

A Survey of Zero and Auto Configurations for Wireless Networks

Zhe Guang Zhou⁺ and Aruna Seneviratne⁺⁺

+ School of Electrical Engineering and
Telecommunications
The University of New South Wales, Sydney, Australia
zheguang@mobqos.ee.unsw.edu.au

++ School of Electrical Engineering and
Telecommunications
The University of New South Wales, Sydney, Australia
a.seneviratne@unsw.edu.au

Abstract – Nowadays, many people have several mobile devices with them and frequently move in and out of small adjacent networks. They also join and leave networks frequently by simply switching their devices on and off. Configuration and reconfiguration of mobile devices are critical processes for these users, due to the different specifications. In order to provide an “easy-to-use network”, future research will focus on the auto-configuration for network systems. This paper will discuss a survey on current research of auto/zero configurations. It will summarize and discuss each of these systems developed today and identify the common designs in context.

Keywords: wireless networks, auto/zero configurations

I. INTRODUCTION

Wireless network become more and more popular with the availability of IEEE 802.11 standard and relatively cheap equipments which conform to it. Today accessing the Internet becomes part of the necessary daily activity for people and they have several different mobile devices with them. Following the broadband technologies being introduced, such as the Cable Modem and ADSL, the structure of the network is not only configured as the large infrastructure such as the LANs and WANs, but also small area networks, like home networks, small business networks and office area networks. Organizations including both large companies and departments, have established their proprietary network to provide intranet accessibilities for internal information sharing and communications. In addition, the network access technology has also changed from wired to wireless, due to convenience in mobility, sufficient bandwidth and high data rate. Not only do the people use the network in offices, but also in private areas such as home area. They often need to carry their works along with different mobile devices which may use different wireless network interface cards. This causes inconsistency of network configurations. Due to the nature of mobility, mobile users are frequently moving in and out of some small adjacent networks. They also join and leave the networks frequently by simply switching their devices on and off. Therefore, installation, configuration and reconfiguration are necessities for mobile users.

Traditional configuration management of IP network is centralized using servers such as DHCP server, DNS, MADCAP. Configuration and maintenance of these server systems are complex and labour-intensive, which often require network administration expert(s). However, manually configuring and maintaining these systems are error prone and inefficient. Network failures are unpredictable, due to human errors. The configuration time of these systems is long and costly. When the configuration changes, there is a limited speed for the mobile devices to perform recovery and reconfiguration. The problem of keeping configuration consistency arises. Furthermore, in small home-area network and office-area network, network system administration may not be feasible. It is impractical to assume individuals have a sufficient understanding in networking in terms of IP address, subnet mask, default gateway and DNS etc. Manually configured network may reach its limit in the foreseeable future, due to the lack of network knowledge of the users. Therefore, future networks will provide some mechanisms for automatic or zero configuration for the network devices. The current labour-intensive and high cost network is likely be replaced by the auto-configured network.

This survey is motivated by the rapid evolution of the wireless network technologies with researches conducting solutions based on self-configuration, auto-configuration, self-organization, etc. As the number of mobile users increase, the commercial demand of self-configurable mobile device will also increase. This survey presented here serves to capture a snapshot of current design trends and techniques. The purpose is not to compare one solution with another, but to identify the common designs and put them in context.

The rest of this paper is organized as follows. Section II presents the basic requirements for establishing a zero configurable network. Section III discusses the protocols that modify the current DHCP server. Section IV presents the auto-configuration in ad-hoc networks. Section V discusses the configuration management systems and Section VI is the conclusion of this paper.

II. ZERO CONFIGURATION NETWORK REQUIREMENTS

There are several essential configuration parameters required by an IP host. A proper IP address is the first

essential configuration parameter for enabling the host on the network. Static IP address configuration is inefficient and not a good solution, since configuring IP address manually involves not only typing the numbers, but also needs to ensure the IP address is not being used. Most IP devices have DHCP client implementations which require a centralized DHCP server on the network. The subnet mask is another parameter for the IP host.

Moreover, router and DNS are two fundamental elements for configuring a simple small network. In traditional network, they are required to be configured manually. The IETF Zero Configuration Working Group (ZCWG) proposed an Internet draft of Zeroconf IP host requirements [12] which is used to establish zero configured (Zeroconf) home area networks. The Zeroconf protocols are mainly used while the centralized services are absent. For example, no DHCP, DNS, MADCAP, LDAP are in the network. These requirements are specified in four different topics [4,11]:

1. IP interface configuration
2. Translation between host name and IP address
3. IP multicast address allocation
4. Service discovery

These requirements attempt to improve network layer (Points 1 and 3) and application layer (Points 2 and 4). The primary purpose of the IETF Zeroconf network is to provide an “easy-to-use” network and make network configuration changes transparent to mobile users. We will briefly discuss these four requirements in this section.

