Looking for a Common View for Mobile Worlds

Maria Gradinariu, Michel Raynal
IRISA/INRIA-CNRS, Université Rennes 1
35000 Rennes, France

Gwendal Simon
France Telecom R & D
92794 Issy Moulineaux, France

Abstract

This paper considers central issues of distributed computing in a mobile environment. Its aim is to light on the first brick of a common view for mobile systems. We pool together mobile systems and analyze them from different angles including architecture and computability aspects. We show that mobile systems (i.e., cellular systems, ad hoc networks, peer-to-peer systems, virtual reality systems or co-operative robotics) are basically confronted with the same problems, hence it is useless to maintain the actual "misleading" barriers. Due to important similarities between the different mobile systems we claim that the next logical step in building a common view is to design a model which conceptually should unify them by providing an abstract description of the parameters which distinguish these systems from classic distributed systems. Moreover, in the new model, the fundamental problems of distributed computing (i.e., (k-)mutual exclusion, leader election or group communication) should find appropriate specifications.

1 Introduction

The distributed computing area includes henceforth mobile systems (i.e., cellular and ad-hoc networks, peer-to-peer systems, mobile virtual reality and cooperative robotics). The classic models and solutions are not adapted to mobile worlds hence new models, new problems and obviously new solutions should be designed. This paper advocates for the creation of a common view for mobile worlds based on the similarities between them. In the sequel we present in an informal manner the main characteristics of what are usually referred as mobile systems, then we describe the elements on which should be based the design of a mobile system model and finally present in an exhaustive manner the models proposed so far in the mobile systems literature. We continue our investigation with the problems solved in mobile systems starting with specific problems. Then we present solutions provided so far for some fundamental problems in distributed computing (i.e., (k-)mutual exclusion and leader election). We conclude by presenting our opinion on open problems or possible future research directions in mobile systems.

2 Mobile Worlds

In this Section, we show that the *Mobile System* appellation is not restricted to wireless mobile systems, but includes several research areas. Typically, we observe that different research communities converge towards an architecture design of systems characterized by the notions of position and mobility.

2.1 Mobile Systems

In this Section, we present characteristics of some popular mobile systems.

Cellular Networks In Cellular Networks, a base station manages an area of world called *cell*. Mobile hosts are connected with the base station of the cell to which they belong. Moreover, the mobile hosts cannot establish a direct connection between them, they rely on a base station to communicate. Interconnected base stations and cell shapes are traditionally fixed in mobile phone systems.

Ad-Hoc Networks Ad-Hoc Networks differ from Cellular Networks in not relying on any specific infrastructure. Each mobile host can send messages to the nodes within its transmission range. Moreover, it can route messages on behalf of others. That is, two mobile hosts can communicate either over a direct wireless link or over a sequence of wireless links including one or more intermediate nodes.

Peer-to-Peer Systems A Peer-to-Peer system is a dynamic and scalable set of peers that distributes the cost of sharing data. Each peer can join or leave the system at any moment and can communicate with any other peer under the only hypothesis that the two peers are aware of each other. The main characteristics of peer-to-peer systems are the ability to pool together and harness large amount of resources, self-organization, load-balancing, adaptation and fault-tolerance.

Shared Virtual Reality Systems In general, a Virtual World means a technology for moving through and interacting with a three-dimensional computer-generated environment such that the experience is perceived to be real. Shared Virtual Environments are inhabited by interconnected *Entities* driven by users (avatars) or by computer (virtual objects). Entities, characterized by a position in the virtual world, enter and leave the world, move from one virtual place to another and interact in real-time. In the real world, which virtual environments emulate, entities have a limited area of interest. Thus, a walking person typically has an area of interest of only several hundred meters. So, in a virtual environment, entities need only to be aware of each other within immediate surroundings.

Cooperative Robotics and Nanorobotic Systems A cooperative robotic system is a collection of robots which are organized in a single coherent team. The main difference between classic distributed mobile systems and cooperative robotic systems is the requirement that robots should control their own motion. The fundamental difference between the nanorobots and the cooperative robots is their ability to reproduce themselves (i.e. produce at any moment physical copies of themselves). [7] surveys cooperative (nano)robotics systems pointing out similarities between these systems and ad-hoc networks.

2.2 Mobile Architectures

From the architectural point of view, we distinguish pure systems (also called uniform systems) and hybrid systems (also referred as hierarchical systems).

