

Proposition of a Hierarchical Service Distribution Architecture for Ad Hoc Networks based on the Weighted Clustering Algorithm

Jorge Nuevo
nuevo@inrs-telecom.quebec.ca

Jean-Charles Grégoire
gregoire@inrs-telecom.quebec.ca

Groupe Logiciels de télécommunications
Institut National de la Recherche Scientifique (INRS)
Université du Québec

Abstract

The highly dynamic conditions of ad hoc networks present several challenges to applications where one of the core functionalities is service discovery. However, existing solutions for service distribution do not work well in these dynamic environments, while other proposals focus on backbone structures at the network layer assuming multicast capabilities which is not always possible. Therefore, we propose a hierarchical architecture based on a very efficient clustering algorithm: the Weighted Clustering Algorithm (WCA). This algorithm has many advantages because not only is invoked on demand by the nodes with higher mobility, but takes into consideration other features such as the ideal number of nodes that a cluster can handle, mobility and battery power. We present and compare our proposed architecture with other strategies such as flooding and an active clustering algorithm.

Keywords: Ad Hoc networks, Hierarchical Architecture, Clustering, Service Distribution, Service Discovery.

1 Introduction

Ad hoc networking is the subject of extensive research, due mainly to the fact that these networks prove to be economical. In emergency services such as disaster recovery, ad hoc networks are the only possible option, because they form a temporary network without any aid from centralized administration. Nodes participating in an ad hoc network communicate on a peer-to-peer basis. A critical functionality in these networks is service distribution. In a service distribution architecture, clients use the discovery process to discover services in the

network, while servers use the advertisement process to periodically announce their registered services. The function of a service discovery protocol is to maintain a client node informed about service information available through scoped attribute queries. We understand by services the entities that perform computation or actions on behalf of client users. Services can be those offered by devices or software resources. Devices include printers, fax machines, cameras, audio-systems, etc. Software resources include games, music, personal information, information-oriented services (maps or weather), commerce based services (such as buying tickets to a movie), and network services (such as 3G Access Points, Network addressing mechanisms, etc).

Service discovery by flooding the network with service requests, as it is used by some peer-to-peer ad hoc networks is not the best option in many cases, since the message overhead generated is not acceptable. Instead, we propose to organize the service distribution architecture over the network layer into a hierarchical structure by using clustering. Section 2 summarizes previous work done within this subject. Section 3 presents possible procedures for disseminating service information in ad hoc networks. Features, advantages and disadvantages are presented for each strategy. In section 4 we review some clustering algorithms proposed in literature, while subsection 4.3 describes the weighted clustering algorithm which will be the base of our architecture. Section 5 describes our proposed hierarchical architecture for service distribution. Section 6 explains our development of the software extensions for the GloMoSim simulation tool and presents some results. Finally section 7 presents some conclusions and directions of future work.

2 Related Work

Existing service discovery architectures such as *Universal Plug and Play Device Architecture (UPnP)*, the *Jini Specification* and the *Service Location Protocol (SLP)* [1] do not present appropriate characteristics for mobile ad hoc networks. Current architectures do not use any performance metrics, and maintain a central directory server containing the necessary information to identify services. Other better tailored solutions have been proposed based on flooding [2] or by dynamically building and maintaining a backbone structure to perform advertisement and discovery [3]. Nevertheless, they suppose that all nodes will have multicast capabilities or will support the backbone formation algorithms, which it is not always true. Yet other proposals focus on the mechanisms for describing and matching services, as well as the user interfaces for discovery applications (such as [4] and [5]). Several routing protocols use clustering at the network layer such as the Cluster Based Routing Protocol (CBRP) [6] and the Hierarchical State Routing (HSR) [7]. But not many clustering algorithms have been proposed for service distribution in ad hoc networks in the best of our knowledge.

