

FRAD-HOC: A FRAMEWORK TO ROUTING AD-HOC NETWORKS

Underlea Corrêa¹, Carlos Montez¹, Vitório Mazzola², and M.A.R Dantas²

Federal University of Santa Catarina (UFSC)

Tecnological Center (CTC)

1 Pos-graduate Program Eletrical Egeineering (PPGEEL)

{underlea, montez}@das.ufsc.br

2 Departant of Informatic and Estatistic (INE)

{mazzola, mario}@inf.ufsc.br

Campus Universitário – Trindade 88040-900 – Florianópolis – SC –
Brasil

Abstract. *This article presents a routing framework for mobile ad-hoc networks, which was called as FRAd-hoc. The main goal of the contribution was the design and implementation of a structure that could gather generic characteristics from hybrid routing algorithm domains. Therefore, it is possible to offer a specializing framework to produce and make available reusable software components. The results present in this research work indicate that the FRAd-hoc environment has reached a successful level, because it was possible to produce others algorithms starting from the proposed framework.*

1 Introduction

The growing interesting in mobile ad-hoc networks (MANETs), has lead the proposal of many routing algorithms. In the literature [1-6] it is possible to verify that many proposals are oriented to some specific target. It is possible to image that these proposals are not suitable to every MANET. As an example, in [4] it is presented the Distributed Dynamic Routing (DDR) algorithm for mobile ad hoc networks, which is efficient for networks with low traffic density. On the other hand, the Zone-based Hierarchical Link State (ZHLS) [3] is high adaptable for dynamic topology and reduce the communication overheads when compared to pure reactive protocols, exemples are [7, 8]. Therefore, all nodes must have a pre-programmed static zone map. This is not feasible in applications where the geographical

boundary of the network is dynamic [9]. The Zone Routing Protocol [10], is an algorithm that it has significantly amount of communication overhead reduction when compared to pure pro-active protocols. It also has reduced the delays associated with pure reactive protocols as DSR [8]. However, it executes perfectly in intermediate networks, because for high density routing zones the protocol can behave in the same fashion as a pure pro-active protocol. On the other hand, for low density zones it behaves as a reactive protocol.

Nevertheless, different network conditions need different routing services [11]. Current routing modes do not allow this feature. It is possible to guess the high level of difficult to change a routing service in a large scale MANET. Characteristics as connectivity, amount of nodes and mobility are dynamic factors. This aspect illustrates the necessity of new approach to consider different network conditions. In this configuration it is expected that many services are static pre-configured in each node. Therefore, in this paper it proposed the development of a framework that could gather generic characteristic from hybrid routing algorithms domains. As a result, new routing algorithms could be developed from the proposed framework.

The paper is structured as it follows: in section 2, related works are presented. In section 3, the proposal for the MANETs routing framework is described in detail. We present, in section 4, the implementation and specialization of the ad hoc routing framework. Finally, in section 5, the conclusions and expectations for future work are exposed.

2. Related Works

Since its formation, the mobile ad-hoc networks group goals, was to develop the peer-to-peer routing ability in a purely mobile wireless domain. From that, a opportunity was opened for various research groups, interested in the development of researches that approached security, energy management and interaction with adjacent layers protocols. Since then, tens of works have been developed aiming at contributing with the research evolution in the ad hoc mobile networks domain. Some works considered relevant for the development of our research are cited below.

The proposed research by He Yu et al [11], present a programmable routing framework that promotes the adaptative in routing services for sensor networks, including a universal routing service allow the introduction of different services through its tunable parameters and programmable componets.

The work in [9] classifies a series of protocols, providing an overview of the great scale of the routing algorithms proposed in the literature. As a major contribution, we believe this work presents a comparison of all the routing algorithms performances approached by it, indicating which of the protocols is capable of better running in large scale networks.

The research introduced by [12] deals with routing algorithms that incorporate the use of mobile agents for the MANET routing. Through a clustering architecture, mobile agents are used to collect and maintain the intra and inter-clustering routing information. This work is very similar to researches that separates a network into zones. However, its differential is adopting mobile agents for it.

