**FLOW a Named Data Networking IoT Demo**

**by UCLA REMAP**

**Documentation for OpenPTrack, Unity and NDN components**

Table of Contents

[Contact Information 3](#_Toc471475736)

[Source Code and Versioning 3](#_Toc471475737)

[Named Data Networking (NDN) 3](#_Toc471475738)

[Dependencies 3](#_Toc471475739)

[Framework 3](#_Toc471475740)

[Application 3](#_Toc471475741)

[RFDuino 3](#_Toc471475742)

[Rasberry Pi 3](#_Toc471475743)

[Mobile Phone 3](#_Toc471475744)

[OpenPTrack (OPT) 4](#_Toc471475745)

[Unity 4](#_Toc471475746)

[Hardware & System Configuration 5](#_Toc471475747)

[System Diagram 5](#_Toc471475748)

[Equipment List 6](#_Toc471475749)

[Interaction 7](#_Toc471475750)

[Visualization 7](#_Toc471475751)

[Networking 7](#_Toc471475752)

[TeamViewer Access 7](#_Toc471475753)

[Startup Guide 8](#_Toc471475754)

[Installation Guide 9](#_Toc471475755)

[Named Data Networking 9](#_Toc471475756)

[Sensors & Devices 9](#_Toc471475757)

[RFDuino 9](#_Toc471475758)

[Gyroscope 9](#_Toc471475759)

[Raspberry Pi 9](#_Toc471475760)

[Mobile Phone 9](#_Toc471475761)

[OpenPTrack 9](#_Toc471475762)

[Calibration 9](#_Toc471475763)

[Calibration Refinement 10](#_Toc471475764)

[Detection and Tracking 11](#_Toc471475765)

[Troubleshooting (Hardware & Software) 16](#_Toc471475766)

[Error: No kinect2 devices found 16](#_Toc471475767)

[OpenPTrack is started, but there is no tracking 18](#_Toc471475768)

[Network Troubleshooting 19](#_Toc471475769)

[NTP Synchronization 19](#_Toc471475770)

[Unity 20](#_Toc471475771)

# **Contact Information**

Jeff Burke: [jburke@remap.ucla.edu](mailto:jburke@remap.ucla.edu)

Zhehao Wang: [zhehao@remap.ucla.edu](mailto:zhehao@remap.ucla.edu)

Zoe Sandoval: [zoe@remap.ucla.edu](mailto:zoe@remap.ucla.edu)

# Source Code and Versioning

The three main software components of FLOW are:

## Named Data Networking (NDN) forwarder and libraries

* [**ndn-cxx**](https://github.com/named-data/ndn-cxx)
* [**NFD**](https://github.com/named-data/nfd)
* **ndn-ccl:** 5 libraries in Python, C++, JS, C# and Java, and needed respectively for each NDN-IoT framework library, explained below)

### Named Data Networking Internet of Things (NDN-IoT) Framework

The NDN IoT framework libraries (Python, C++, JS, and C#) are in Github under “framework” folder: <https://github.com/remap/ndn-flow/tree/master/framework>.

An overview of functionalities can be found here: <https://github.com/remap/ndn-flow/tree/master/framework#functionalities>, and a detailed interface description can be found here: https://github.com/remap/ndn-flow/tree/master/design/docs.

Each piece’s own dependencies, installation guide, and examples can be found in their own folders respectively.

* **ndn-iot-dot-net (C# library) :** https://github.com/remap/ndn-flow/tree/master/framework/ndn\_iot\_dot\_net
* **ndn-iot-js (JavaScript library):** https://github.com/remap/ndn-flow/tree/master/framework/ndn\_iot\_js
* **ndn-iot-cpp (C++ library):** https://github.com/remap/ndn-flow/tree/master/framework/ndn\_iot\_cpp
* **ndn-iot-python (Python library):** https://github.com/remap/ndn-flow/tree/master/framework/ndn\_iot\_python

## Application

The *FLOW* application components are described in Github under “application folder”: <https://github.com/remap/ndn-flow/tree/master/application>. Each component’s functionalities, required devices, and installation guide can be found in their own folders respectively.

