Genetic Evolution and Adaptation of Advanced Protocols for Ad Hoc Network Hardware Systems

Jennifer Jackson[1], Mark Leeson[2]

Abstract. The diversity of future technologies requiring ad hoc networks to operate within unpredicted situations will mean an increase in the required flexibility of the actual protocols used for communicating information. A methodology is proposed to genetically evolve the optimum ad hoc network communication protocol under any given network scenario. The methodology creates and dynamically adapts the communication protocol based upon an alphabet of characteristics and performance metrics using simple protocol mapping techniques and minimisation of a fitness function via a genetic selection process. A scenario has been created to evaluate the performance of the methodology in finding the optimum solution. Preliminary results show that the methodology is able to find the global optimum within several runs. The methodology could be enhanced using Field Programmable Gate Array (FPGA) hardware nodes for real time performance and distributed control.

1 Introduction

A predicted explosion in the demand for mobile services will mean that ad hoc networks of the future must have the ability to interconnect diverse technologies such as wearable computers, and home robots, as well as accommodating environmental conditions that were not premeditated, such as malicious security attacks, failures within the network and sudden changes in topology.

An ad hoc network is characterised by a number of devices, often mobile, connected in an arbitrary manner to form a network without a central controller. Their development began in the 1970s with the appearance of static wireless networks, but they were increasingly adapted, particularly during the 1990s to enable wireless mobility [1]. Today, a number of wireless protocols are in commercial use, but despite this nearly forty year development there are still

[1] Complexity Science, University of Warwick, UK
[2] Engineering, University of Warwick, UK

challenges facing ad hoc communication protocol design. Current protocols are fixed for a given application, but ad hoc networks need to encompass a growing list of requirements that cannot be satisfied by a single fixed protocol. There is therefore a need for network adaptability to cope with the environment and application by choosing the optimum protocol for the given situation. This work exploits the powerful search capabilities of the genetic algorithm, together with simple mapping techniques to evolve optimum protocol designs for a given scenario.

The remainder of this paper is organised as follows: section 2 begins with some background regarding communication protocols, and highlights relevant work. Section 3 details the proposed methodology including the characteristic alphabet, the protocol mapping technique, and the genetic selection process. Section 4 describes the scenario used to test the methodology and section 5 presents the results. Section 6 gives the conclusions of the work followed by acknowledgements and references.

2 Background

2.1 Protocol stack

Communicating from one device to another in an ad hoc network involves a number of layers of interacting processes, from the physical medium such as radio waves to the user software such as a web page. These combined layers form the protocol stack, commonly analysed using the Open Systems Interconnection (OSI) model as shown in Fig.1.

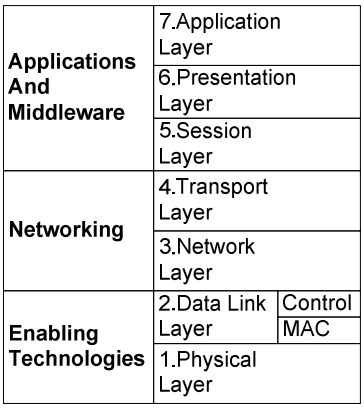

Fig. 1. OSI protocol stack model

Each of the seven layers can contain one or more different sub-protocols. There are many wireless protocol stacks, often only defined for the physical and data

link layers because it is these two layers that are mostly concerned with, and affected by, the transmission medium used. In ad hoc networks particular attention needs to be given to the network layer and how the data will be routed due to the constantly changing nature of the ad hoc topology which is not present in other types of networks. Above these layers, where the transmission medium used is of no concern to the application, it is advantageous to share a common language such as Transmission Control Protocol (TCP) when bridging across wired and wireless networks to access information from the internet.

