Unity Network, how-to setup the platform guide

This guide will cover the platform setup and may help an admin setup unity network either on the **Internet (WAN)** or on **a local network (LAN)** under two modes: one is **full network** and the other is with a **standalone blue node**.

A. Using a Standalone Blue Node

A standalone Blue Node is the easiest mode to setup. However, this network form may only be used a limited number of hosts as there is only one bluenode to carry out the traffic.

Your network should distribute addresses under the range of 192.168.x.x this is due to the virtual network which makes use of 10.x.x.x network addresses for the virtual hosts.

Let's say our LAN network distributes IP addresses of 192.168.1.x

In one host ex. 192.168.1.6 **Download** a Blue Node by following the instructions on Blue Node's git page:

https://github.com/kostiskag/unitynetwork-bluenode

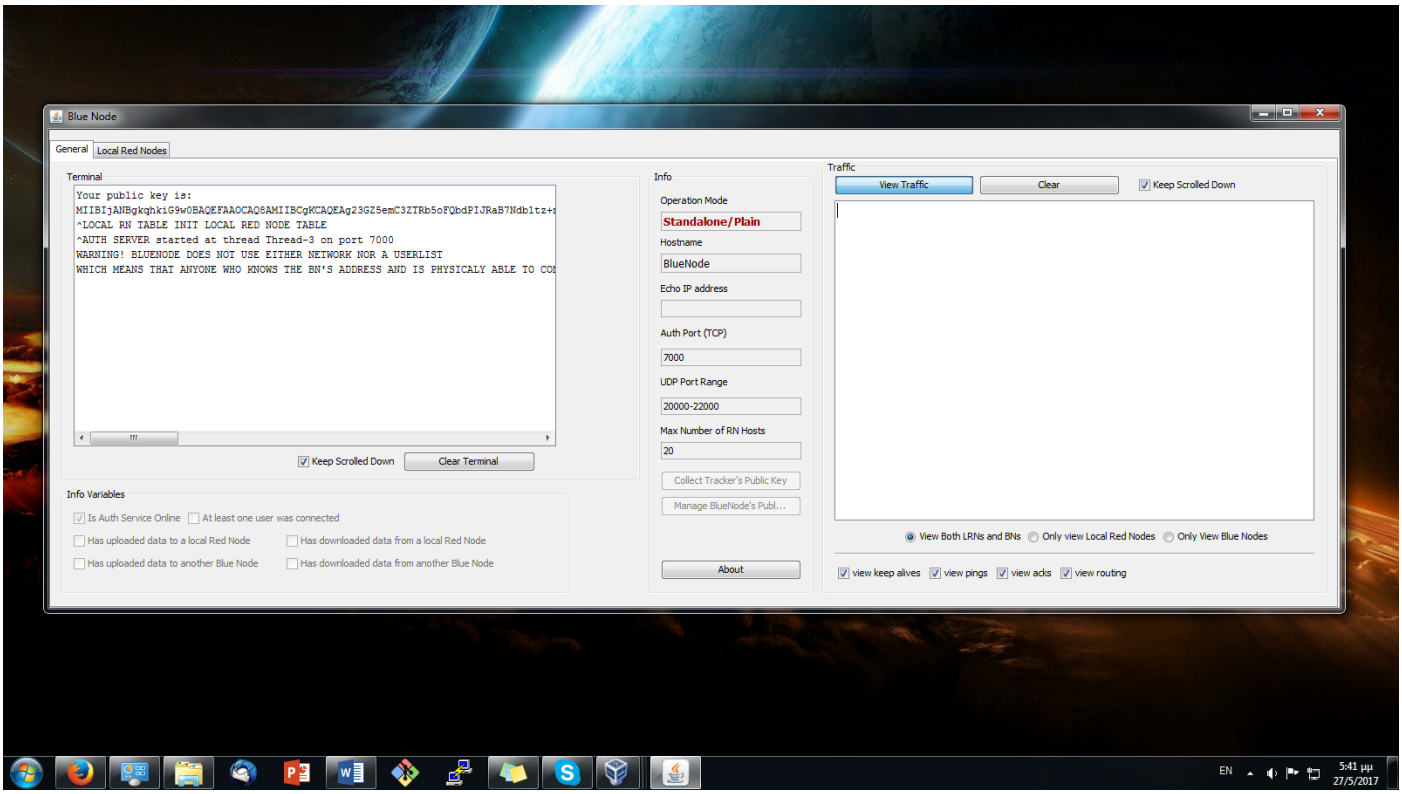
When you have your setup complete, open **bluenode.conf** file with a text editor and select:

