Automatic end-host configuration

Research Project 1

Sebastian Dabkiewicz

System And Network Engineering
University of Amsterdam

8th February 2012

Research Question

Research Question

How can one create an automatic end-host confguration?

Sub-Research Questions

- What are the requirements for a fast establishment of the connection?
- What is the current situation?
- What kind of implementations are available?
- What kind of configuration is needed?
- Is there support for a cross platform solution?

Environment

- Circuit-based networks
- Built for long time
- Time intensive to build
- Automated GLIF Open Lightpath Exchanges (GOLE) with Network Service Interface (NSI) from Global Lambda Integrated Facility (GLIF)

Goal

- Quick online
- Almost no configuration (Zeroconf)
- Cross-platform

Zerconf

- Link local addresses
- Multicast DNS
- DNS Service Discovery

IPv4 Link Local address

- IP range 169.254/16
- Not routable
- 3 ARP probes to verify address is available
- RFC 3927 - Dynamic Configuration of IPv4 Link-Local Addresses

IPv6 Link Local address

- fe80::/64
- Derived from MAC-Address
- Present on every interface
- RFC 4291 - IP Version 6 Addressing Architecture
- RFC 4862 - IPv6 Stateless Address Autoconfiguration

MAC to IPv6ll

MAC-Address: `00:15:c5:e1:41:bf`

Becomes: `fe80::215:c5ff:fee1:41bf/64`

multicast DNS

- `221.0.0.254` & `ff02:0:0:0:0:0:0:fb`
- Listen on port 5353
- `.local.` like vanilla.local.
- IETF Draft - Multicast DNS - Dec 9, 2011

DNS Service-Discovery

- Discover services
- DNS SRV Service Types
- e.g. _ssh._tcp
- IETF Draft - DNS-Based Service Discovery - Dec 9, 2011

Implementations

Two main implementations

- Bonjour
  - Shipped with MAC OS X
  - Available for Windows
- Avahi
  - Open source implementation
  - Linux, BSD

Implementations

Two main implementations

- Bonjour
  - Shipped with MAC OS X
  - Available for Windows
- Avahi
  - Open source implementation
  - Linux, BSD

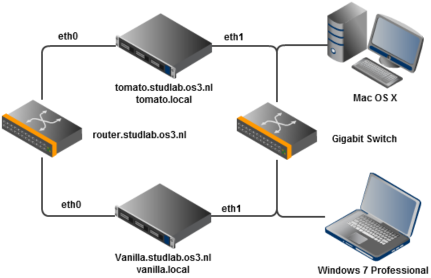
Test-Environment

- Servers
  - Ubuntu 11.10
- Clients
  - MAC mini with MAC OS X
  - Windows 7 Professional laptop

Test-Environment

- Servers
  - Ubuntu 11.10
- Clients
  - MAC mini with MAC OS X
  - Windows 7 Professional laptop

Test-Environment



avahi-daemon

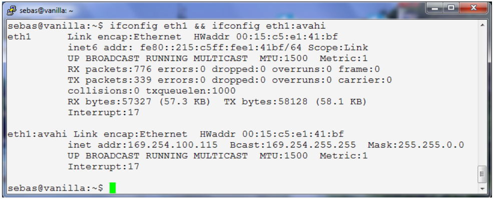
- Main part of Avahi
- Configuring parameters

avahi-daemon configuration file

```
use-ipv4=yes
use-ipv6=no
allow-interfaces=eth1
deny-interfaces=eth0
```

avahi-autoipd

- creates interface
- ethX:avahi
- ipv4ll address

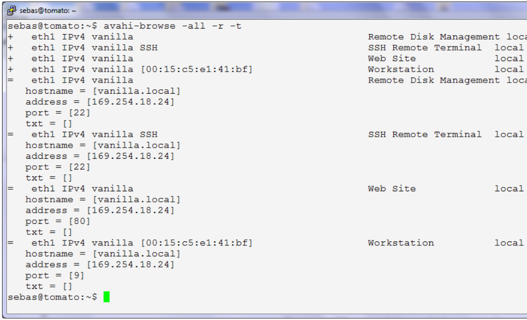


```
sebas@vanilla:~$ ifconfig eth1 && ifconfig eth1:avahi
eth1      Link encap:Ethernet  HWaddr 00:15:c5:e1:41:bf
          inet6 addr: fe80::215:c5ff:fee1:41bf/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:776 errors:0 dropped:0 overruns:0 frame:0
          TX packets:339 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:57327 (57.3 KB)  TX bytes:58128 (58.1 KB)
          Interrupt:17

eth1:avahi Link encap:Ethernet  HWaddr 00:15:c5:e1:41:bf
          inet addr:169.254.100.115  Bcast:169.254.255.255  Mask:255.255.0.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          Interrupt:17

sebas@vanilla:~$
```