2.2 Related Research

Related research focuses on automated protocol design. Ocenasek and Sveda [2] propose the use of genetic algorithms to develop security protocols. Xue et. al. [3] apply an artificial immune algorithm to make the design of security protocols more secure and reliable. Perrig and Song [4] use an automated technique for security protocol design involving minimising a cost function based upon a set of requirements. Virtanen et. al. [5] suggest the idea of a programmable processor capable of processing several different protocols. Oberg et. al. [6] use a grammar based specification method for hardware synthesis of data communication protocols. None of these ideas however create a protocol dynamically in real time. They are concerned with developing optimum protocols for a set of pre-generated criteria where the network environment is known. Pavlosoglou et. al. [7] however use Selfridge's Pandemonium concept to dynamically emerge an optimum routing protocol for the security of wireless ad hoc networks. Limitations with this method meant that global solutions were not always found. The methodology proposed within this paper improves upon this by using a genetic algorithm approach which is good at finding global solutions, and additionally focuses on multiple layers of the protocol stack to address the most important constituents of a wireless ad hoc protocol.

3 Protocol Methodology

3.1 General Concept

The general concept of the proposed methodology is the creation and adaptation of a communication protocol for a wireless ad hoc network, where the chosen protocol is based upon feedback of the current network performance. The decision making process has been made at a global level where there is a centralised

controller monitoring the network. This allows a *first step* in the investigation of the concept of dynamically creating a communication protocol.

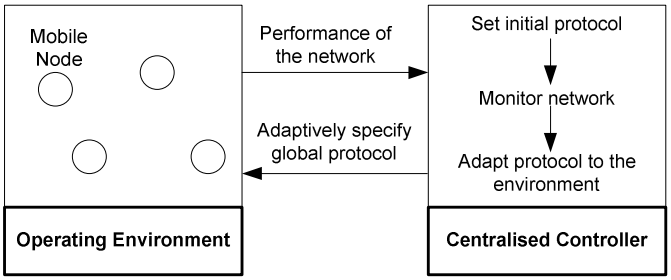

Fig. 2. General concept of the network operating environment

3.2 Alphabet of Characteristics

The functionality of each layer within the protocol stack can be defined by a set of characteristics through classification of all the sub-protocols within it. For example within the physical layer the sub-protocols could be classified according to their transmission frequency or the type of modulation schemes they use. Many such characteristics could be used to classify the sub-protocols, but there is a minimum number needed to uniquely distinguish one sub-protocol from another. This minimum set of characteristics is represented by an alphabet, where each letter of the alphabet represents one particular characteristic. To demonstrate the principle of the methodology three layers of the OSI model have been optimised: a) the physical layer, b) the Media Access Control (MAC) sub-layer, and c) the network layer routing, with the remaining layers fixed. The protocol generation algorithm is used to find the optimum set of characteristic values which map to the optimum protocol.

3.3 Physical Layer Characteristics

The classification of the physical layer sub-protocols available within the simulation tool can be simplified to two independent characteristics as given in Table 1, with each characteristic assigned an alphabet letter. These characteristics allow the solution space to be represented by a two-dimensional vector space as shown in Fig. 5a, where each available protocol for a defined set of internal parameters can be uniquely represented by a point within the vector space. The *range type* indicates the kind of network that the wireless protocol was designed for. At one end of the characteristic scale is the Personal Area Network (PAN)

designed for the interaction of nodes within close proximity around a person such as communication between a PC and a video camera. In the middle range is the Local Area Network (LAN) designed for interconnecting computers, printers and scanners within office buildings. At the other end of the scale is the Wide Area Network (WAN) designed for connecting devices on a larger scale such as connecting homes and cities to the World Wide Web. The *maximum bit rate* indicates how fast data can be transferred across the network and encompasses the frequency and modulation type of the protocol because at the low end of the characteristic scale low frequencies are used leading to lower bit rates. At the high end of the scale high frequencies are used often with modulation techniques for multiple channels resulting in high bit rates.