### RFDuino

* **The RFduino code :** <https://github.com/remap/ndn-flow/tree/master/application/rfduino>

### Rasberry Pi

* **The Raspberry Pi helper for RFduinos:** https://github.com/remap/ndn-flow/tree/master/application/rfduino/rpi\_helper
* **The Raspberry Pi controller:** https://github.com/remap/ndn-flow/tree/master/framework/ndn\_pi

### Mobile Phone

* **The mobile website:** <https://github.com/remap/ndn-flow/tree/master/application/website>

### OpenPTrack (OPT)

OpenPTrack can be found on Github: <https://github.com/OpenPTrack/open_ptrack>. For this installation the **development** branch was used. Specifically commit [55fa9ef0697a335f5279329d4b72c38bec442324](https://github.com/OpenPTrack/open_ptrack/commit/55fa9ef0697a335f5279329d4b72c38bec442324) was used in this installation.

### Unity

The installation was tested on Unity version 5.3.2.f1 and running on Mac OS X version 10.11.5.

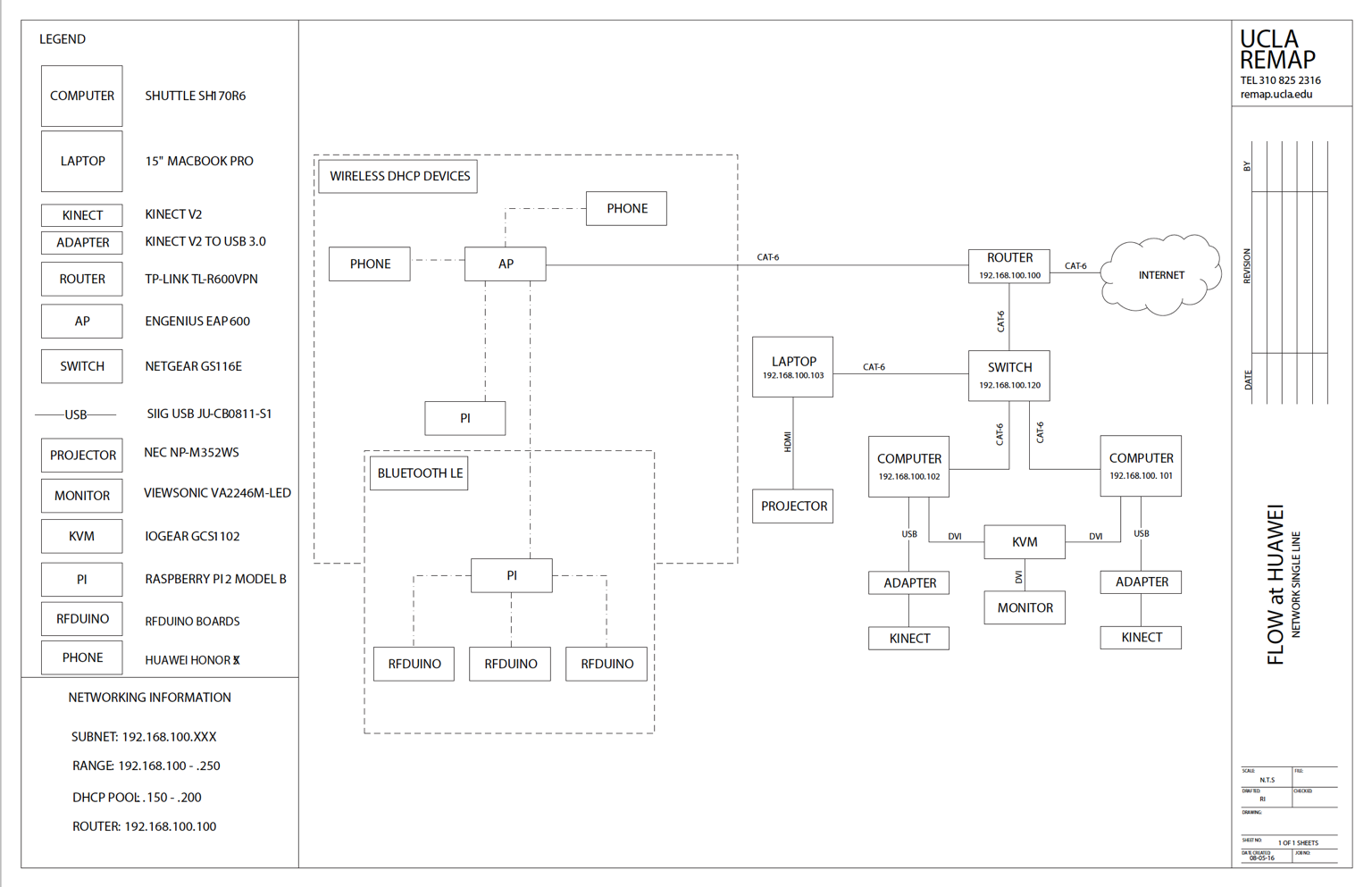
Detailed information can be found on Github: <https://github.com/remap/ndn-flow/tree/master/application/unity/WWBlimp>

