# Chapter 2
# Self-Configuring, Self-Organizing, and Self-Healing Schemes in Mobile Ad Hoc Networks

**Doina Bein**

**Abstract** The evolution of technology, the expansion of the Internet, and the tendency of systems to become more software-dependent make computing environments and networks more complicated and less humanly controlled. In this chapter, we consider the problem of organizing a set of mobile nodes, with unique IDs, that communicate through a wireless medium, into a connected network, in order to obtain a *self-configuring* or *self-organizing* network. Additionally, we address the issue of how a reliable structure, once acquired by self-configuring, can be maintained when topological changes occur, due to node failure, node motion, or link failure, in order to obtain a *self-healing* network. We discuss these concepts and present a brief history of self-configuring or self-healing algorithms, respectively, for wireless mobile networks. We detail a number of representative algorithms used in practice. We then go on to address the current theoretical results on self-configuring networks for which we propose directions for future research.

## 2.1 Introduction

Beginning as a military application, Mobile Ad hoc Networks (MANETs) had become largely used for personal use: e.g., personal area network (PAN), for short-range communication of user devices, wireless local area network (WLAN), and in-house digital network (IHDN), for video and audio data exchange.

A MANET is a peer-to-peer, multihop connected network, and is composed usually of tens to hundreds of mobile nodes. The nodes have transmission ranges of up to hundreds of meters and each individual node must be able to act both as a host, which generates user and application traffic, and as a router,

D. Bein (✉)
Department of Computer Science, University of Texas at Dallas, 800 W. Campbell Road; MS EC31, Richardson, TX 75080, USA
e-mail: siona@utdallas.edu

which carries out network control and routing protocols. The mobility of the nodes makes self-configuring of the network much harder. Although the nodes are battery-powered, the energy consumption is of second importance (a difference to wireless sensor networks), since each device could have its battery recharged or replaced when needed. Providing quality of service (QoS), which is the routing of packets, is the most important issue and has to be scalable in the context of a changing topology, limited bandwidth, and limited transmission power. In contrast, a cellular network is a large network consisting of stationary and mobile nodes, where the mobile nodes largely outnumber the stationary ones. The stationary nodes are called *base stations*; they have unlimited power supply, and are placed to cover a large area with little overlap. For cellular networks, the self-configuring aspect is needed for handling mobile-to-mobile node communication when the mobile nodes move within the area.

In a MANET, the nodes must share the wireless communication medium efficiently and be able to perform routing for the transmitted data. Channel access poses a difficult problem, namely the so-called *hidden terminal problem* (Tobagi and Kleinrock [55]). In this problem, a node is considered to be a terminal: transmissions from different nodes that use the same communication channel at the same time may interfere by colliding with one another, and as a result either a truncated message or corrupted data are received.

Inspired by biological systems, IBM introduced the term *autonomic system* for a system that is less dependent on human intervention and able to cope by itself with the complexity and heterogeneity of its life cycle (Horn [25]) – in other words is *self-manageable*. As defined by IBM, an autonomic system has four major and four minor characteristics [27]. There are four major characteristics, referred to as "self-CHOP":

- *Self-configure* is the property to implement specific strategies to change the relations among the components to guarantee either survivability in changing environments or a higher performance.
- *Self-heal* is the property to detect (or predict) faults and automatically correct faults (events that cause the entire system or parts of it to malfunction).
- *Self-optimize* is the property of monitoring its components and fine-tune the resources automatically to optimize the performance.
- *Self-protect* is the property of anticipating, detecting, identifying, and protecting itself from attacks in order to maintain overall integrity.

Additionally, the four minor characteristics are:

- *Self-aware* means knowing itself (its components, resources, the relations among them, and the limits) in a detailed manner.
- *Self-adapt* means automatically identifying the environment, generating strategies on how to interact with neighboring systems, and adapt its behavior to a changing environment.
- *Self-evolve* means generating new strategies and implementing open standards.