3. FRAd-hoc Model

The dynamic topology nature of MANETs makes the multi-hop routing difficult [2]. Due to this factor, various research works, such as [1, 3, 5, 13] have been developed to offer, among others, a routing algorithm that defines the network topology, fulfilling the best qualitative and quantitative features demanded. Thus, in [14] we verified that the algorithms aim at proposing different solutions using similar techniques, showing advantages and disadvantages according to specific network situations. Thus, willing to offer a solution, we analyzed the possibility of offering a framework that not only aggregates two or more routing algorithms, but also determines the protocol to be used in according of the network's profile. For that, we introduce the proposal for a structure called FRAd-hoc (Ad-hoc Routing Framework), as shown in Figure 1.

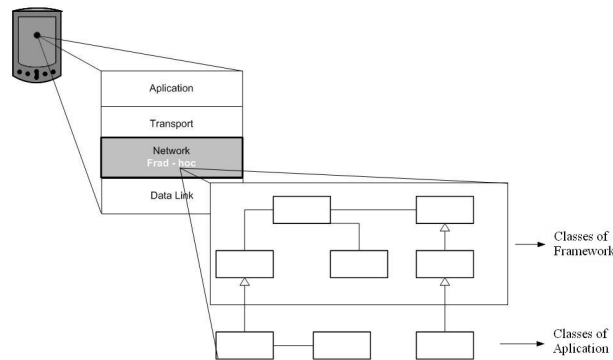


Fig. 1. Ad-hoc Framework Routing

The main goal was the design and implementation of a structure that could gather generic characteristics from hybrid routing algorithm domains. Therefore, it is possible to offer a specializing framework to produce and make available reusable software components.

The development approach for the proposed framework used the example-driven design methodology [15-18], which is threefold. The first stage approaches the domain analysis, where the existing applications are the main information source. The second stage approaches the hierarchy definition of classes that generalize the investigated domain, and the third stage, called framework test, uses it to develop examples of applications that approach the studied domain.

3.1 Domain Analysis

Routing algorithms for MANETs use the routing methodologies quoted above, may be classified, according to [9], in three main characteristics: proactive, reactive and hybrid. In the first case, there are the routing algorithms characterized for trying to continuously evaluate the network keeping updated knowledge of all the routes, for when a package needs to be forwarded, the route is already known. In the second case, there are routing algorithms denominated for establishing routes to be used on

demand, that is, only when the route to a destination is required these algorithms initiate the route finding process. In the third and last case, there are algorithms that are proactive and reactive by nature. Designed to increase scalability, allowing the close nodes to work together for forming a kind of backbone to reduce route finding overhead, they are called by [9] the new generation of MANET routing algorithms.

According to the analysis made in [14] we defined, generically, that the domain involves two types of routing protocols: the intra-zone and inter-zone levels routing algorithms and the algorithms that use mobile agents. In the first case, the routing made in the intra-zone relies on a pro-active mechanism, while the one run in the inter-zone relies on a reactive mechanism. In the second case, the routing is initially trusted to mobile agents responsible for both finding and keeping updated the node routing tables, being the nodes able to being finding a route to a certain destination when necessary. Based on this definition, we determine the FRAd-hoc has, as its initial responsibility, to compose and to provide a structure that will work as a base to aggregate the common features to the routing algorithms, providing the support necessary for the development of other routing protocols.

3.2 Class hierarchy definition

FRAd-hoc (Figure 2) is composed by five classes that offer basic mechanisms for developing hybrid approach routing algorithms. Each class in this structure performs a sequence of methods responsible for the correct routing communication and running. Among them, we have: Node Class, responsible for the mechanisms of communication through message exchange between the nodes, made by the *sendMsg()* method. This Node class also has methods that indicate a possible routing failure, in case a route is not valid at the time of a data transmission, for instance. Besides, the framework has methods responsible for updating the route table and for establishing the finding of a path to a certain destination, besides methods that give back an answer from a valid route when asked by a destination node, as it is for the *repRoute()*, *updateTabRoute()*, *updateIntraZT()* and *pathDiscovery()* methods. The *IntraZT* and *InterZT* classes are responsible for aggregating methods that run zone construction and updating functions, as it is for the *buildIntraZT()*, *buildInterZT()* and *updateIntraZT()* methods.