Pure Systems A mobile system is referred as *pure* when all participants have equal or similar role. Nodes have identical capabilities and responsibilities, and all communications are symmetric. Among the different mobile systems, peer-to-peer implementations like Gnutella [12] or Freenet [6] are relevant examples of pure systems. Despite the heterogeneity in transmission ranges, ad-hoc networks are pure systems, in which all nodes are simultaneously server, router and client. An example of virtual reality system based on a pure architecture is Solipsis [18].

In pure systems only a local view is accessible, hence the application design is more fastidious due to the system symmetry. Instead, a pure system offers as main advantages its scalability, no additional cost for the infrastructure maintenance and its adequacy to the anonymous applications.

Hybrid Systems In *hybrid* systems, nodes do not have equal roles. Cellular networks are typical hybrid systems in which base stations and mobile hosts have different functionalities. The majority of shared virtual reality systems are based also on a hybrid architecture. Some peer-to-peer systems are based on the “super-peers” notion [34]. A super-peer network operates like a pure system for super-peers, while non super-peers are connected to one unique super-peer following a classic client/server paradigm. Since the number and the type of clients per super-peer can vary, super-peer networks are not symmetric.

3 Modeling a Mobile World

A very first step in modeling a distributed system is to find a model for the communication between system entities. In classic static distributed systems, communications are modeled by a graph whose nodes represent the processors and the edges represent the physical links between the processors. In the following we first show that in mobile systems the communication graph may not reflect the physical connections between system entities. We then present some models designed for particular mobile systems.

3.1 Preliminaries

3.1.1 Position and Communication Graph

In order to define the communication graph of a mobile systems it is important to precise the notion of position and its impact on the system topology. Hence, we distinguish two classes of mobile systems: systems subject to physical or applicative constraints, and free (or non-constrained) systems.

Constrained Systems In mobile systems based on wireless transmission, the notion of position corresponds to the physical location of the mobile hosts. Due to the limited transmission radius of the wireless links, a mobile host can communicate only with physically closed mobile hosts. Hence, there is a natural correlation between physical neighboring and the links in the communication graph.

Entities in a shared virtual reality environment are interconnected through non-constrained links. They have the ability to connect with any other entity present in the system. In this case the application does impose constraints in order to be conform to the reality. Since the entities must be aware of their virtual neighbors, there is a correlation between the distance which separates two virtual positions and the edges of the communication graph.

Overall, a node in the communication graph is characterized by its position and the edges are function of the nodes position. The communication graph of the constrained systems could be modeled by the neighborhood graphs of a finite planar set [14].

Non-Constrained Systems Users of peer-to-peer applications have no user-specified position and are able to connect with any other users. The notion of position is based on a choice of the application. Applications like Freenet give to the users a position depending on the keys of requested files, while in CAN or Chord the position is based on a key identifier in a virtual d-dimensional Cartesian space. In Gnutella [12], the position cannot be assimilated to any determined variable. The only assumption is that a new user is the neighbor of its connection point(s).

Non-constrained systems do not limit connections to close nodes. Since connections between nodes do not depend on the physical proximity, communication graphs are not conform with the neighborhood graph characteristics. and virtual structures/models are more appropriate [1].

3.1.2 Mobility and Communication Graph

In classic distributed systems the communication graph is static. In mobile systems, the participants can leave or join the system at any time and the connections between participants may be transient due to position changes. Consequently, the communication graph should reflect these changes. On the other hand the classic notion of failure should change of meaning since in some mobile systems there is no difference between a crash and a disconnection. In the following we discuss the impact of both connection/disconnection and node motion on the communication graph representing diverse mobile systems.

Connexion/Disconnection In pure systems, the disconnection of a node may induce important modifications of the communication graph or the lost of system data. For example, a mobile host in an ad-hoc network may be the only point connecting two partitions of the network: the disconnection of this particular node induces network partitioning. In the case of peer-to-peer file sharing systems, some data files can be found only at one location. If the node at this position disconnects, the data stored in here are lost.

In hybrid systems, node disconnection induce heterogeneous modifications of the communication graph or of the structure and the content of the system. For instance if the client of a super-peer disconnects the consequences are limited, while the disconnection of a super-peer implies that the set of its clients is suddenly disconnected and a recovery procedure should be executed.

Node Motion In mobile systems, nodes may move by changing their position. This originates from the physical motion or to the application layer.