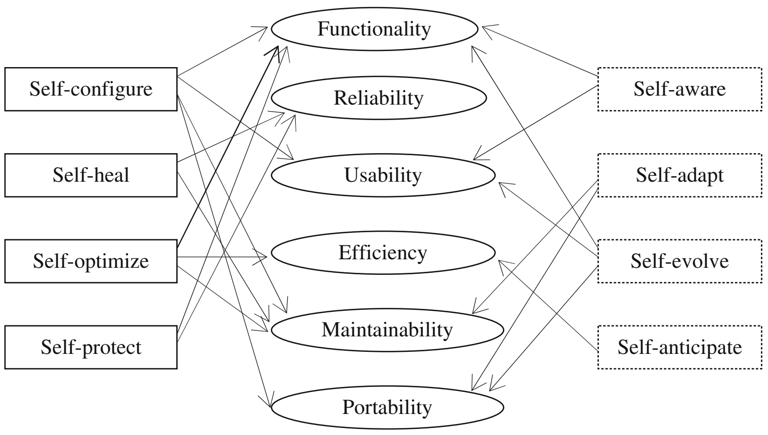

**Fig. 2.1** A self-managing system and the relationship between the basic factors of software evaluation and the characteristics of an autonomous system

- *Self-anticipate* means anticipating the requests for resources from the users without involving them in the complexity of its functionality.

The international standard ISO 9126 evaluates the software. It is divided into four parts, which address the quality model, external metrics, internal metrics, and quality in use metrics. The quality model established in the first part of the standard, ISO 9126-1, classifies software quality in a structured set of six characteristics (see e.g., the wikipedia page [58]). The relations between these characteristics and the quality factor of a system are depicted in Fig. 2.1.

## 2.2 Background

In this context we cite Robertazzi and Sarachik [48]: "A self-organizing network based on radio communications will create its own connections, topology, transmission schedules, and routing patterns in a distributed manner. It may establish local hubs, a backbone network, gateways, and relays."

The hidden terminal problem, presented in Section 2.2.1, creates a common pitfall for any self-configuring, self-healing, or self-optimizing MANET.

A self-configuring and self-organizing wireless network has two mechanisms implemented: *discovery* of routes between pair of nodes and *update* the current topology, by first detecting the node or link failures and secondly by optimizing the routes obtained through discovery. The discovery mechanism can be done proactively, when routes between any pairs of nodes are sought, periodically, or on-demand, when only certain routes are required. On updating the current topology, either single or multiple routes are maintained between a pair of nodes (more details are given in Section 2.2.2).

A self-optimizing mechanism attempts to improve the current topology with respect to either route length (*path-aware*) or energy consumption (*energy-aware*), with low overhead in terms of the transmission of control, discovery, or update messages. Gui and Mohapatra [23] presented a Self-Healing and Optimizing Routing Technique (SHORT) in which nodes monitor the on-demand routes to obtain a better local subpath.

Some researchers consider self-healing systems as a particular case of a *fault tolerant system*. Be aware that adaptive systems and self-healing systems are considered closely related. A *fault tolerant system* is able to sustain a certain number of faults, provided that enough time for recovery is given. They come in two different varieties: masking and non-masking. *Masking fault-tolerance* guarantees that the system continues to keep its functionality in the presence of faults. In contrast, *non-masking fault-tolerance* merely guarantees that when faults stop occurring, the system converges to configurations from where it continues to function. (More details are given in Section 2.2.3.)

## 2.2.1 Hidden Terminal Problem

For medium access control (MAC), the hidden terminal problem can be formulated as follows. Given four nodes, A, B, C, and D, such that A and C are not in the communication range of each other, A sends a packet to B on the same channel and at the same time when C sends a packet to D. Neither A nor C is able to determine, by itself, that a collision has occurred (see Fig. 2.2).

Numerous solutions [2, 7, 14, 19, 21, 29, 42, 47, 50, 53, 54] have been proposed for this problem. We present two representative solutions, namely the RTS/CTS approach and the timeslot approach.

The RTS/CTS (Request-to-Send/Clear-To-Send) approach [2, 7, 29], used in IEEE 802.11 wireless LAN standard, is similar to the Ethernet model. When node A wants to send a packet to node B, it first sends a RTS message to B, and

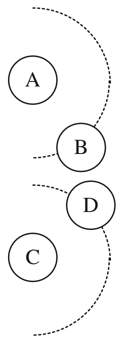

**Fig. 2.2** Hidden terminal problem

waits for B's reply as a CTS message. If B is busy or the channel is dedicated for other communication, A will not receive a CTS message, and will keep retrying until some timeout limit is reached or a CTS message is received from B. On receiving B's CTS message, A sends the data and waits for the acknowledgment.

The timeslot approach (TDMA) [21, 47, 50] provides a transmission schedule for each node in the network, which needs to be recalculated each time a node joins or leaves the network. Similar approaches concern the frequency bands and the spread spectrum code (CDMA).

The RTS/CTS approach works better for networks with unpredictable load transmission, while the timeslot approach works better for networks with more uniform load transmission per node. The timeslot approach guarantees quality of service, at the cost of high computational time for computing the schedule (when a group of nodes joins, the schedule for the whole network needs to be recomputed) and large time difference between two transmissions of the same node for densely distributed networks.