3 Dissemination Strategies for Service Information

Dissemination of service information constitutes an important issue of service distribution, because it influences enormously the performance of discovering a service. *Flooding* is a simple mechanism by which every incoming packet at a node is sent out on every outgoing link except the one it arrived on. Nevertheless, it is not possible to exclude physically the incoming links in wireless environments. Instead, each node rebroadcasts a message to its neighbors upon receiving it for the first time. This is achieved with control mechanisms that avoid that messages circulate in the network without end. One control mechanism consists in tracking the flooding messages based on a unique source identifier and a sequence number. However, flooding is far from optimal and generates a high number of redundant messages that may cause a serious problem referred as the *broadcast storm problem* [8]. Furthermore, flooding overhead corresponding to replicated, redundant messages increases with connectivity. Other strategies aim to reduce broadcast redundancy to save scarce resources by selecting a number of forward nodes in

order to define a connected set. This set is referred to as the backbone, and will carry out a broadcast or flooding process. In this approach, a structure (a *tree* or a *mesh*) is built, such that any other node in the network is adjacent to a node in the tree or the mesh structure. Backbone structures are used to disseminate changes in link state information and send probes for route discovery in multicast protocols. Backbones have also been proposed for service distribution, for example [3] considers the use of protocols at the network layer such as the Virtual Dynamic Backbone Protocol [9], while [10] proposes the use of the On-Demand Multicast Routing Protocol (ODMRP) for service discovery within ad hoc networked appliances. Although it is true that multicast is very important in mobile environments where bandwidth is scarce and hosts have limited power (like ad hoc networks), it is also true that multicast capabilities are not always present while significant work in the area has to be done [11]. Another approach consists in building hierarchical structures based on clustering. Cluster-based structures promote more efficient use of resources in controlling large dynamic networks as referred by [12], which shows that 75% or more of transmissions can be saved with clustering. A cluster may have one or more controllers (cluster-heads) acting on their behalf to make decisions for cluster members. A cluster may also define one or more *gateways* and *ordinary nodes* as shown at Fig[1]. The cluster-head schedules transmissions and allocates resources within the clusters, while gateways connect adjacent clusters. The argument made against clustering is that the rearrangement of the clusters and the assignment of the nodes to clusters may require excessive processing and communications overhead. Therefore, an algorithm that minimizes overhead is essential to maintain the benefits that clustering schemes provide.

4 Clustering Algorithms

Formation and maintenance of a cluster-based control structure requires algorithms to initiate clustering nodes and for adjusting clusters as nodes join, leave or move within the network. Choosing cluster-heads optimally is an NP-hard problem [13], yet several algorithms exist for building the initial clustering hierarchy and for restructuring the hierarchy as network connectivity changes. Some are the following [14].

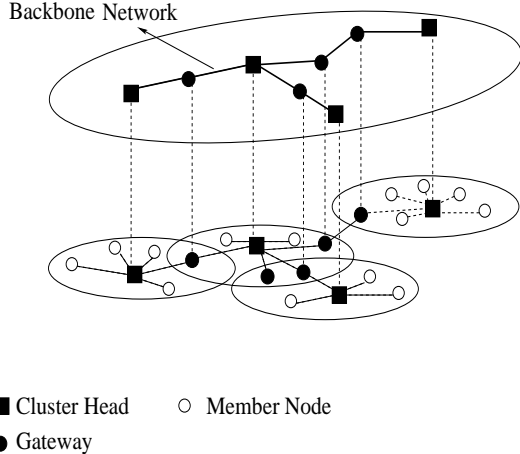


Figure 1: Cluster-Based Control Structure

4.1 Highest Degree.

Also known as *connectivity-based clustering*, was originally proposed by Gerla [15] where each node broadcasts its ID to the nodes that are within its transmission range. The node with maximum number of neighbors (maximum degree) is chosen as a cluster-head and any tie is broken by the unique node IDs. The neighbors of a cluster-head become members of that cluster and can no longer participate in the election process. Since no cluster-heads are directly linked, only one cluster-head is allowed per cluster. Any two nodes in a cluster are at most two-hops away since the cluster-head is directly linked to each of its neighbors in the cluster.

