**NDNDroid User Manual**

This is an application that enables the NDNLP protocol to run on Android devices. NDNLP is a link layer protocol that enables CCNx applications to run directly on Ethernet without relying on TCP or UDP tunnels. (<http://www.named-data.net/techreport/TR006-LinkProtocol.pdf>).

NDNDroid enables launching of the ***ndnld*** user space daemon and also to create create remote faces and register prefixes.

NDNDroid requires a rooted phone to run due to the need to use raw sockets. Our implementation was tested on CyanogenMod 9 and CyanogenMod 10.

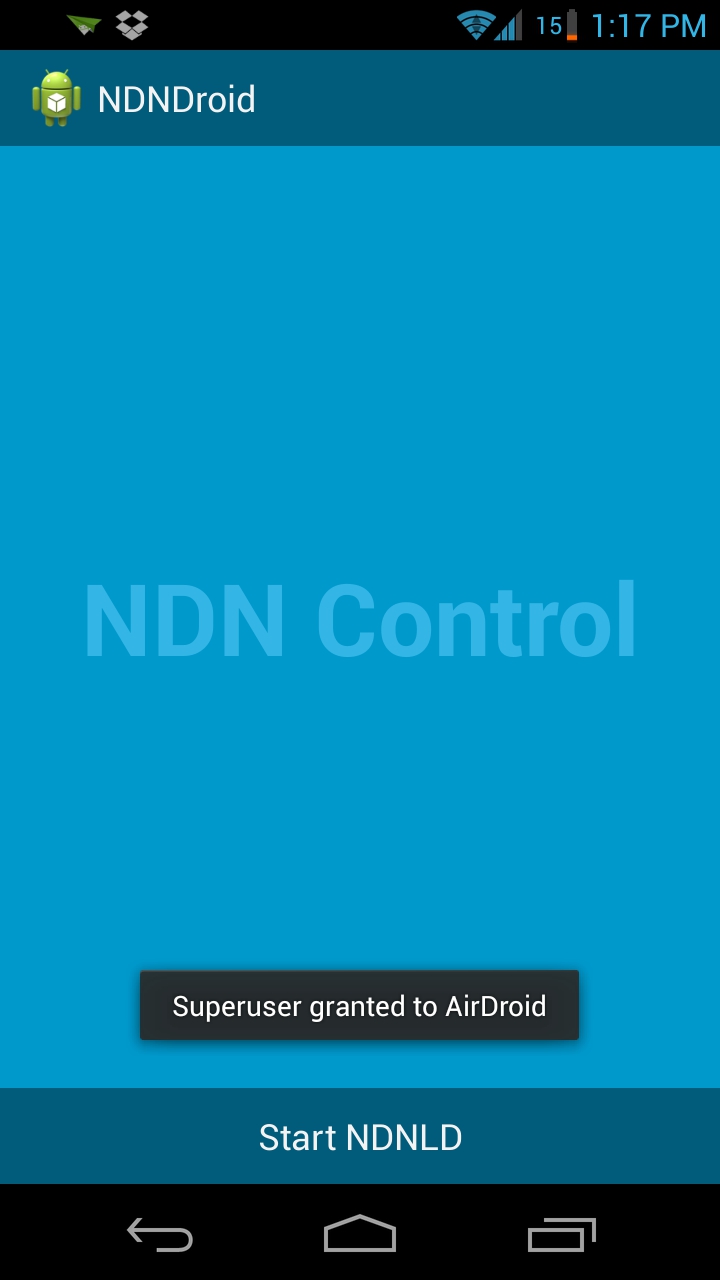
NDNDroid Code repository:

NDNDroid Code apk:

The method to use NDNDroid is as follows:

