Autoconfiguration procedures for multiradio wireless mesh networks based on DHCP protocol

Krzysztof Grochla, Walter Buga, Jarosław Dzierżęga, Piotr Pacyna, Aleksander Seman
Proximetry Poland, Katowice, Al. Rozdzienskiego 91
{kgrochla,wbuga,jdzierzega,ppacyna,aseman}@proximetry.com

Abstract

We propose procedures for autoconfiguration of wireless mesh networks based on DHCP protocol. The proposed mechanism provides automatic address and radio parameters set-up for nodes with multiple radio interfaces, allowing us to fully automatically form and to start a wireless mesh network, and to route packets. It supports mesh networks comprised of multiple IP subnets build with multiple networking technologies. Some performance issues of the autoconfiguration procedure are discussed. The autoconfiguration scheme was implemented and tested on OpenWRT platform.

1. Introduction and motivation

For the last few years we have been experiencing a rapid growth of interest in mobile ad-hoc networking. The wireless mesh networks, comprised of nodes with multiple radio interfaces routing the packets, are a promising technology for example for broadband residential internet access or to provide connectivity to temporal events [8]. In order to simplify network deployment, the autoconfiguration procedures providing automatic network start-up with minimum manual configuration of the nodes are increasingly important.

The autoconfiguration component in a wireless mesh network provides a method for automatic mesh start-up, joining a node into an existing mesh and automatic repair of temporary connectivity outage. The main objective of this component is to simplify the node configuration process as much as possible. The autoconfiguration will provide networks with methods to set up transmission parameters while joining a node into an operational mesh, to merge two disjoint networks, and to sustain the network transmission in case of link or node failure.

The autoconfiguration component should provide:

- simple and fast node attachment into the mesh network,
- IP address assignment for node interfaces,
- automatic network boot-up,
- joining of two disjoint networks or a partitioned one.

We propose novel autoconfiguration procedures, based on extensions to DHCP server and client implementations and BOOTP relays, to support both automatic and predefined configurations of multiple radio interfaces in the mesh nodes. The underlying autoconfiguration method is applicable to networks with technology heterogeneity and it is independent of routing protocol in use. To optimize the routing it can automatically partition the IP address space into subnets. The DHCP server may be preconfigured by a network operator or can be automatically started on one of the mesh nodes, to provide a fully automatic network setup.

The proposed autoconfiguration schema was implemented and tested in laboratory network on Linux OpenWRT platform and Mikrotik routerboards.

The paper is organized as follows: in Chapter 2 we provide a brief review of the existing auto-configuration solutions. Then, in Chapter 3 we give a description of the autoconfiguration procedure. In Chapter 4 we briefly analyze the performance and overhead incurred by the autoconfiguration. We end up the paper with a short summary in Chapter 5.

2. Existing autoconfiguration approaches

The wireless mesh networks can be organized to exchange the packets with use of Layer 2 forwarding or Layer 3 routing, based on a destination node address. An example of a Layer 2 mesh is the IEEE 802.11s network [9]. The 802.11s mesh network uses MAC address assigned to the network interface card by its manufacturer, or configured by a user. A mesh network is identified by a Mesh ID (e.g. Mesh1), which is

978-1-4244-4439-7/09/$25.00 ©2009 IEEE

similar in purpose to SSID. In this case the user needs to configure only the Mesh ID, since the IP addressing is not discussed in the standard. However, Layer 2 mesh networks have some known limitations, such as support of only one networking technology (e.g. IEEE 802.x) and they do not propagate the routing information between radio interfaces.

The Layer 3 mesh networks route the packets at the IP layer and may support multiple types of radio technology like e.g. WiMAX and WIFI. They are able to forward routing information through multiple interfaces and provide better integration with wired networks. For such a network to be operational, a unique IP address needs to be assigned to network interfaces, together with establishment of the layer 2 connectivity between the connected interfaces. In the paper we concentrate on layer 3 wireless mesh networks.

