Dibbler – a portable DHCPv6
config files

Tomasz Mrugalski

12 stycznia 2004 roku

version 0.1

1 Config files

This short document describes Dibbler server and (optional) client configuration. Square brackets denotes optional values: mandatory [optional]. Alternative is marked as |. A | B means A or B.

1.1 Tokens and basic informations

Config file parsing is token-based. Here's list of tokens used:

IPv6 address – IPv6 address

32-bit decimal integer – string containing only numbers, e.g. 123456

32-bit hex integer – string constaining only numbers and letter a–f, ended with h letter, e.g. 23afh

string – string of arbitrary characters enclosed in single quotes, e.g. 'this is string'

DUID identifier – hex number enclosed in double quotes, e.g. "deadbeaf"

There are also some client or server specified tokens. Here's the list:

client – iface, no-config, address, ia, no-ia, log-level, work-dir, append, prepend, require, request, send, default, supersede, prefered-lifetime, valid-lifetime, t1, t2, dns-servers, domain, ntp-servers, time-zone, reject-servers, prefered-servers, rapidcommit

server – iface, no-config, class, log-level, work-dir, dns-servers, domain, ntp-server, time-zone, accept-only, reject-only, t1, t2, prefered lifetime, valid-lifetime, unicast, preference, pool, rapidcommit, max-lease, client-max-lease

Comments are also allowed. All common comment styles are supported:

- C++ style one-line comments: // this is comment
- C style multi-line comments: /* this is multiline comment */
- bash style one-line comments: # this is one-line comment

Parses is case-insensitive, so Iface, IfAcE, iface and IFACE mean the same. This does not apply to interface names, of course. eth0 and ETH0 are dwo diffrent interfaces.

2 Client config file

Client config file should be named *client.conf.* After successful startup, old version of this file is stored as *client.conf-old.*

2.1 Scopes

There are four scopes, in which options can be specified:

- global
- inteface
- IA
- address

Global scope is the largest. It covers the whole config file and applies to all intefaces, IAs, and addresses, until some lower level options override it. Next comes inteface scope. Options defined there are inteface-specific and apply to this interface, all IAs in this interface and addresses in those IAs. Next is IA scope. Options defined there are IA-specific and apply to this IA and to addresses it contains. Least significant scope is address. Every option is specific for one scope. For example, T1 is defined for IA scope. However, it can be also used in more common scopes. In this case – in interface or global. Defining T1 in interface scope means: „for this interface default value for T1 is ...". The same applies to global scope. Options can be used multiple times. In that case value defined later is used.

2.2 Global scope

Every option can be declared in global scope. Config file has this form:

```
 [interface declaration  |
  global options         |
  interface options      |
  IA options             |
  address options        ]
```

2.3 Interface declaration

Interface can be declared this way:

```
iface 'name_of_this_interface'
{
  interface options      |
  IA options             |
  address options
}
```

or

```
iface number
{
  interface options      |
  IA options             |
  address options
}
```

or

```
iface 'name_of_this_interface' no-config
```

or

```
iface number no-config
```

In every case, number denotes interface number. It can be extracted from „ip l" (Linux) or „ipv6 if" (Windows). *no-config* means that this interface should not be configured. Client retrives interface list from the system. If it detects interface and does not find information about it in config, it assumes that one address should be requested for this interface. (Note that interfaces with *loopback* flag set are not configured. That applies also to interfaces without flags *up* and *running*).

If interface contains no IA keyword, one IA with one address is assumed. If interface should not request for address, *no-ia* should be used.

2.4 IA declaration

IA is declared this way:

```
ia [number]
[{
  [address declaration   |
   IA options            |
   address options       ]
   ...
}]
```

where number is optional number, which describes how many such IAs should be requested. If this number is not equal 1, then address options are not allowed. That could come in handy when someone need serveral IAs with the same parameters. If IA contains no addresses, client assumes that one address should be configured.

