

SEAMS ARTIFACT

---

## **DeltaIoT Installation Guide**

---

March 21, 2017

Gowri Sankar Ramachandran  
Muhammad Usman Iftikhar

## Introduction

DeltaIoT is an artifact of Internet-of-Things (IoT) for research on self-adaptation. The artifact is hosted at DistriNet, KU Leuven, Belgium. We encourage researchers working on self-adaptation to use the artifact for validating their approaches. Currently, we provide access to the artifact via a VPN connection.

VPN is an expansion for *Virtual Private Network*, and it connects your computer to the network of DeltaIoT artifact. Once the VPN connection is established, your computer can interact with DeltaIoT artifact following the user's guide <sup>1</sup>.

We start with describing the steps involved in setting up the VPN connection on your machine. Then we provide details about how to get the credentials to get access to the DeltaIoT network. Finally, we explain how to install the DeltaIoT package.

## Setting up OpenVPN in your own machine

To setup OpenVPN at your machine, the user has to follow the guidelines given below.

- **Step 1:** For accessing the DeltaIoT server, the user's machine has to be on the same network as the DeltaIoT server. In our current version, we use OpenVPN to configure the user's machine and to connect to the network of DeltaIoT server. To use OpenVPN, the user has to download OpenVPN client. An appropriate client must be downloaded and installed on the user's machine. We provide the options for linux, windows and MAC operating systems below.
  - **Linux (debian) :** OpenVPN client can be installed by issuing the following command.

```
sudo apt-get install openvpn
```

The above command will install both the OpenVPN server and client.

- **MAC:** MAC users can download and install TunnelBlick from the following link:

```
https://tunnelblick.net/downloads.html
```

The above link provides the installation image for TunnelBlick.

---

<sup>1</sup>[https://people.cs.kuleuven.be/~danny.weyns/software/DeltaIoT/DeltaIoT\\_User\\_Guide.pdf](https://people.cs.kuleuven.be/~danny.weyns/software/DeltaIoT/DeltaIoT_User_Guide.pdf)

- **Windows:** The OpenVPN client for Windows can be downloaded from the following link:

<https://openvpn.net/index.php/open-source/downloads.html>

The user has to download the installation files from the above link and follow the instructions on the web page to install OpenVPN in Windows.

- **Step 2:** After installing the OpenVPN client, the user needs to start the client and choose the OpenVPN configuration files (check the Downloads section). We provide the operating system specific instructions below.

- **Linux (debian) :** OpenVPN creates a configuration folder in `\etc \openvpn`. The user should copy the contents of the `openvpn_keys` folder (downloaded folder) to `\etc \openvpn`. Then, start `openvpn` by issuing the following command in the terminal:

```
\etc \init.d \openvpn start
```

- **MAC:** MAC users have to create a folder `\etc\openvpn` and copy the contents of the `openvpn_keys` folder to it. Then, the user should open TunnelBlick and choose `\etc\openvpn \lora05.conf`. TunnelBlick is now configured. By clicking "connect to Lora05", the user's MAC connects to the network of the DeltaIoT server.
- **Windows:** In case of Windows, the user should open the `openvpn_keys` folder from the `openvpn-client`. The client software will guide the user and will establish a connection to the network of DeltaIoT server.

We recommend the user to contact one of the maintainers of DeltaIoT in case of technical issue (see the DeltaIoT website<sup>2</sup>).

- **Step 3:** The user has to check that OpenVPN is running and the machine has the following ip address: `172.21.0.15`. This can be verified by issuing `ifconfig` command in the terminal. The output of `ifconfig` should show `tap0` interface with `172.21.0.15` as its inet addr, as shown in Figure 1. Furthermore, the user should be able to ping `172.21.0.11` from its terminal to make sure that the network is correctly setup. Otherwise check the installations steps 1 and 2 again.

<sup>2</sup><https://people.cs.kuleuven.be/~danny.weyns/software/DeltaIoT/>

```
tap0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    ether
    inet 172.21.0.12 netmask 0xffffffff broadcast
    inet :          netmask 0xffff0000 broadcast 169.254.255.255
    media: autoselect
    status: active
    open (pid 15146)
```

**Figure 1:** Example *tap0* configuration.

Once the client machine is connected to DeltaIoT, credentials are required that allows access. In the next section, we discuss how the credentials can be acquired.