4.2 Lowest ID.

Also known as *identifier-based clustering* or *Linked-Cluster Algorithm (LCA)* was originally proposed by Baker and Ephremides [16]. This heuristic assigns a unique ID to each node and chooses the node with the minimum ID as a cluster-head. The clusters are formed around cluster-heads, and are identified by the cluster-head ID. Gateway nodes are also defined to establish connectivity among clusters. To enable the cluster formation and maintenance, all nodes keep the information about their neighbors in a “neighbor table”. Each node periodically broadcasts Hello packets which contain the Node-ID and the Node’s cluster-head ID. The Hello packets are only delivered to the immediate neighbors and are not passed any further. When a node receives a Hello packet, it updates its neighboring table. If the received packet does not exist in the table, a new entry is added. When a cluster-head hears a Hello

packet from another cluster-head, it compares its ID with the one of the heard cluster-head. The node with higher ID will then resign as a cluster-head, and will advertise its new status on the network. If an ordinary node does not hear Hello packets from any cluster-head in a period of time, then it becomes cluster-head. To ensure quick formation of clusters, this period of time between Hello broadcasts should be chosen rather small.

4.3 Weighted Clustering Algorithm

The problem with the schemes reviewed on the previous section is that they are “active”. They require a constant refresh rate of cluster-dependent information, and therefore introduce significant control overhead, even in the case that no data is being sent. Passive clustering has been investigated in the past to reduce computation and communication costs by invoking the clustering algorithm *on-demand*. The *weighted clustering algorithm (WCA)* [17] is a recent proposal of on-demand clustering that can be used for service distribution. Unlike other algorithms that are invoked periodically resulting in high communication overhead, WCA is *adaptively* invoked based on the mobility of the nodes. Even more, this algorithm takes into consideration the ideal number of nodes that a cluster can handle, the mobility (speed of nodes) and the battery power, in order to assign weights that will be used to determine the cluster-head nodes. The cluster-head election procedure is invoked at the time of system activation, and also when the current dominant set is unable to cover all the nodes. Every invocation of the election algorithm *does not* necessarily mean that all the cluster-heads in the previous dominant set are replaced with the new ones. If a node detaches itself from its current cluster-head and attaches to another cluster-head, then the involved cluster-heads update their member list instead of invoking the election algorithm. The algorithm for the weighted clustering algorithm as specified by [17] is the following:

1. Find neighbors of each node v which defines its *degree* (d_v) as

$$d_v = |N(v)| = \sum_{v' \in V, v' \neq v} \{dist(v, v') < tx_{range}\}$$

2. Compute the *degree-difference* $\Delta_v = |d_v - \delta|$ for every node v , where δ is the number of nodes (pre-defined threshold) that a cluster-head can handle ideally.

3. For every node, compute the sum of the distances D_v with all its neighbors

$$D_v = \sum_{v' \in N(v)} \{dist(v, v')\}$$

4. Compute the running average of the speed for every node until current time T . This gives a measure of mobility M_v

$$M_v = \frac{1}{T} \sum_{t=1}^T \sqrt{(X_t - X_{t-1})^2 + (Y_t - Y_{t-1})^2}$$

5. Compute the cumulative time P_v during which a node v acts as a cluster-head. P_v indicates how much battery power has been consumed, which is assumed more for a cluster-head than an ordinary node.
6. Calculate the combined Weight (W_v) for each node v where
$$W_v = w1 * \Delta_v + w2 * D_v + w3 * M_v + w4 * P_v$$
7. Choose that node with the smallest W_v as the Head-Cluster. All neighbors of the chosen cluster-head are no longer allow to participate in the election procedure.
8. Repeat steps 2 to 7 for the remaining nodes not yet selected as a cluster-head, or assigned to a cluster.

All nodes continuously monitor their signal strength as received from the cluster-head. When the mutual separation between the node and its cluster-head increases, the signal strength decreases. In that case, the mobile has to notify its current cluster-head that it is no longer able to attach itself to that. The node tries then to hand-over to a neighboring cluster. If the node goes into a region not covered by any cluster-head, then the WCA election procedure is invoked and the new dominant set is obtained.