Network = false

You may either use a host list or not with the option:

UseHostList = true

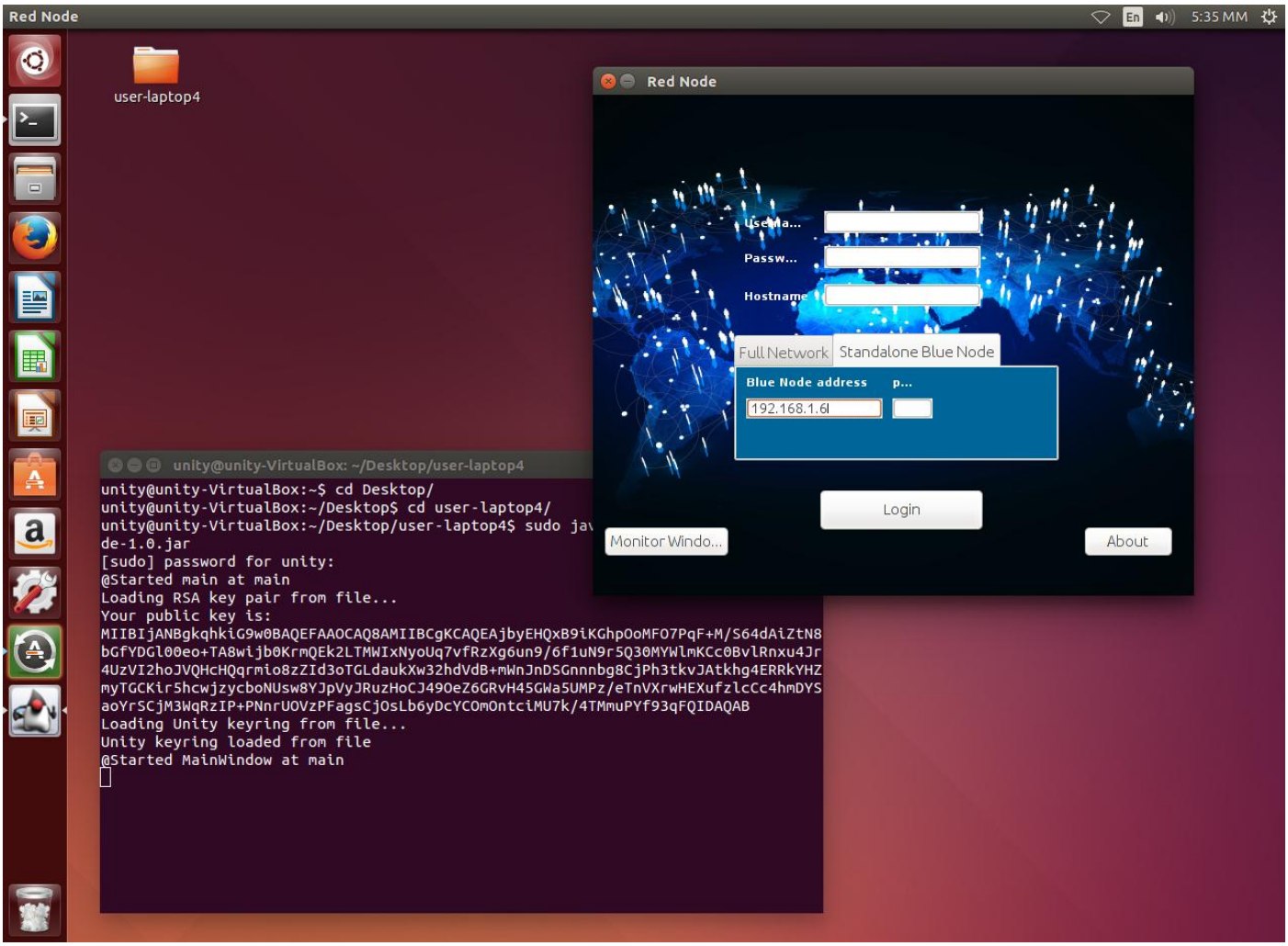
If you decide to use a list of allowed hosts you may define them in **host.list** file otherwise the bluenode may allow all connecting rednodes to join.