Firstly, the IP interface configuration is the essential problem for configuring a small simple network. When mobile devices (i.e. mobile hosts or mobile nodes, MNs) move between different networks, they need to use different IP addresses correspondingly. ZCWG requires the mobile device with automatic IP address configuration when there are no DHCP, MAP servers available. In IPv4, the requirements are in terms of configuring appropriate net mask, unique IP address, resolving IP address conflict etc. In IPv6, the solution exists already since it is able to auto configure the IP address using the link-local IP address auto configuration [14].

Secondly, numerical IP address is a very poor representation of the destination host. Translation between the numerical IP address and the domain name in alphabet is also an interesting research topic. In this requirement, there is no difference between IPv4 and IPv6. Currently, DNS is the solution for the traditional centralized networks. This, however, is not suitable for the Zeroconf environment since the DNS may be absent. Moreover, the translation of IP address and host name is a bi-directional mapping. When MN connects to different networks due to the mobility of the device, the mapping will be difficult since different IP address will be used. However, the host name remains the same while the MN moves. The difficulty is not in the unique configuration of the IP address, but the host name can be easily reused, because it could exist already in the new IP network. Therefore, this conflict must be solved during the configuration phase. Furthermore, when the MN is

corrupted, its host name should be reallocated to other new MN. A similar solution to Time-to-Live (TTL) is needed to solve this problem.

Thirdly, the requirements for IP multicast address allocation are that the protocol selects a multicast address and prevents the conflicts. Currently, the Internet draft of Zeroconf Multicast Address Allocation Protocol (ZMAAP) [9] provides the solutions for those requirements which is a peer-to-peer protocol for multicast addresses allocation.

Finally, the service discovery requirement is for discovering the services in the current network such as printer service etc. Service Location Protocol (SLPv2) [10] is one of the solutions being used. It is defined to operate at administrative local scope for IPv4 and site scope for IPv6. It is designed to scale up to a single administration as it uses administrative-scope multicast. DNS SRV [1] is another possible solution for service discovery. The MNs look up the services via the DNS. ZCWG specifies the requirement for the services to be discovered without the use of a service-specific protocol. Therefore, these two protocols may not be the good solutions for this requirement.

Since configuring a proper IP address is the essential process for the IP hosts, this paper will focus on the IP interface configurations. In the following sections, we will discuss each system and outline the common designs.

III. MODIFICATION OF DHCP

In IP address configuration, there is no difference between the wireless network and wired network. In the past few years, considerable efforts have been devoted to develop protocols for auto configuring MNs. These works can be broadly categorized into the modification of existing DHCP and ad-hoc auto-configuration. In this section, we will discuss the modified DHCP systems and the ad-hoc systems will be discussed in next section.

Traditional network architecture uses a centralized DHCP server for allocating IP address to the requesting MN. Several proposals are put forward to modify the existing DHCP server to achieve auto-configuration functionality.

A. Enhanced DHCP Server (DRCP)

Maculey et al [2] proposed to use a Dynamic Registration and Configuration Protocol (DRCP) which is an enhanced version of DHCP server. Some additional features are included on top of the current DHCP technology. They stated that in the current DHCP mechanism, the message size and the number of messages are larger than it needs. Moreover, there is no configuration-change-recovery mechanism. They proposed that several additional messages are used to extend the DHCP mechanism. Five messages are used from a DRCP client (i.e. the network device) and four messages are sent to the client from the DRCP server. The DRCP server inter-operates with the original DHCP server. It configures the new client by advertising itself periodically. The DRCP client detects

the advertisement message and then requests for an IP address. The DRCP server continues to transmit an “offer” message until it gets an “accept” or “decline” message from the client. The client also continues to transmit a “discover” or “request” message until it gets an “offer” or “ack” message from the DRCP server. Obviously, these messages are the overhead compared to the original DHCP configuration. However, with the combination of DRCP and their proposed Dynamic Address Allocation Protocol (DAAP), domain auto-configuration is achievable as follows. In traditional DHCP mechanism, the IP address is allocated by using one single IP address pool. They proposed that any device within the same domain is able to act as the DHCP server which reserves a small IP address pool for address allocation. A device is able to switch itself from client mode to server mode using DAAP. For example, a node A configured itself by using DRCP with an address pool which is a range of available IP addresses. Then node A is in server mode. Suppose node B has two network interfaces, it configures the IP address of one of these two network interfaces from node A using DRCP. It then requests an address pool from node A which sends half of its address pool to node B. Then node B switches to server mode and it is able to configure the remaining network interface and supply the addresses for the new incoming network devices.