There are two IP address autoconfiguration mechanisms in frequent use: DHCP[1] and Zeroconf[2]. Unfortunately, in their basic form these are not directly applicable in multi-hop wireless networks because in such a network setup either reachability is the problem, or address uniqueness, or both. There have been proposed autoconfiguration schemes for MANET, too. Most of them are based on Duplicate Address Detection [5, 10-11] and cannot be used in a network which is not a single broadcast domain, as is the case for multiple radio wireless mesh network.

The No Overhead Autoconfiguration [4] is an interesting proposition, however it can only be used in connection with OLSR routing protocol. Ancillotti et al. [3] propose an AH-DHCP protocol to assign a globally routable IPv4 address to the mobile nodes of a multi-hop WLAN using the DHCP-based mechanisms already implemented in the wired part of the network. The limitation of this solution is that it does not divide the network into subnets, thus can only work with routing protocol that spread routing information about every host. Another solution is proposed by Templin et al [6] - a layer of virtual enterprise traversal is added to allow automatic configuration of the network nodes, but it does not support automatic setup of wireless parameters for basic connectivity. The same limitation applies to LUNARng [12] protocol, which implements a virtual DHCP server to assign the IP addresses to the MANET nodes and use DNS names as a node identifier.

3. Description of the proposed solution

We propose an autoconfiguration algorithm based on DHCP servers with minor extensions. It is based on the following general assumptions:

- the autoconfiguration must provide IP addressing and basic radio parameters,
- the network is routed,
- nodes are equipped with one or more radio interfaces,
- packet forwarding can occur between interfaces or with use of the same (multi-access) interface,

Each mesh node should have at least one configuration profile providing some basics bootstrapping parameters. The profile should include:

- identification of mesh network that the node will connect to (e.g. by a mask on ESSID),
- authorization credentials.

The above two parameters are the only input required, the addressing and all other parameters being provided by the mesh network, although for the centrally managed system it can be manually inserted by the operator in the server.

The proposed method is based on DHCP protocol and uses BOOTP relays to forward DHCP protocol packets in a multihop network. The DHCP protocol was selected because it is relatively lightweight, easy to deploy, with numerous server and client implementations existing and will require only minor changes to support wireless mesh autoconfiguration. It is widely used and well tested. It is extensible by introducing new options or sending additional information as vendor specific options. We assume, that the DHCP server may be preconfigured by network operator, or is automatically started on one of the mesh nodes.

The IP address configuration of nodes having direct communication with DHCP server (within 1 hop distance) obtained by direct exchange of DHCP packets. For the autoconfiguration of further nodes the BOOTP relays are used to forward DHCP requests to the server. The relays are working on all nodes that take part in packet forwarding. Each relay learns the address of the DHCP server as the address of the server that provided IP addresses for the relay node.

3.1 Initial scanning

The goal of the initial scanning procedure is to find any reachable nodes in the mesh network. The mesh node must scan all possible frequencies, on all its radio interfaces in order to detect if there are any networks within range that match the mesh profile (e.g. the mask on ESSID). During scanning the node creates a list of

978-1-4244-4439-7/09/$25.00 ©2009 IEEE

available networks – a reachable networks with the ESSID of its interest.

After the initial scanning the mesh node will have also some basic information about the channels used by non-mesh traffic to give some input to the channel assignment procedures.

3.2 Connecting to the network

The basic task of the connection procedure is to try to join any possible mesh, set the IP address of the first interface and try to establish communication with a central management server. After the scanning procedure the node should try to connect to any of the discovered networks or simply try to connect to any SSID that appears to be part of a mesh network. The node starts the joining procedure with only one radio interface turned on. It should start from the networks with strongest signal level. If required, the node must authorize to the current network (with a pre-shared key), using the credentials provided within the profile. The configuration text file will include encryption type (e.g. WPA) and keys. For simple, self-managed networks the authorization procedure is not required.

