

# Understanding Netdot

## Contents

<b>1</b>	<b>Installation</b>	<b>2</b>
<b>2</b>	<b>Netdot “Device”</b>	<b>2</b>
<b>3</b>	<b>Naming of devices</b>	<b>2</b>
<b>4</b>	<b>End user PCs are not “Devices”</b>	<b>3</b>
<b>5</b>	<b>Polling of devices</b>	<b>3</b>
<b>6</b>	<b>“Address” (a.k.a. “ipblock”)</b>	<b>3</b>
<b>7</b>	<b>Address views</b>	<b>4</b>
<b>8</b>	<b>VLANS</b>	<b>4</b>
<b>9</b>	<b>DNS</b>	<b>4</b>
<b>10</b>	<b>DHCP</b>	<b>4</b>
<b>11</b>	<b>Not convered in this presentation</b>	<b>5</b>
<b>12</b>	<b>Documentation</b>	<b>5</b>
<b>13</b>	<b>Platform</b>	<b>5</b>
<b>14</b>	<b>Database structure</b>	<b>5</b>

## 1 Installation

- Option 1: install the prebuilt VM image
- Option 2: install under Ubuntu 14.04 (LTS):
  - See the lab worksheet provided
  - Quite a few steps and dependencies. Read carefully.
  - Get help from the netdot-users mailing list!
- I recommend you set your timezone
  - `sudo dpkg-reconfigure tzdata`

## 2 Netdot “Device”

- Represents a piece of *infrastructure* which you manage via SNMP
  - e.g. switch, router, AP, server
- Can add individual devices manually via the GUI (SNMP or not)
- Add SNMP devices via command line
  - `updatedevices.pl -H 10.10.0.254 -I -c NetManage`
- Can sweep a subnet for SNMP devices
  - `updatedevices.pl -B 10.10.0.0/24 -I -c NetManage`
  - Warning: everyone will see your community string!
  - And you might want to use the loopback address anyway
- Or add a list of devices from a text file (see manual)

## 3 Naming of devices

- Each device must have a primary IP and name
- Several sources, in configurable order, e.g.
  - resolve SNMP sysName
  - reverse DNS of the snmp target
- Unfortunately doesn’t work well unless your DNS is already working
  - Not good if you are planning to use Netdot to fix your DNS!
- The device name is a key in Netdot’s DNS tables, but may not be associated with an A record (RR)
  - This is confusing

- Device serial numbers are checked, but you only get a warning if they overlap
  - Sometimes we see the same serial number more than once, e.g. with emulated hardware (dynamips) or faulty equipment

## 4 End user PCs are not “Devices”

- They are auto-detected from ARP tables and forwarding tables
- You can search for them by IP or MAC address
- You can see which switch port they were last plugged into
- You can see when the IP was last active

## 5 Polling of devices

- A cron job polls your devices periodically (say) every hour
  - Don’t forget to `cp netdot.cron /etc/cron.d/netdot`
- `updatedevices.pl -DIFAT`
  - Poll only devices in databases
  - Refresh basic information (sysName, sysLocation etc)
  - Read forwarding tables from switches
  - Read ARP tables from routers
  - Read CDP/LLDP topology
- This can be turned on/off per device
  - e.g. if you don’t want to poll forwarding table on a particular switch

## 6 “Address” (a.k.a. “ipblock”)

- Represents a single IP, or a prefix
- Prefix can be “container” or “subnet”
  - A “subnet” is something actually seen on a router interface
  - A “container” can contain subnets or other containers (recursively)
  - A “subnet” *cannot* contain subnets or containers
  - A “container” will be turned into a “subnet” if it’s seen on a router interface
- Single IPs have different states
  - Static - belongs to a device interface, or manually set to static

- Discovered - seen in ARP table
- Dynamic - defined as part of a DHCP pool
- Reserved - should not be assigned

## 7 Address views

- View as list, tree, or graphical block form
- Clicking on an individual IP marks it as “static”
  - One-click IP allocation!
  - But it doesn’t have a confirmation dialog

## 8 VLANS

- Discovered from switches/routers
- Right now, a dot1q tag must be unique across your network

## 9 DNS

- Must create a DNS zone first, then the records
- Create reverse DNS zones, or you won’t get PTR records created
- Netdot can create unique DNS names for each device interface, based on e.g. interface type and number
  - This is configurable, and you can write a module for your own naming convention if you wish
- Using “DNS ... new” actually allocates a new IP address
  - Perhaps this is in the wrong place in the UI
  - You need to link one or more zones to a subnet to make this work

## 10 DHCP

- Can create config files for ISC DHCP server, e.g. to give static IPs to particular hosts
- Create a DHCP “scope”
- Then you mark a row individual IPs as “dynamic”, using the bulk update facility
  - So that the DHCP range doesn’t have to fall on subnet boundary
  - Probably won’t scale to v6 :-)

## 11 Not covered in this presentation

- Cable plant
- Contacts and People
- Reports
- Exporting (e.g. DNS zone files, Nagios configs, dhcpd configs)

## 12 Documentation

- Finding stuff can be difficult
  - “I know that Netdot can show my network topology, but where was it again?”
- It's open source, you can contribute to the documentation
  - The manual
  - wiki

## 13 Platform

- apache + mod\_perl
- mysql
- rrdtool
- graphviz
- ... lots of perl dependencies

## 14 Database structure

- DB schema diagram: linked from <https://osl.uoregon.edu/redmine/projects/netdot/wiki/Documentation>
- Traditional SQL design
- Most important tables: device, ipblock