Dependencies

* [ndn-cxx](https://github.com/named-data/ndn-cxx), [NFD](https://github.com/named-data/nfd)
* [PyNDN](https://github.com/named-data/PyNDN2), [IoT framework (Python device bootstrap, C# library)](https://github.com/remap/ndn-flow/tree/master/framework)

# Hardware & System Configuration

## System Diagram



SUMMARY ABOUT SYS CONFIG

## Equipment List

|  |  |  |
| --- | --- | --- |
| Quantity | Part Number | Manufacturer |
| **OpenPTrack** | | |
| 2 | SH170R6 16gb ram | Shuttle |
| 2 | [One Kinect Sensor](https://www.amazon.com/Xbox-One-Kinect-Sensor/dp/B00NMST9G8/ref=sr_1_2?ie=UTF8&qid=1465944392&sr=8-2&keywords=kinect+v2) | Microsoft |
| 2 | [Kinect Adapters for Windows](https://www.amazon.com/Microsoft-9J7-00001-Kinect-Adapter-Windows/dp/B00NMSHT7E/ref=sr_1_1?ie=UTF8&qid=1465944391&sr=8-1&keywords=kinect+v2) | Microsoft |
| 2 | [60’ USB 3.0 Extensions](https://www.amazon.com/SIIG-Active-Repeater-20-Meters-JU-CB0811-S1/dp/B0053YLYLC/ref=sr_1_1?ie=UTF8&qid=1465944449&sr=8-1&keywords=SIIG+USB+JU-CB0811-S1) | SIIG |
| 1 | [Gigabit Broadband VPN Router](https://www.amazon.com/TP-LINK-TL-R600VPN-Gigabit-Broadband-Supports/dp/B007B60SCG/ref=sr_1_1?ie=UTF8&qid=1465944493&sr=8-1&keywords=tp-link+vpn+router) | TP-LINK |
| 1 | [GS316 Network switch](https://www.amazon.com/NETGEAR-16-Port-Gigabit-Ethernet-Desktop/dp/B01AX8XHRQ/ref=sr_1_2?ie=UTF8&qid=1465944519&sr=8-2&keywords=16-port+netgear+switch) | Netgear |
| 2 | [Pavilion HDMI VGA Monitor](https://www.amazon.com/HP-Pavilion-21-5-Inch-HDMI-Monitor/dp/B015WCV70W/ref=sr_1_1?s=pc&ie=UTF8&qid=1468523513&sr=1-1&keywords=monitor) | HP |
| 2 | [M500 Corded Mouse](https://www.amazon.com/Logitech-910-001204-Corded-Mouse-M500/dp/B002B3YCQM?ie=UTF8&keywords=logitech%20wired%20mouse&qid=1464966736&ref_=sr_1_1&sr=8-1) | Logitech |
| 2 | [K120 Keyboard](https://www.amazon.com/Generic-920-002478-Logitech-Keyboard-K120/dp/B003ELVLKU/ref=sr_1_1?ie=UTF8&qid=1468953299&sr=8-1&keywords=keyboard) | Logitech |
| 4 | [Power Strips](https://www.amazon.com/gp/product/B000UD4LIY/ref=oh_aui_search_detailpage?ie=UTF8&psc=1) | Tripp Lite |
| 4 | [Extension Cords](https://www.amazon.com/gp/product/B000I97ME0/ref=oh_aui_search_detailpage?ie=UTF8&psc=1) | CableWholesale |
| 1 | [Cat6 Ethernet Cable](https://www.amazon.com/Cable-Matters-5-Pack-Snagless-Ethernet/dp/B00C2CA3N8/ref=sr_1_4?ie=UTF8&qid=1468685261&sr=8-4&keywords=cat+6+10+ft) | Cable Matters |
| 6 | [Cat6 Ethernet Cable](https://www.amazon.com/gp/product/B00G9BN9KW/ref=oh_aui_search_detailpage?ie=UTF8&psc=1) | Cable Matters |
| 2 | [VGA Cables](https://www.amazon.com/Fullink-Premium-Connectors-Computer-Projector/dp/B00JK093S6/ref=sr_1_2?s=electronics&dd=hCB56rF2DSH53n8jGFqrkg%2C%2C&ddc_refnmnt=pfod&ie=UTF8&qid=1468523993&sr=1-2&keywords=vga+cable&refinements=p_97%3A11292772011) | Fullink |