At the beginning only one radio interface is used to get the connectivity. Then the DHCP protocol is used to obtain an IP address – the node should try to find a DHCP server by sending the DHCP Discover packet. When there is no response the node will retry 3 (or more) times and re-send the query. If there is still no response from DHCP server, this part of network should be considered a non-mesh and the node has to try another network, i.e. the one with the next-strongest signal level. When the node receives the DHCP Offer packet it checks for a vendor specific option with the information about mesh autoconfiguration. More details about the options are given in the 3.5 subsection. The node should reply with DHCP Request and wait for DHCP Ack, or else it must try to find another DHCP server by switching to another SSID.

When the first interface is configured the node starts DHCP procedure to get IP addresses for all the other interfaces. It is done by creating a "self-relay" for DHCP packets – i.e. packets with DHCP Relay option set, but which are created by the node itself. To inform the server that this is a query for address of another interface of the same node, the node puts a flag in vendor specific option in DHCP packet and adds the IP address of the first interface in the `giaddr` field as if it would be forwarded through a relay. The DHCP server may create a new subnet for these interfaces and wait for new clients to join, or to assign these interfaces to an existing IP cloud. It is done by setting the SSID, the IP address and subnet respectively.

When a node cannot find any mesh networks within range or it does not receive any DHCP Offer packets after scanning all possible SSIDs the node should switch to the distributed mode. It turns on all of its interfaces, configures them to separate channels and separate SSIDs, and boots up a DHCP server. It becomes a start-up point of a new mesh and is the server providing addresses for any new node willing to join the mesh.

3.3 BOOTP Relay Operation

Every node that is forwarding packets should work as a BOOTP Relay and forward DHCP packets from, or for some other nodes. The node should start the relay on every interface that takes a part in packet forwarding. Multiple relays within an ad-hoc cloud is not a problem for the protocol, since the packets are transmitted to the DHCP server using unicast and the protocol is capable of handling this situation, at the cost of slight increase of the autoconfiguration traffic overhead.

The BOOTP relay works with the configured DHCP server address (implicitly, the server which provided the IP to the node) and forwards all packets to the DHCP server using its address or unicast address of the DHCP server network. It must fill the `giaddr`, as required in the RFC 1542[7]. The DHCP server responds with a unicast packet addressed to the network as in the `giaddr` field. The transmission between the BOOTP relay agent and the DHCP server is supported by a routing protocol used in the mesh.

3.4 Starting new network with distributed management

The distributed start-up procedure is launched when a node cannot detect any operational mesh networks within its coverage on all radio interfaces. When a node does not sense any suitable mesh networks (or does not sense any signal, at all) or when it cannot join any of existing networks (e.g. because of security policies) it shall start a new mesh network. The node should turn on all interfaces and switch each of them to different frequency. The frequency selection should try to minimize interference. The automatic rate control should be used. Then the node should assign IP addresses to each interface, by selecting a different IP network address from a private range – eg. 10.0.1.0 for the first interface, 10.0.2.0 for the second and so on, all with the same network mask, e.g. 255.255.255.0. The assumption of no more then 254 in the host part can be made for most of the networks, however a bigger

978-1-4244-4439-7/09/$25.00 ©2009 IEEE

address pool can be defined if required by large networks.

The node that starts a new mesh network should bring up a DHCP server. In the vendor specific option within DHCP packets information should be propagated to indicate that it is a disconnected network. The node must assign address pools to advertise on each interface (from the same subnetworks as the addresses of those interfaces). The detailed procedure of DHCP server configuration is given in the chapter defining the centralized server. As the last part of the initial configuration the routing protocol should be started.

3.5. Additional functionality provided by the DHCP server

The proposed autoconfiguration algorithm requires some modification of the server implementation. New fields are added to DHCP packets as a vendor specific option: SSID, channel (providing information about 802.11 radio parameters to the nodes) and flags, used for marking a packet as "self-relayed" and for detection of duplicate DHCP server.