The *Agent* and *TabRoute* classes contain the methods to be used by algorithms that apply the mobile agent paradigm. Both aggregate methods that require and answer the path and maintenance and path update, as established in Figure 2.

The framework dynamics may begin with the Node class, through forwarding messages that allow one node to know other nodes, which share the same frequency channel, calling them its direct neighbors. In case of algorithms that divide the network into zones (see Figure 3), the information received by the neighbor nodes, in a general way, are stored in a table called *intraZT* after the running of the zone building method, *buildIntraZT()*, which can run the inter-zone building method called *buildInterZT()*, in case a gateway node is detected. When a node knows its neighbors, it can run the route requisition method, *reqRoute()*, initially consulting its intra-zone table. In case the route required by the node is not known by its *IntraZT* table, it can run a route requisition method for its *InterZT* table. If *InterZT* cannot

obtain information from the node to which one wants to establish a communication, the node may initiate a new message sending process, to check if there is any change in the network.

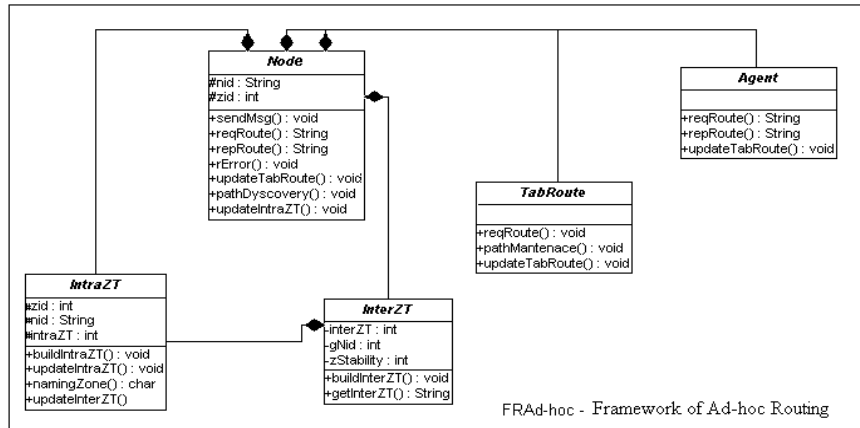


Fig. 2. Class diagram’s ad-hoc routing framework

In case of algorithms that use mobile agents, Figure 4 shows that a node, knowing its neighbors, may receive constant visits from mobile agents, which compare their routing table with the visited nodes’ routing tables, thus offering the update of the valid routes to a destination. However, if the node needs a route that is not defined in its routing table, it may send a route requisition to its neighbor routes, being able to abort this operation in case a mobile agent updates it before a valid answer is received or if it offers a shorter path route.

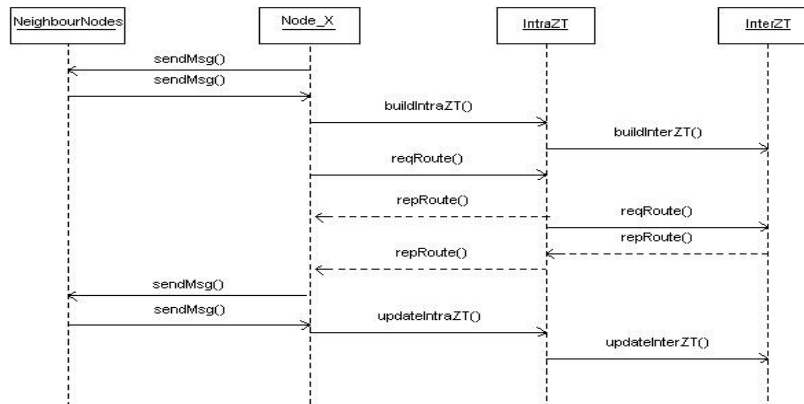


Fig. 3. Sequential FRAd-hoc diagram of algorithms that divide the network in levels of zones