### *2.2.2 Self-Configuring and Self-Organizing MANETs*

A self-configurable system must be able to extract the necessary information supporting its software intelligence from the data it collects. The need for automatic changes (self-configuring) ought to take the following design issues for a self-configuring network in consideration:

- *Ad hoc deployment*: the nodes may not be positioned in a regular pattern (grid, honeycomb, 3D grid, 3D honeycomb, etc.).
- *Error-prone wireless medium*: the wireless medium is more error-prone than the wired medium, and collisions could occur more frequently.
- *Limited resources and energy constraint*: a unit has limited resources (battery, memory, computational power). The number of actions a node executes and the time consumed by an action must be minimized, in order to prolong its battery lifetime.

Adapting to the changing environment can be done periodically or gradually. The first proposed self-configuring protocols for a MANET are protocol LCA of Baker and Ephremides [4, 5, 12], protocol DEA by Post et al. [45, 46], and protocol Layer Net, proposed by Bhatnagar and Robertazzi [8]. They periodically discard the network topology information and rebuild the network from scratch. Later protocols consider a gradual approach to self-configure a MANET, for example, the protocol SWAN by Scott and Bambos [49].

The first self-configuring (self-organizing) wireless network was proposed as a two-tier hierarchical model by Baker and Ephremides [4, 5, 12]. The nodes are classified as ordinary, cluster heads, and gateways, with the restriction that a node belongs to a single cluster and it is one hop away from its cluster head.

Since selecting the minimum number of such cluster heads is NP hard, they proposed a link cluster algorithm (LCA) for categorizing the nodes and a link activation algorithm (LAA) to schedule (activate) the links between nodes.

The LCA algorithm uses a dominating set partitioning of the network based on node ID and works as follows: the node with the highest identity number among a group of nodes without a cluster head within one hop declares itself as a cluster head. The other nodes become either gateways (if there are connected to two or more cluster heads) or ordinary nodes. Variations of the LCA algorithm are to consider either the lowest ID or the highest connected node [19, 39] instead of the highest ID node.

The distributed evolutionary algorithm (DEA) proposed by Post et al. [45, 46] is based on a clique partitioning of the network and is uniform (the same for each node in the network). It works as follows: a *starter* node activates all its neighbors, which are part of some clique as itself (so-called clique neighbors), to begin communication based on a schedule decided by itself. Then these nodes become the starter nodes for the rest of the network.

A tree layer model proposed by Bhatnagar and Robertazzi [8] builds a spanning tree starting at some starter node. Each node has a layer number equal to its distance from the root (measured in number of hops). The schedules for transmission are made one layer at a time in a kind of a breadth-first search.

In protocol SWAN (Scott and Bambos [49]), new connections are sought during random access periods; and after a timeout, connections that do not respond to a control call are declared unusable.

Konstantinou et al. [30] propose a four-layer architecture for a self-configuring network, which could be applied to a MANET. From top to down, the *Application Layer* is the one where the request for data is initiated and the collected data are processed. The *Self-configuration Layer* generates the rules (methods) to be applied in case of disconnection. The *Configuration Model Layer* is responsible for the discovery and maintenance of the network topology, and monitoring the network elements. The mobile nodes play the role of *Network Elements*.

Sehrabi and Pottie [52] present a self-configuration scheme, similar to a TDMA schedule. Li and Rus [37] present a scheme where mobile nodes modify their trajectory to transmit messages in case of disconnected ad hoc networks. Gao et al. [15, 17] propose a randomized algorithm for constructing and maintaining a CDS with low overhead, by splitting the area into grids and selecting one active node per grid. Another connected topology proposed by Gao et al. [16] is the restricted Delaunay graph (RDG) where only Delaunay edges with a limited fix transmission range are included. Alzoubi et al. [3] describe a distributed construction of a minimum connected dominating set (MCDS) for the unit-disk graphs with a constant approximation ratio. Krishnamachari et al. [33] examine some self-configuration problems, specifically a partition of the network into coordinating cliques, Hamiltonian cycle formation, and conflict-free channel allocation, related to the formation of specialized structures on the network connectivity graph.

### *2.2.3 Self-Healing*

The concept of self-healing was inspired by the study of immune systems in biology (Forrest *e*t al. [13]). *Self-healing* is the property of a system to detect that it is not operating correctly and, with or without user intervention, makes the necessary adjustments to restore itself to normal. Healing systems that require external intervention are called *assisted-healing systems*. Self-healing systems (Ghosh et al. [20]) are generally sought for software agents, grid and middleware computing. The first type of a self-healing network was proposed by Grover [22] for a digital cross-connect system (DCS). The network is a dynamic restoration-control, employing autonomous restoration algorithms. The paper of Zhang and Arora [59, 60] focuses specifically on the self-healing algorithms for cellular networks.