2.5 Address declaration

Addres is declared like this:

```
address [number]
[{
[address options   |
IPv6 address    ]
}]
```

where number denotes how many addresses with those values should be requested. If it is diffrent than 1, then IPv6 address option is not allowed.

2.6 Options

Options are declared this way:

```
OptionName option-value
```

Every option has a scope in which it can be used, default value and sometimes allowed range.

Name	Scope	Values (default)	default	Description
valid-lifetime	address	32-bit integer	(4294967296)	valid lifetime for address (specified in seconds) (H)
prefered-lifetime	address	32-bit integer	(4294967296)	after this amount of time(in seconds) address becomes depreciated (H)
T1	IA	32-bit integer	(4294967296)	client should renew addresses after T1 seconds (H)
T2	IA	32-bit integer	(4294967296)	client should send REBIND after T2 seconds (H)
reject-servers	IA	IPv6 addrs or DUID list, coma separated	(null list)	list containing servers which should be discarded in configuration of this IA
prefered-servers	IA	IPv6 addrs or DUID list	(null list)	Prefered servers list. ADVERTISE messages received by client are sorted according to this list.
RapidCommit	IA	0 or 1	0	should we use Rapid Commit (SOLICIT–REPLY)?
work-dir	global	string	(null)	working directory
log-level	global	1-5	5	log-level (5 is most verbose)
dns-servers	interface	IPv6 addrs list, coma separated	(null list)	prefered DNS servers list (H)
domain	interface	domain	(null)	prefered domain (H)
ntp-servers	interface	IPv6 addrs list, coma separated	(null list)	prefered NTP servers list (H)
time-zone	interface	timezone string	(null)	time zone (H)

Note that timezone format is described in file draft-ietf-dhc-dhcpv6-opt-tz-00.txt and domain format is described in RFC 3646.

2.7 Client Examples

In simplest case, client config can be empty. Client will try to assign one address for every up and running interface (except loopback, of course).

Let's conside more coplicated case. Let's say there are 4 interfaces, numbered 1 thru 4. Interfaces 1,2 and 3 are not to be configured. Interface 4, named eth0 should have 3 IAs. Two of them are supposed to contain one address each. This IA should contain 3 addresses. Addresses assigned to first and second IA should have prefered-lifetime 1 hour and valid-lifetime 2 hours. This IA should have 3 specific addresses: 2000::1, 2000::2 and 2000::3. Information about NTP servers,our current timezone, available DNS servers and our domain should also be retrived,

Here's config file:

```
iface 'eth0'
{
  valid-lifetime 7200    // default value - 2 hours
  prefered-lifetime 3600 // default value - 1 hour
  T1 600                 // request 10 minutes interval
  T2 1200  /* trouble begins after 20 minutes
                         of server's silence */
  IA 2                   // 2 IAs with just specified values (1h/2h/10min/20min)
  IA  // third IA is more specific
  {
    valid-lifetime 3600    // valid lifetime changed to 1 hour
    prefered-lifetime 1800 // prefered lifetime changed to 30min

    address
    {
      2000::1   // request those addresses
      2000::2
      2000::3
    }
  }
  NTP-Servers       // ask for NTP servers
  Time-Zone         // ask for timezone
  DNS-Servers ''    // ask for DNS servers
  Domain            // ask for domain
}
iface 3 no-config
iface 2 no-config
iface 1 no-config
```

Note that dns-servers option requires a string. If there is no hint to send to server, empty string has to be specified. (Client code contains some nasty bug, which we cannot find. Sorry.)

2.8 Server config file

Client config file should be named *server.conf*. After successful startup, old version of this file is stored as *server.conf-old*.

2.8.1 Scope

There are four scopes, in which options can be specified:

- global
- inteface
- IA
- address

Scopes has the same meaning as been described in client config files.

2.8.2 Global scope

Every option can be declared in global scope. Config file has this form:

```
interface declaration |
global options         |
interface options      |
class options
```

2.9 Interface declaration

Interface can be declared this way:

```
iface 'name_of_this_interface'
{
  interface options      |
  class options
}
```

or

```
iface number
{
  interface options      |
  class options
}
```

where 'name_of_this_interface' denotes name of the interface and number denotes it's number.