Table 1. Physical layer classification; those in italics were not used during simulations

Sub-protocol	A. Range Type	B. Max bit Rate
PHY IEEE802.11a [8]	LAN	High - 54 Mb/s
PHY IEEE802.11b [8]	LAN	Medium - 11Mb/s
PHY IEEE802.16 [9]	*WAN*	*High - 30Mb/s 75Mb/s*
PHY IEEE802.15.4 [10]	*PAN*	*Low - 250kb/s*

3.4 Media Access Control Layer Characteristics

The classification of the MAC layer protocols available within the simulation tool can be represented by three independent characteristics and is given in Table 2. The three characteristics allow representation by a three-dimensional vector space as shown in Fig. 5b. *Contention* is concerned with the ability of the protocol to avoid or resolve collisions when more than one node is attempting to access the channel at the same time. At one end of the characteristic scale are contention-free methods where certain assignments are used to avoid contentions altogether. Contention-based schemes on the other hand are aware of the risk of collisions and take steps to resolve them. Random access methods apply a random wait time if a collision occurs before re-trying, whereas collision resolution or avoidance methods tend to listen to the channel or make an announcement before sending data which subsequently reduces the probability of a collision. *Quality of Service* is a measure of the level of service that data receive when they transfer across the network. The network is expected to guarantee a set of measurable pre-specified service attributes such as end-to-end delay, available bandwidth, and probability of packet loss. At one end of the characteristic scale are “best effort” protocols that do not guarantee any kind of service quality, at the other end of the scale are protocols that do guarantee a service quality, and then there are some protocols in between that guarantee some specific attributes. *Number of Channels* indicates the number of channels the protocol uses to coordinate connection sessions between

sending and receiving nodes. At one end of the characteristic scale are single channel methods and at the other end are multiple channel methods. There are some protocols that can operate using single or multiple channels depending upon the mode.

Table 2. MAC layer classification; those in italic were not used during simulations

Sub-protocol		C. Contention	D. Quality of service	E. Number of channels
MAC IEEE802.11	[8]	Resolution	None	Multiple
MAC IEEE802.11e	[11]	Resolution	Yes	Multiple
MAC IEEE802.16	*[9]*	*Resolution*	*Yes*	*Multiple*
MAC IEEE802.15.4	*[10]*	*Resolution*	*None*	*Single/Multiple*
CSMA	[12]	Random Access	None	Single
MACA	[12]	Resolution	None	Single
TDMA	[12]	Contention Free	None	Multiple
ALOHA	[12]	Random Access	None	Multiple

3.5 Network Layer Routing Characteristics

The routing protocols available within the simulation environment allow their classification to be simplified to three independent characteristics, as detailed in Table 3. The orthogonality of the alphabet characteristics allow the solution space to be represented by a three-dimensional vector space as shown in Fig. 5c. *Route Computation* specifies how the routes between nodes within the network are calculated. In this case, one end of the characteristic scale is represented by the reactive method whereby the route from source to destination is computed only at the point when data are to be sent. At the other end of the scale is the proactive method whereby routes to all nodes are pre-computed and the information is usually stored within a table. In-between these two characteristic extremes are methods where routes are partially pre-computed and partially computed when data are to be sent. *Update Period* specifies the method by which route information is updated. At one end of the characteristic scale is the event driven update such as a node entering or leaving the network. The periodic update where updates are carried out at pre-defined times regardless of the state of the network is at the other end of the scale. *Source Routing* defines how the routing information is transmitted across the network. At one end of the characteristic scale is the source method whereby the complete route is sent along with the data from the source node. The other extreme is the hop-by-hop method where only enough route information is sent with the data to traverse to the next node.

Table 3. Routing protocol classification

Sub-protocol	F. Route Computation	G. Update Period	H. Source Routing
OLSR-INIA [13]	Proactive	Hybrid	Hybrid
FISHEYE [14]	Proactive	Periodic	Hybrid
DSR [1]	Reactive	Event	Source
AODV [1]	Reactive	Event	Hop-by-hop
ZRP [14]	Hybrid	Periodic	Source
STAR [15]	Proactive	Event	Source

3.6 Interfacing Sub-Protocols

The decision regarding which sub-protocol to choose in each layer is carried out sequentially starting from the bottom physical layer. There are inevitably some sub-protocols that can only be interfaced to a subset of other sub-protocols in the next layer due to compatibility problems, leading to a reduced set of possible communication protocol stacks. After the choice of sub-protocol has taken place within the current layer, a simple masking method is used to reduce the available choice of sub-protocols at the next layer based upon the current layer's choice.

