in ad hoc mobility. Having the goal of implementing our global virtual data structures will feed this process, increasing our chances of success.

6 Conclusions

Ad hoc networks represent a complex technology which poses many challenges to computing. By completely removing the traditional wired network, many assumptions about the community of distributed components are fundamentally changed. In this proposal, we have set forth the goal of developing an environment for the rapid development of dependable applications in the ad hoc mobile environment. Our approach will include the development of *virtual global data structures*, the definition of a *programming model* to employ these structures, and an *evaluation* of the structures and how the constraints of the ad hoc environment are addressed. Our work will include the formal specification of the structures themselves, algorithms necessary to support the implementation of these structures, and prototype implementations to demonstrate feasibility. Our initial work on LIME has shown that our approach through distributed virtual global data structures is promising, and we expect that our previous work on algorithm development in the base station environment will lead to success in the ad hoc environment.

References

[1] A. Acharya, B.R. Badrinath, and T. Imielinski. Checkpointing distributed applications on mobile computers. In *Proceedings of the Third International Conference on Parallel and Distributed Information Systems*, October 1994.

[2] A. Acharya, B.R. Badrinath, T. Imielinski, and J.C. Navas. A WWW-based location-dependent information service for mobile clients. In *Fourth International WWW Conference*, July 1995.

[3] A. Acharya, A. Bakre, and B.R. Badrinath. IP multicast extensions for mobile internetworking. Technical Report LCSR-TR-243, Rutgers University, 1995.

[4] B.R. Badrinath, A. Acharya, and T. Imielinski. Structuring distributed algorithms for mobile hosts. In *Proceedings of the Fourteenth International Conference on Distributed Computing Systems*, pages 21–28, Poznan, Poland, 1994.

[5] B.R. Badrinath, A. Acharya, and T. Imielinski. Designing distributed algorithms for mobile computing networks. *Computer Communications*, 19(4):309–320, April 1996.

[6] M. Baldi and G.P. Picco. Evaluating the tradeoffs of mobile code design paradigms in network management applications. In *Proceedings of the Twentieth International Conference on Software Engineering (ICSE'98)*, pages 146–155, Kyoto, Japan, April 1998. IEEE CS Press.

[7] J. Broch, D.B. Johnson, and D.A. Maltz. The dynamic source routing protocol for mobile ad hoc networks. Internet Draft, March 1998. IETF Mobile Ad Hoc Networking Working Group.

[8] V. Chikarmane, R. Bunt, and C. Williamson. Mobile IP-based multicast as a service for mobile hosts. In *Proceedings of the Second International Workshop on Services in Distributed and Networked Environments*, Los Alamitos, CA, USA, June 1995. IEEE Computer Society Press.

[9] P. Ciancarini, R. Tolksdorf, F. Vitali, D. Rossi, and A. Knoche. Coordinating multiagent applications on the WWW: A reference architecture. *IEEE Transactions on Software Engineering*, 24(5):362–375, May 1998.

[10] D. Coore, R. Nagpal, and R. Weiss. Paradigms for structure in an amorphous computer. A.I. Memo No. 1614, Massachusetts Institute of Technology Artificial Intelligence Laboratory, October 1997.

[11] S. Corson and J. Macker. Mobile ad hoc networking (MANET): Routing protocol performance issues and evaluation considerations. Internet Draft, March 1998. IETF Mobile Ad Hoc Networking Working Group.

[12] Ericsson, IBM, Intel, Nokia, and Toshiba. Bluetooth. http://www.bluetooth.com, 1999.

[13] A. Fuggetta, G.P. Picco, and G. Vigna. Understanding code mobility. *IEEE Transactions on Software Engineering*, 24(5):342–361, 1998.

[14] M. Fukuda, L.F. Bic, M.B. Dillencourt, and F. Merchant. Intra- and inter-object coordination with MESSENGERS. In *Proceedings of Coordination Languages and Models. First International Conference (COORDINATION '96)*, pages 179–96, Cesena, Italy, April 1996. Springer-Verlag.

[15] M. Fukuda, L.F. Bic, M.B. Dillencourt, and F. Merchant. Distributed coordination with MESSENGERS. *Science of Computer Programming*, 31(2), May 1998. (To appear).

[16] R.S. Gray. Agent Tcl: A transportable agent system. In *Proceedings of the CIKM Workshop on Intelligent Information Agents*, Baltimore, MD, USA, December 1995.

[17] Z.J. Haas and M.R. Pearlman. The zone routing protocol (ZRP) for ad hoc networks. Internet Draft, August 1998. IETF Mobile Ad Hoc Networking Working Group.

[18] M. Jiang, J. Li, and Y.C. Tay. Cluster based routing protocol (CBRP) functional specification. Internet Draft, March 1998. IETF Mobile Ad Hoc Networking Working Group.

[19] D.B. Johnson and D.A. Maltz. Truly seamless wireless and mobile host networking. Protocols for adaptive wireless and mobile networking. *IEEE Personal Communications*, 3(1):34–42, 1996.

[20] A.D. Joseph, A.F. de Lespinasse, J.A. Tauber, D.K. Gifford, and M.F. Kaashoek. Rover: A toolkit for mobile information access. *Operating Systems Review*, 29(5):156–71, 1995.