Then, each other user with a LAN host may **download** and **run** the rednode app to connect to the network from its git repository.

https://github.com/kostiskag/unitynetwork-rednode

Users may start the application and click the **Standalone Blue Node** tab. When a user list is selected, users should give their credentials, otherwise they may leave the fields empty or provide a specific hostname.



Port Forwarding for WAN use

To let hosts from the Internet running rednodes join in the network you should configure your router/gateway's NAT to forward the below bluenode's ports.

The TCP auth port which Is by default 7000.

The UDP port range from 20000 – 22000.

You may change the default ports from the **bluenode.conf** file.

You do not need port forwarding for the individual rednodes.

The rednode users should know the bluenode's public IP address or domain in order to login to the virtual network.

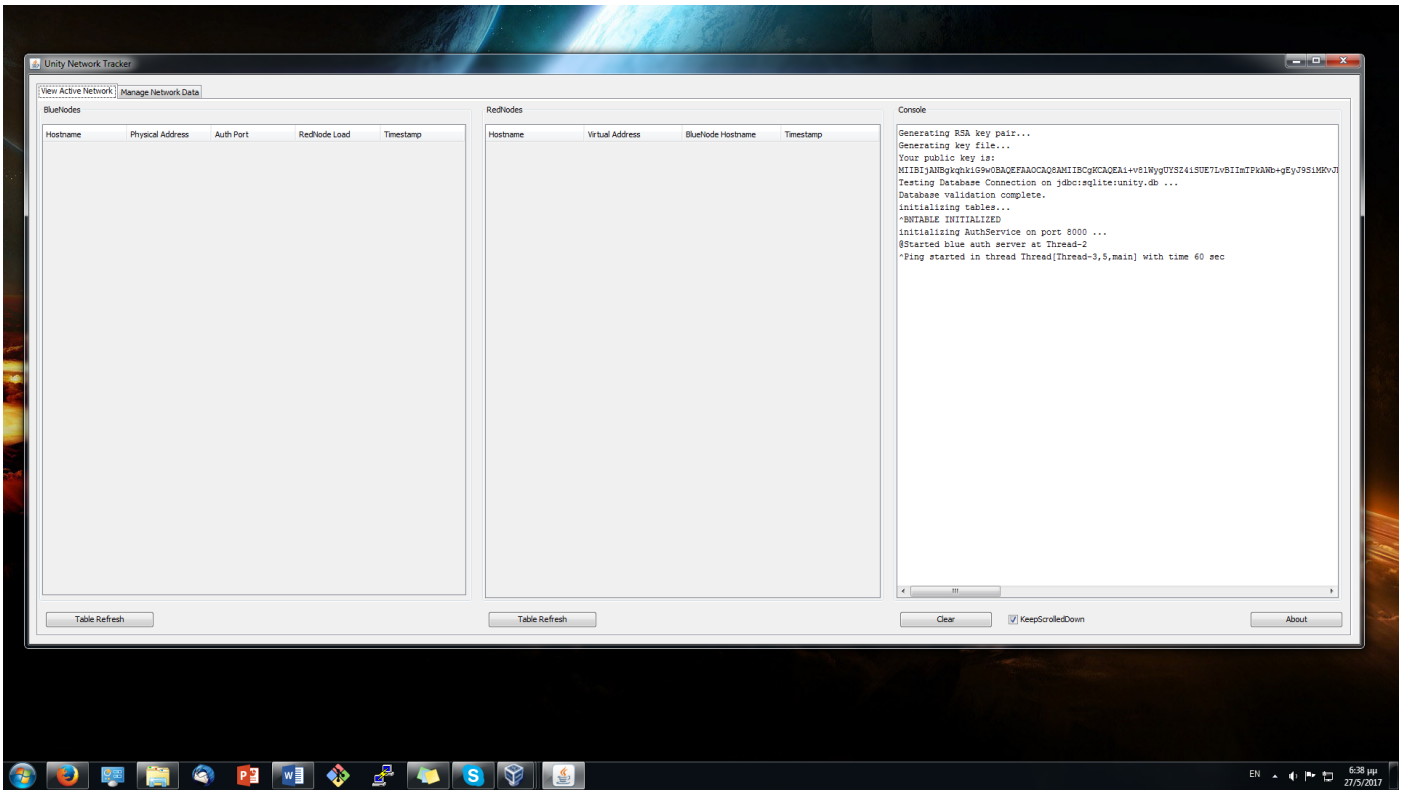
B. Use Unity in full network mode

Unity network is managed by a **tracker** app which is responsible to authenticate bluenodes and rednodes, keep track of logged in members and distribute public keys through the platform. However, the tracker is not responsible to forward any kind of network traffic. Unity as a platform offers resilience in the form of: bluenodes are responsible to forward traffic and may be connected, disconnected in any given moment. The network needs at least one bluenode to be operational and the more bluenodes join in the bigger the network's capacity for virtual hosts becomes.

Therefore, the first task at hand is to **download**, and **run** the tracker application in a host. For this task, you may consult the readme guide in the tracker's repository.