The DHCP server organizes the topology of the network by assigning IP subnets to which an interface should join. While it provides the IP addresses it can arrange which nodes should communicate with which directly. For the 802.11 networks it also provides the SSID and channel, dividing the mesh network into ad-hoc clouds. The topology management, in its simplest form, is the following: the first interface acquires an IP address from the range of the subnet that it has connected to and all other interfaces create new subnets, letting new nodes to join them. Distinct ESSID and channels are assigned to every interface of a node to increase the available bandwidth by reducing interferences between interfaces. This method guarantees connectivity but may lead to inefficient resource utilization. More advanced algorithms for topology management based on measurements may be developed, but are beyond the scope of this paper.

3.6 Periodical address updates

Connectivity failures can be detected by nodes through monitoring the received frame count on the radio interface. If it remains constant for a longer period (e.g. a couple of minutes) the node may assume that there is connectivity loss on the interfaces, and consequently to the network. In this situation the nodes should restart the autoconfiguration procedure to provide basic self-healing functionality.

Every node must also periodically check for connectivity with DHCP server from which it received the IP addresses (which was automatically started on the first node that created the mesh), to detect the partitioning or respond to failure of the initial node. It is done by applying the DHCP renewal procedure. The renewal timers must be adjusted to size of the network

3.7 Additional DHCP server functionality and algorithms

Some additional functions can be added to the autoconfiguration procedures by modifying the DHCP server topology management, and by integrating it with the network management, and collecting the information about radio parameters scanned:

- advanced topology management (by channel management and eventually signal strength adjustment)
- Signal strength control (to limit the interferences)
- QoS policies management
- Topology aware configuration updates
- Self-healing by changing the network topology or channel assignment in case of distortions experienced

These algorithms need further study (as some of them are np-complete problems the sub-optimal algorithms must be provided) and evaluation and may improve the efficiency and resilience to network failures, but are not required for the baseline autoconfiguration.

3.8 Detecting the DHCP server failure

The mesh nodes must detect the failure of the DHCP server to trigger autoconfiguration procedures in case of any server failure or upon network partitioning. In the distributed mode the DHCP server is automatically started on one of the nodes. If this node is not available a DHCP server should be started on another node and continue to provide new the addresses. The network addressing scheme should not be changed to sustain the connectivity – the new server should rather gather the information about current network configuration. The server can get the topology state from the DHCP renew requests.

The mesh nodes must periodically renew the IP addresses from the DHCP server. It is done by the DHCP Renew packets. Due to the lower reliability of the wireless links a mesh node should try 3 times to resend the renew packet before going to rebind state. If the server did not responded to any of these packets the node should assume that is down. In this case the node must go to the rebind state and start to broadcast DHCP Request messages. It should mark within the vendor

978-1-4244-4439-7/09/$25.00 ©2009 IEEE

specific option the address of the DHCP server. This packet, when received by the surrounding relays will inform them about the failure of the server. When a relay receives such a packet it should go to the renewing state and try to reconfirm its IP. It should not believe blindly in the information provided by other node, but it can try just one time to not increase the delay. If a relay does not receive a response it should go to rebinding state and broadcast its DHCP Request packets to all interfaces but that on which it received the first DHCP Request. While the relay is in rebinding state it should store all incoming DHCP Request and keep them in the cache. When it receives the response to DHCP Request all these packets should be sent to DHCP server.

3.9 Detecting the DHCP server duplication

It is possible that there will be two (or more) autoconfiguration DHCP servers within the mesh e.g. in case of bringing up adjacency between two networks or after a temporal communications failure between a DHCP server and some other nodes. In such case one of the servers should remain active and continue IP address assignment, while all others should be turned off. To detect the duplication each server should periodically send DHCP Discover packets, again with the additional flag added to the vendor specific options, and a sequence number. This packet should be resend by all nodes in the network to all interfaces except the one on which it has been received. Each node must store the most recent sequence number; if a packet has already been resent it should be ignored.