[21] R.H. Katz. Adaptation and mobility in wireless information systems. *IEEE Personal Communications*, 1(1):6–17, 1994.

[22] J.J. Kistler and M. Satyanarayanan. Disconnected operation in the coda file system. *ACM Transactions on Computer Systems*, 10(1):3–25, 1992.

[23] P. Krishna, N.H. Vaidya, M. Chatterjee, and D.K. Pradhan. A cluster-based approach for routing in dynamic networks. *Computer Communication Review*, 27(2):49–64, April 1997.

[24] S. Kryukova, B. Massingill, and B. Sanders. Specification and proof of an algorithm for location management for mobile communication devices. In *International Workshop on Formal Methods for Parallel Programming: Theory and Applications*, Geneva, Switzerland, 1997.

[25] J. Matocha. Distributed termination detection in a mobile wireless network. In *36th Annual ACM Southeast Conference*, Marietta, GA, April 1998.

[26] Lucent Technologies. Wavelan. http://www.wavelan.com, 1999.

[27] Sun Microsystems. Jini. http://www.sun.com/jini, 1999.

[28] J. McLurkin. Using cooperative robots for explosive ordnance disposal. Massachusetts Institute of Technology Artificial Intelligence Laboratory.

[29] A.L. Murphy, G.-C. Roman, and G. Varghese. An exercise in formal reasoning about mobile computations. In *Proceedings Ninth International Workshop on Software Specification and Design*, pages 25–33, Ise-Shima, Japan, 1998. IEEE Computer Society Press.

[30] A. Myles and D. Skellern. Comparing four IP based mobile host protocols. *Computer Networks and ISDN Systems*, 26(3):349–355, 1993.

[31] J. Mysore and V. Bharghavan. A new multicasting-based architecture for Internet host mobility. In *Proceedings of The Third Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'97)*. ACM/IEEE Press, September 1997.

[32] B. Noble, M. Satyanarayanan, D. Narayanan, J.E. Tilton, J. Flinn, and K. Walker. Agile application-aware adaptation for mobility. In *Proceedings of the Sixteenth ACM Symposium on Operating System Principles*. ACM Press, October 1997.

[33] V. Park and M.S. Corson. Temporally-ordered routing algorithm (TORA) version 1. Internet Draft, August 1998. IETF Mobile Ad Hoc Networking Working Group.

[34] V.D. Park and M.S. Corson. A highly adaptive distributed routing algorithm for mobile wireless networks. In *Proceedings of IEEE INFOCOM '97*, Kobe, Japan, April 1997.

[35] C.E. Perkins. IP mobility support. Technical Report RFC 2002, IETF Network Working Group, October 1996.

[36] C.E. Perkins and D.B. Johnson. Mobility support in IPv6. In *Proceedings of the Second Annual International Conference on Mobile Computing and Networking (MobiCom'96)*, Rye, NY, USA, November 1996. ACM Press.

[37] G.P. Picco. μCODE: A lightweight and flexible mobile code toolkit. In *Mobile Agents: Second International Workshop MA '98, LNCS 1477*, pages 160–171, Stuttgart, Germany, September 1998. Springer-Verlag.

[38] R. Prakash and M. Singhal. Dynamic hashing + quorum = efficient location management for mobile computing systems. In *Proceedings of the Sixteenth Annual ACM Symposium on Principles of Distributed Computing (PODC)*, page 291, Santa Barbara, CA, USA, August 1997. ACM Press.

[39] G.-C. Roman, P.J. McCann, and J.Y. Plun. Mobile UNITY: Reasoning and specification in mobile computing. *ACM Transactions on Software Engineering and Methodology*, 6(3):250–282, 1997.

[40] M. Satyanarayanan. Mobile information access. *IEEE Personal Communications*, 3(1), February 1996.

[41] M. Satyanarayanan, J.J. Kistler, P. Kumar, M.E. Okasaki, E.H. Siegel, and D.C. Steere. Coda: a highly available file system for a distributed workstation environment. *IEEE Transactions on Computers*, 39(4):447–459, April 1990.

[42] B.N. Schilit, N. Adams, and R. Want. Context-aware computing applications. In *Proceedings of the Workshop on Mobile Computing Systems and Applications*, pages 85–90, Santa Cruz, CA, USA, 1994. IEEE.

[43] C.D. Tait and D. Duchamp. An efficient variable consistency replicated file service. In *Proceedings of the USENIX File Systems Workshop*, pages 111–126, Ann Arbor, MI, USA, 1992.

[44] D. Terry, M. Theimer, K. Petersen, A. Demers, M. Spreitzer, and C. Hauser. Managing update conflicts in Bayou, a weakly connected replicated storage system. *Operating Systems Review*, 29(5):172–183, 1995.

[45] G.M. Voelker and B.N. Bershad. Mobisaic: An information system for a mobile wireless computing environment. In *Proceedings of the Workshop on Mobile Computing Systems and Applications*, pages 185–90, Santa Cruz, CA, USA, 1994. IEEE.

[46] J.Z. Wang. A fully distributed location registration strategy for universal personal communication systems. *IEEE Journal on Selected Areas in Communications*, 11(6):850–860, August 1993.