5 Proposed Architecture

We have decided to locate the service distribution mechanisms on top of the network layer, in order to take advantage of either the service discovery mechanisms or the topological structures defined at that layer. For example, if the routing protocol used is multicast, then the backbone already defined at

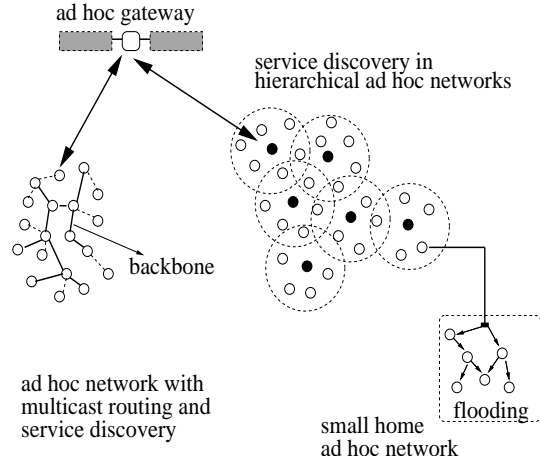


Figure 2: Heterogeneity of service distribution mechanisms

the network layer, can be re-used for service distribution as specified at [3]. If a hierarchical routing protocol is being use such as Cluster-Based Routing Protocol (CBRP) or Hierarchical State Routing (HSR), then the topological structure defined at the network layer can also be re-used for service distribution because cluster-nodes and gateways are already defined. Only when no mechanism is defined for service distribution at the network layer, then a hierarchical structure is defined based on the WCA algorithm. This approach provides service distribution in networks where different discovery techniques are used. Assuming that the appropriate ad hoc gateways exist to relate those clusters, then our architecture allows heterogeneity of service distribution and routing protocols as shown at figure 2. The gateways that connect the networks may be airborne relay nodes, low earth orbit/geostationary satellites [18], or traditional fixed-based infrastructure [18]. This approach provides interoperability, reduced overhead and aggregation.

5.1 Service Distribution.

In this section we focus on the hierarchical architecture based on WCA-clustering. When a client node needs a service, it browses its cache memory, and if it is not found, it sends a service request to its cluster-head. The cluster-head verifies if the service is available in its group to reply to the client node. If the service does not exist in the cluster, then the cluster-head re-sends the service request to other available cluster-heads nodes until the service is found; in which case, the client node receives a reply-message from the cluster-head where the ser-

vice is available. When a service request is sent, a timer is triggered to limit the time the client node waits for a reply-message. If the timer expires and no reply was received, it can re-send the service request until a maximum number of times (the timer may be increased in subsequent requests). At this moment, if no reply-message is received, the client node assumes the requested service is not available. When a service becomes available, the server node advertises it to its neighbors by using a *distance-limited flooding mechanism* (using a $TTL=n$, where n represents the maximum number of hops that a packet can traverse). The server node periodically announces its services to its designated cluster-head, so it can keep track of the services in its group. If the cluster-head does not receive these updating messages in a period of time, it assumes that the service is no more available and erases it from its registry. Furthermore, the combined weights (W_v) for each node by the weighted clustering algorithm could be included in the service messages in order to indicate the stability of potential servers. A client would select the server with the smallest weight if more than two servers are found for a service request.

5.2 Service Model.

In this section, we present the different components that define our service model (shown at Figure 3). We define a Service Application Module (SAM) at the application layer which is responsible for service distribution. In ad hoc networks, each node may behave either as a service client or server, therefore we define a client manager (called *SAC*) and a server manager (called *SAS*) at the application layer:

- *SAC (Service Application Client)*. This module is responsible of sending service requests to the network, when a service is required. It is also responsible to keep a status of such requests.
- *SAS (Service Application Server)* This module registers services available by a node and announces them to the node’s neighbors.