In pure systems where the nodes are connected to their closest neighbors, when a node moves, its neighbors change. In this case the moving node opens connections with its new neighbors and closes the connections with the old neighbors which are further away. The mechanisms that allows new neighbors discovery could be automatic in wireless networks while it requires collaboration between nodes in a shared virtual environment [18].

In hybrid systems, a variation of position may have two different impacts. If the moving node does not leave its cluster, then only the node position will be changed in the information table of the super-peer. Instead, if the node goes out of its cluster, it has to connect with the super-peer that is responsible of the host cluster. In this case, the node requires the recuperation of informations on this super-peer and, eventually, all the routing tables of the cluster are modified.

Note that in some systems the speed is bounded (e.g., ad-hoc networks), while others facilitate abrupt change of the position (peer-to-peer networks or shared virtual reality systems admitting teleportation). In the former case some adjacent nodes remain adjacent after the motion, in other words the motion impact on the communication graph is minimal. In the last case, the motion may be assimilated to a disconnection followed by a reconnection in some other point of the network which could completely change the topology of the communication graph.

In mobile systems the communication graph should be able to reflect both nodes and edges deletion or addition. The mathematical model the most appropriate is the dynamic graphs [8] model.

Overall, a model for the communication graph of a distributed mobile system should be a mixture between neighborhood graphs, dynamic graphs and logical graphs.

3.2 Models for Mobile Systems

In mobile systems models were proposed function on the specificity of each system. For example in ad hoc networks models refer mainly the communication while in peer-to-peer systems models are oriented in order to optimize the search function.

Adhoc Networks Models in the context of ad hoc networks are motion based or application based. The first motion based model for ad hoc networks is presented in [15]. In this model, hosts are moving in a three dimensional space and each host has a transmission range which could be approximated by a regular polyhedron (authors choose the cube as regular representation). A mobile system is a graph whose nodes are the cube modeling the range transmission for every host and an edge between two nodes occurs in the graph if the corresponding cubes are adjacent. In this system authors define two classes of protocols — the compulsory protocols which impose to mobile hosts a specific motion schema in order to meet the protocol demands, and non-compulsory protocols that take advantage of the natural mobile host movement and are based on the communication between hosts which are meeting incidentally.

In [25] the authors present a layered based architecture for group communication in ad hoc networks. The group membership layer uses the informations provided by a proximity layer which main role is to find all nodes at distance d from a mobile host.

Shared Virtual Reality The most inclusive models for the communications in the shared virtual reality is presented in [3, 13]. Similar to ad-hoc networks, a spatial model is proposed which uses the properties of the space as the basis for mediating interactions. The entities carry a sub-space called *aura* which acts as an enabler for the potential interactions. When two auras collide, interactions between the entities become possible. Each entity may owns different auras for different media.

The entities measure their awareness by manipulating two key notions referred as *focus* and *nimbus*, subspaces within which an object chooses to direct either its presence or its attention. Specifically, the more an entity p is within the focus of an entity q, the more q is aware of p. Conversely, the more an entity is within your nimbus, the more he is aware of you. So, the level of awareness that entity A has of entity B is some function of A's focus and B's nimbus.

Peer-to-Peer Networks Most popular models for peer-to-peer systems are CAN [27], Pastry [29] and Chord [30].

In the Pastry system [29], each peer has an unique identifier. The routing in a Pastry system is realized in a greedy fashion. With concurrent node failures, eventual delivery is guaranteed unless $l/2$ (where l equals 16) or more nodes with adjacent identifiers fail simultaneously.

The CAN design [27] is based on a virtual d-dimensional Cartesian coordinate space. At any point in time the space is dynamically partitioned among all the nodes in the system. Each CAN node maintains a coordinate routing table that holds the IP address and virtual coordinate zone of each of its neighbors in the coordinate space.

The Chord system [30] maps keys onto nodes using a consistent hashing technique. A node's identifier is chosen by hashing the node's IP address, while a key identifier is produced by hashing the key. Each Chord node needs "routing" information about only a few other nodes. Because the routing table is distributed, a node resolves the hash function by communicating with a few other nodes.

General Model Based on the observation that the recurrent factor in all the aforementioned approaches is the virtual organization, in [1] the authors proposed a model which represents the network organization as a multi-layer system, each layer being characterized by its own communication graph and the logical orientation of the communication links. In this representation a node can simultaneously belong to more than one layer at a time.