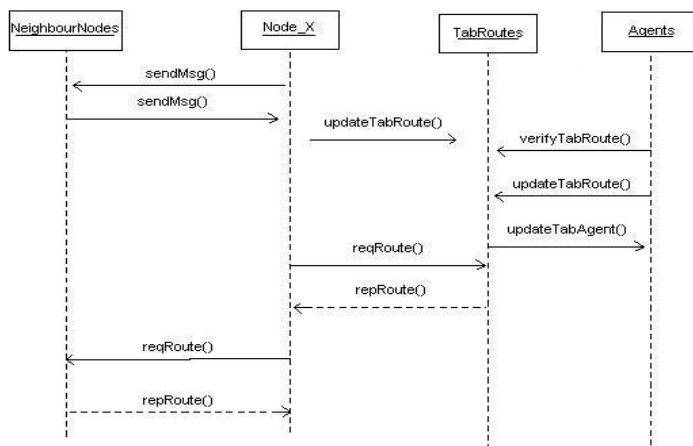


Fig. 4. FRAd-hoc sequence diagram executing algorithms that use mobile agents

4. Experimental Results

In this stage we show the usability of the proposed framework, verifying through the specializing routing algorithms, if it offers the functionalities planned. As an example of developed applications, we implemented the DDR [4], and HARP [5] algorithms among the algorithms studied in [14], because it presents features that may be applied to the other test algorithms.

Figure 5 presents the FRAd-hoc class structure approached in the previous section and the class structure of the application developed under it. One may immediately see the specializing of four concrete classes created by the user to obtain the functionalities demanded by the algorithm, in which three of them are inherited from FRAd-hoc, showing, as we wished, a clear evidence of reuse.

The activity sequence of the methods (see Figure 6) implemented by the DDR algorithm classes [4] begins with the *NodeDdr* class, where it starts running a message exchange method with its neighbor nodes, called *sendMsg()*, responsible for the communication among the nodes. When a node knows its neighbors, it is then able to run a series of methods, beginning with the method responsible for determining the choice of the favorite node, called *determinePN()* (see Figure 7), according to. Then, the *createBeacon()* method is run. This method is responsible for generating a message to be forwarded to the neighbor nodes, containing the zone identification information, node identification, node degree and favorite node (the node with the most neighbors).

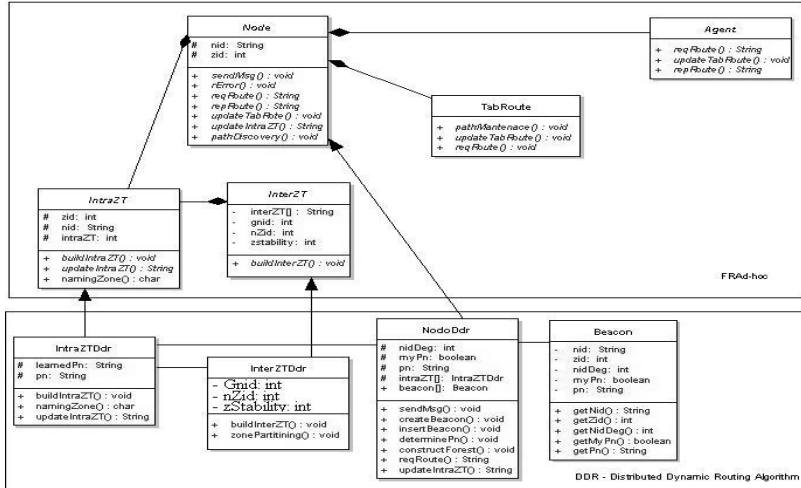


Fig. 5. DDR’s implementation algorithm under the FRAd-hoc

The information forwarded by the neighbor nodes through beacon messages are stored in the intra-zone table in each node through the *insertBeacons()* method (see Figure 8). Having this information about its neighbors, a node may then continue building its intra-zone through the *buildIntraZT()* method (see Figure 8). In case the node already has a valid intra-zone table, it will only run an update, adding or removing nodes that no longer belong to this intra-zone table. After building the intra-zone, the IntraZT class runs the method that generates the name of the zone through the *namingZone()* method and consequently builds the inter-zone through the remaining nodes in the intra-zone table, which are called gateway nodes, which can be moved to the intra-zone table whenever they can join an *x* node tree [7].

