Projects for Wireless Ad Hoc Networking

**Self-Configuring, Self-Organizing, and Self-Healing Schemes in Mobile Ad Hoc Networks**

By

John Jegede and Hassan Said

Table of content

**ABSTRACT**

A mobile ad hoc network (MANET), often called a mobile mesh network, is a self-configuring network of mobile devices connected by wireless connections. The Ad hoc networks are a new wireless networking standard for mobile devices. Unlike the usual mobile wireless networks, ad hoc networks do not rely on any fixed infrastructure. Instead, ad hoc devices rely on each other to keep the network connected. It represents a complex distributed systems that comprise of wireless mobile nodes that can freely and dynamically self-organize into arbitrary and temporary, ‘‘ad-hoc’’ network topologies, allowing people and devices to seamlessly internetwork in areas with no pre-existing communication infrastructure.

The changes in technology, expansion of the Internet, and the tendency of systems to become more dependent on software make computing environments and networks more difficult and less humanly controlled. In this paper, we consider the problem of organizing a set of mobile nodes, with unique identification number, that communicate through a wireless medium, into a connected network, so as to get 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 for wireless mobile networks. We detail a number of representative algorithms used in practice.

In this paper we talk about the schemes used for self-configuring in the first section followed by self-organizing then self-healing.

**BACKGROUND**

A self-configuring and self-organizing wireless network has two methods on which they are applied these methods are the discovery of routes between pair of nodes and updating the current topology, by first detecting the node or link failures and secondly by optimizing the routes obtained through discovery

Some researchers consider self-healing systems as a particular case of a fault tolerant system that can recover when anything happens. Adaptive systems and self-healing systems are considered similar. A fault tolerant system can sustain a certain number of faults, provided there is enough time for recovery is given.

With the application of MANET in many fields such as Tactical networks (Military Communication), Sensor Network (remote weathers for sensors), Emergency Services (Disaster recovery), Educational Applications (Setup virtual class), Entertainment (Multi-user games), these Self-Configuring, Self-Organizing, and Self-Healing Schemes will be very useful and will allow for proper communication between nodes without human intervention which will lead to speedy recovery of the network in case any node is down.

The characteristics of MANET are: Network is not dependent on any fix infrastructure for its operation, Ease of deployment, Speed of deployment, Dynamic Changing Topology of nodes Multi-hop network.

**INTRODUCTION**

A MANET is a peer-to-peer, multi hop connected network, and is composed mainly of tens to hundreds of mobile nodes. The nodes have transmission ranges that can be up to hundreds of meters and each individual node must be able to act both as a host, which generates user and application traffic data, and as a router, which carries out network control and routing protocols, sending the data from one node to another.

Inside the ad hoc networking field, wireless sensor networks take a special role. A sensor network is composed of a large number of small sensor nodes, which are typically densely (and randomly) deployed inside the area in which a phenomenon is being monitored. Wireless ad hoc networking techniques also constitute the basis for sensor networks. However, the special constraints imposed by the unique characteristics of sensing devices, and by the application requirements, make many of the solutions designed for multi-hop wireless networks (generally) not suitable for sensor networks

The primary objectives of Self-Configuring, Self-Organizing, and Self-Healing are Functionality, Reliability, Usability, Efficiency, Maintainability, scalability, Portability, and availability of systems composed of a huge number of subsystems. These are also characteristics of an autonomic system. Often Self-Configuring, Self-Organizing, and Self-Healing are referenced in conjunction with other self-X capabilities such as

* Self-configure: This involves appling specific strategies to change the connections among the components to guarantee either survivability in changing environments or a higher performance.
* Self-heal: This involves detecting or predicting faults and automatically correct faults or event that can cause the entire system to shut down
* Self-optimization: This involves monitoring the system components and fine-tuning the resources automatically to optimize the performance.
* Self-protection: This the property of anticipating, detecting, identifying, and protecting itself from attacks in order to maintain overall integrity.

Additionally, the minor characteristics are:

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

**SELF CONFIGURING SCHEMES**

A self-configurable system must be able to collect the necessary information supporting its software intelligence from the data it collects from other nodes in the network. When designing self configuring systems the following should be considered:

1)Ad hoc deployment: The nodes could be scattered around and may not be positioned in a regular pattern (grid ).

2) Wireless medium been error prone: the wireless medium is more error-prone than the wired medium, and collisions could occur more often.

3) Limited resources and energy constraint: a node has limited resources such as battery, memory, computational power etc. The number of actions a node takes and the time consumed by an action must be minimized, in order to prolong its battery lifetime.