4 Mobile Computing

This section first gives some pointers to specific problems in each class of mobile systems. Then it describes some general problems. Finally, it considers a few fundamental problems in the distributed computing area and their mobile solutions.

4.1 Specific Problems

The specific problems in the cellular and ad hoc networks are in relation with the autonomy of mobile hosts, routing and control of energy.

The autonomy of a mobile host depends on its transmission range, so minimizing energy costs affects network connectivity and routing efficiency. Current research in ad-hoc and cellular networks focuses on associating efficient *routing* [17, 31] and *energy reduction* [19].

The *routing problems* are also studied in the context of peer-to-peer networks. Moreover, real applications (file or data sharing) of these networks impose the study of methods for improving *search efficiency* [20, 30] and *anonymity preservation* [6, 12].

In virtual reality systems a specific problem is the optimal *partitioning of the shared virtual world*. Techniques in partitioning the shared virtual world and giving to each segment its own multicast group are presented in [22]. All entities belonging to the same segment communicate through multicast, so virtual neighbors are connected. Informations on segments are given to entities by a server or an entity acting as a server. If the partitioning was initially presented as static, different research projects have investigated the matching between segment shapes and virtual world characteristics (rooms, buildings, etc...) [2, 26], adapting segment size with entity density or organizing the virtual world in a hierarchical database [9].

4.2 General Problems

Resource Discovery Resource discovery is a goal for almost all mobile applications. When a mobile host in ad-hoc or cellular networks decides to open the communication with another mobile host, the underlying distributed algorithms are generally identical to those used in order to discover/search data in peer-to-peer system. The simplest algorithm consists in flooding the network. Even though the ends successfully this mechanism is too resource-costly to be implemented. Among alternatives, the greedy algorithm is very popular. When a node receives a message containing the position of some resource, the message is sent to the node that is the closest to the carried position among its adjacent neighbors. Despite the fact that the message may not follow the optimal path on a graph, this routing algorithm is efficient in a graph with no global view. Note that this algorithm can fail when the path followed by the message does not reach the resource.

Current research in ad-hoc networks focuses on modeling the network and the routing scheme together, attempting to optimize some of the relevant parameters of both simultaneously [14, 18].

Information Dissemination In mobile systems the common techniques in order to diffuse information are flooding and dynamic multicast trees. Both techniques are expensive, the former in the number of messages to be sent while the latter in the additional cost in order to maintain the tree structure in a completely dynamic environment. Recently, another information dissemination scheme, namely the publish/subscribe scheme, was proved to be well-suited to the mobile systems.

In the case of peer-to-peer networks publish/subscribe schemes are constructed on top of two popular object location and routing substrates, Pastry and CAN. The SCRIBE system [4] builds a multi-cast tree per group on top of a Pastry overlay and relies on Pastry in order to optimize the routes from the root to each group member. The CAN multi-cast [28] does not built multi-cast trees. The messages are flooded to all nodes in a CAN overlay network. Multi-groups are supported by creating a separate CAN overlay per group.

[16] was the first work which adapted the classic static publish/subscribe schemes to the cellular networks. In [1] the authors proposed a publish/subscribe scheme for general mobile systems.

Connectivity The graph connectivity is another common problem for all mobile systems. In hybrid systems, the management of communications between key elements of the network — super-peers in super-peer based networks, servers in shared virtual reality systems or base station in cellular networks — reduce the risk of the disconnection.

In pure systems and mainly in ad-hoc networks, some geometric solutions issued from the 2-dimensional neighborhood graphs theory are recently proposed in [19]. The connectivity of ad-hoc networks relies in most general case on hypothesis of potentially large transmission range in area with dense population. Because of their non-constrained characteristics, peer-to-peer systems are more flexible. In particular, when a new node joins the system, the application can use it to perform another redundant path between several nodes. [1] addresses explicitly the connectivity problem in the context of mobile systems.

4.3 Fundamental Problems

Classic problems of distributed computing (e.g., mutual exclusion, leader election or k-mutual exclusion) are also studied in the context of mobile systems. Implementing these classic problems in a mobile environment is a hard problem since in a completely dynamic system it is easy to prove that these problems have no deterministic solution, e.g., the node disconnection could generate for instance that the unique token/privilege in the network. Moreover the network partitioning imposes the use of a partition detection mechanism.

Impossibility Results In the following we show intuitively that it is impossible to provide a deterministic solution to the mutual exclusion problem in a completely dynamic environment (a similar scenario could be exhibit for k-mutual exclusion or leader election problems).