If a DHCP server receives this DHCP Discover packet it should response with DHCP Offer packet with the same flag marked. When the DHCP server receives the DHCP Offer with a flag it responds with DHCP Request only if its MAC address is higher than the MAC of the offering server (by simple numerical comparison). After getting the IP the DHCP server should terminate. To inform the mesh nodes about the existence of new DHCP server it should send the DHCP NACK packet to all of its leases with the IP of the new server added as an vendor specific option.

3.10 Sample address assignment

Below we describe the sample procedure of autoconfiguration of a small mesh network. The topology of the network and assigned addresses are presented in Fig. 1. In the example there are 3 nodes: A, B and C, each of them with 3 WIFI interfaces. Let's assume that node A was started first. It scanned for a mesh nodes within range, could not find any existing network, so it configured the network interfaces to private IP addresses 10.0.0.1, 10.0.1.1 and 10.0.2.1, thus creating 3 subnets. The interfaces were configures to SSIDs *mesh0*, *mesh1* and *mesh2* respectively. The DHCP autoconfiguration server was started automatically on node A. Then, node B was powered on. It discovered node A, connected with one of the interfaces to the *mesh0* subnet and asked for the IP. The DHCP server assigned the address 10.0.0.2 to the first interface of node B. As the next step, node B's requests for the IPs for the second and third interface were forwarded to the server. Two more subnets were created, and the 10.0.4.1 and 10.0.5.1 addresses were assigned, thus creating two new ad-hoc subnets: *mesh4* and *mesh5*. After that the DHCP Relay was started at node B. The third node that joined the mesh is the node C. Assume that it sensed the 10.0.4.1 network, as the interface with the strongest signal. The request for the IP was forwarded by the relay to and the 10.0.4.2 IP was assigned. When node C sent request for its second and third interface the DHCP server decided to create one new subnet (10.0.6.0) and join the second interface to the 10.0.2.0 subnet.

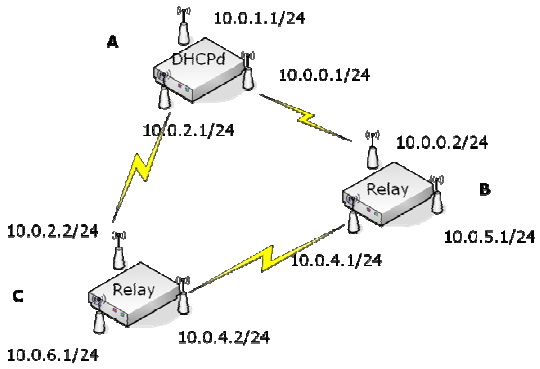

Figure 1. Sample address assignment

4. Performance considerations

The autoconfiguration procedures will generate control traffic in the network. The speed of DHCP protocol exchanges will depend on the network size and packet propagation and processing delay. In the following, we will evaluate average values for autoconfiguration delay and amount of traffic generated.

The average size of DHCP packet used for the proposed autoconfiguration schema is 130 bytes. The exact value depends on options like e.g. the domain name. A typical node join operation involves 4 transmissions of a packet per interface on the path from the new node to DHCP server. If we assume that every

978-1-4244-4439-7/09/$25.00 ©2009 IEEE

node has i interfaces it gives, $4i*1040$ bits transmitted to configure a node. In typical wireless 802.11g network with 54MBit/s datarate less than 1‰ of bandwidth is consumed by autoconfiguration if a node performs a join to the network every second.