These modules may present a friendly user front-end (maybe in Micro-Java or WML) that will capture the needs of users and present services found in ad hoc networks. Nevertheless, we have not developed such front-end and we suppose the existence of it. We propose the definition of a *topology manager module* that will create and maintain clusters with the WCA procedure (default), or will interface to the appropriate network managers if a mechanism

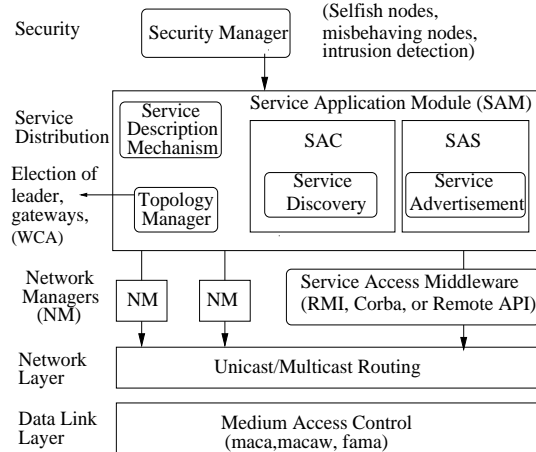


Figure 3: Service Model

is defined at the network layer. We also consider that service distribution mechanisms have to interface with a security module that protects from passive eavesdropping, active impersonation, message replay and message distortion. Intrusion detection proposals for ad hoc networks, such as [19], describe policies for neighboring nodes to collaborate and detect signs of intrusion. An intruder detected by the security module might be excluded from offering and demanding services in the service distribution module. We have not focused our research in security issues due to the extensive of the field, but we consider it must be taken into account in the architecture.

6 Simulations

The two main simulators for wireless ad hoc networks (NS-2 [20] and GloMoSim [21]) do not include any mechanisms for service distribution. Therefore, we have developed the necessary extensions for GloMoSim to support some strategies for service distribution. The GloMoSim extensions, collected in the form of a software patch, can be found at [22]. The patch implements the Service Application Module (SAM) and uses three different strategies for service distribution: *flooding*, *two-hop clustering with LCA* and *clustering with WCA*. We have performed different simulation tests to evaluate the performance of service discovery with WCA clustering. We began by comparing the service discovery strategies under different node densities. For this purpose, we defined a scenario with size of 800 x 800 m. The random waypoint mobility model (RWPM) [23] was used with a maximum speed of 3 m/s and a pause time between movements of zero. Simulation time

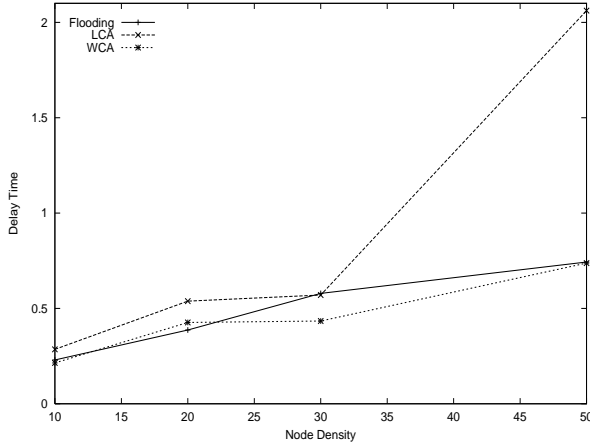


Figure 4: Delay Comparison

was defined as 50 seconds and nodes had a radio range of 250 meters. The propagation model used was the free space model; 802.11 DCF was used as the MAC layer protocol and AODV was selected as routing protocol. The service distribution scenario consisted of tens of nodes including five servers and five client nodes that demanded different services only after such services became available. Therefore, no re-transmissions of service requests due to unavailable services were considered. Simulations with different initialization seeds were repeated until the variance obtained was less than 10 % of the mean value for each test. The number of nodes considered for this example were 10, 20, 30 and 50 nodes. Figure 4 shows the service time response comparison for flooding, LCA and WCA strategy; while figure 5 shows service overhead between flooding and hierarchical (WCA) strategies. Figure 6 shows the overhead for the hierarchical approach using LCA and WCA as clustering algorithm.