Let us consider the simplest topology (the ring) and assume that the number of processors in the system is 3, in clockwise order p_1, p_2 and p_3. Now, assume that the privilege is held by the processor p_1 and that the processor p_2 changes frequently its position. Assume also that p_2 requests the token and just before receiving the token p_2 changes again its position. The new configuration of the network is p_1, followed in clockwise sens by p_3 and p_2. In order to be passed to p_2 the token is send now by p_1 to p_3. The communication graph changes again such that the new position of p_2 is between p_1 and p_3 and the token is held by p_3. Moreover, the processor p_2 keeps requesting the token. Using this scenario the processor p_2 even if it is the only processor which is requesting the token may never be the privileged processor.

Basically, when the classic specifications of the fundamental problems are imported without modifications the main risk is to not be able to provide deterministic or even probabilistic solutions. Hence, the necessity to redefine the specifications/problems in order to capture the characteristics of the environment (i.e., dynamicity or mobility) and to be implementation free. Unfortunately, the specifications provided so far for classic distributed problems in mobile environments are merely orientated to specific implementations loosing in this way their generality.

Mobile Solutions In the following we visit the main solutions proposed for the mutual, k-mutual exclusion and leader election. Part of the solutions ([23, 32, 33]) are based on the reversal schemes of [11] implemented on top of a rooted DAG architecture. The idea is very simple: in a rooted DAG every sink node other than the root totally or partially reverts its edges. Hence in a finite number of computation steps the only sink of the network is the root. These two schemes are also referred direction orientated schemes and designed in order to define efficient routing in mobile systems. The main feature of the two schemes is that, for every node in the system, there is at least a directed path from the node to the system root. The main drawbacks is that the schemes use unbounded memory and that they require a fixed (always connected) root.

The mutual exclusion algorithm constructed on top of the partial reversal scheme assumes that the unique privilege in the network is held initially by the root. When a processor needs the token it reverts its adjacent edges in order to become the new unique sink of the network. The k-mutual exclusion uses merely the same idea the only difference between the two implementations is the fact that in the latter the number of tokens in the network is k and the tokes are held by the first k nodes. When a non privileged node requests the token it reverts its edges till it obtains the first available token.

The partial reversal scheme [11] was modified in [24] in order to cope with network partitioning. On top of this new scheme [23] design their leader election algorithm. Another solution for leader election is provided in [15], the solution uses a regular communication graph. Thus, the provided solutions and the proposed analysis are based on network regularities. This solutions are not suited to pure mobile systems since their communication graph are not regular.

5 Concluding Remark and Open Problems

This paper pointed out similarities among common mobile systems (i.e., cellular networks, ad hoc networks, peer-to-peer systems, reality virtual systems and cooperative robots). Moreover, based on this observation we claim that there should be a common model which characterizes their common parameters.

In the following we draw up a list containing problems which, in our opinion, should be further investigated in the context of mobile systems.

What is the mobility, dynamicity or scalability of a mobile system? In the area of mobile systems the notions mobility, dynamicity or scalability are often used without any formal definition. Moreover, these terms are used to describe the same behavior. In order to provide a unified view on mobile systems these notions should find a formal definition and eventually the different degrees of dynamicity, mobility and scalability should be appropriately measured.

What is the appropriate model for mobile systems? Mobile systems need a complete model which should characterize not only the communications between entities but also the executions of a mobile system. The I/O automaton [21] is a formal model for distributed systems. It is obvious that the classic I/O automaton should be augmented with the mobility notion in order to obtain a complete model for mobile systems. Exhibiting general properties of the new automaton and defining new techniques of composition could be a next step in mobile systems modelisation area.

What are the specifications of the distributed fundamental problems? In the previous section we pointed out that the specification of classic problems in distributed computing should be revisited. We also showed that it is impossible to provide solutions for these problems when there is no information on the system dynamicity. Hence, there are two possible approaches: provide new specifications for these problems or exhibit the minimal hypothesis on the system dynamicity in order to solve these problems. Moreover, designing algorithms solving these problems that cope with network dynamicity and partitioning could be a new directions in mobile computing area.

