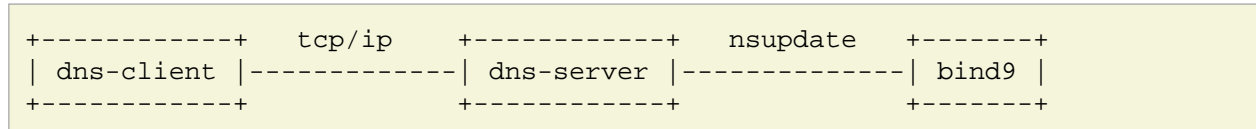


# Presentation

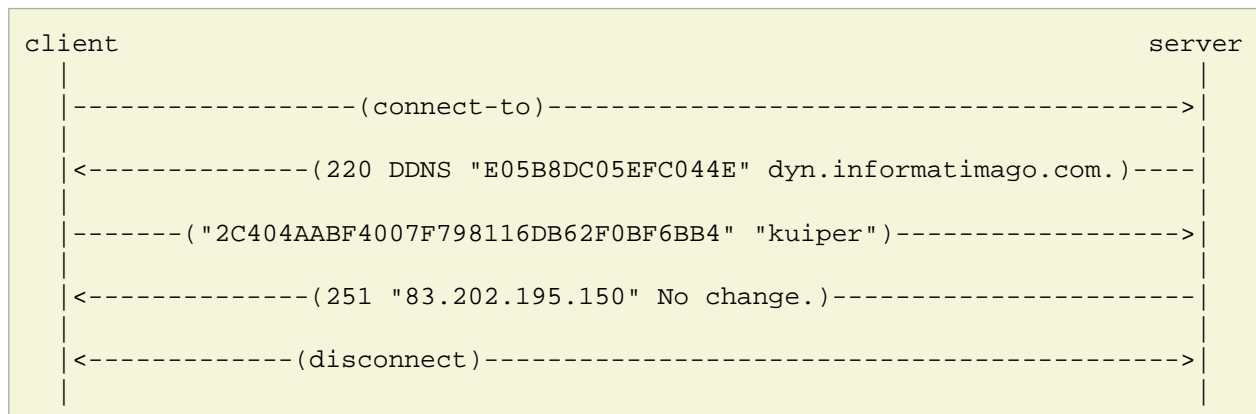
DDNS is a couple of client/server programs, where the client sends a request to the server to update the dns record of the host with its current address using nsupdate.

The DDNS server is not necessary running on the DNS server:



# Protocol

The protocol is quite simple, and hopefully secure:



1. The client connects.
2. The server sends an answer:
  - status code: 220
  - protocol: DDNS
  - a random seed: "E05B8DC05EFC044E"
  - its dynamic domain name \$ORIGIN: dyn.informatimago.com.
3. The client sends a request:
  - opening parenthesis: (
  - hashed secret: "2C404AABF4007F798116DB62F0BF6BB4"
  - host name: "kuiper"
  - closing parenthesis: )

The hashed secret is the MD5 hash of the concatenation of the hostname, "/", the seed, "/" and a common secret.

4. The server checks the hashed secret, and if it matches, updates the DNS records with the address of the client and the host name. It sends an answer:

- status code,
- address of the client,
- message
- optionally, continuation lines with output from the nsupdate process.

The status code is:

- 250 for successful update,
- 251 for no change, the host has already this address,
- 501 for an Invalid token.
- 530 for an failure to authenticate.
- 553 for an update error.