avahi-utils 1

avahi-browse

- browse the network for services
- resolve services

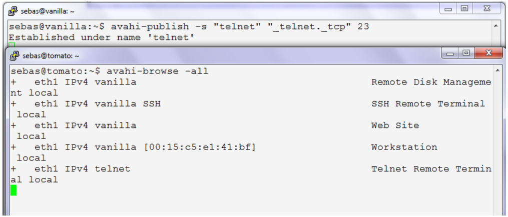


```
sebas@tomato:~$ avahi-browse -all -r -t
+   eth1 IPv4 vanilla                                       Remote Disk Management loca
+   eth1 IPv4 vanilla SSH                                   SSH Remote Terminal  local
+   eth1 IPv4 vanilla                                       Web Site             local
+   eth1 IPv4 vanilla [00:15:c5:e1:41:bf]                   Workstation          local
=   eth1 IPv4 vanilla                                       Remote Disk Management loca
   hostname = [vanilla.local]
   address = [169.254.18.24]
   port = [22]
   txt = []
=   eth1 IPv4 vanilla SSH                                   SSH Remote Terminal  local
   hostname = [vanilla.local]
   address = [169.254.18.24]
   port = [22]
   txt = []
=   eth1 IPv4 vanilla                                       Web Site             local
   hostname = [vanilla.local]
   address = [169.254.18.24]
   port = [80]
   txt = []
=   eth1 IPv4 vanilla [00:15:c5:e1:41:bf]                   Workstation          local
   hostname = [vanilla.local]
   address = [169.254.18.24]
   port = [9]
   txt = []
sebas@tomato:~$
```

avahi-utils 2

avahi-publish

- Publish services
- Name
- SRV-type
- Port



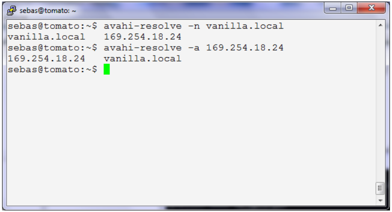
```
sebas@vanilla:~$ avahi-publish -s "telnet" "_telnet._tcp" 23
Established under name 'telnet'
```

```
sebas@tomato:~$ avahi-browse -all
+   eth1 IPv4 vanilla                                        Remote Disk Manageme
nt local
+   eth1 IPv4 vanilla SSH                                    SSH Remote Terminal
 local
+   eth1 IPv4 vanilla                                        Web Site
 local
+   eth1 IPv4 vanilla [00:15:c5:e1:41:bf]                    Workstation
 local
+   eth1 IPv4 telnet                                         Telnet Remote Termin
al local
```

avahi-utils 3

avahi-resolve

- Resolve Hostname
- Resolve IP-address



```
sebas@tomato:~$ avahi-resolve -n vanilla.local
vanilla.local   169.254.18.24
sebas@tomato:~$ avahi-resolve -a 169.254.18.24
169.254.18.24   vanilla.local
sebas@tomato:~$
```

Clients

- Mac Mini
  - MAC OS X
  - No additional installation
- Windows Laptop
  - Windows 7 Professional
  - Installation of Safari browser
  - Bonjour control panel

Interface configuration

- Interface configuration is needed
- No DHCP! Time-out takes 5 minutes
- Use ipv4ll

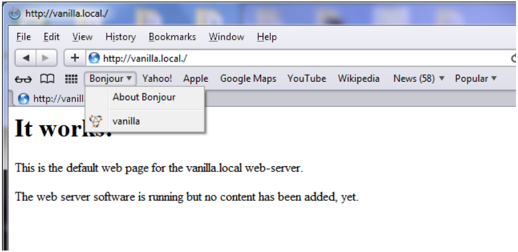
Interface configuration

```
/etc/network/interfaces
(...)
auto eth1
iface eth1 inet ipv4ll
```

Cross platform

avahi-publish

- Windows 7
  - installation of Bonjour needed
  - Bonjour SDK
- Mac OS X
  - Works out of the box

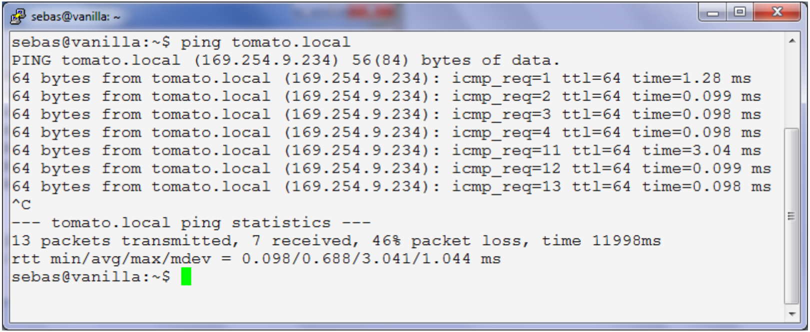


It wor

This is the default web page for the vanilla.local web-server.