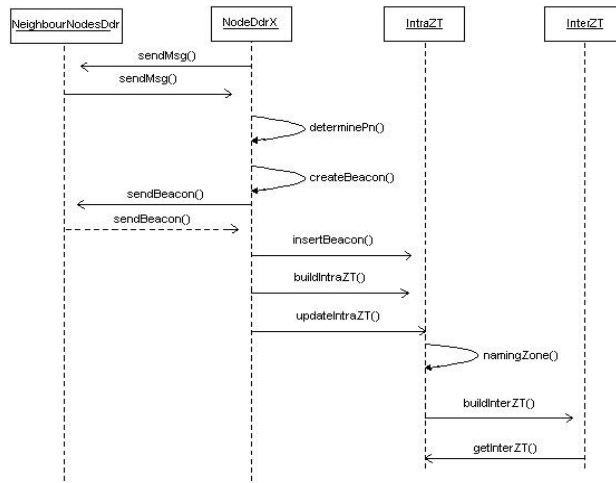


Fig.6. Sequential diagram algorithm DDR

Through Figure 7, we may observe more precisely the dynamics of the *determinePN()* method responsible for electing the favorite neighbor node. For this method three cases are defined: the first case verifies if the neighbor node set (pnX) of the *x* node equals zero, indicating it doesn't have any neighbor nodes, and consequently no favorite node.

However, the second case evaluates if the neighbors set (pnX) equals 1 (one); if true, then this will be defined as the *x* node's favorite node. Finally, if none of the above information is true, we have the case in which if the neighbor node set has more than one member, the node must elect the member with the larger identifier number (NID).

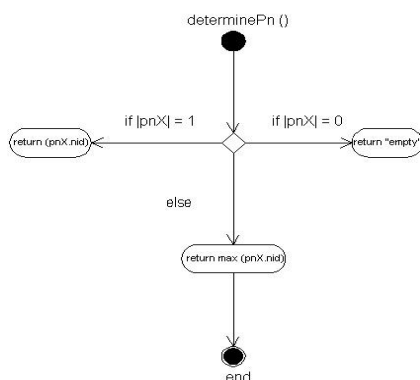
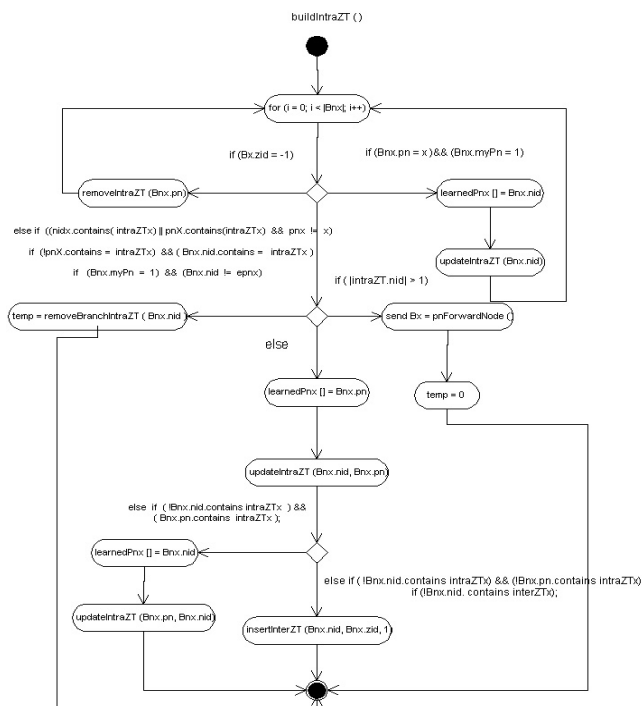