We will be talking about three schemes in self configuration they are Link cluster algorithm, Adaptive self configuring sensor networks topologies algorithm (ASCENT) and Kernel tree routing algorithm (KTRA).

**Link cluster Algorithm**

The LCA algorithm uses a dominating set partitioning of the network based on node ID and works as follows: the node with the highest ID 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 instead of the highest ID node. Link cluster algorithm (LCA) for categorizing the nodes and a link activation algorithm (LAA) to schedule (activate) the links between nodes.

**Adaptive self configuring sensor networks topologies algorithm**

ASCENT adaptively selects “active” nodes from all nodes in the network. Active nodes stay awake all the time and perform multi-hop packet routing, while the rest of the nodes remain “passive” and periodically check if they should become active.

Initially, only some nodes are active. The other nodes remain idle passively listening to packets but not transmitting. The source starts transmitting data packets toward the sink because the sink is at the limit of radio range, it gets very high message loss from the source. We call this situation a communication hole, the receiver gets high packet loss due to poor connectivity with the sender. The sink then starts sending help messages to signal neighbors that are in listen-only mode –also called passive neighbors– to join the network. When a neighbor receives a help message, it may decide to join the network. When a node joins the network it starts transmitting and receiving packets, i.e. it becomes an active neighbor. As soon as a node decides to join the network, it signals the existence of a new active neighbor to other passive neighbors by sending a neighborhood announcement message. This situation continues until the number of active nodes stabilizes on a certain value and the cycle stops. When the process completes, the group of newly active neighbors that have joined the network make the delivery of data from source to sink more reliable. The process will re-start when some future network event (e.g. node failure) or environmental effect (e.g. new obstacle) causes message loss again.

ASCENT state transitions

In ASCENT, nodes are in one of four states: *sleep, passive, test, and active* states*.* Initially, a random timer turns on the nodes to avoid synchronization. When a node starts, it initializes in the *test* *state*. Nodes in the *test state* exchange data and routing control messages. In addition, when a node enters the *test state*, it sets up a timer Tt, and sends neighbor *announcement* messages. When Tt expires, the node enters the *active state*. If before Tt the number of active neighbors is above the *neighbor* *threshold* (NT), or if the average *data loss rate* (DL) is higher than the average loss before entering in the *test state*, then the node moves into the *passive state*. If multiple nodes make a transition to the test state, then we use the node ID in the announcement message as a tie breaking mechanism (higher IDs win). When a node enters the *passive state*, it sets up a timer Tp. When Tp expires, the node enters the *sleep state*. If before Tp expires the number of neighbors is below NT, and either the DL is higher than the *loss threshold* (LT) or DL is below the *loss threshold* but the node received a *help message* from an neighbor, it makes a transition to the *test state*.

*ASCENT parameters tuning*

ASCENT has many parameters that can affect its final behavior. Parameters like the timers Tt, Tp, Ts, *neighbor* (NT), and *loss threshold* (LT) are choices left to the applications. In this section, we explain the choices made in the current ASCENT algorithm. A particular application may select different parameter settings, for instance, perhaps trading energy savings for greater sensing coverage. The *neighbor threshold* (NT) value determines the average degree of connectivity of the network. When the sensing range of the micro sensors is lower than the radio communication range, the minimum number of nodes necessary to provide communication coverage may not be sufficient for sensing coverage. There is a trade-off between the energy consumed and/or the level of interference (packet loss) vs. the desired sensing coverage. An application could adjust this value dynamically depending on the events occurring in a certain area of the network. The *loss threshold* (LT) determines the maximum amount of data loss an application can tolerate before it requests help to improve network connectivity. This value is very application dependent.

**Kernel tree routing algorithm**

The KTRA (Kernel Tree Routing Algorithm) is for self-configuring mobile Ad hoc networks. KTRA reconstructs the actual network topology as a logic tree topology, thus the network nodes need only to maintain partial routing information, and the update requirement is restricted around the branch. Furthermore, KTRA has the advantages of both low delay and high flexibility comparable to that of other self-configuring algorithm.

Our proposed KTRA (Kernel Tree routing algorithm) is a unique self configuring and routing algorithm for mobile Ad Hoc network. It applies the routing information exchange among the network nodes to design the construction algorithm for the tree topology and the communication protocol, and therefore reconstructs the logical tree topology. In KTRA, the kernel tree grows from the root of the tree and grows into a tree gradually. New node joins in the kernel tree under the criterion that the path between this new node and the root is the shortest path. All the communications between the nodes, such as traffic flow and topology maintenance, are restricted on the tree topology. It is well known that routing algorithm on tree topology is very simple. Inherently, there is no need to perform routing computation. However, the tree topology has to be updated in time when the network changes. Therefore, the network routing algorithm is reduced to formation and maintenance of kernel tree, and communication can be performed easily in this dynamic topology environment.