https://github.com/kostiskag/unitynetwork-tracker

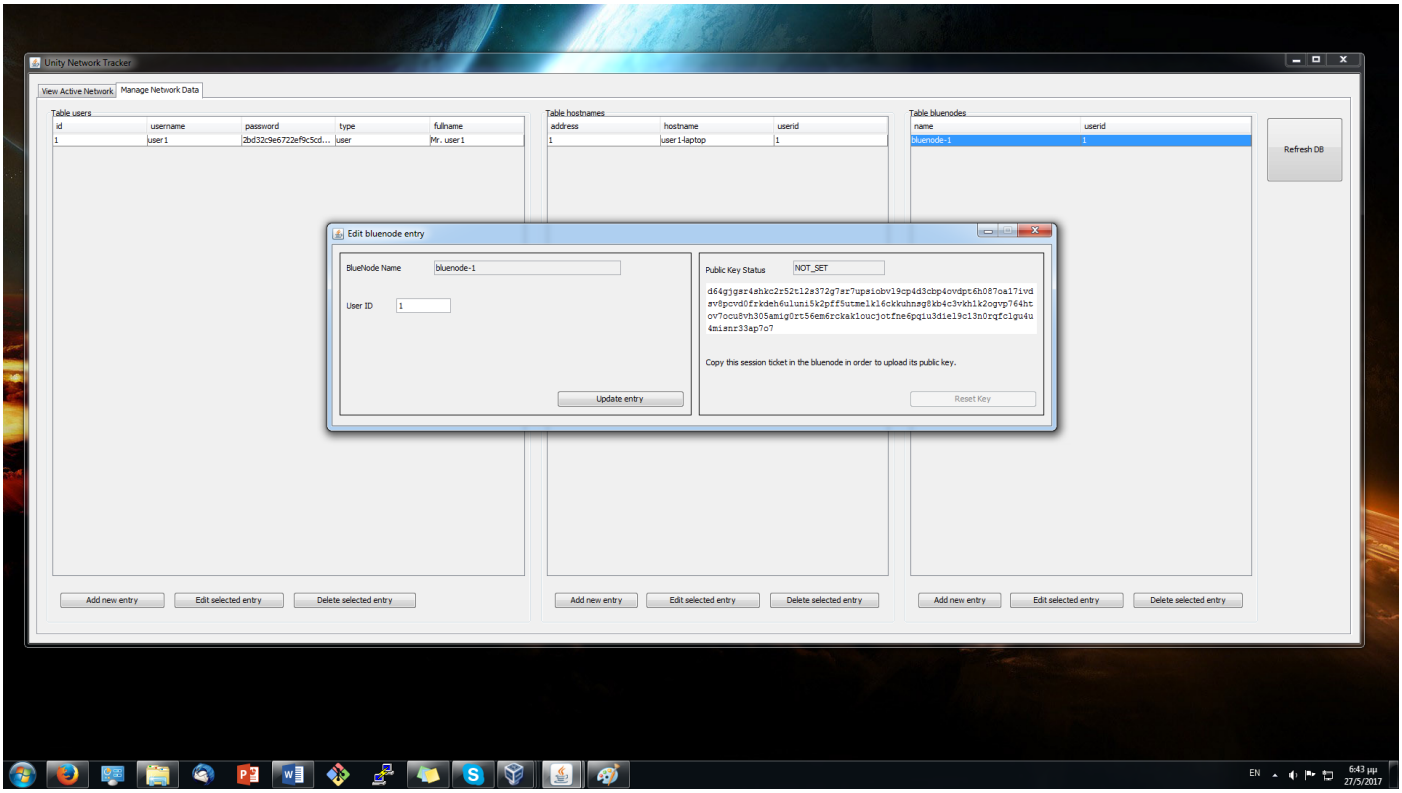
The tracker is operational by default and does not require extensive configurations. You may select to define another port or to use an external database as well as other options from the **tracker.conf** file.



There are two tabs, in the first, an admin may view the active network and the connected bluenodes and rednodes in real time. In the second, an admin may define user accounts, bluenode and renode entries. For each new entry, a dialog box may be popped to let the admin configure the entry.

Tracker collecting Public Keys

In order for a rednode or a bluenode to be operational: admins should send the generated ticker in a bluenode's or a rednode's user in order to upload the node's public key as the keypair is being generated into the client's side. This task would have easily been done through a web interface should the admin choose to build one. However, in our case the admin may send the tickets via email. A ticket will be burned after a blue or red node has uploaded its public key to the tracker and a new one will be generated if the user decides to revoke the public key. A Blue or red node is operational only when a public key has been uploaded.



Port Forwarding for WAN use

In order to let the tracker be visible on WAN you need to forward its TCP auth port default: 8000 through your router's NAT towards the tracker's host IP address.

Each bluenode should forward its ports as described under section A.

Rednodes do not need to forward any ports.

Blue Node setup for full network

On the tracker's side:

- Create or use an existing user account for the bluenode's holder.
- Create a bluenode entry.
- From the new entry, copy and transfer the bluenode ticket to the bluenode's user.

On the bluenode's side:

Edit **bluenode.conf** and set:

Network = true

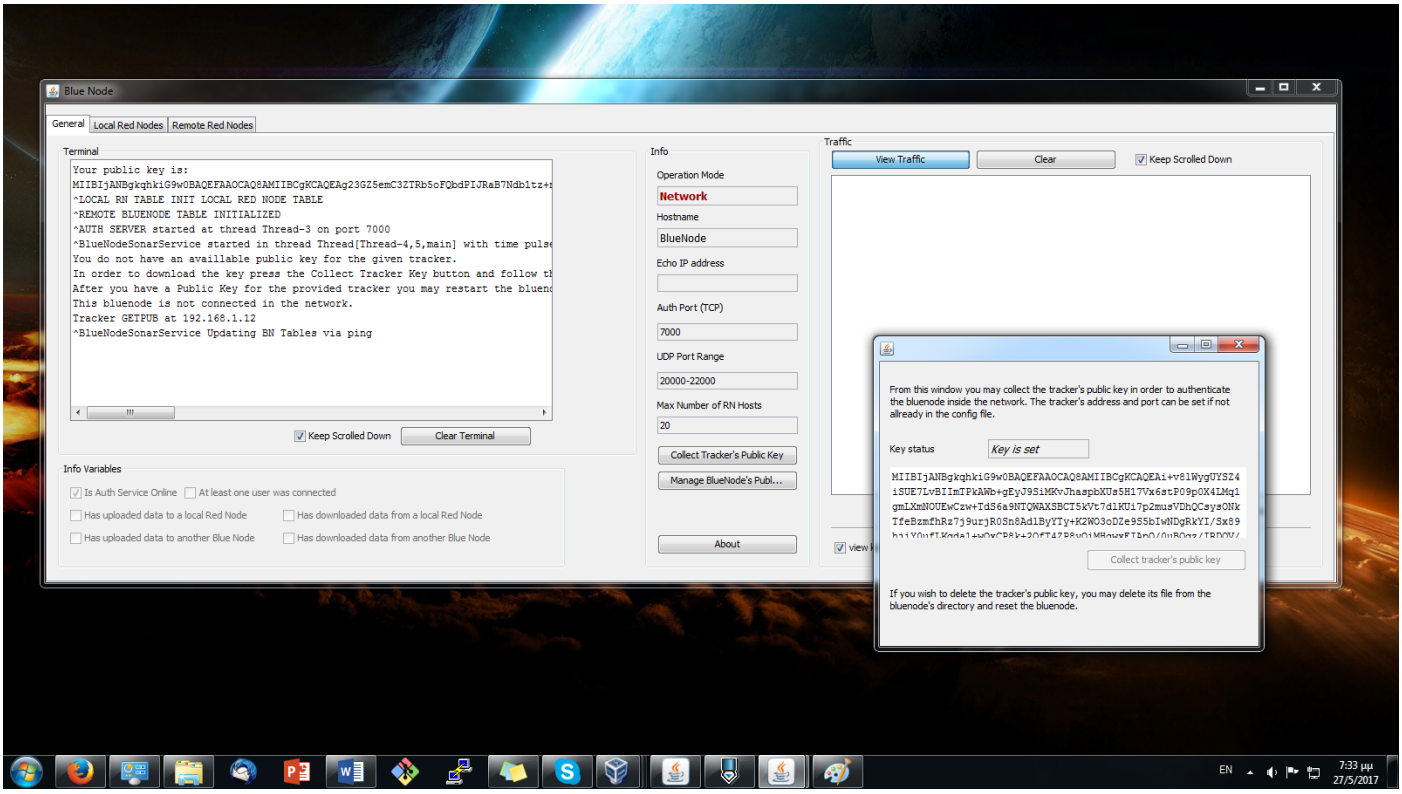
UnityTrackerAddress = [the tracker's address]

UnityTrackerAuthPort = [the tracker's port]