The states through which a system transits during its lifetime are depicted in Fig. 2.3. When the system functions properly we say that it is in an acceptable state. When a fault occurs, generally considered to be a disconnection of the network due to node movement or crashes, then the system enters a degraded state, where some parts of the network continue to function properly but some parts do not. If the functionality of the network depends on the failed portion, then the system enters a failed state. A recovery mechanism is able to bring the network into an acceptable state, for example, by moving nodes to the affected part.

A recent survey of self-healing systems of Ghosh *et al.* [20] identifies three critical issues for self-healing systems: *(a)* maintenance of the system health, *(b)* detection of system failure mechanism, and *(c)* recovery to an acceptable ("healthy") state from a degradable or failed state.

a) Maintenance of the system health can be done by:
   - *Maintaining redundancy*: Replicating components.
   - *Probing*: A special component of the system collects updated information about the other components. Michiels et al. [41] propose a DMonA architecture in which two types of sensors gather information from the functional layer of a system: *State sensors (SS)* collect data related to the internal state, and *analysis sensors (AS)* collect data related to the messages flowing through the system.

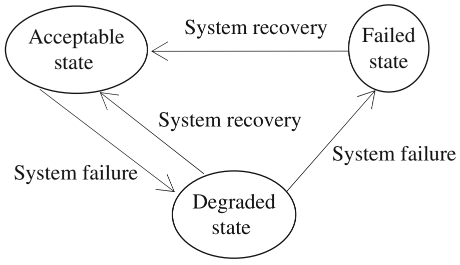

**Fig. 2.3** Transitions from the functional point of view of a MANET

  - *Performance log analysis*: Self-evaluate the performance and self-fine-tune (Kaiser et al. [28]), monitor periodical for typical symptoms (Hong et al. [24])

b) Detection of system failure can be divided into:
  - *Something is missing* (i.e., a missing component, a missing response): Nagpal et al. [43] propose a strategy in which the agents can produce replications when they sense the disappearance of their neighbors. System failure can be assessed when messages (George et al. [18]) or responses to a query are not received (Aldrich et al. [1]).
  - *Some monitored value is out of range*: Merideth et al. [40] propose a model in which proactively probing a system and collecting data when a fault is detected could improve the survivability of it.
  - *A foreign element is detected*: A proactive containment strategy notifies to the system the presence of a malicious replica (Merideth et al. [40]).

c) Recovery mechanism could employ:
  - *Redundancy techniques as replicating components*: Nagpal et al. [43] propose a strategy in which the agents can produce replications when they sense the disappearance of their neighbors. During the healing process, a biological system produces more cells than necessary to combat the intrusion, so that some may survive the attack (George et al. [18]).
  - *Repair strategies*: In some cases, the faulty components are isolated and the system is reconfigured (de Lemos et al. [36]).
  - *Byzantine agreement*: For some systems, where the function of the system is to produce output results and the non-faulty components always produce the same value for the output, voting could take place and majority is applied to the output results of all processes. In that way, faulty processes can be detected and tolerated (Merideth et al. [40]). Another strategy is to delegate several agents to do the same task, and either select directly or vote on the agent that achieves the best complexity (on time, space, etc.) (Huhns et al. [26]).

Periodically, the system may check for malfunctions and, based on the existing threshold values of certain parameters, may decide that a failure occurs or a malicious agent is present. The criteria of what is considered an acceptable state may vary in time (Shaw [51]), and the key properties of a system such as its performance and accuracy may have dynamic thresholds. The steps are given in Fig. 2.4.

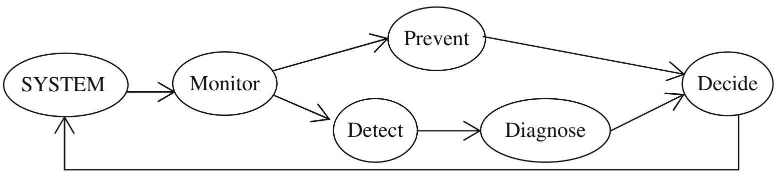

**Fig. 2.4** Steps taken by a self-healing system when a fault occurs

*Monitor*: observe the behavior of a system and the values of specific parameters (called *indicators*).
*Detect*: automatically establish when the observed behavior and indicators deviate from an established, acceptable range.
*Diagnose*: determine whether a detected deviation of a set of detected deviations could be characterized as a *fault*.
*Decide*: automatically change or intervene to repair a diagnosed fault.
*Prevent*: anticipate possible faults based on monitored indicators.