## Get credentials to access the DeltaIoT network

As the DeltaIoT network can only be self-adapted by a single client at any point in time, the access to the network is regulated for users. To obtain the right to access the network a user needs to obtain credentials that she/he has to include in a `config.properties` file (the **User Guide** provides the details to do this). Credentials provide in principle access to the DeltaIoT network for 24 hours. Credentials can be requested by sending an email to contact persons provided on the DeltaIoT website. A request should contain: name and affiliation of the user, a short description of the usage scenario, and the date and time required for the experiment.

In the next section , we discuss contents of DeltaIoT package and how to use these credentials .

## DeltaIoT Package

The DeltaIoT package is available for download from the Download section of the DeltaIoT website. The package is available as a zip file, containing the source and executable files of DeltaIoT. To perform experiments with DeltaIoT, two sub-packages are provided:

1. **DeltaIoTClient:** The `DeltaIoTClient` provides the software to implement your own adaptation logic and remotely connect with the physical IoT system using probes and effectors to perform adaptations.
2. **Simulator:** The Simulator as the name suggests enables simulation of the DeltaIoT network (on your local machine). Thus, the simulator enables offline experiments without connecting with the real DeltaIoT.

The `Simulator` and the `DeltaIoTClient` have the same interface, which allows seamless integration with the simulator and the physical IoT system. The simulator is useful for debugging, testing and performing offline experiments before using the real set up (which is more time consuming).

Both the `DeltaIoTClient` and the `Simulator` are included in the zip file of the DeltaIoT package. The zip file contains the following folders:

1. `DeltaIoT-Source`
2. `DeltaIoT-Executables`

We now briefly discuss the sources and how to use the executables.

## Source files

The source folder is an Eclipse workspace that contains several Java projects. The source folder can easily be used by Eclipse software ([www.eclipse.org](http://www.eclipse.org)). To use the source folder, after installing Eclipse, run Eclipse and select the source folder as workspace. The source folder contains the following projects:

1. **DelataIoTClient:** `DeltaIoTClient` is a framework that enables the adaptation logic to monitor and adapt the real IoT system.
2. **DelataIoTGUI:** A graphical user interface that allows visualising the current topology of the DeltaIoT.
3. **SimpleAdaptationWithDeltaIoTClient:** A simple example of adaptation logic that uses the `DeltaIoTClient` framework to demonstrate how the real IoT system can be adapted.
4. **Simulator:** Simulator software that enables offline experiments of the simulated DeltaIoT network on a local machine.
5. **SimpleAdaptationWithSimulation:** A simple example of adaptation logic that uses the `Simulator` for performing adaptation experiments in simulation.
6. **SimulatorGUI:** A Graphical user interface that allows to visualize simulation results with and without adaptation. `SimpleAdaptationWithSimulation` shows the adaptation results graphically.

The `DeltaIoTClient` folder contains a `config.properties` file. As we already discussed above, access to the DeltaIoT network requires credentials that can be obtained via email to contact persons of the DeltaIoT network. The credentials including the authorization token and URL should be saved in the `config.properties` file before using the DeltaIoT network.

## Executable files

The Executables folder contains two subfolders `DeltaIoT` and `Simulator`, which separates the executables of both the `DeltaIoTClient` and the `Simulator`.

### DeltaIoT folder:

The `DeltaIoT` folder contains the executable of `DeltaIoTGUI`. Before executing this file, the user should make sure that the credentials are saved in the `config.properties` file in the folder as explained above.

To execute the `DeltaIoTGUI`, open a terminal and go to the `DeltaIoT` folder inside the `DeltaIoT-Executables` folder using the following command:

```
cd DeltaIoT-Executables/DeltaIoT
```

After that run the following command to execute the `DeltaIoTGUI`.

```
java -jar DeltaIoTGUI.jar
```

Once the above command is executed, a graphical user interface (GUI) will appear. In the **User Guide** we discuss the graphical user interface of `DeltaIoTGUI` in detail.

### Simulator folder:

The `Simulator` folder contains the executable of the `SimulatorGUI` software. The `Simulator` graphical user interface (GUI) enables visualising the results of the simulated DeltaIoT network with and without adaptation. To view the results of adaptation, we use the `SimpleAdaptationWithSim` example.

To run the GUI, open the terminal and go to the `DeltaIoT-Executables/Simulator` folder using the `cd` command.

```
cd DeltaIoT-Executables/Simulator
```

After that run the following command in a terminal:

```
java -jar SimulatorGUI.jar
```

Once this command is executed a graphical user interface will appear which is further discussed in the **User Guide**.