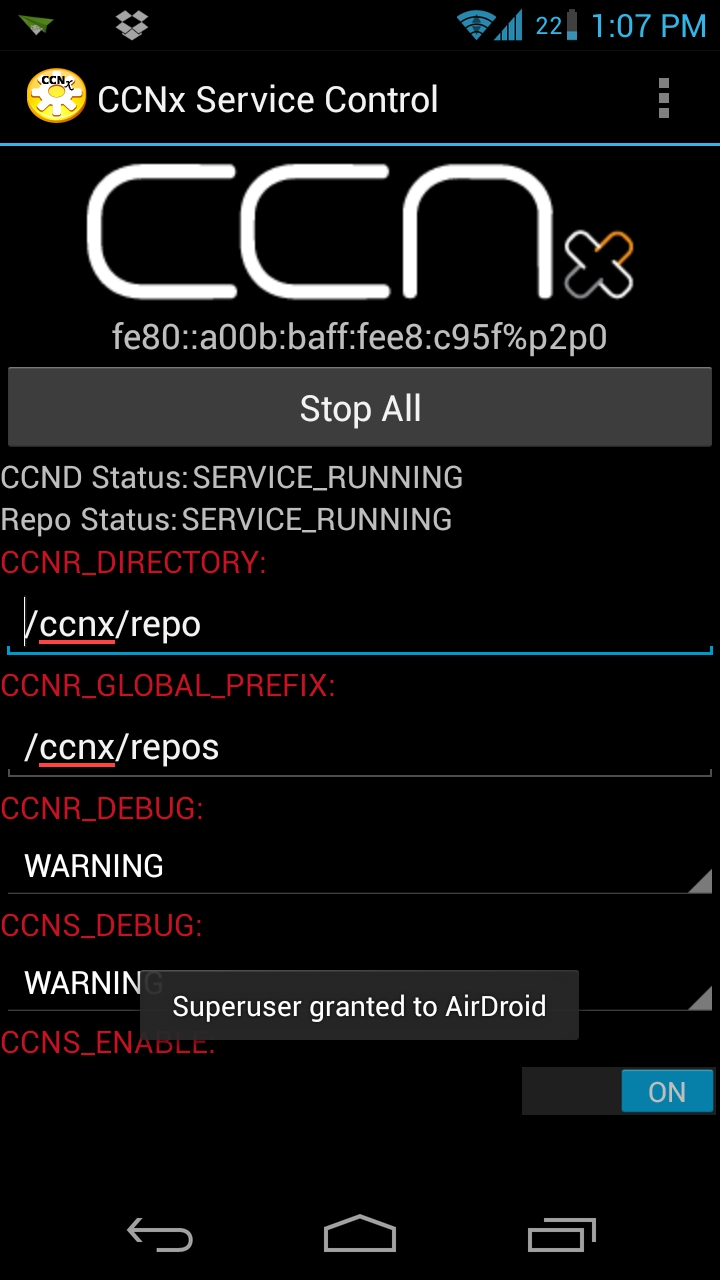
**Step 2:**

Launch NDNDroid application and start NDNLD. This creates a background process that other applications can connect to perform various operations supported by ***ndnldc***