The renew procedure requires transmission of 2 additional packets every renew interval. there will be additional 260*i bytes to transfer per every node. For example, in a network of 100 nodes with 3 interfaces each, with average distance to the DHCP server equal 5, when we assume the renew interval is equal 1000s, there will be additional 3900 bytes to transmit every second. The DHCP server duplication detection adds only a very small value to this, because the packet used for discovery are send only by one node - the DHCP server. In the above calculation we omit the MAC and PHY layer overhead, as they introduce only a small overhead

The DHCP procedure requires 4 packets to travel from the node to server and back. If we assume that the network is symmetric, the time required for the autoconfiguration procedure is equal to 4 times the time required to one of the DHCP packets to travel through the network. The average travel time of a packet is a sum of average one hop travel time multiplied by distance, the queuing time and the computation time required to process the DHCP packet. Thus the average time required for the autoconfiguration procedures to finish T_A may be calculated using the equation:

$$T_A = 4\left(DT_p + DT_q + T_c\right)$$

where

D is an average distance to DHCP server

T_p is a propagation delay between neighboring nodes

T_q is an average queuing time

T_c is computation time

As can be seen the distance D is bounded by the network diameter, which is growing less than the number of nodes n – the autoconfiguration time is growing much slower than the number of nodes.

5. Summary

We have presented a relatively simple autoconfiguration scheme for multi-hop, multi-radio wireless networks, based on the existing standards and well known technology. Te solution is suitable for both centrally managed IP address and parameter assignment, as well as for disconnected scenarios. The system requires some new, but simple protocol options. In the scheme both the DHCP server and client function are used. Some hints for performance evaluation have also been provided.

Acknowledgement

This work was supported in part by the European Commission in the 7th Framework Programme through project EU-MESH (Enhanced, Ubiquitous, and Dependable Broadband Access using MESH Networks), ICT-215320, www.eu-mesh.eu.

References

[1] R. Droms, "RFC 2131 – Dynamic Host Control Protocol", *http://www.faqs.org/rfcs/rfc2131.html*

[2] S. Cheshire, B. Aboba, E. Guttman, "RFC 3927 – Dynamic Configuration of IPv4 Link-Local Addresses", *http://www.faqs.org/rfcs/rfc3927.html*

[3] E. Ancillotti, R. Bruno, M. Conti, A. Pinizzotto, "Dynamic address autoconfiguration in hybrid ad hoc networks", *Pervasive and Mobile Computing*, Elsevier, doi:10.1016/j.pmcj.2008.09.008

[4] K. Mase, C. Adjih, "No Overhead Autoconfiguration OLSR", *Internet-Draft*, http://tools.ietf.org/html/draft-mase-manet-autoconf-noaolsr-01

[5] J. Reibel, "An IP Address Configuration Algorithm for Zeroconf. Mobile Multi-hop Ad-Hoc Network", *Proceedings of the International Workshop on Broadband Wireless Ad-Hoc Networks and Services*, Sophia Antipolis 2002.

[6] F. Templin et al., "Virtual Enterprise Traversal (VET)", Internet Draft, February 2009, *Internet-Draft*, http://tools.ietf.org/html/draft-templin-autoconf-dhcp-33

[7] W. Wimer, "RFC 1542 – Clarifications and Extensions for the Bootstrap Protocol", *Internet Standard*, http://www.ietf.org/rfc/rfc1542.txt

[8] E. Hossain, K. Leung, *Wireless Mesh Networks*, Springer 2008

[9] D. E. Eastlake (ed.), "IEEE P802.11s Draft STANDARD for Information Technology D1.09", IEEE 2008

[10] K. Weniger, "PACMAN: Passive autoconfiguration for mobile ad hoc networks", *IEEE Journal on Selected Areas in Communications*, March 2005.

[11] N. Vaidya, "Weak duplicate address detection in mobile ad hoc networks", *Proceeding of ACM MobiHoc* 2002, Lausanne, Switzerland.

[12] C. Jelger, C. Thusdin, "Dynamic Names and Private Address Maps: Complete Self-Configuration for MANETs", *Proceedings of the 2nd Conference on Future Networking Technologies*, December 2006, Lisboa, Portugal.

978-1-4244-4439-7/09/$25.00 ©2009 IEEE