What are the appropriate communication primitives? The design of the communication primitives was the first step in mobile computing research. Hence, the classic communication primitives (e.g., broadcast, multicast) were modified in order to cope with a mobile environment and new ones (e.g., gossip) were proposed. It is obvious that every of aforementioned communication primitives has its limitation (For instance the construction of the multicast tree is very expensive in terms of control messages and the main drawback of the gossip technique is the network flooding). Designing hybrid communication primitives overcoming these drawbacks should be one of the priorities in mobile computing.

What are the most adapted simulators? Testing algorithms in the mobile computing area is a hard task which motivated the design of specialized simulators. Recently, in [5] the authors show that, surprisingly, for the same set of tests the simulators responses are completely different. Designing an accurate simulator could be another new direction for practical mobile computing.

What is the specification of mobile publish/subscribe? The publish/subscribe is a service designed in order to disseminate in an intelligent manner the information in large systems. Hence, the information is not diffused to all the members of the system but to part of them which previously manifested a real interest to the information. The literature proposes algorithms and techniques in order to implement the publish/subscribe scheme but there is no agreement on its formal specification. A formal definition of the publish/subscribe scheme should take into account both the notification semantics and the publishing semantics.

How to efficiently manage caches in a mobile system? The classic solutions for cache management do not take into account the cache user mobility. [10] pointed out that in a mobile system new caching techniques should be designed in order to increase the availability of the cached services to mobile users. Moreover, the new caching algorithms should cope with the nature of mobile systems (with the communication cost and the storage cost).

How to measure the complexity of mobile algorithms? Generally, the algorithms provided as solutions of mobile problems have no proofs or are “proved” by means of simulations. Furthermore, the complexity measures are imported from static distributed theory without any modification which is inappropriate for a dynamic/mobile system. New complexity measures suited to system mobility and new proof techniques should be defined.

What is the self-organization of a mobile system? Among the features of mobile algorithms usually is invoked the self-organization. Surprisingly, the notion of self-organization has no formal definition and there are no formal proofs in order to verify if a mobile system is self-organizing or not. Hence, providing a formal definition for the notion of self-organization and techniques in order to verify the self-organization of a mobile system could be one of the important steps that should be addressed in the future.

References

[1] E. Anceaume, A. K. Datta, M. Gradinariu, and G. Simon. Publish/subscribe scheme for mobile networks. *Proc. 2nd Int. Workshop of Principles of Mobile Computing (POMC'02)*, pages 74–81, 2002.

[2] J. W. Barrus, R. C. Waters, and D. B. Anderson. Locales and Beacons: Efficient and Precise Support for Large Multi-User Virtual Environment. Technical report, Mitsubishi Electric Research Laboratory, August 1996.

[3] S. Benford, J. Bowers, L. E. Fahlén, and C. Greenhalgh. Managing Mutual Awareness in Collaborative Virtual Environments. In *VRST'94*, August 1994.

[4] M. Castro, P. Druschel, A.-M. Kermarrec, and A. Rowstron. Scribe: A large-scale and descetralized application-level multicast infrastructure. *IEEE Journal on Selected Areas in Communications*, 20(8):100–111, 2001.

[5] D. Cavin, Y. Sasson, and A. Schiper. On the Accuracy of MANET Simulators. *Proc. 2nd Int. Workshop of Principles of Mobile Computing (POMC'02)*, pages 38–43, 2002.

[6] I. Clarke, O. Sandberg, B. Wiley, and T. Hong. Freenet: A distributed anonymous information storage and retrieval system. In *Proceedings of the ICSI Workshop on Design Issues in Anonymity and Unobservability*, Berkeley, CA, 2000.

[7] X. Defago. Distributed computing on the move: from mobile computing to cooperative robotics and nanorobotics. *Proc. 1st Int. Workshop of Principles of Mobile Computing (POMC'01)*, pages 45–55, 2001.

[8] D. Eppstein, E. Galil, and G. F. Italiano. Dynamic Graph Algorithms. Technical report, University Ca'Foscari di Venezia, october 1996.

[9] E. Frécon and M. Stenius. DIVE - A Scaleable network architecture for distributed virtual environments. *Distributed Systems Engineering Journal (special issue on Distributed Virtual Environments)*, Vol. 5, September 1998.

[10] R. Friedman. Caching web services in mobile ad-hoc networks. *Proc. 2nd Int. Workshop of Principles of Mobile Computing (POMC'02)*, pages 90–96, 2002.

[11] E. Gafni and D. Bertsekas. Distributed algorithms for generating loop-free routes in networks with frequently changing topology. *IEEE Transactions on Communications*, C-29(1):11–18, 1981.