We note that WCA has a better behavior and presents less overhead as the number of nodes increases, while service overhead for flooding constantly increases with the number of nodes. We also note that even LCA is a clustering solution, it presents the worst behavior as node density increases (around 30 nodes) in this scenario. This is due the pro-active nature of the algorithm and because a higher density implies a more frequent exchange of control messages to preserve the hierarchical structure. On the other hand, on-demand clustering solutions such as WCA present better results in response time and overhead comparison.

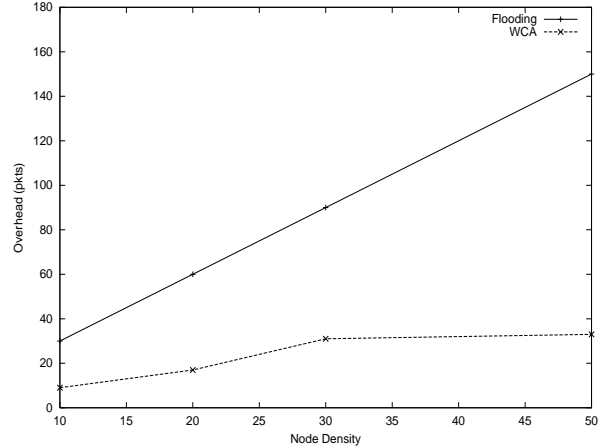


Figure 5: Overhead Comparison

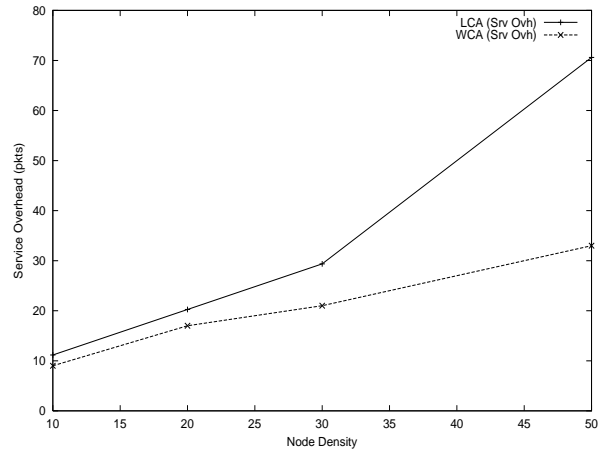


Figure 6: Clustering overhead

7 Conclusions

In this paper we have proposed a hierarchical architecture for service distribution in ad hoc networks that use an efficient on-demand algorithm for clustering. The architecture allows interoperability with mechanisms of the network layer such as different routing algorithms. We have also developed the necessary software extensions for the GloMoSim simulation tool, in order to evaluate the proposed architecture. Our results indicate that important advantages can be obtained by using the weighted clustering algorithm (WCA) in a hierarchical architecture for service distribution. Nevertheless, further research has to be done in order to optimize and validate our model. We continue to evaluate scenarios with different mobility models and traffic patterns in order to determine the optimal configuration para-

meters of our service distribution architecture. Simulations with different scenario conditions (area size, node density, mobility, etc.) have to be performed in order to determine the situations where clustering is better than other strategies. Nevertheless, we are convinced that hierarchical architectures are essential for service distribution, in order to provide aggregation and scalability in ad hoc networks.