The failure detection mechanism, besides detecting the fault, can accomplish some or all of the following: gauge the degree of malfunction, assert whether the recovery mechanism is needed, assert whether the recovery mechanism is able to correct the state, and bring the system to an acceptable state.

## 2.3 Thoughts for Practitioners

To provide high-quality Internet access to mobile users using portable devices such as cell phones, laptops, PDAs, when the user is not in the proximity of a stationary access point (such as a base station), it is desirable to reach the access point via the portable devices of other users in a multihop fashion. In this way, extended coverage and better throughput for the same bandwidth are provided; consider, for example, a small area where a large number of users try to use their cell phones at the same time. The route to that access point, and the way the network of portable devices adapts when a particular access point is removed from the network, is just an example of how the current theoretical research on MANET can help design self-configuring MANETs.

In an ad hoc environment, the set of neighbors can change at arbitrary moments of time. Thus in order to increase the throughput on routing the packets, nodes should use fewer routes; these routes form the *communication backbone* of the network. At the same time, since in an ad hoc network a node can fail or move somewhere else with high frequency, the nodes that are part of the backbone must be provided with enough redundancy for communication. This is done by letting them act as alternative routers, and alternative routes have to be available before crashes affect the communication backbone. Unlike in the wired environment, bidirectional communication is not guaranteed between any pairs of nodes, since their communication range is not fixed and can also vary based on node power.

There is a tradeoff between selecting a smaller set of nodes and the power consumption of the selected nodes. Maintaining the routing infrastructure to only a subset of nodes reduces the routing overhead (the schedule of the MAC) and excessive broadcast redundancy. At the same time, to save energy, unused non-backbone nodes can go into a sleeping mode and wake up only when they have to forward data or are selected due to the failure or movement of nodes. Selecting fewer nodes to act as routers has a power consumption disadvantage:

The routers will deplete their power faster than the non-router nodes. Thus once the power level of these nodes falls below a certain threshold, those nodes can be excluded as routers, and some other nodes have to replace them.

Selecting the nodes in the communication backbone can be modeled as a linear program where for each node *i* in the network a binary variable $y_i \in \{0,1\}$ is associated, such that $y_i = 1$ if and only if node *i* is selected to be part of the backbone. The objective is to minimize the sum of all *y*-variables such that some properties are achieved for every node *i* (e.g., for the connected dominating set the sum of $y_i$ and all the *y*-variables of *i*'s neighbors is greater than 1; and if two *y*-values are 1, then there is a connection between the corresponding nodes).

For all the above reasons, hierarchical structures such as link-cluster architecture and dominating sets (with varieties as connected dominating set, weakly connected dominating set [11], *k*-fold dominating set [31, 32, 34, 38, 56], *k*-dominating set [35, 44, 57], and the *k*-connected *k*-dominating set [9, 10]) are not able to provide sufficiently fast redundancy or fault-tolerance for high-speed or real-time networks, where the latency in every node should be very short. The decision time at every node needed to decide how to route messages should be comparable to the time of the propagation delay between neighboring nodes.

The *k*-fold cover *t*-set [6] provides a favorable alternative to hierarchical structures by selecting not only nodes from the immediate neighborhood but also nodes located at two or more hops. Thus the network can tolerate up to *k* node failures without losing the routing infrastructure. Unfortunately, the *k*-fold cover set model can fit better for wireless networks with low traffic or without power constraints.

Since most of the dominating set-related problems are NP-complete, heuristics or approximation algorithms are given for some, but no implementation in an actual architectural model. It is for practitioners to implement them and compare them for the practical point of view by simulations.

## 2.4 Directions for Future Research

The topological connectivity of a MANET is variable due to the mobility of the nodes and the limited battery power at each node. This variation motivates the necessity of generating rules or actions to be executed when a change occurs in the network, and preference is given to the distributed algorithms that are able to cope with the change and adapt to it, in shorter time, with the minimum cost. It is preferred that the time to correct the failure should be proportional to the number of nodes involved in the failures.

Algorithms that require a large amount of time or number of messages for a small number of faults, thus an excessive amount of energy and bandwidth, do not scale well. At the same time, algorithms that have a low cost, but require reactivation each time a failure occurs, may disrupt the functionality of the network more frequently. Thus an amount of redundancy is required in selecting in selecting the nodes that are part of the backbone communication of the network and the routing procedure at each node, at a time when the self-healing actions are not executed, unless the network is already in a degraded state.