[12] Gnutella. Gnutella website. http://gnutella.wego.com.

[13] C. Greenhalgh and S. Benford. Supporting Rich and Dynamic Communication in Large-Scale Collaborative Virtual Environments. *MIT Presence*, February 1999.

[14] Y. Hassin and D. Peleg. Sparse communication networks and efficient routing in the plane. *Distributed Comp.*, 2001.

[15] K. Hatzis, G. Pentaris, P. Spirakis, V. Tampakas, and R. Tan. Fundamental control algorithms in mobile networks. *Proc. 11th Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA '99)*, pages 251–260, 1999.

[16] Y. Huang and H. Garcia-Molina. Publish/subscribe in a mobile environement. *Proc. Workshop on Data Eng. for Wireless and Mobile Access (MOBIDE'01)*, pages 27–34, 2001.

[17] B. Karp and H. T. Kung. GPSR: Greedy Perimeter Stateless Routing for Wireless Networks. In *6th ACM Conference on Mobile Computing and Networking (Mobicom '00)*, 2000.

[18] J. Keller and G. Simon. Toward a Peer-to-Peer Shared Virtual Reality. In *IEEE Workshop on Resource Sharing in Massively Distributed Systems*, July 2002.

[19] L. Li, J. Halpern, P. Bahl, Y. M. Wang, and R. Wattenhofer. Analysis of a Cone-Based Distributed Topology Control Algorithm for Wireless Multi-Hop Networks. In *ACM Symposium on Principles of Distributed Computing (PODC'01)*, pages 264–273, August 2001.

[20] D. Liben Nowell, H. Balakrishnan, and D. Karger. Analysis of the Evolution of Peer-to-Peer Systems. In *ACM Symposium on Principles of Distributed Computing (PODC'02)*, pages 233–242, July 2002.

[21] N. Lynch. Distributed algorithms. *Morgan Kaufmann*, 1996.

[22] M. Macedonia. *A Network Software Architecture for Large Scale Virtual Environments*. PhD thesis, Naval Postgraduate School, June 1995.

[23] N. Malpani, J. Welch, and N. H. Vaidya. Leader election algorithms for mobile ad hoc networks. *Proc. 4th Int. Workshop on Discrete Algo. and Methods for Mobile Computing and Com. (DialM'00)*, pages 96–103, 2000.

[24] V. Park and M. Corson. A highly adaptive ditributed routing algorithm for mobile wireless networks. *IEEE INFOCOM*, pages 7–11, 1997.

[25] R. Prakash and R. Baldoni. Achitecture for group communication in mobile systems. *Proc. 8th Symposium on Reliable Distributed Systems (SRDS'89)*, pages 235–242, 1998.

[26] J. Purbrick and C. Greenhalgh. Extending Locales: Awareness Management in MASSIVE-3. *IEEE Virtual Reality 2000*, March 2000.

[27] S. Ratnasamy, M. Handley, P. Francis, and R. Karp. A Scalable Content-Addressable Network. In *ACM SIG/COMM*, pages 161–172, 2001.

[28] S. Ratnasamy, M. Handley, R. Karp, and S. Shenker. Application level muticast using content addressable networks. *Proc. of the Third International Workshop on Networked Group Communication*, pages 14–29, 2001.

[29] A. Rowstrom and P. Druschel. Pastry: Scalable distributed object location and routing for large-scale peer-to-peer systems. In *ACM International Conference on Distributed Systems Platforms (Middleware'01)*, 2001.

[30] I. Stoica, R. Morris, D. Karger, K. M., and B. H. Chord: A scalable peer-to-peer lookup service for internet applications. *ACM SIG/COMM*, pages 149–160, 2001.

[31] I. Stojmenovic. Position-based Routing in Ad-hoc Networks. *IEEE Communications Magazine*, 40(7):128–134, 2002.

[32] J. Walter, G. Cao, and M. Mohanty. A k-mutual exclusion algorithm for ad-hoc mobile networks. *Proc. 1st Int. Workshop of Principles of Mobile Computing (POMC'01)*, 2001.

[33] J. Walter, J. Welch, and N. Vaidya. A mutual exclusion algorithm for ad-hoc mobile networks. *ACM and Baltzer Wireless Networks journal, special issues on DialM*, 2001.

[34] H. Yang and B. Garcia Molina. Designing a Super-Peer Networks. Technical report, Stanford University, 2002.