2.10 class scope

Address class is declared as follows:

```
class
{
     class options |
     address poll
}
```

address poll has this format:

```
poll minaddress-maxaddress
```

2.11 Options

Name	Scope	Values (default)	default	Description
valid-lifetime	class	32-bit integer	(4294967296)	valid lifetime for address (specified in seconds)
prefered-lifetime	class	32-bit integer	(4294967296)	after this amount of time(in seconds) address becomes depreciated
T1	class	32-bit integer	(4294967296)	client should renew addresses after T1 seconds
T2	class	32-bit integer	(4294967296)	client should send REBIND after T2 seconds
reject-clients	class	IPv6 addrs or DUID list, coma separated	(null list)	list containing servers which should be discarded in configuration of this IA
accept-only	class	IPv6 addrs or DUID list	(null list)	these are the only clients allowed to use this class
RapidCommit	class	0 or 1	0	should we allow Rapid Commit (SOLICIT–REPLY)?
work-dir	global	string	(null)	working directory
log-level	global	1-5	5	log-level (5 is most verbose)
dns-servers	interface	IPv6 addrs list, coma separated	(null list)	DNS servers list
domain	interface	domain	(null)	domain
ntp-servers	interface	IPv6 addrs list, coma separated	(null list)	NTP servers list
time-zone	interface	timezone string	(null)	time zone
unicast	class	0 or 1	0	is unicast communication allowed?
preference	class	0-255	0	server preference value (higher is more prefered)
max-lease	class	32-bit integer	4294967296	how many addresses can be leased by clients?
client-max-lease	class	32-bit integer	4294967296	how many addresses can be leased by one client?

2.12 Server example

In opposite to client, server uses only interfaces described in config file. Examine this common situation: server has interface named *eth0* (which is 4th interface in the system) and is supposed to assign addresses from 2000::100/124 class. Simplest config file looks like that:

```
iface 4
{
  class
  {
    pool 2000::100-2000::10f
  }
}
```

Let's extend this with numerous requirements:

- add new class on fifth interface named *eth1*, for example 2000::fe00/120, for which Rapit commit will be allowed

- server preference set to maximum (255)
- assign 2000::20/124 class on 4th interface named eth0. Preference value for this interface should be 0.
- don’t ignore client with DUID “00001231200adeaaa” in the class 2000::20/124.
- valid and prefered lifetimes are 1 hour and 30 minutes respectively. T1 and T2 set to 10 minutes and 20 minutes.
- For class 2000::100/124 valid and prefered lifetimes are 2 hours and 1 hour.
- in class 2000::100/124 one client can request up to two addresses
- do not assign more than 10 addresses from 2000::20/124 class.
- client with fe80::200:39ff:fe4b:1abc address should get his static address 2000::2f
- we shall support DNS and NTP servers on interface 5. Timezone and domains information is also supported.
- working directory and log level is also set

Here’s config to do all this stuff:

```
log-level 5
work-dir ’/var/lib/dibbler’
valid-lifetime 3600
prefered-lifetime 1800
T1 600
T2 1200
iface 4
{
  preference 0
  class
  {
    reject-clients ‘‘00001231200adeaaa’’
    2000::2f-20::20  // it’s in revers order, but it works.
                     // just a trick.
    max-lease 10
  }
  class
  {
    accept-only fe80::200:39ff:fe4b:1abc
    pool 2000::2f
  }
}
iface 5
{
  dns-server 2000::123:456,2000::456:1234
  ntp-server 2000::1111:2222
  RapidCommit 1
  time-zone ’’EST’’
  domain ’’example.com’’
  preference 255
  class
  {
    pool 2000::fe00-2000::feff
  }

  class
  {
    valid-lifetime 7200
    prefered-lifetime 3600
    pool 2000::0-2000::f
    client-max-lease 2
  }
}
```