## 2.5 Conclusions

In this chapter, we have presented the problem of organizing a set of mobile, radio-equipped nodes, nodes that communicate through a wireless medium, into a connected network, in order to obtain a *self-configuring* or *self-organizing* network. We address the issue of how a reliable structure, once acquired by self-configuring, can be maintained when topological changes occur due to node failure, node motion, or link failure, in order to obtain a so-called *self-healing* network. In Section 2.2, we have detailed some protocols; simulations that illustrate different cases are available on the papers that have proposed them. A number of theoretical results presented in Section 2.3 have not been implemented in real networks; no protocols have been proposed; thus there is work left for practitioners to implement, and compare, from the practical point of view by simulations.

## Terminologies

*Assisted-healing system*: a healing system that requires external intervention.

*Autonomous system*: a system that is less dependent on human intervention and able to cope by itself with the complexity and heterogeneity of its life cycle.

*Energy-aware optimizing mechanism*: mechanism that attempts to improve the current topology with respect to energy consumption.

*Hidden terminal problem*: transmission from different nodes that use the same communication channel at the same time may interfere by colliding with each other, and as a result either a truncated message or some garbage is received.

*On-demand discovery mechanism*: only certain routes, required by the application layer, are sought by the discovery mechanism.

*Path-aware optimizing mechanism*: mechanism that attempts to improve the current topology with respect to route length.

*Proactive discovery mechanism*: the routes between any pairs of nodes are sought by the discovery mechanism, periodically.

*Self-configuring system*: system that changes the relations among the components to guarantee either survivability in changing environments or higher performance.

*Self-healing system*: system that detects or predicts faults, and automatically corrects faults (events that cause the entire system or parts of it to malfunction).

*Self-optimizing mechanism*: mechanism that improves the current topology with respect to either route length (*path-aware*) or energy consumption (*energy-aware*), with a low overhead in terms of the transmission of control messages; that is, messages used by discovery or updating mechanism.

## Questions

1. What are the self-CHOP characteristics of an autonomous system?
2. Exemplify one difference between a self-configuring and a self-healing system.
3. Can the hidden terminal problem occur when the message sent by node A reaches node C as well? What if the nodes B and C are the same?
4. Exemplify one difference between discovery mechanism and updating mechanism for self-configuring networks.
5. For the discovery mechanism, which is more energy-consuming, proactive or on-demand discovery?
6. A self-optimizing mechanism, which is part of an updating mechanism for self-configuring networks, attempts to improve the current topology with respect to what?
7. A scalable self-configuring mechanism must have a low overhead in terms of transmission of control messages. Why?
8. LCA is a heuristic approach to the dominating set problem, and the dominating set problem is NP-complete. DEA is a heuristic approach to the clique partition problem, and the clique partition problem is NP-complete as well. Which of the two, LCA or DEA, is better for a good approximation?
9. Exemplify two critical issues for the self-healing systems, as identified by Ghosh et al. [20].
10. What are the steps a self-healing mechanism takes when a fault occurs?

## References

1. Aldrich J, Sazawal V, Chambers C, Notkin D (2002) Architecture-centric programming for adaptive systems. Proc 1st Workshop on Self-healing Syst, 93–95
2. Alwan A, Bagrodia R, Bambos N, Gerla M, Kleinrock L, Short J, Villasenor J (1996) Adaptive mobile multimedia networks. IEEE Pers Comm **3**(2):34–51
3. Alzoubi KM, Wan P-J, Frieder O (2002) Message-optimal connected-dominating-set construction for routing in mobile ad hoc networks. Proc 3rd ACM Intl Symp on Mobile Ad Hoc Networking and Computing, 157–164