Fig. 7. Preferred neighboring node election method



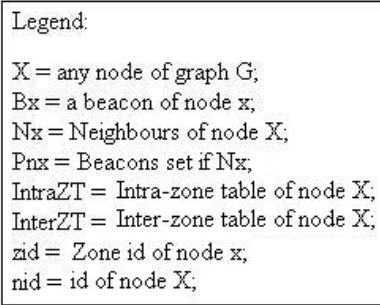


Fig. 8. Construction intra-zone method

Figure 8 represents in a generic way the activity run by the method that establishes the zone construction, as stated by [4]. For its construction, a node needs to know basically two levels: its NID (node id) of neighbors, and the the NID of the elected favorite node neighbors, defined by [4] as *learnedPn* of node.

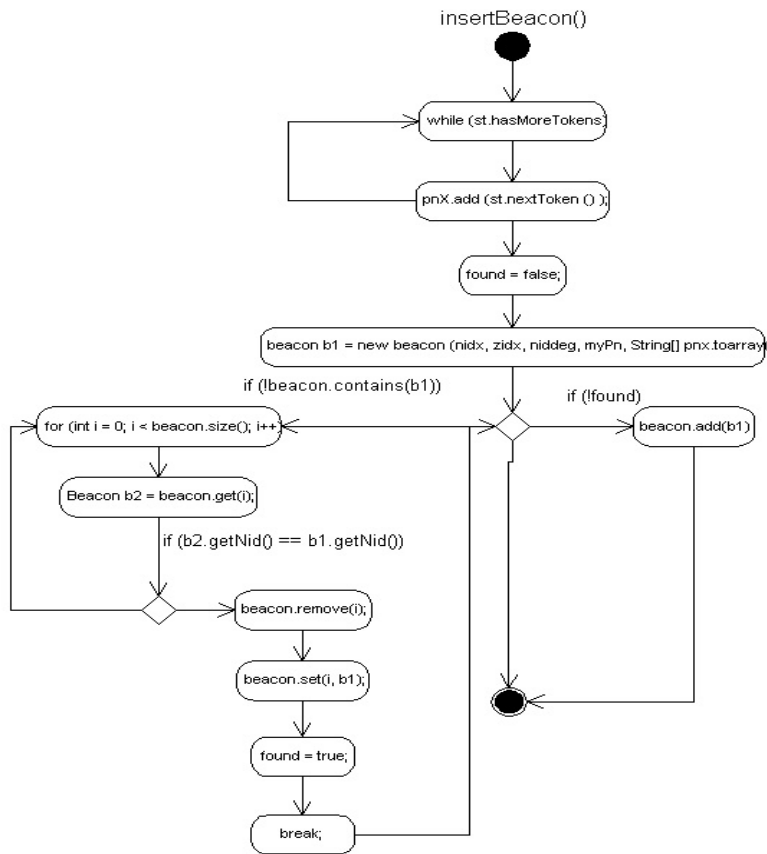


Fig. 9. Beacon insertion method

Figure 9 illustrates the *insertBeacon()* method, responsible for adding the information received through messages forwarded by the nearby nodes.

5. Conclusions and Future Works

In this paper we presented the design and implementation of a framework called FRAd-hoc. The goal of the framework was to create a structure that could offer support to the development of hybrid routing algorithms for mobile ad-hoc networks. The first step was a careful research related to MANET routing algorithms and analysis. In this phase, we observed the absence of a unique algorithm that aggregates every possible MANET state. Therefore, we presented in this work a new routing approach that uses the oriented-object frameworks mechanism. The next effort was the design and implementation of many classes and methods, which were important to attend the primary goals of the present research. Along this phase, we verified that results from the framework reached some efficiency, when compared to the native proposal. In this environment an appropriated routing algorithm was chosen more appropriated to the related network configuration.