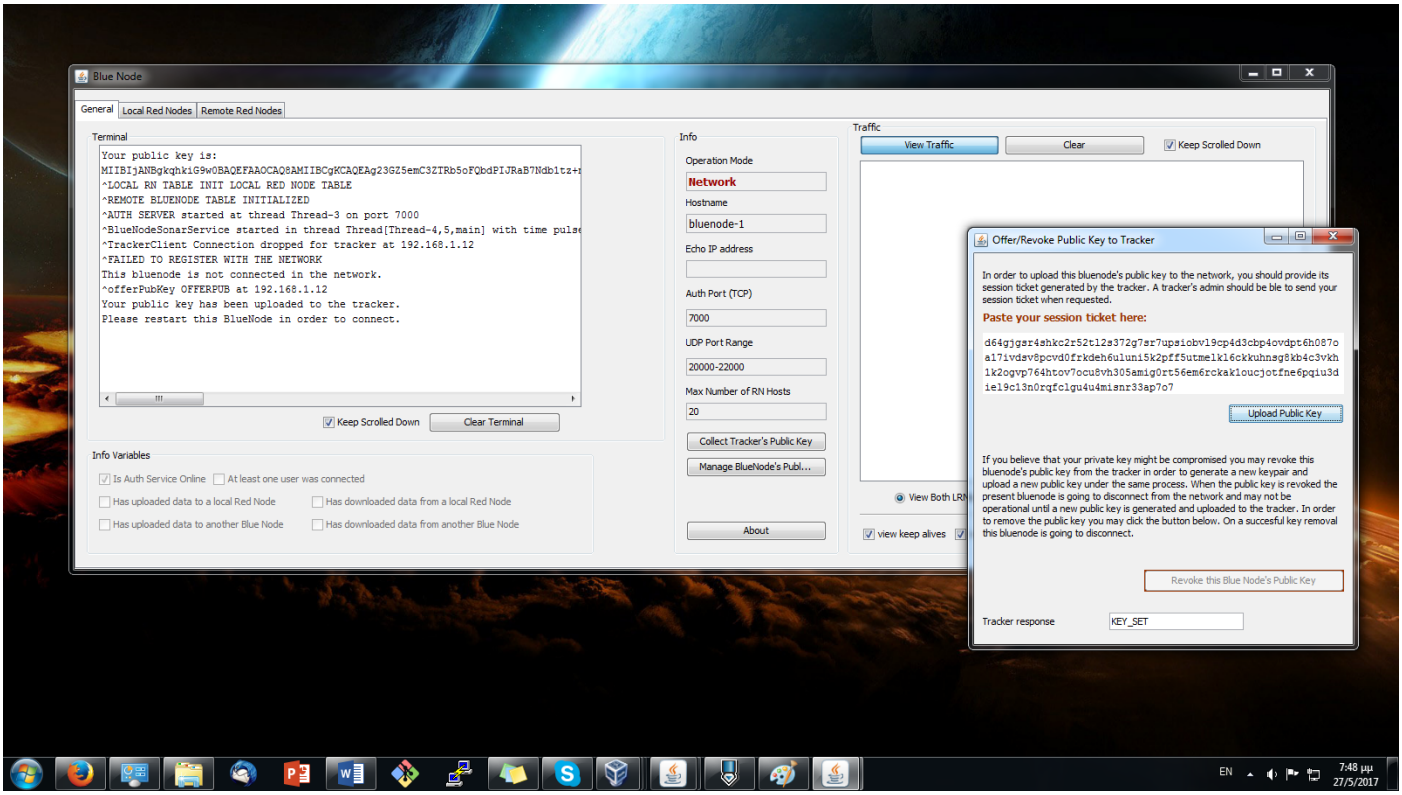
Name = [the registered bluenode name]

Start the bluenode in order to generate a keypair, you may delete the keypair file in order to generate another.

Click the button **Collect Tracker's public key** to collect the tracker's public key.



After that, click manage **bluenode's public key** button.



Restart the bluenode to join it in the network.

Red Node setup for full network mode

One rednode may connect to multiple networks and multiple standalone bluenodes. For this reason and for the user's convenience a keyring was created to hold and manage the networks.

Users should:

A. Select the **Full Network** tab and click the **Keyring** button.
B. Create a new keyring instance for each network by specifying a tacker's address and port.
C. Collect a tracker's public key for a given network instance.
D. Upload the rednode's public key to the selected tracker by making use of a provided ticket.

After the above setup is done, users may connect to the same network by filling in only their credentials each time.