3.7 The Genetic Algorithm and Fitness Function

As shown in Fig. 3, this methodology uses a genetic algorithm [16] with an initial population of N random protocols which are simulated in turn, each returning performance measurements. These are then used by the fitness function to obtain a fitness score for each protocol. The selected fittest protocols then undergo crossover and mutation to create a new population of fitter protocols. This is repeated until an optimum solution is found. The aim of the genetic algorithm is to minimise a fitness function. The fitness function (F) is a sum of the chosen performance metrics which allow the network to be evaluated for a given protocol stack. The first performance metric (P1) is calculated within a defined period of time and given that the aim is to *minimise* the fitness function, the ratio of the two numbers is inverted from the normal calculation used for throughput. The subsequent three performance metrics (P2, P3 and P4) add a small penalisation factor for specifying a set of characteristic values a long way from the chosen protocol by taking the length of the shortest distance from the nearest protocol into account at each layer of the protocol stack. This is necessary due to the limited protocol choice meaning that some protocols took up a very large volume within

the solution space increasing the probability of being selected even when there were other equally fit protocol choices available.

$$F = P1 + P2 + P3 + P4. \quad (1)$$

$$P1 = \text{Number of packets sent / Number of packets received} \quad . \quad (2)$$

$$P2,P3,P4 = \text{Shortest distance in layer / Maximum distance in layer} \quad . \quad (3)$$

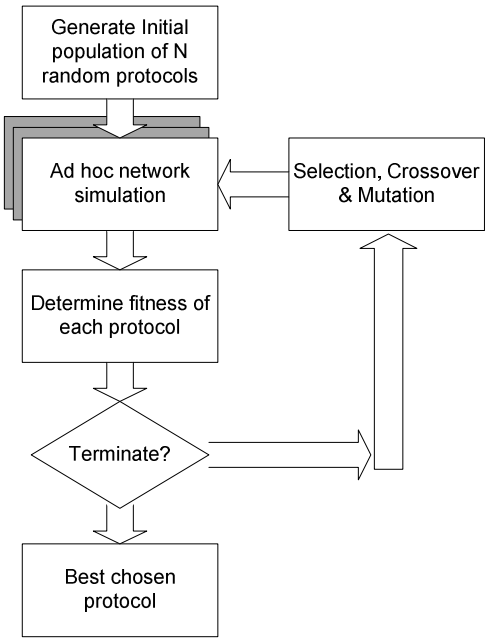

Fig. 3. Genetic algorithm flow

4 Network Scenario

A network scenario was generated to determine how well the methodology performed under changing network conditions by applying faults to the network and monitoring how the protocol adapted. Simulations were run five times for each scenario case generated. QualNet [17] was used for the operating environment and Matlab was used for the centralised controller. The protocol stack model used within QualNet closely resembles the OSI model and used a Constant Bit Rate (CBR) at the application layer, and User Datagram Protocol (UDP) at the transport layer. The parameters for each of the sub-protocols were assigned their default QualNet values.