|  |  |  |
| **FLOW** | | |
| 1 | [15" Macbook Pro & Ethernet Dongle](http://www.apple.com/shop/buy-mac/macbook-pro?product=MJLQ2LL/A&step=cto_accessories) | Apple |
| 1 | [EAP600 Wireless Access](https://www.amazon.com/gp/product/B009V17BYC/ref=oh_aui_search_detailpage?ie=UTF8&psc=1) | Engenius |
| 2 | [Cat6 Ethernet Cable](https://www.amazon.com/Cable-Matters-5-Pack-Snagless-Ethernet/dp/B00C2CA3N8/ref=sr_1_4?ie=UTF8&qid=1468685261&sr=8-4&keywords=cat+6+10+ft) | Cable Matters |
| 2 | [Cat6 Ethernet Cable](https://www.amazon.com/gp/product/B00G9BN9KW/ref=oh_aui_search_detailpage?ie=UTF8&psc=1) | Cable Matters |
| 2 | [Power Strips](https://www.amazon.com/gp/product/B000UD4LIY/ref=oh_aui_search_detailpage?ie=UTF8&psc=1) | Tripp Lite |
| 2 | [Extension Cords](https://www.amazon.com/gp/product/B000I97ME0/ref=oh_aui_search_detailpage?ie=UTF8&psc=1) | CableWholesale |
| 2 | [Rasberry Pi 2 - Model B](https://www.amazon.com/CanaKit-Raspberry-Complete-Starter-9-Items/dp/B008XVAVAW/ref=sr_1_6?s=pc&ie=UTF8&qid=1469551173&sr=1-6&keywords=raspberry+pi+2+model+b) | Rasberry Pi |
| 2 | [Bluetooth Dongle](https://www.amazon.com/Plugable-Bluetooth-Adapter-Raspberry-Compatible/dp/B009ZIILLI/ref=sr_1_4?s=electronics&ie=UTF8&qid=1469465297&sr=1-4&keywords=bluetooth+dongle) | Plugable |
| 1 | [8 GB SD Cards](https://www.amazon.com/SDSDU-008G-U46-SanDisk-Ultra-Class-Memory/dp/B00812K4V4/ref=pd_sim_147_2?ie=UTF8&dpID=41OFENyKxzL&dpSrc=sims&preST=_AC_UL160_SR160%2C160_&psc=1&refRID=FWJ649684PEABA34TK3X) | Sandisk |
| 3 | [Rfduino boards](https://www.amazon.com/Bluetooth-802-15-1-Modules-RFduino-Module/dp/B0131V8BA0) | RF Digital Wireless |
| 1 | [USB Shield](https://www.amazon.com/Interface-Development-Tools-Rfduino-Shield/dp/B0131UWML0) | RF Digital Wireless |
| 3 | [AAA Battery Shields](https://www.amazon.com/Power-Management-Development-Rfduino-Shield/dp/B0131UNZZC) | RF Digital Wireless |
| 1 | [Alkaline AAA Batteries](https://www.amazon.com/procell-Alkaline-Battery-aaa-Volt/dp/B004E2KQ0O/ref=sr_1_1?ie=UTF8&qid=1469492429&sr=8-1&keywords=1.5+AAAbatteries) | Procell |
| 3 | [Gyroscopes](https://www.amazon.com/gp/product/B00NH8Z6BU/ref=oh_aui_search_detailpage?ie=UTF8&psc=1) | SMAKN |
| 1 | [Breadboards](https://www.amazon.com/Frentaly%C2%AE-Solderless-BreadBoard-tie-points-power/dp/B01258UZMC/ref=sr_1_6?s=pc&ie=UTF8&qid=1469465667&sr=1-6&keywords=breadboard) | Frentayl |
| 1 | [120pc Multicolored wires](https://www.amazon.com/Kalevel%C2%AE-120pcs-Multicolored-Female-Breadboard/dp/B00M5WLZDW/ref=sr_1_7?s=electronics&ie=UTF8&qid=1469482189&sr=1-7&keywords=wires) | Kalevel |
| 1 | [Honor 5X](https://www.amazon.com/Honor-5X-unlocked-smartphone-Warranty/dp/B019O8YX1K/ref=sr_1_10?s=wireless&ie=UTF8&qid=1469485960&sr=1-10&keywords=huawei+cell+phone) | Huawei |

