

# Service advertizing with Avahi Datto Infrastructure

• Fred Mora

# Agenda

2

- **Problem to solve**
- **Solution**
- **Apple's Bonjour**
- **Linux Avahi**
- **Dynamic service definition**
- **Static service advertizing**

# Problem to solve

- **Ad-hoc installations don't want to deal with DNSes. Examples:**
  - Home with several machines and devices
  - Test lab
- **Yet, dealing with IP addresses is:**
  - uncomfortable for consumers
  - a sign of unpreparedness for pros – what do we do in IPv6 environment, spend our life typing 128-bit IPs?
- **Additionally, some devices are "special": They offer a service. Example:**
  - Printers
  - Media servers

# Solution: Zeroconf

- **Zeroconf is a set of protocols that let a machine**
  - **Get an IP address from DHCP**
  - **Advertize its name and IP using multicast DNS (mDNS)**
  - **Announce the availability of a service at a certain port**
- **Users can then browse a list and discover machines and service types.**
- **Common service names are standardized**
- **No previous knowledge of the network is required.**

# What is mDNS

- **DNS: point-to-point queries and response on port 53**
- **mDNS: Multicast queries on port 5353**
- **Used for DNS Service Discovery**
  - **Queries should be specific to avoid a response explosions, e.g., query for service**
  - **Responders should be optimized to filter queries**
  - **mDNS daemons should maintain caches**

# Apple Bonjour protocol

- **Apple implements Zeroconf with Bonjour**
  - **Born in 2002 as Rendezvous, a name copyrighted by someone else**
  - **Renamed with another pretentious French word in 2005**
- **Uses mDNS and Zeroconf concepts**
- **Local machines are on the .local domain**
- **Users and programs are encouraged to use names in the .local domain, not transient IPs**

# Apple Bonjour protocol (cont'd)

- Application can publish and search for standard service names, e.g.: Apple TimeMachine, iTunes Audio Access, Internet Printer, LDAP Server, etc.
- Revolves around mDNS announcements and responses
- Performance optimization: caching, suppression of duplicate responses, query frequency tapering.
- Still very chatty: Datto at Norwalk see > 1 Mb/s mDNS traffic.

# Linux Avahi

- **Avahi is an OSS implementation of Bonjour**
- **Created in 2004 because Apple's code was under a not-so-open-source license**
- **Shipped with most Linux and \*nix distros**
- **An avahi is a type of Madagascar lemur.**



# Testing Avahi

- **Start the service as root:**
  - `/etc/init.d/avahi-daemon start`
  - If it fails, check `syslog`.
  - If you see a `syslog` message about `DBus`, start it first:  
`dbus-daemon -system`  
and retry
- **Check the `/etc/avahi/services/` dir**
  - Most distro contain at least an entry. Ubuntu has `udisks.service`
  - Service definition files are XML files. Look at the service type (e.g., `_udisks-ssh._tcp`)

# Testing Avahi (cont'd)

- From a shell, run  
`avahi-browse _udisks-ssh._tcp -t -r |less`
- Browse the output, looking for your hostname
- Now other machines can see you on the `.local` domain

# Dynamic service definition

- Very useful for tests
- You can run an avahi responder that will advertize a service defined on the command line:  

```
# avahi-publish-service -v FredsTest _datto._tcp 22 &  
Server version: avahi 0.6.30; Host name: fmora-dtp.local  
Established under name 'FredsTest'
```
- In another terminal:  

```
# avahi-browse _datto._tcp -t -r -p  
+;eth0;IPv4;FredsTest;_datto._tcp;local  
=;eth0;IPv4;FredsTest;_datto._tcp;local;fmora-dtp.local;10.0.71.101;22;
```

# Static service definition

- For persistent definition, create an XML file in `/etc/avahi/services/`

- Example:

```
<?xml version="1.0" standalone='no'?>
<!DOCTYPE service-group SYSTEM "avahi-service.dtd">
<service-group>
  <name>Test Service</name>
  <service>
    <type>_datto_test._tcp</type>
    <port>22</port>
    <txt-record>I can put whatever here</txt-record>
    <txt-record>More that 1 record is OK</txt-record>
  </service>
</service-group>
```

- To view this service in your scripts:

```
# service avahi-daemon restart
# avahi-browse _datto_test._tcp -t -r
+ eth0 IPv4 Test Service _datto_test._tcp local
= eth0 IPv4 Test Service
_datto_test._tcp local
  hostname = [fmora-dtp.local]
  address = [10.0.71.101]
  port = [22]
  txt = ["More that 1 record is OK" "I can put whatever here"]
```