Zero Configuration Networking

Tim Niemueller
tim@niemueller.de

Abstract: In today's mobile world inter-machine communication has become the key to information exchange. Until now at best networking was an annoyance, at worst it was a show stopper. Zero Configuration Networking aims to improve this by defining a set of protocols that can be used to assign IP addresses automatically, resolve names and discover services.

1 Introduction

In today's mobile world inter-machine communication has become the key to information exchange - be it file transfers, chats or web surfing.

Imagine an average business traveller who is about to attend a conference important for his company. There are several tasks he faces involving network communication. Back in the company he needs to copy the slides created on the workstation to his laptop and print them out. Later at the airport he meets a company fellow who wants to copy the newest statistics that should be incorporated into the slides. At the conference questions can be send via a chat program to a discussion panel. The beamer accepts image streams from the notebook so no cable connection to the beamer is required.

Until now in an ad-hoc network with a few peers you had to set most information manually. You agreed on an IP subnet and exchanged information about the services. At best networking was an annoyance, at worst it was a show stopper.

There comes Zero Configuration Networking into the arena (ZeroConf). ZeroConf claims to solve these issues. It describes a set of standards (or proposed standards) that will make networking an experience. It includes how basic addressing settings are negotiated, names are resolved and services discovered on the network.

2 Instant Networking

For two hosts to be able to communicate with each other the basic information they need is addressing information. On the physical layer these addresses are fixed. But logical IP addresses have to be configured. The *Internet Protocol (IP)* is the basis for almost all modern networking. In this document we will look at IPv4, version 4 of the IP.

In the given scenario configuring the IP settings of the workstation and the laptop in the company is no problem. This is done via the Dynamic Host Configuration Protocol (DHCP) where a central server distributes the configuration data. The same applies to the conference where there is usually a managed network in place.

But when the two company fellows meet at the airport there may not be a central infrastructure. So they form a spontaneous ad-hoc wireless network. Before ZeroConf IP addresses where configured manually by agreeing on a given IP subnet usually in a private range assigned by the Internet Assigned Number Authority (IANA).

Today this problem can be solved using *Link-Local IPv4 addressing* defined in [CAG05].

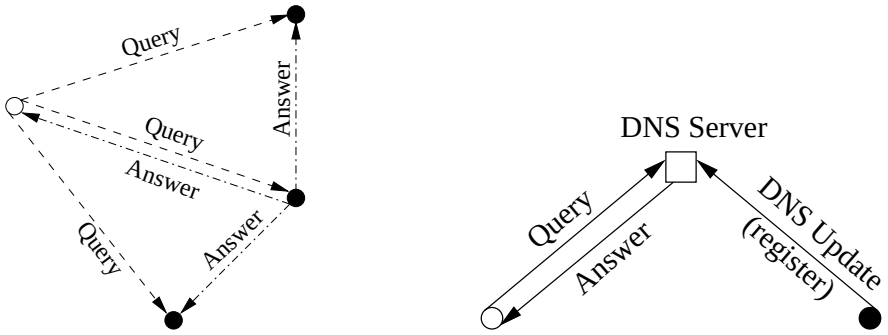
The local link defines a local, closed and interconnected set of machines. Two machines are on the same local link if exchanged packets using unicast, multicast or broadcast that arrive with unmodified link-layer package payload and if a broadcast sent by one station can be received by all other stations. This implies that there are no routers on the path a packet takes.
Every machine joining a network takes an IP address from a uniform random distribution over the range 169.254/16 assigned by IANA. Then it checks using the Address Resolution Protocol (ARP) if that address has already been taken. This is done as long as there are IP addresses left and a conflict happened. After that the host has a valid IP configuration. This scales well to about 1300 hosts where the chance of selecting a valid address after two tries rises to 99.96%.
So by using this method the two laptops would take an IP from the same subnet automatically because there is no DHCP server on the ad-hoc network and after that the machines can communicate and send data.