The web server software is running but no content has been added, yet.

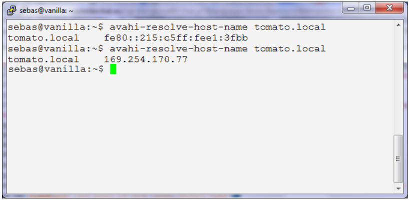
Timing



```
sebas@vanilla:~$ ping tomato.local
PING tomato.local (169.254.9.234) 56(84) bytes of data.
64 bytes from tomato.local (169.254.9.234): icmp_req=1 ttl=64 time=1.28 ms
64 bytes from tomato.local (169.254.9.234): icmp_req=2 ttl=64 time=0.099 ms
64 bytes from tomato.local (169.254.9.234): icmp_req=3 ttl=64 time=0.098 ms
64 bytes from tomato.local (169.254.9.234): icmp_req=4 ttl=64 time=0.098 ms
64 bytes from tomato.local (169.254.9.234): icmp_req=11 ttl=64 time=3.04 ms
64 bytes from tomato.local (169.254.9.234): icmp_req=12 ttl=64 time=0.099 ms
64 bytes from tomato.local (169.254.9.234): icmp_req=13 ttl=64 time=0.098 ms
^C
--- tomato.local ping statistics ---
13 packets transmitted, 7 received, 46% packet loss, time 11998ms
rtt min/avg/max/mdev = 0.098/0.688/3.041/1.044 ms
sebas@vanilla:~$ 
```

Avahi-resolve

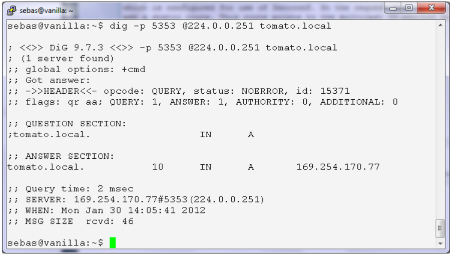
- Avahi standard publishes AAAA-record on IPv4
- Shows first IPv6 record then IPv4 record
- Use −4 or −6 switch to get direct the correct record



```
sebas@vanilla:~$ avahi-resolve-host-name tomato.local
tomato.local    fe80::215:c5ff:fee1:3fbb
sebas@vanilla:~$ avahi-resolve-host-name tomato.local
tomato.local    169.254.170.77
sebas@vanilla:~$ 
```

Multicast-DNS

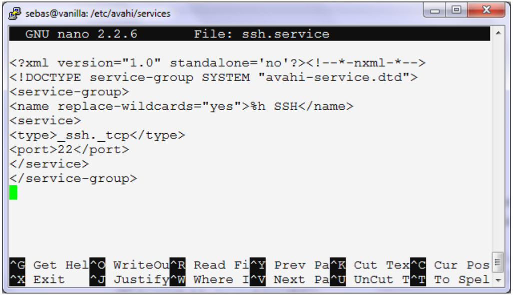
- Problems using dig to resolve hostname
- Packet send out on eth0 (internet interface)
- Add static route for 224.0.0.251/32



```
sebas@vanilla:~$ dig -p 5353 @224.0.0.251 tomato.local

; <<>> DiG 9.7.3 <<>> -p 5353 @224.0.0.251 tomato.local
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 15371
;; flags: qr aa; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;tomato.local.                  IN      A

;; ANSWER SECTION:
tomato.local.           10      IN      A       169.254.170.77

;; Query time: 2 msec
;; SERVER: 169.254.170.77#5353(224.0.0.251)
;; WHEN: Mon Jan 30 14:05:41 2012
;; MSG SIZE  rcvd: 46

sebas@vanilla:~$
```

DNS Service Discovery

- Not every service is announced
- Avahi provides XML-file format
- *.service



sebas@vanilla: /etc/avahi/services

```
 GNU nano 2.2.6          File: ssh.service

<?xml version="1.0" standalone='no'?><!--*-nxml-*-->
<!DOCTYPE service-group SYSTEM "avahi-service.dtd">
<service-group>
<name replace-wildcards="yes">%h SSH</name>
<service>
<type>_ssh._tcp</type>
<port>22</port>
</service>
</service-group>
```

^G Get Hel^O WriteOu^R Read Fi^Y Prev Pa^K Cut Tex^C Cur Pos
^X Exit   ^J Justify^W Where I^V Next Pa^U UnCut T^T To Spel

Conclusion

- Zeroconf fits in the situation
- On servers some configuration needed
- Quick online, about 6 seconds

Demo

Demo

Demo with Mac mini and evocam

Questions?

Questions?

Report & Presentation available at:

`http://goo.gl/FmfKf`

and

`http://goo.gl/ruWEQ`