*KTRA* Tree Formation Algorithm

KTRA comprises two sub-algorithms, the formation and maintenance algorithm of tree and the routing algorithm. Let denote the characteristics of the (N,P, L) kernel tree, where N and P are the identifications of node and its parent node, respectively, and L is the level of node. The level of the node that has not joined in the kernel tree is 0 and the level of root node is 1. For the node that has joined in the kernel tree, the hops between this node and its root node plus 1 is its corresponding level value.

The procedure of the tree formation is as follows.

Step 1. Node initiation: All the nodes set to zero

Step 2. Root node confirmation: The root node can be set manually or through the consulting between the nodes.

Step 3. Nodes joining in the tree: The new node N broadcasts its joining-in request, then according to the reply messages from its neighbor nodes which have joined in the tree, select a lowest-level neighbor node as its parent node.

Step 4. After confirm *P*i, node *N* announces its joining-in message to *P*i

*KTRA* Tree Maintenance Algorithm

The following limiting conditions are added to the nodes. Each node only has one parent node and must announce its children nodes to its parent node, but parent node need not announce its upper nodes to its children nodes. The changes of children nodes must announce to parent node. Above limiting conditions produce results as follows. The higher the node level is, the less topology information the node has to maintain. On the contrary, the lower the node level is, the more comprehensive topology information the node has to maintain. The root node maintains the entire network topology information.

**SELF ORGANIZING SCHEMES**

Self-organization can be summarized as the interaction of multiple components on a common system objective. In computer networks, self-organization is especially important in ad hoc networking because of the spontaneous interaction of multiple heterogeneous components over wireless radio connections without human interaction. There are two main terms that will represent and form the shape all self-organizing methodologies for ad hoc networks they are self-organization and emergence. Self-organization is a process in which pattern at the global level of a system emerges solely from numerous interactions among the lower-level components of that system. Emergent behavior of a system is provided by the apparently meaningful collaboration of components (individuals) in order to show capabilities of the overall system (far) beyond the capabilities of the single components. Self organization is often referred to as the multitude of algorithms and methods that organize the global behavior of a system based on inter-system communication.

Classification of self-organization methods

Here we will talk about two main classification of self-organizing methods and give some examples.

Location based methods: Geographical positions or affiliation to a group of surrounding nodes, i.e. clustering mechanisms, are used to reduce necessary state information to perform routing decisions or synchronizations. The primary goal for employing clustering algorithms is always to reduce the maintenance overhead. Zone Routing Protocol (ZRP) and HEED (Hybrid, Energy-Efficient, Distributed Clustering Approach) are examples for power-aware cluster based communication approaches for ad hoc networks

Neighborhood Information: Neighborhood information is available to perform necessary decisions such as decreasing the size of previously mentioned clusters to a one-hop diameter. The basic idea is to periodically exchange some hello or sync messages that include all necessary information for the particular algorithm to take decisions based on its local state and the state of its neighbors examples are AODV (Ad-Hoc On Demand Distance Vector Routing) and its successor DYMO (Dynamic MANET On Demand)

Self Healing schemes (5 pages)

The different schemes

Conclusion (1 page)

In coming years, mobile computing will keep flourishing, and an eventual seamless integration of MANET with other wireless networks, and the fixed Internet infrastructure, appears inevitable. In this paper we have presented the methods 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. There are a lot of other schemes out there but this are the easy to implement schemes. There is work left for researchers to still come up with more ways for a network to become reliant on itself with little to human intervention.

References (1 page)

Doina Bein, Self-Configuring, Self-Organizing, and Self-Healing Schemes in Mobile Ad Hoc Networks in S. Misra et al. (eds.), Guide to Wireless Ad Hoc Networks. 2009, pp. 27-39,

Falko Dressler, Self-Organization in Ad Hoc Networks: Overview and Classification. Autonomic Networking Group, Dept. of Computer Science, 2006.

Alberto Cerpa and Deborah Estrin, ASCENT: Adaptive Self-Configuring Sensor Networks Topologies.IEEE INFOCOM 2002

Qin Yu and Yuming Mao, A Novel Self-Configuring and Routing Algorithm for Mobile Ad Hoc Networks in IEEE International Conference on Mechatronics and Automation. 2006.

Pravin Ghosekar, Girish Katkar, Dr. Pradip Ghorpade, Mobile Ad Hoc Networking: Imperatives and Challenges in IJCA Special Issue on “Mobile Ad-hoc Networks” 2010