B. DCDP

Misra et al [6] proposed a mechanism, Dynamic Configuration and Distribution Protocol (DCDP), that automates the distribution of IP address pools to a hierarchy of DHCP server. Their algorithm is very similar to [2] which proposes the combination of DRCP [3] and DCDP to perform the network auto-configuration. Initially, one node in the network runs DCDP and contains a range of available IP addresses in the address pool. When the new node joins the network, DCDP does not allocate the IP address for the new node immediately. It splits its IP address pool into two parts; the first part is for the initial node and the second part is forwarded to the node which is directly connected to the new node. Hence the IP address of the new node can be allocated by choosing any one address from the second part of the address pool. The DCDP distributes the address pools to each link of the network that connects the network devices by recursively splitting the address pool down the distribution hierarchy.

C. Mini DHCP Server

Many companies are developing wireless network technologies in the current market. Each of these vendors has its own strengths in terms of QoS, power consumption and coverage area towards its own targeted application demands. Customers will choose their network devices depending on their needs. They may add different devices designated to their needs in their sub-networks and may join them at a particular time. Thus, the future small area wireless networks will consist of multiple small segments and platform technologies that

are integrated through a set of gateways and shared devices. In order to auto-configure these devices in multi-segment Zeroconf networks by a single router, [7] suggested to run multiple mini-DHCP servers in the routers for those small sub-networks.

The mini-DHCP server in each router for IPv4 should be able to activate or deactivate itself upon lost or reach contact with an existing mini-DHCP server as follows. The mini-DHCP server sends out the “dhcpdiscover” message periodically to detect the existing DHCP server. If it gets a positive response, it deactivates itself and turns on its timer. After time out, it sends out the “dhcpdiscover” message again for DHCP server removal detection. If the removal is detected, then it activates itself. Otherwise, the timer is reset. For IPv6, the solution exists already since it allows a MN to select an appropriate IP address and subnet mask using available routing information.

IV. AD-HOC NETWORK AUTO-CONFIGURATIONS

The systems mentioned above either modify the existing DHCP, or insert a mini-DHCP server in the routers. For the former case, the problem could be the IP address releasing. For instance, if one of the server MNs failed or corrupted, the IP address pool cannot be released for new MNs. For the latter case, the mini-DHCP server is still centralized by the particular router.

According to the requirements of ZCWG, the Zeroconf protocols are used whenever the centralized servers are absent. Therefore, it is necessary to look at the research in ad-hoc network since it is aimed for the mobile nodes communication without any network infrastructure.

A. IP Auto-configuration for Ad-hoc Networks

Perkins et al [8] proposed a configuration scheme for auto configuring the IP address in Ad-hoc networks for both IPv4 and IPv6.

For IPv4, when a new MN has joined the network, it randomly picks up one IP address from the range of 169.254/16 [16] as the requesting IP address. Then it also randomly picks up one IP from the range of the temporary source address pool as its source IP address. After that it broadcasts the message, AREQ, which includes the request IP address for asking the neighbour nodes whether they are using that requesting IP address or not. Meanwhile, it turns on its timer. It will send out several broadcast messages if time out occurs in the case of message lost. If it still cannot receive any reply message, AREP, from the neighbourhoods, then it assumes that the requesting IP address is unused. Thus it uses it as its own IP address and release the temporary source IP. In the case of the requesting IP address is used, it will receive a notification. Then the new MN will randomly pick up another IP address and repeat the above process.

One of the problems needed to be considered is the hidden node. For instance in Figure 1, MN A is the new node that just joined the network and broadcasts the IP

request message. However, an existing node, MN C, may use the same IP as the requesting IP and MN C is not in the radio coverage of MN A, it is not able to receive the broadcast message from MN A. This IP conflict problem is solved by MN B acting as an intermediate node. When MN B receives the second broadcast message from MN A, it propagates the message to its neighbours.