## Interaction

The interaction system is OpenPTrack (OPT), which is an open source distributed people tracking system. This OPT installation comprises of (2) Shuttle SZ170r6 V2 running NVIDIA GeForce GTX750 Ti graphics cards, and (2) [Microsoft Kinect Ones](http://www.xbox.com/es-AR/xbox-one/accessories/kinect-for-xbox-one).

## Visualization

The visualization system is Unity a 3D game engine used to create the application for *FLOW.* The engine will run on a 15” Macbook Pro,and the output of the visualization can be an external display, either a projection or additional monitor.

# Networking

The networking configuration for the system comprises a router, a Netgear 16-port switch, and an EnGenius 600 wireless access point. All components of the system will be connected to the network via Cat-6 wired connections.

Needs chart w/ name, model number, and IP addresses.

## TeamViewer Access

This is a placeholder / might not be used.

# Startup Guide

This is a placeholder.

1. Turn on computers
2. Verify Kinects are powered on
3. Start OPT tracking
4. Verify Tracking
5. Check NDN Publisher
6. ……

# Installation Guide

## Named Data Networking forwarder

* **NFD installation:** <https://github.com/named-data/NFD/blob/master/docs/INSTALL.rst>
* **NFD getting started:** <https://named-data.net/doc/NFD/current/INSTALL.html>

## Sensors & Devices

### RFDuino

* **Setting up the gyroscope on RFduino:** <https://github.com/remap/ndn-flow/tree/master/application/rfduino/rfduino-flow-producer#installation>

### Raspberry Pi

* **Setting up the Pi and running the gyroscope helper:** <https://github.com/remap/ndn-flow/tree/master/application/rfduino/rpi_helper#installation>
* **Running the ndn-pi home controller:** https://github.com/remap/ndn-flow/tree/master/framework/ndn\_pi

### Mobile Phone

* **Setting up the mobile website:** https://github.com/remap/ndn-flow/tree/master/application/website#installation

## OpenPTrack

This section will explain how to calibrate; start detection and tracking; refine calibration; troubleshoot OPT issues. Generally, the calibration should not be necessary, but it will be outlined in this guide. If one is unfamiliar with OPT, a good place to start is the Github Wiki, [here](https://github.com/OpenPTrack/open_ptrack/wiki/)[[1]](#footnote-1), which has all of the current installation, calibration, tracking, and troubleshooting information. Additionally, there is a deployment guide as part of the Wiki, [here](https://github.com/OpenPTrack/open_ptrack/wiki/Deployment-Guide)[[2]](#footnote-2), which has photos and additional information on calibration and resolving detection and tracking information. If calibration is not needed and only information on how to start tracking, please proceed to the detection and tracking section.