3 Name Resolution in Ad-hoc Networks

It is inconvenient for users to remember IP addresses of hosts and in general it is a lot easier to read host names instead of IP addresses in configuration files. This mapping from addresses to names and vice versa is accomplished via the Domain Name System (DNS).
In the corporate and conference networks where there are DHCP servers there is usually also a DNS server. In the company this may even map the laptop's IP address to the correct name, in the conference network this is very unlikely, as is in the ad-hoc network at the airport. So for these cases we need a new peer-to-peer approach that hosts can resolve the names on the local network without a central server.
This is accomplished with *Multicast DNS* as defined in [CK05b]. It proposes a slight change how DNS is used – via multicast networking. It describes what has to be taken care of if DNS responders start sending and answering queries via multicast networking. Multicast means that a packet sent by one station is received by a group of hosts on the network. You put packets in at one end, and the network conspires to deliver them to anyone who asks.
The basic idea is to have a new top-level domain called .local. In this space all names are freely available. In a managed network a machine has a fully qualified domain name (FQDN) like laptop.example.com. On the local link the machine takes a name in the .local domain. Whenever a machine joins a network and has an IP address it claims a name. For this it queries for the desired name. If there is an answer another host already has claimed that name and the conflict has to be resolved by choosing another name. When a name was found that has not been taken the station announces the name and thereby claims the name as its own.
Queries are then sent to the local link via multicast networking. Every station answers for the records it is authoritative for (for instance the address record for its name), again via multicast so all stations on the network can cache this information. Known answers are sent with the query to reduce the traffic on the network.
A difference to regular DNS is how failures are detected. A DNS server responds with an error if the record did not exist (like NXDOMAIN, not existent domain). With mDNS this is not the case. The station has to deduct from not getting an answer after a given number of tries that there is no such record available.

4 Dynamic Service Discovery

The number of services offered in networks grows rapidly. The traditional way was to configure each service explicitly on each host.



(a) DNS-SD over mDNS

(b) DNS-SD with DNS server

Figure 1: Service discovery 1(a) on an ad-hoc network, 1(b) on a managed network, • Service, ○ User station, □ DNS Server, – Unicast, - - Multicast Query, - · - Multicast Answer

In our scenario this works for the corporate network. But at the airport and at the conference this does not work. The user does not know about services like the file serving facility for copying files, the chat for asking questions on the discussion panel or using the beamer before he actually starts using the service.

So an approach is needed to discover services as they are required. Service discovery greatly simplifies the task of finding and utilizing services on a network.

ZeroConf proposes *DNS-based Service Discovery (DNS-SD)* (confer [CK05a]) to solve this very problem. It is build on top of mDNS for ad-hoc networks, but can also be used with conventional DNS servers to work in managed networks.

DNS-SD describes a convention for naming and structuring DNS resource records (RRs). Given a type of service that a client is looking for, and a domain in which the client is looking for that service (.local in the case of an ad-hoc network), this convention allows clients to discover a list of named instances of that desired service, using only standard DNS queries.

The returned list is specific to the service requested by the user and the application. In the office and at the airport the user would request a file service. At the conference there may be an iChat system in place for the discussion panel. The beamer would offer a kind of remote display service via DNS-SD that the presentation software of the business man would search for. So for all those services the user just searches for the service and does not configure them.

5 Conclusion

We presented all major components needed for a Zero Configuration Networking – IP address configuration, name resolution and service discovery.

With ZeroConf the business traveller could have used all services without knowing anything about IP configuration and the network specific details of the services. The application just has to search for the services applicable for the current task and if there is more than one present a list to the user to choose from.

This techniques can also be used in a corporate network to administrate a set of services in a DNS zone so that the administrator does not have to configure the services on each workstation but only once on the server (see figure 1).

ZeroConf is about searching and using services, and not about configuring them.

References

[CAG05] Stuart Cheshire, Bernard Aboba, and Erik Guttman. Dynamic Configuration of IPv4 Link-Local Addresses. RFC 3927, Internet Engineering Task Force, May 2005.

[CK05a] Stuart Cheshire and Marc Krochmal. DNS-Based Service Discovery. Internet-Draft, Internet Engineering Task Force, June 2005.

[CK05b] Stuart Cheshire and Marc Krochmal. Multicast DNS. Internet-Draft, Internet Engineering Task Force, June 2005.