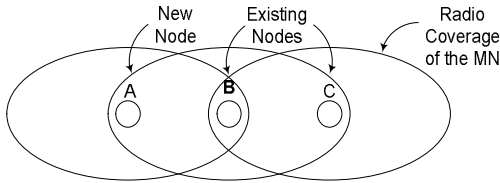


Figure 1: Hidden Node Problem

For IPv6, the additional messages, AREQ and AREP are the modified versions of the original messages, the Neighbour Solicitation and Neighbour Advertisement, respectively. The difference between IPv4 and IPv6 is that in IPv6 the MN performs the auto-configuration process by obtaining a non-link-local prefix. It also uses a temporary IP address as its source address for broadcasting the address request message. The rest of the procedure is the same as in IPv4.

The time delay is one of the potential problems in the above scheme. The new MN needs to wait for one (single hop case) or more (multi-hop case) time out periods for the replying from the nodes in the network if the requesting IP address is not used. Moreover, when two different MNs request the IP address simultaneously, a collision may occur for them to pick up the same temporary IP address as the source address. This may easily happen if there is more than one intermediate node between these two requesting nodes and they cannot directly communicate with each other.

B. MANETconf

Nesargi et al [15] proposed a distributed dynamic host configuration protocol designed to configure nodes in mobile ad-hoc networks.

They stated that the requesting IP address block, 169.254/16, used in [8] is not a good range since this block is registered with IANA (Internet Assigned Numbers Authority) for link-local unique addressing. Therefore, the MNs cannot communicate since each node will act as a router. Thus, they assume the MNs in their system use the following IPv4 address blocks: 10.0.0.0–10.255.255.255, 172.16.0.0–172.31.255.255, 192.168.0.0–192.168.255.255. These IP blocks are equally applicable for IPv6 address space.

The IP address allocation process in MANETconf system is described in the following. The new MN that requests the IP address is called a requestor. It broadcasts the request message to its neighbours. If the requester is the only node in the network (i.e. timed out), then it becomes an initiator. If it receives a reply message (i.e. at least a node is in the network), then it selects one of the reachable node as the initiator, which performs the

address allocation. The initiator works as a proxy server. All other MNs know the routing path to communicate with the initiator but may not have a direct communication path to the requestor. The initiator selects an IP address and floods the requesting message to all other MNs. If the IP address is used, then the initiator selects another IP address and repeats the process, otherwise, it allocates this IP address to the requester.

It is possible that two initiators are attempting to allocate the same IP address for two different requestors simultaneously. In order to solve this problem, priority of the lower IP address (smaller number) of the initiator is used to prevent the conflict between two initiators.

All MNs in MANETconf must be powerful in terms of computing capabilities since any one of them could be the initiator. In addition, since the initiator is also a MN, the mobility of the initiator will cause the request process failure in half way. This will result in the problem of initiator migration. The authors provide a solution that the requester selects a new initiator. In this case, however, the old initiator will need to forward the current status to the new initiator. Furthermore, all other MNs that without a direct path to the new MN will need to be notified about initiator change and update their routing tables. The solution for this problem was not clearly explained.

V. CONFIGURATION MANAGEMENT SYSTEMS

Apart from the above systems, several researchers have made an attempt to achieve auto-configuration and maintain the configuration information in higher layers. There are two types of self-configuration system available for the configuration management. One is directory-based and another is policy-based. The directory is a special database designed for fast information lookup, while the policy-based system passes the configuration policy to the MNs for determining the configurations.

A. Directory-enabled System

Boutaba et al [13] proposed a architecture, SELFCON, for self-configuring networks. This architecture is a directory-based system. A directory server is used to maintain the configuration information. The network elements and services are modelled by the Directory-enabled network (DEN) specification. The standard Lightweight Directory Access Protocol (LDAP) is used for the communications between the directory and network elements.

Each network device, which is called a self-configurable network element, has three entities in it, the LDAP client, the notification unit and the Local Decision Point. The network elements are registered in the directory server. In the initialization phase, the network element downloads the initial configuration information from the directory server using the LDAP communication. In order to keep the consistency of the configurations, the directory server will notice the network elements if any changes occur.

B. NESTOR

The NESTOR system [5,17] is a self-managed and organization network architecture, a combination of directory-based and policy-based system. The two layers, self-configuration management and configuration modelling, are inserted between the application layer and the network layer. NESTOR uses protocol proxies instead of the real DHCP server. These protocol proxies communicate between the hosts and the real DHCP server. The network elements are monitored by the configuration modelling layer.

A resource directory server (RDS) is used in NESTOR for maintaining the object repository that stores the network element configuration information. Directory Access Protocol (DAP) is used as the communication protocol. NESTOR models the network by using the object-oriented programming language, Resource Definition Language (RDL). It establishes the configuration management by using the policy scripts. The self-configuration management layer is used for handling the policy constraints and maintaining the consistency of the configuration changes.