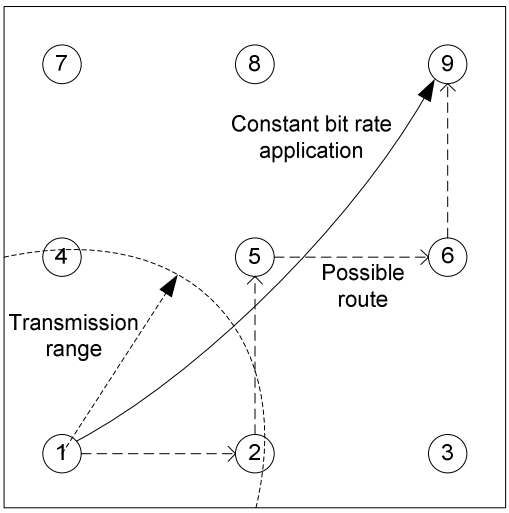

Fig. 4. Layout of the 3 x 3 mesh scenario

Within this scenario nine nodes were positioned in a 3 by 3 mesh arrangement as shown in Fig. 4. The distances between the nodes were set close to the maximum transmission range so that the probability of data packets transmitted diagonally, for example from node 1 to node 5, or even directly to node 9 was very low. This forced multiple possible routing paths when data was transmitted from node 1 to node 9 using a constant bit rate application. The simulation was run for 25 generations to find the optimum protocol. At the 25th generation faults were applied to the network and the simulation then ran for a further 25 generations to determine how the protocol adapted. The number of data packets received from each protocol stack combination was assessed independently to determine how well the algorithm performed. The simulation parameters for the scenario are given in Table 4. Three faults were applied to the network at nodes 2, 3 and 5. Intermittent faults were applied to nodes 2 and 3 whereby the faults prevented the node from operating for a short period of time at random intervals. A static fault was applied to node 5 which lasted for the first 5 seconds of the simulation.

Table 4. Simulation parameters

Parameter	Details
Sending node	1
Receiving node	9
CBR details	10Mbits/sec
Simulation time per protocol selected	15seconds
Population Size	25
Generation number when faults applied	25

Total generations	50
Fault 1	Node 2 intermittent
Fault 2	Node 3 intermittent
Fault 3	Node 5 static 0-5s
Maximum mobile speed of nodes	10m/s

Table 5. Scenario test cases

Case number	Mobility	Mutation rate
1	static	0.2
2	static	0.5
3	static	0.7
4	Random Waypoint	0.2
5	Random Waypoint	0.5
6	Random Waypoint	0.7

The scenario was run six times by varying two parameters. The first parameter, mobility, was set to either static where the nodes remained in a fixed position or set to random waypoint where the nodes could move about in a random fashion as an ad hoc network might behave in practice. The second parameter, mutation rate, was varied to investigate whether changing the diversity of the population was able to improve the ability of the algorithm to find the global optimum. Test cases are given in Table 5.

5 Results

Fig. 5 shows the output from a single run over 25 generations of the genetic algorithm under a mobile environment at the maximum mutation rate of 0.7 (first 25 generations of case 6 in Table 5), with no faults set. The crosses show the points generated by the genetic algorithm of chosen characteristic values. After 25 generations there is clustering around chosen protocols for each of the three optimised layers. For this particular case it correctly chose PHY 802.11a, CSMA, and FISHEYE as the optimum protocol selection.

Fig. 6 shows how the mean fitness score of the population changes over the generations. The mean fitness score diminishes quickly to a minimum at the 10th generation long before it approaches the 25^{th} generation where the optimum protocol is established. After the 25^{th} generation faults are applied and the mean fitness score rapidly increases as the current population is no longer optimal. At the 37th generation the mean fitness score diminishes again as the protocol adapts to the environment. For this particular case it correctly chose PHY 802.11a, MAC 802.11, and AODV as the optimum protocol selection.

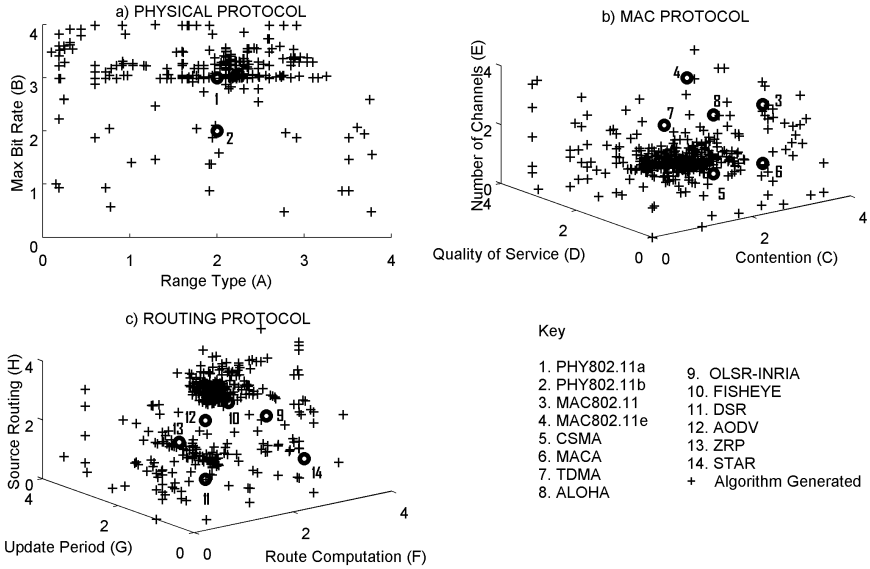

Fig. 5. Optimisation for the 3 by 3 mesh scenario for case 6 with no faults set.

