Demonstration of automatic address and radio parameters assignment in MANET using DHCP protocol extensions

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

Abstract

We propose procedures for autoconfiguration of multihop wireless ad-hoc networks based on DHCP protocol. The proposed mechanism provides automatic address and radio parameters set-up for nodes with multiple radio interfaces, allowing to fully automatically form and to start a wireless mesh network and to route packets. The autoconfiguration schema was implemented and tested on OpenWRT platform. The demonstration shows how multiple Mikrotik routerboard with three 802.11g interfaces can fully automatically create a multihop wireless network, assign IP addresses, radio channels. It will also include the automatic visualization of network topology by network management software.

1. Motivation of work

Our goal was to create autoconfiguration component for wireless multihop networks that provides a method for automatic network start-up, joining a node into an existing network 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 the 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 transmission in case of link or node failure.

Although there exists some autoconfiguration algorithms designed for MANET networks [1-6] we were not able to find a procedure that fully automatically assigns the radio parameters (like channels and SSID for 802.11 networks) together with IP addresses, divides the network into IP subnets corresponding to the ad-hoc networks created and is independent of routing protocol.

We propose novel autoconfiguration procedures, based on extensions to DHCP[8] server and client implementations and BOOTP[7] 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 the autoconfiguration should 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 fully automatic network setup. The DHCP protocol is extended by adding some additional options required to configure the MANET that are transferred as vendor specific extensions and some modifications to the packet relaying procedure.

The proposed autoconfiguration schema was implemented and tested in laboratory network on Linux OpenWRT[9] platform and routerboards platform manufactured by Mikrotik.

2. Description of proposed autoconfiguration procedures

Each node has preconfigured a profile, providing information about authorization required to join the MANET (eg. WPA keys), interfaces to be used and SSID mask or other method to select radio parameters. After boot-up the node starts to scan for a networks in range. It creates a list of possible networks to which it can try to connect that are compatible with its profile.

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 join procedure with

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

only one radio interface turned on. It should start from the networks with strongest signal level. 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. If there is no response from DHCP server, this part of network should be considered a not compatible and the node has to try another network, i.e. the one with the next-strongest signal level.

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.

Every node that is forwarding packets should work as a BOOTP Relay and forward DHCP packets from, or for some other nodes. 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.

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. All these parameters are transferred using vendor specific extension added to DHCP packets. The channel assignment algorithm is centralized and works on the DHCP server. 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. This methods guarantees connectivity but may lead to inefficient resource utilization. More advanced algorithms for topology management based on measurements of perceived may be developed, but are beyond the scope of autoconfiguration procedures.

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. The mesh nodes must periodically renew the IP addresses from the DHCP server. It is done by the DHCP Renew packets. If the server did not respond to the renew request the node starts to scan for another DHCP server in the network. It is done by flooding the network with DHCP Discovery packets. When there is no response to Discovery flooding the node should start DHCP server functionality. Then it responds to renew request from other nodes, assigning the same IPs as were before and gathering in this way the state of the network.

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. A procedure to select and disable one of the servers is implemented.

3. 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.

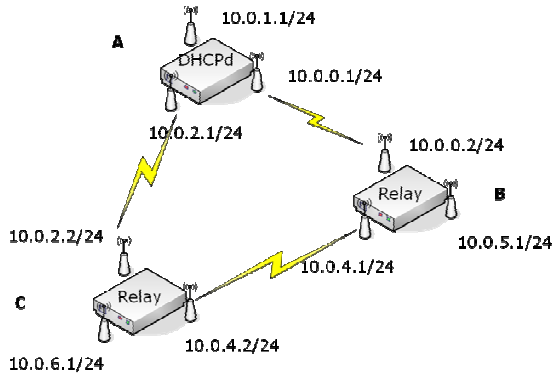

Figure 1. Sample address assignment

4. The demonstration

The main goal of the demonstration is to provide the audience with proof of the correct behavior of proposed enhanced DHCP scheme, but it will also show the time required for the autoconfiguration procedures to finish and visualize the integration of autoconfiguration procedures with network management software.

The demonstration set-up will consist of 4 Mikrotik router boards with three 802.11g interfaces each and a laptop computer. All the equipment for the demonstration will be delivered by Proximetry. To perform the presentation two tables with access to electricity sockets (220V) are required to place the equipment. An Internet connection (via WIFI or Ethernet) would be a plus.

The OpenWRT software together with autoconfiguration agent will be installed on the routerboards. The laptop computer will serve as a autoconfiguration and network management server. The DHCP server with autoconfiguration extensions will be installed on the laptop together with AirSync[10] NMS system. The AirSync software will visualize the address and radio parameters configuration and network topology, to provide the instant view of the network state for the audience.

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

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

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

[9] Linux OpenWRT project, *http://openwrt.org/*

[10] AirSync network management platform, *http://www.proximetry.com/product/airsync.html*