VI. CONCLUSION

The idea and demand of auto and zero configurations become more and more popular since it simplifies the labour-intensive network administrations. By surveying previous researches on automatic and zero configurations, we notice that the common idea of automatic or zero configuration networks is to upgrade the existing network to a "plug-and-play" network. Each of those system presented in this paper has its own strength and weakness. However, the common idea of these systems is to provide an easy-to-use network for the network users. The IETF specifications have a very limited idea and coarse-grained solutions for starting the development of zero configured networks. This survey mainly focused on the IP address auto-configuration system, since it is the essential process for devices connecting to the network. Our survey discussed about the centralised systems, i.e. modification of DHCP servers, the distributed systems, i.e. ad-hoc auto-configuration and the configuration management systems.

From our survey, it can be seen that there remains a number of open issues for researchers. The research frontier in automatic or zero configurations will be rely on developing suitable protocols more efficiently, robustly and scalable to configure the MNs onto small and administration absented networks.

ACKNOWLEDGEMENTS

This work was carried out through an Australian Research Council Linkage Postgraduate Research Award with Singtel Optus Communications, Australia. The authors also wish to thank Chak Tong Kam and Nga Man Kam for reviewing this paper.

References

- [1] A.Gulbrandsen, P.Vixie and L.Ebisov, "A DNS RR for Specifying the Location of Services (DNS SRV)", RFC2782, 2000.
- [2] A.J.McAuley and K.Manousakis, "Self-configuring Networks", 21st Century Military Communications Conference Proceedings, 2000.
- [3] A.McAuley, "Dynamic Registration and Configuration Protocol", draft-itsumo-drcp-00.txt, Work in progress, 2000.
- [4] A.Williams, "Requirements for Automatic Configuration of IP Hosts", draft-ietf-zeroconf-reqts-12.txt, Work in progress, 9-19-2002.
- [5] Alexander V.Konstntinou, Danilo Florissi and Yechiam Yemini, "Towards Self-Configuring Networks", In Proceedings of the DARPA Active Networks Conference and Exposition (DANCE.02), 2002.
- [6] Archan Misra, Subir Das, Anthony McAuley and Sajal K.Das, "Autoconfiguration, Registration, and Mobility Management for Pervasive Computing", IEEE Personal Communications, 2001.
- [7] Bernard Aboba, "Auto-Addressing in Multi-segment Networks", draft-aboba-zeroconf-multi-00.txt, Work in progress, 1999.
- [8] Charles E Perkins, Jari T Malinen, Ryuji Wakikawa, Elizabeth M Belding-Royer and Yuan Sun, "IP Address Autoconfiguration for Ad Hoc Networks", draft-perkin-manet-autoconf-01.txt, Work in progress, 2001.
- [9] Dave Thaler, Bernard Aboba and Erik Guttman, "Zeroconf Multicast Address Allocation Protocol (ZMAAP)", draft-ietf-zeroconf-zmaap-02.txt, Work in progress, 2003.
- [10] E.Guttman, C.Perkins, J.Veizades and M.Day, "Service Location Protocol, Version 2", RFC2608, 1999.
- [11] M.Hatting, "Zeroconf IP Host Requirements", draft-ietf-zeroconf-reqts-09.txt, Work in progress, 8-31-2001.
- [12] M.Hatting, "Zeroconf IP Host Requirements", draft-ietf-zeroconf-reqts-09.txt, Work in progress, 8-31-2001.
- [13] Raouf Boutaba, Salima Omari and Ajay Pal Singh Virk, "SELFCON: An Architecture for Self-Configuration of Networks", IEEE Journal of communications and networks, 2001.
- [14] S.Thomson and T.Narten, "IPv6 stateless Address Autoconfiguration", RFC1971, 2001.
- [15] Sanket Nesargi and Ravi Prakash, "MANETconf: Configuration of Hosts in a Mobile Ad Hoc Network", In Proceedings of INFOCOM, 2002.
- [16] Stuart Cheshire, Bernard Aboba and Erik Guttman, "Dynamic Configuration of IPv4 Link-Local Address", draft-ietf-zeroconf-ipv4-linklocal-07.txt, Work in progress, 8-23-2002.
- [17] Yechiam Yemini, Alexander V Konstntinou and Danilo Florissi, "NESTOR: An Architecture for Network Self-Management and Organization", IEEE Journal on selected areas in communications, 2000.