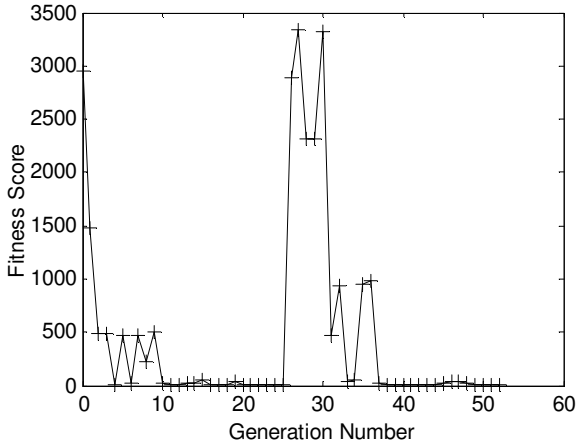

Fig. 6. Mean fitness score for a single run of case 6

Fig. 7 shows the number of times the correct protocol was generated over 5 runs for each of the 6 scenario cases. For the static node cases there appeared to be some improvement when a high mutation rate was used after faults were applied to the network. For the mobile case however the opposite was true and could be due to the fact that moving nodes is a harder problem to solve. Further testing would be needed before drawing more conclusions from these results. Out of the total 60 runs conducted for this scenario, 43 of them resulted in the identification

of the correct optimum protocol suggesting a preliminary identification rate of 72%.

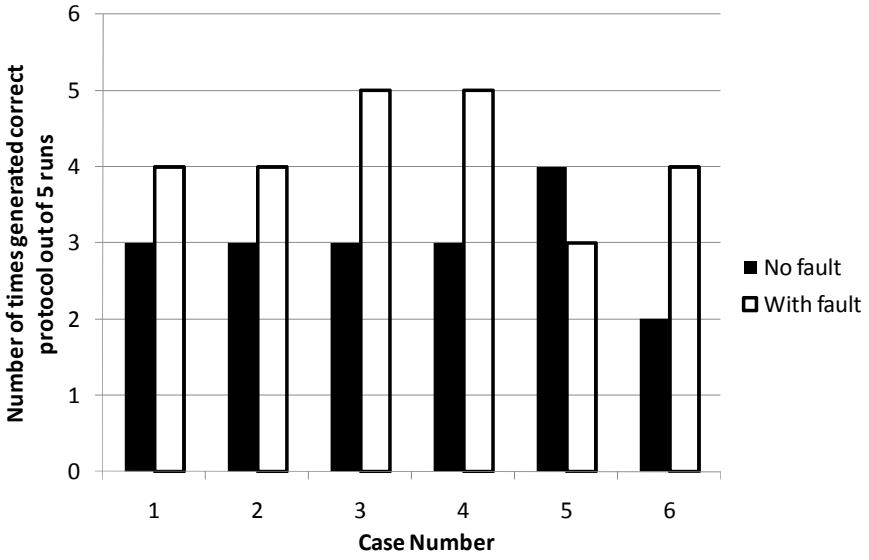

Fig. 7. Effect of varying the mutation rate on the ability of the genetic algorithm to find the optimum protocol for each of the 6 cases.

