

Installing and Configuring NAC Client: Manual Version 1.0

1 Installing NAC Client

The NAC client is developed based on the **CCNx** package and **c-ares** package. To compile the NAC client, CCNx and c-ares are required to be installed in default directory, ex: CCNx commands MUST be installed in /usr/local/bin. Please refer to the following link to download packages:

CCNx package: <http://www.ccnx.org/>
c-ares package: <http://c-ares.haxx.se/>

After you install both CCNx and c-ares, you can compile, and install it:
compile: **make**
install: **sudo make install**

2 Checking \$HOME directory

You MUST check if you have ".ccnx" directory in the "\$HOME" directory. If you don't have it, please run command "**ccninitkeystore**" to create. Otherwise, the nac_client will report error "unable to create face for ..."

3 Configuring NAC Client

You can get NDN parameters from the NAC server in three different levels. Please read the rest sections, find your scenario and figure out what you need to do.

3.1 Manually triggered Client

For the user who wants to run the NAC client manually, the only thing he needs to do is

running script "**join-ndn.py**"

in directory "**user**". This script will call NAC client to locate the NAC server(s), get NDN parameters, and connect to local NDN network.

Note that root privilege is required to run this script.

3.2 Automatically triggered Client

A more convenient way to launch the NAC client is triggering the client when the computer is booted.

Ubuntu Assuming that your Ubuntu OS uses NetworkManager to manage the network, and you have **installed** the `nac_client`, then all you need to do is running the script "**ubuntu-NM.sh**" in folder "**user/script**".

This script will automatically add a NetworkManager script for you. Then once the network devices are up, the modified script will launch NAC client to get NDN parameters, and connect to local NDN network. If you did not install the `nac_client`, you **MUST** modify the NetworkManager script to provide the `nac_client`'s right location.

Fedora Assuming that your Fedora OS uses NetworkManager to manage the network configuration, and you already **installed** the `nac_client`, then when you run the script "**fedo-NM.sh**" in folder "**user/script**", the OS will launch the NAC client automatically as soon as you open your machine.

MacOS Assuming that you have installed the NAC client on your Mac OS, you **MUST** install third-party package **pymacadmin** to enable reaction to network events.

(1) Installing Pymacadmin:

- Download pymacadmin from <http://code.google.com/p/pymacadmin/>.
- Run the script "`install-crankd.sh`" to install crankd library.

(2) Installing NAC client script:

Run the script "**macscriptgen.sh**" in "`user/scripts`". It will ...

- Install the NAC client script "`/usr/local/bin/nac_clt_starter_Mac.py`",
- Add configuration file "**net.ndn.keepalive.crankd.plist**" to directory "`/Library/LaunchDaemons/`",
- Add configuration file "**net.ndn.nac_client.crankd.plist**" to directory "`/Library/Preferences/`".

Scripts for other operating systems will be released latter.

3.3 NAC optimization

You can make the process faster. To achieve this goal, in addition to the configuration mentioned in the last section, you need to let the DHCP client request NDN parameter in the IP address allocation process, and let the DHCP client accept the DHCPOFFER that contains NDN parameters with higher priority.

Currently this method only support Ubuntu. Please exactly follow the two steps:

(1) Modify the `dhclient.conf` Run the script "**ubuntu-dhclient-gen.sh**" in "`user/scripts/`". This script will change the configuration file "`dhclient.conf`". Then the DHCP client will always requests NDN parameters in the DHCPDISCOVER message.

(2) Replace local dhclient program We provide a ISC DHCP client patch to users. This program enables users prefer the DHCPOFFER with NDN parameters when they collect multiple DHCPOFFERS.

To apply the dhclient patch, please find "dhclient.patch" in the folder "user/patch", then apply this patch file to "dhclient.c" in ISC DHCP 4.2.4-P1.

Note that this optimization method only successes when the local NAC server provide both the IP address and NDN parameters. If the DHCPOFFER which is provided by the local NAC server does not contain both the IP address and NDN parameters, the NAC client will accept that DHCPOFFER, and fall back to the scenario in section 2.2.

Now, you have done everything you need. And you can find your NDN name space in the configuration file "/etc/ndn_namespace.conf".

Enjoy your NDN trip!

4 How to test if it works?

After you run the NAC client, you have two ways to test if you are already connected to the NDN network in netsec.colostate.edu:

(1) If you have installed the ccngetfile, you can use **ccngetfile** to download the test file:

```
ccngetfile ccnx:/ndn/colostate.edu/netsec/NAC-test test
```

This file contains text "This is a test for NAC"

(2) You can also use the **ccnls** to check if this file exists:

```
ccnls ccnx:/ndn/colostate.edu/netsec/NAC-test
```

you will see the following result:

```
%FD%05%02Vi%CFP [unverified]
```

* Operators are encouraged to publish content in your own domain, helping users to test if they are connected to your NDN network.

5 FAQ

(1) The nac_client cannot receive any DHCPACK messages when it broadcasts DHCPINFORM messages, why?

Please check your host firewall policies, and add/remove the associated one.

On Fedora, the iptables is running in default, and it blocks unrelated udp packet. Therefore, you need to add a policy to allow DHCP packets, the following is a option:

```
sudo iptables -I INPUT 1 -p udp --sport 67 --dport 68 -j ACCEPT
```

(2) The nac client reports error "unable to create face for ..."

Please check if the ".ccnx" directory exists in the "\$HOME" directory. If you don't have it, please run "**ccninitkeystore**" to create the directory and the key store.

*More questions please send email to chengyu@cs.colostate.edu