Examples:

```
250 "83.202.195.150" Update successful.
250-Outgoing update query:
250-;; ->HEADER<<- opcode: UPDATE, status: NOERROR, id:      0
250-;; flags:; ZONE: 0, PREREQ: 0, UPDATE: 0, ADDITIONAL: 0
250-;; UPDATE SECTION:
250-kuiper.dyn.informatimago.com. 0      ANY      A
250-kuiper.dyn.informatimago.com. 10 IN A      83.202.195.150
250-

251 "83.202.195.150" No change.

501 Invalid token.

530 Authentication invalid.

553 "83.202.195.150" Update failed.
553-update failed: SERVFAIL
```

## Compilation

- clisp is used to compile the programs. <http://clisp.cons.org/>
- alternatively, ccl can be used to compile the programs. <http://ccl.clozure.com/>
- quicklisp is required to compile and download dependencies. If it's not already installed in the user account compiling the program, then you may install it with:

```
pushd ~
wget https://beta.quicklisp.org/quicklisp.lisp
sum=( $(sha256sum quicklisp.lisp) )
if [ ${sum[0]} != 4a7a5c2aeb0716417047854267397e24a44d0cce096127411e9ce9ccfeb2c17 ] ; then
  echo 'BAD QUICKLISP!'
fi
yes|clisp -norc -x '(load "quicklisp.lisp")' -x '(quicklisp-quickstart:install)' -x '(ql:add-to-init-file)'
popd
```

- tune the configuration in `server.lisp`:

```
(defparameter *default-port*      8053)
(defparameter *default-origin*    "dyn.informatimago.com.")
```

```
(defparameter *default-dns-server* "localhost")
```

- and in `client.lisp`:

```
(defparameter *default-port* 8053)  
(defparameter *default-ddns-server* "hubble.informatimago.com")  
(defparameter *default-secret-file* #P"dnskeys/ddns.secret")
```

the ports shall be the same in both files.

- then you may compile the programs:

```
make
```

Note: there's a bug with the latest clisp-2.49+ that prevents the client to run. Instead, we'll compile it with ccl.

When compiled with clisp, the client usage is:

```
ddns-client -- $host
```

When compiled with ccl, the client usage is:

```
ddns-client $host
```

## Installation

### Creation of keys, and configuration of bind to receive dyndns updates

See:

1. <http://linux.yyz.us/dns/ddns-server.html> and
2. <http://linux.yyz.us/nsupdate/>

### On the server

1. Create a ddns user account:

```
groupadd --system ddns  
useradd --system -d /home/ddns/ -m -c 'DDNS Server' -s /bin/bash -g ddns ddns  
mkdir /home/ddns/dnskeys /home/ddns/etc  
# install:  
# ddns.secret  
# Kdyn.informatimago.com.+157+24639.key  
# Kdyn.informatimago.com.+157+24639.private  
# in /home/ddns/dnskeys
```

2. The server is installed in `/home/ddns/bin/` with:

```
make install-server
```

3. Edit `/etc/rc.local`, adding:

```
su -c 'screen -d -m -S ddns -t server -c /home/ddns/etc/screenrc \  
      bash -c "export LC_CTYPE=en_US.UTF-8 ; while sleep 2 ; do \  
              /home/ddns/bin/ddns-server ; done"' \  
- ddns
```

4. Launch it immediately with:

```
/etc/init.d/rc.local restart
```

## On the clients

Currently we run the client as root, but it could and should be run under a ddns user account like the server.

1. The client is installed in `/usr/local/sbin/` with:

```
sudo make install-client
```

2. Install the keys:

```
mkdir /root/dnskeys/  
# install:  
#   ddns.secret  
# in /root/dnskeys/
```

3. `/usr/local/sbin/ddns-client` is run from root with cron, fcron, dcron, etc, a few times a hour. With cron, `/etc/crontab` entry:

```
5 * * * * root /usr/local/sbin/ddns-client kuiper
```

## License, Development, Contact

This program DDNS (ddns-client and ddns-server), is distributed under the AGPL3 license.

Copyright 2012 - 2015 Pascal J. Bourguignon

The sources are obtainable with git:

```
git clone http://git.informatimago.com/public/domains  
cd domains/ddns/
```

Send requests, bug reports, patches: <mailto:pjb@informatimago.com>.

Known Bugs:

1. We use MD5 in the protocol, this is not safe anymore, SHA256 should be used instead.