4. Baker DJ, Ephremides A (1981) A distributed algorithm for organizing mobile radio telecommunication networks. Proc 2nd Intl Conf Distrib Computing Syst
5. Baker DJ, Ephremides A (1981) The architectural organization of a mobile radio network via a distributed algorithm. IEEE Trans Commun **29**:11
6. Bein D (2008) Fault-tolerant k-fold pivot routing in wireless sensor networks. Hawaii Intl Conf. Syst. Sci (HICSS), 245
7. Bharghavan V, Demers A, Shenker S, Zhang L (1994) MACAW: a media access protocol for wireless LAN's. Proc Conf Comm Architectures, Protocols, and Applications, 212–225
8. Bhatnagar A, Robertazzi TG (1990) Layer Net: a new self-organizing network protocol. Mil Comm Conf (Milcom) **2:**845–849
9. Dai F, Wu J (2005) On constructing k-connected k-dominating set in wireless networks. Proc IEEE Intl Parallel and Distributed Processing Symp (IPDPS)
10. Dai F, Wu J (2006) On constructing k-connected k-dominating set in wireless ad hoc and sensor networks. J Parallel Distrib Computing **66**(7):947–958
11. Dunbar JE, Grossman JW, Hattingh JH, Hedetniemi ST, McRae AA (1997) On weakly-connected domination in graphs. Discrete Math **167/168:**261–269
12. Ephremides A, Baker DJ (1981) An alternative algorithm for the distributed organization of mobile users into connected networks. Conf Inf Sci Syst
13. Forrest S, Hofmeyr SA, Somyaji A (1997) Computer immunology. Comm ACM **40**(10):88–96
14. Gallager R (1985) A perspective on multiaccess channels. IEEE Trans Inf Theory **31**(2):124–142
15. Gao J, Guibas LJ, Hershberger J, Zhang L, Zhu A (2001) Discrete Mobile Centers. Proc 7th Annual Symp on Computational Geom (SCG), 188–196
16. Gao J, Guibas LJ, Hershberger J, Zhang L, Zhu A (2001) Geometric spanner for routing in mobile networks. Proc 2nd ACM Intl Symp on Mobile Ad Hoc Networking and Computing (MobiHoc), 45–55
17. Gao J, Guibas LJ, Hershberger J, Zhang L, Zhu A (2003) Discrete mobile centers. Discrete Comput Geom **30**(1):45–65
18. George S, Evans D, Marchette S (2003) A biological programming model for self-healing. 1st ACM Workshop on Survivable and Self-regenerative Syst, 72–81
19. Gerla M, Tsai JT-C (1995) Multicluster, mobile, multimedia radio network. Wireless Networks **1**(3):255–265
20. Ghosh D, Sharman R, Rao HR, Upadhyaya S (2007) Self-healing systems – survey and synthesis. Decis Support Syst **42**:2164–2185
21. Goodman DJ, Valenzuela RA, Gayliard KT, Ramamurthi B (1989) Packet reservation multiple access for local wireless communications. IEEE Trans Comm **37**(8):885–890
22. Grover WD (1987) The selfhealing network: a fast distributed restoration technique for networks using digital cross-connect machines. IEEE Globecom
23. Gui C, Mohapatra P (2003) SHORT: self-healing and optimizing routing techniques for mobile ad hoc networks. Proc 4th ACM Intl Symp on Mobile Ad Hoc Networking and Computing (MobiHoc), 279–290
24. Hong Y, Chen D, Li L, Trivedi K S (2002) Closed loop design for software rejuvenation. Workshop on Self-healing, Adaptive and Self-managed Syst (SHAMAN), 159–170
25. Horn P (2001) Autonomic computing: IBM's perspective on the state of information technology. http://www-1.ibm.com/industries/government/doc/content/bin/auto.pdf. Accessed 20 February 2008
26. Huhns MN, Holderfield VT, Gutierrez RLZ (2003) Robust software via agent-based redundancy. Proc 2nd Intl Joint Conf on Autonomous Agents and Multiagent Syst (AAMAS), 1018–1019
27. IBM (2008) Autonomic computing: The 8 Elements. http://researchweb.watson.ibm.com/autonomic/overview/elements.html. Accessed 20 February 2008