6 Conclusion

The methodology proposed in this paper is a first step at dynamically evolving and adapting an ad hoc communication protocol under changing network conditions. It uses simple protocol mapping techniques and a genetic algorithm to select the optimum protocol for a given scenario using a simple fitness function to provide feedback regarding the network's current performance. Preliminary results show that the methodology is able to find global optima for a network scenario under varying conditions, and has a global optimum identification rate of 72%. The methodology is by no means complete and there are areas which can be developed further. For example if the operating environment, which is currently simulated in QualNet, was directly replaced with a real-time environment then it would take a minimum of two and a half hours (plus computation and interfacing time) to establish an optimum protocol if all population trials were carried out in a sequential manner (25 populations x 15 seconds of run-time x 25 generations). This response time could be optimised down to a few minutes, making it more realistic, with higher data rates to capture throughput information for the fitness function in a shorter run-time, together with a fitness threshold to optimise and reduce the number of generations. Alternatively, or in addition to the above

optimisation, the instantaneous state of the network could be captured at regular intervals and input into high speed offline parallel processors to predict the optimum protocol before sending a global protocol update and minimising disruption to the network. For realistic application within the distributed architecture of an ad hoc network however the methodology would need to be designed for real-time performance with distributed rather than centralised control. This would require each of the nodes acting as simple interacting elements evolving the optimum communication protocol through local interactions and decisions. Future work would include using FPGAs to provide this hardware architecture with parallel processing and run-time reconfiguration capability to allow dynamic protocol changes.

Acknowledgments

This work was supported by the Complexity Science Doctoral Training Centre at the University of Warwick under EPSRC funding. The authors would like to thank Professor Sadie Creese of the University of Warwick for helpful review comments.

References

1. E. Royer, and C.-K. Toh, "A review of current routing protocols for ad-hoc mobile wireless networks," *IEEE Personal Communications Magazine,* vol. 6, no. 2, 1999.
2. P. Ocenasek, and M. Sveda, "An approach to automated design of security protocols," in Proceedings of the International Conference on Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies (ICNICONSMCL'06), 2006.
3. H. Xue, H. Zhang, and S. Qing, "A schema of automated design security protocols," in International Conference on Computational Intelligence and Security Workshops, 2007.
4. A. Perrig, and D. Song, "On a first step to the automatic generation of security protocols."
5. S. Virtanen, J. Isoaho, T. Westerlund, and J. Lilius, "A programmable general protocol processor - a proposal for an expandable architecture," in *URSI/IEEE XXIV Convention on Radio Science*, 1999.
6. J. Oberg, A. Kumar, and A. Hemani, "Scheduling of outputs in grammar-based hardware synthesis of data communication protocols," *IEEE.*
7. I. Pavlosoglou, M. S. Leeson, and R. J. Green, "Applying emergence to the design of routing protocols for the security of wireless ad hoc networks," in Proceedings of the First International Conference on Security and Privacy for Emerging Areas in Communications Networks, 2005.
8. R. Jurdak, C. V. Lopes, and P. B. Baldi, "A survey, classification and comparitive analysis of medium access control protocols for ad hoc networks," *IEEE Communications Surveys,* vol. 6, no. 1, 2004.
9. F. Wang, A. Ghosh, C. Sankaren *et al.*, "Mobile wimax systems: Performance and evolution," *IEEE Communications Magazine,* vol. 46, no. 10, 2008.

10. E. D. Pinedo-Frausto, and J. A. Garcia-Macias, “An experimental analysis of zigbee networks,” in IEEE Conference on Local Computer Networks, 2008.
11. E. Ferro, and F. Potorti, “Bluetooth and wi-fi wireless protocols: A survey and a comparison,” *IEEE Wireless Communications,* vol. February, 2005.
12. A. C. V. Gummalla, and J. O. Limb, "Wireless medium access control protocols," *IEEE Communications Surveys & Tutorials,* Second Quarter, 2000.
13. P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, and L. Viennot, “Optimized link state routing for ad hoc networks,” in Technology for the 21st Century Multi Topic Conference, 2001.
14. X. Zou, B. Ramamurthy, and S. Magliveras, “Routing techniques in wireless ad hoc networks - classification and comparison,” in Proceedings of the Sixth World Multiconference on Systemics, Cybernetics, and Informatics, 2002.
15. J. J. Garcia-Luna-Aceves, and M. Spohn, “Source-tree routing in wireless networks,” in Seventh International conference on Network Protocols, 1999.
16. D. Whitley, “A genetic algorithm tutorial,” *Statistics and Computing,* vol. 4, 1994.
17. "Qualnet." http://www.scalable-networks.com.