**Step 1:**

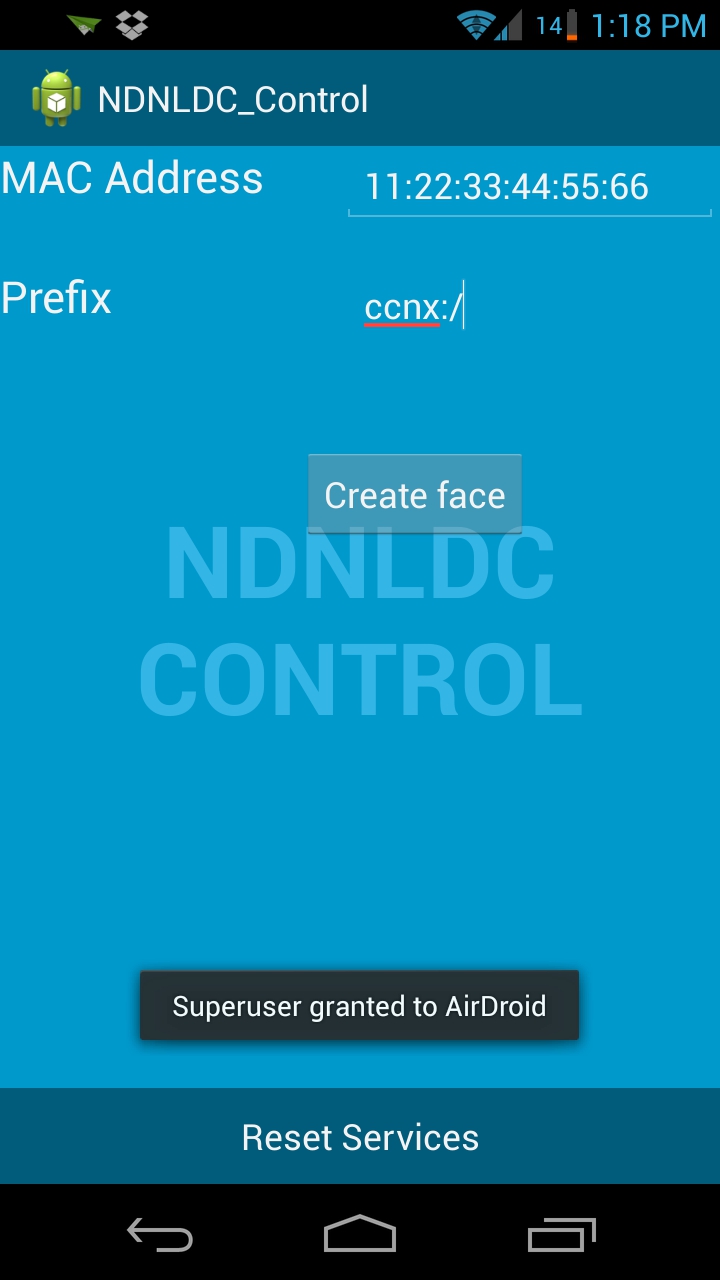
Start CCNx services



**Step 3:**

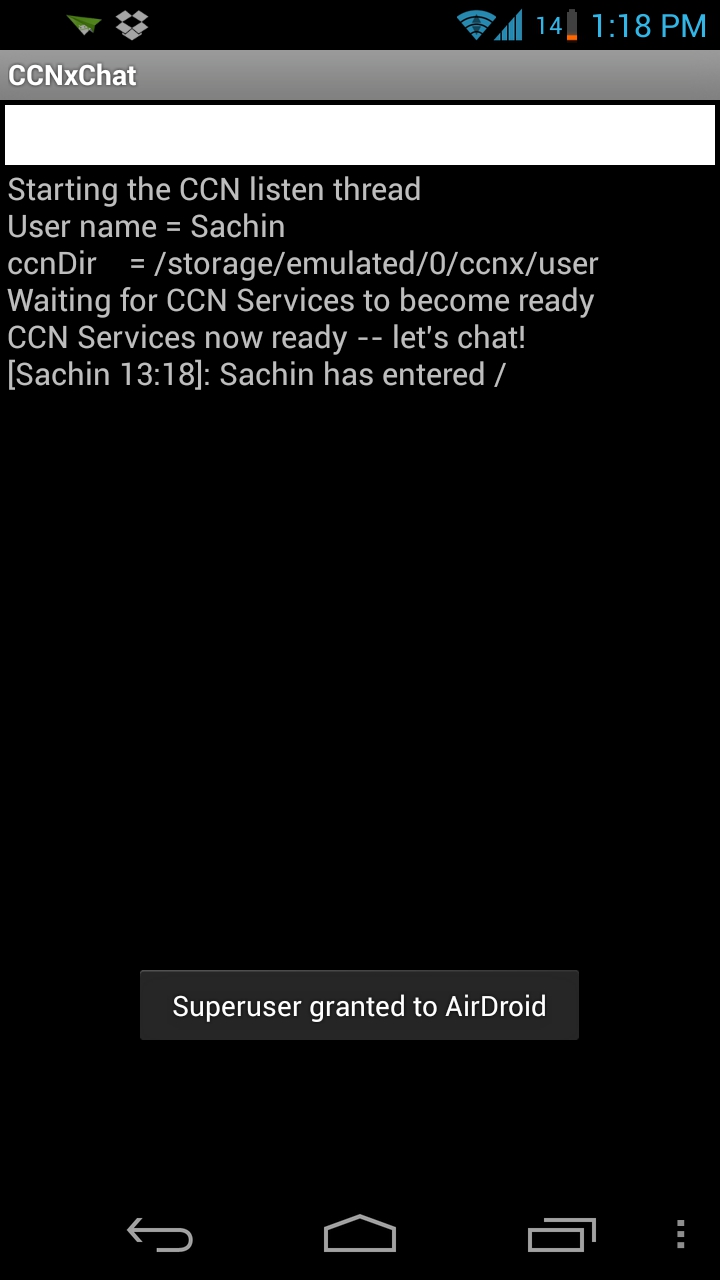
You are now brought to the NDNLDC control screen as shown below.

In this screen you can create remote faces by providing the MAC address of the remote machine. You can also mention the prefixes to be registered on the created face.



**Step 4:**

Launch your application (ccnChat is shown below) and use the same prefix that was reistered on NDNdroid in Step 3. Packets sent for these prefixes will now be sent to the destination over Ethernet packets.



**Interfacing NDNDroid with other applications**

NDNDroid launches a background service that other applications can connect to for performing different NDNLP operations.

Reference: <http://mindtherobot.com/blog/37/>

The following APIs are exposed by the background service:

**string startNDNBackgroundService()**: This function checks if the ***ndnld*** user-space daemon is running and launches it if it is not. It returns the output of launching the user daemon.

**string addNewConnection(String mac, String prefix):** This function creates a face for sending packets directly over ethernet(wifi) and registers the requested prefix. It returns the output of the operation.

**boolean resetServices():** Thisfunction relaunches the ***ndnld*** user-spacedaemon and re-creates all the faces that were created using addNewConnectionand also registers their prefixes. It returns a boolean value to indicate success or failure.

**void stopServices():** This function removes all the faces that were registered using addNewConnection.

**String runNdnldcCommand(String command) :** This function takes the arguments to the *ndnldc* command as a single string and runs the *ndnldc* command on the phone. We may execute any supported *ndnldc* command using this API. The result of running the command will be returned as a string.