28. Kaiser G, Gross P, Kc G, Parekh J, Valeto G (2002) An approach to autonomizing legacy systems. Workshop on Self-healing, Adaptive and Self-managed Syst (SHAMAN)
29. Karn P (1990) MACA-a new channel access method for packet radio. 9th Computer Networking Conf- ARRL/CRRL Amateur Radio
30. Konstantinou AV, Florissi D, Yemini Y (2002) Towards self-configuring networks. Proc DARPA Active Networks Conf and Exposition (DANCE), 143
31. Kratochvil J (1995) Problems discussed at the Workshop on Cycles and Colourings, http://univ.science.upsj.sk/c&c/rhistory/cc95prob.htm, Accessed 20 February 2008
32. Kratochvil J, Manuel P, Miller M, Proskurowski A (1998) Disjoint and fold domination in graphs. Australas J Combinatorics **18**:277–292
33. Krishnamachari B, Wicker S, Bejar R, Fernandez C (2003) On the complexity of distributed self-configuration in wireless networks. Telecomm Syst **22**(1–4): 33–49
34. Kuhn F, Moscibroda T, Wattendorf R (2006) Fault-tolerant clustering in ad hoc and sensor networks. Proc 26th IEEE Intl Conf on Distributed Computing Syst (ICDCS), 68
35. Kutten S, Peleg D (1998) Fast distributed construction of small k-dominating sets and applications. J Algorithms, 40–66
36. de Lemos R, Fiadeiro JL (2002) An architectural support for self-adaptive software for treating faults. Proc 1st Workshop on Self-healing Syst, 39–42
37. Li Q, Rus D (2000) Sending messages to disconnected users in disconnected ad hoc mobile networks. Proc 6th Annual ACM/IEEE Intl Symp on Mobile Computing and Networking (MobiCom), 44–55
38. Liao CS, Chang GJ (2003) k-tuple domination in graphs. Inf Processing Letters **87**:45–50
39. Lin CR, Gerla M (1997) Adaptive clustering for mobile wireless networks. IEEE J Selected Areas of Comm **15:**1265–1275
40. Merideth MG, Narasimhan P (2003) Proactive containment of malice in survivable distributed systems. Proc Intl Conf on Security and Management, 3–9
41. Michiels S, Desmet L, Janssens N, Mahieu T, Verbaeten P (2002) Self-adapting concurrency: the DMonA architecture. Proc 1st Workshop on Self-healing Systems, 43–48
42. Morrow Jr. RK, Lehnert JS (1992) Packet throughput in slotted ALOHA DS/SSMA radio systems with random signature sequences. IEEE Trans Comm **40**(7):1223–1230
43. Nagpal R, Kodancs A, Chang C (2003) Programming methodology for biologically-inspired self-assembling systems. AAAI Spring Symp on Computational Synthesis
44. Penso LD, Barbosa VC (2004) A distributed algorithm to find k-dominating sets. Discrete Applied Math **141**(1–3):243–253
45. Post MJ, Kershenbaum AS, Sarachik PE (1985) A distributed evolutionary algorithm for reorganizing network communications. Mil Comm Conf (Milcom)
46. Post MJ, Kershenbaum AS, Sarachik PE (1985) A biased greedy algorithm for scheduling multi-hop radio networks. Proc Conf Inf Sci Syst, 564–572
47. Rajendran V, Obraczka K, Garcia-Luna-Aceves JJ (2006) Energy-efficient, collision-free medium access control for wireless sensor networks. Wireless Networks **12**(1):63–78
48. Robertazzi TG, Sarachik PE (1986) Self-organizing communication networks. IEEE Comm Magazine **24**(1):28–33
49. Scott K, Bambos N (1997) Formation and maintenance of self-organizing wireless networks. 31st Asilomar Conf on Signals, Syst and Computers **1**:31–35
50. Shakkottai S, Rappaport TS, Karlsson PC (2003) Cross-layer design for wireless networks. IEEE Comm Magazine **41**(10):74–80
51. Shaw M (2002) Self-healing: softening precision to avoid brittleness. Workshop on Self-Healing Syst (WOSS), 111–114
52. Sohrabi K, Pottie G (1999) Performance of a novel self-organizing protocol for wireless ad hoc sensor networks. Proc IEEE Vehicular Technology Conf, 1222–1226
53. Sousa ES, Silvester JA (1988) Spreading code protocols for distributed spread-spectrum packet radio networks. IEEE Trans Comm **36**(3):272–281

54. Storey JS, Tobagi FA (1989) Throughput performance of an unslotted direct-sequence SSMA packet radio network. IEEE Trans Comm **37**(8):814–823
55. Tobagi FA, Kleinrock L (1975) Packet switching in radio channels: Part II – The hidden terminal problem in carrier sense multiple access and the busy tone solution. IEEE Trans Comm **23**:1417–1433
56. Vazirani V (2001) Approximation Algorithms. Morgan Kaufmann Publishers, Springer Verlag
57. Wang FH, Chang JM, Wang YL, Huang SJ (2003) Distributed algorithms for finding the unique minimum distance dominating set in split-stars. J Parallel Distrib Comput **63**:481–487
58. Wikipedia (2008) ISO 9126. http://en.wikipedia.org/wiki/ISO_9126. Accessed 20 February
59. Zhang H, Arora A. (2002) GS$^3$: scalable self-configuration and self-healing in wireless sensor networks. 21st ACM Symp on Principles of Distributed Computing (PODC)
60. Zhang H, Arora A (2003) GS$^3$: scalable self-configuration and self-healing in wireless sensor networks. Computer Networks **43**(4):459–480