Currently we are implementing other algorithms under the proposed framework structure. One of the future extensions is to build a management tool that will define which of the implemented routing protocols should run under the FRAd-hoc. We find this tool necessary to improve the management of the routing algorithms in the future.

References

1. S. Marwaha, C. K. Tham, and D. Srinivasan, A Novel Routing Protocol using Mobile Agents and Reactive Route Discovery for Ad Hoc Wireless Networks, Networks ICON, 10th IEEE International Conference, p. 311 – 316.
2. R. Onishi, S. Yamaguchi, H. Morino, and T. Saito, The Multi-agent System for Dynamic Network Routing, Autonomous Decentralized Systems, 5th IEEE International Symposium, p. 375-382.
3. M. Joa-Ng, and L. I-Tai, A peer-to-peer zone-based two-level link state routing for mobile ad hoc networks, IEEE Journal on selected areas in communications, vol 17, n° 8, p. 1415-1425.
4. N. Nikaein, H. Labiod, and C. Bonnet, DDR – Distributed dynamic routing algorithm for mobile ad hoc networks, Mobile and Ad Hoc IEEE Networking and Computing. MobiHOC 2000 First Annual Workshop, p19- 27.

5. N. Nikaein, and C. Bonnet, HARP- Hybrid Ad Hoc Routing Protocol, In: IST2001 – International Symposium on Telecommunications <http://www.eurecom.fr/~nikaeinn/harp.ps>.
6. R. Roychoudhury, S. K. P. Bandyopadhyay, A distributed mechanism for topology discovery in ad hoc wireless networks using mobile agents, Mobile and Ad Hoc Networking and Computing, MobiHOC IEEE 2000 First Annual Workshop, p. 145-146.
7. S. Das, C. Perkins, E. Royer, Ad hoc On Demand Distance Vector (AODV) routing, Internet Draft, draft-ietf-manet-AODV-11.txt, (2002).
8. D. Johnson, D. Maltz, J. Jetcheva, The dynamic source routing protocol for mobile ad hoc networks, Internet Draft, draft-ietf-manet-dsr-07.txt, (2002).
9. M. Abolhasan, T. Wysocki, Eryk Dutkiewicz, A review of routing protocols for mobile ad hoc networks, Elsevier Computer Science, www.elsevier.com/locate/adhoc, (2003).
10. Z.J. Hass, R. Pearlman, Zone routing protocol for ad-hoc networks, Internet Draft, draft-ietf-manet-zrp-02.txt, (1999).
11. Yu He, Cauligi S. Raghavendra, Steven Berson, Roberts Braden, A programmable Routing Framework for Autonomic Sensor Networks, Proceedings of the Autonomic Computing Workshop Fifth Annual International Workshop on Active Middleware Services (AMS'03), (2003).
12. K. Denko Mieso, The use of mobile agents for clustering in mobile ad-hoc networks, Proceedings of annual research conference of the South African institute of computer scientists and information technologists on Enablement through technology, ACM International Conference Proceeding Series. p. 241 – 247.
13. N. Minar, K. H. Kramer, and P. Maes, Cooperating Mobile Agents for Dynamic Networking, Software Agents for Future Communications Systems, Chapter 12.
14. U. C. Corrêa, Vítório B. Mazzola, M.A.R Dantas, Analise Comparativa de Protocolos de Roteamento de Redes Ad-Hoc, Anais da 2a. Escola Regional de Redes de Computadores - ERRC, 149-155 (2004).
15. Ralph E. Johnson, and Brian Foote, Designing Reusable Classes, Journal of Object-Oriented Programming, Volume 1, Number 2, pages 22-30.
16. S. Srinivasan, Design Patterns in Object-Oriented Frameworks, Computer, Vol.:32, Issue: 2, p.:24-32, Injournal, (1999).

17. Ralph E. Johnson, Frameworks – Components and Patterns, Communications of the ACM, vol. 40, Number 10.

18. D. Parson, A. Rashid, A. Speck, A. Telea, A Framework for Object-Oriented Frameworks design, Technology of Object-Oriented Languages an Systems, p.: 141-151, (1999).