References

- [1] Christopher Dabrowski and Kevin Mills. Analyzing Properties and Behavior of Service Discovery Protocols using an Architecture-based Approach. In *Proceedings of Working Conference on Complex and Dynamic Systems Architecture*, December 2001.
- [2] Yaron Y. Goland et al. Simple Service Discovery Protocol, 1999. IETF Internet Draft (work in progress).
- [3] U.C. Kozat and L. Tassiulas. Network Layer Support for Service Discovery in Mobile Ad Hoc Networks. In *Proceedings of IEEE INFOCOM*, 2003.
- [4] Sumi Helal, Nitin Desai and Varum Verma. Konark - A Service Discovery and Delivery Protocol for Ad-hoc Networks. In *Proceedings of the Third IEEE Conference on Wireless Communication Networks (WCNC)*, New Orleans - United States, March 2003.
- [5] Todd D. Hodes et al. Composable ad-hoc Mobile Services for Universal Interaction. In *Mobile Computing and Networking*, pages 1–12, 1997.
- [6] Jinyang Li Mingliang Jiang and Y.C. Tay. Cluster Based Routing Protocol, 1999. IETF Internet Draft (work in progress).
- [7] A. Iwata et al. Scalable Routing Strategies for Ad Hoc Wireless Networks. In *IEEE Journal on Selected Areas in Communications*, August 1999.
- [8] Sze-Yao Ni et al. The broadcast storm problem in a mobile ad hoc network. In *Proceedings of the Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom-99)*, pages 151–162, N.Y., August 15–20 1999. ACM Press.
- [9] U.C. Kozat, G. Kondylis, B. Ryu and M.K. Marina. Virtual Dynamic Backbone for Mobile Ad Hoc Networks. In *Proceedings of IEEE International Conference on Communications (ICC)*, June 2001.
- [10] Liang Cheng and Ivan Marsic. Service Discovery and Invocation for Mobile Ad Hoc Networked Appliances. In *Proceedings of the 2nd International Workshop on Networked Appliances (IWNA0)*, New Brunswick, N.J - United States, November 2000.
- [11] Katia Obraczka and Gene Tsudik. Multicast Routing Issues in Ad Hoc Networks. In *Proceedings of the International Conference on Universal Personal Communication (ICUPC)*, Florence, Italy, October 1998.
- [12] Taek Jin Kwon and Mario Gerla. Efficient Flooding with Passive Clustering (PC) in Ad Hoc Networks. *ACM Computer Communication Review*, 32(1), January 2002.
- [13] I. Chlamtac S. Basagni and A. Farago. A Generalized Clustering Algorithm for Peer-to-Peer Networks. In *Proceedings of Workshop on Algorithmic Aspects of Communication, Satellite Workshop of ICALP*, Bologna Italy, July 11–12 1997.
- [14] Charles E. Perkins. *AD HOC Networking*. Addison - Wesley, 2001.
- [15] M. Gerla and J. Tsai. Multiclusterc, Mobile, Multimedia Radio Networks. *Wireless Networks*, 1(3):255–265, 1995.
- [16] D. J. Baker and A. Ephremides. A distributed algorithm for organizing mobile radio telecommunication networks. In *2nd International Conference on Distributed Computing Systems*, pages 476–483, Paris, France, April 1981. IEEE.
- [17] Mainak Chatterjee, Sajal K. Das, and Damla Turgut. A Weighted Clustering Algorithm for Mobile Ad hoc Networks. In *Proceedings of the 7th International Conference on High Performance Computing (HiPC)*, pages 511–521, December 2000.
- [18] M. Ahmed, S. Dao and R. Katz. Positioning Range Extension Gateways in Mobile Ad Hoc Wireless Networks to Improve Connectivity and Throughput. In *Proceedings of the IEEE Military Communications Conference (MILCOM)*, Washington D.C. - United States, October 2001.
- [19] Yongguang Zhang and Wenke Lee. Intrusion detection for wireless ad hoc networks. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MOBICOM-00)*, pages 275–283, N. Y., August 6–11 2000. ACM Press.
- [20] Kevin Fall and editors Kannan Varadhan. “NS notes and documentation”. The VINT Project, UC Berkeley, LBL, USC/ISI and Xerox PARC, November 1997. Available at <http://www.isi.edu/nsnam/ns>.
- [21] *GloMoSim: Global Mobile Information Systems Simulation Library*. <http://pcl.cs.ucla.edu/projects/glomosim/>.
- [22] Jorge Nuevo. Internet Homepage. <http://www.inrs-telecom.uquebec.ca/users/nuevo>.
- [23] Vanessa Davies. Evaluating Mobility Models within an Ad-Hoc Network. Master’s thesis, Colorado School of Mines, 2000.