### Calibration

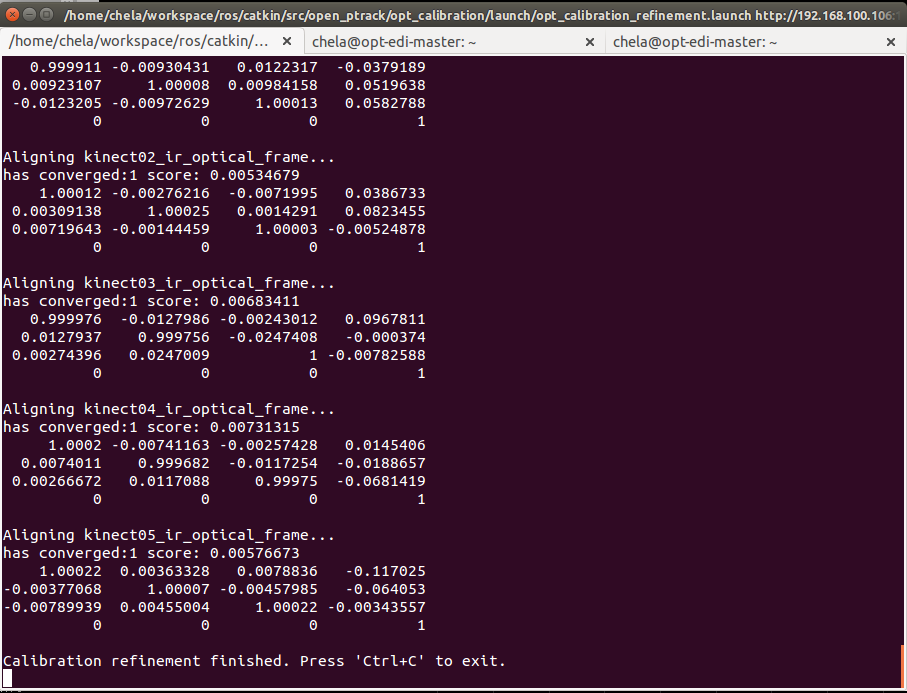
Calibration should only be performed if calibration refinement cannot resolve track splitting, loss in tracking, or issue related to ground planes. Regarding the process of calibration, much of the information can be found on the Wiki, [here](https://github.com/OpenPTrack/open_ptrack/wiki/Camera-Network-Calibration)[[3]](#footnote-3), and in the deployment guide, [here](https://github.com/OpenPTrack/open_ptrack/wiki/Calibration-in-Practice)[[4]](#footnote-4). The general calibration process is:

1. On opt-flow-01, the desktop of the master computer, there is a file named 20170113\_OPT\_FLOW. This file contains basic information and the commands for starting and completing calibration. Run as follows:
   1. roscore
   2. ./optterms
   3. ./optstartcal
2. On all nodes run:
   1. sudo ~/workspace/ros/catkin/devel/lib/kinect2\_bridge/.kinect2\_bridge
3. On master, run:
   1. ./optcalgui
4. Calibrate the system by:
   1. Placing the checkerboard on the ground in front of each of the (2) Kinects, allowing ~10 images to be captured
   2. Then, place the checkerboard in the center of the space with the (0,0) at the center maker.
5. Now, on the master with the checkerboard still in the center of the space with the X-Y coordinates on the top right corner closest to screen three, with the Y arrow pointing towards the entrance, and the X arrow pointing towards screen 1 and 2:
   1. ./optcalsave
   2. ./optcalwrite
6. Last on all the nodes, run:
   1. ./optcalread
7. All the operations can now be stopped

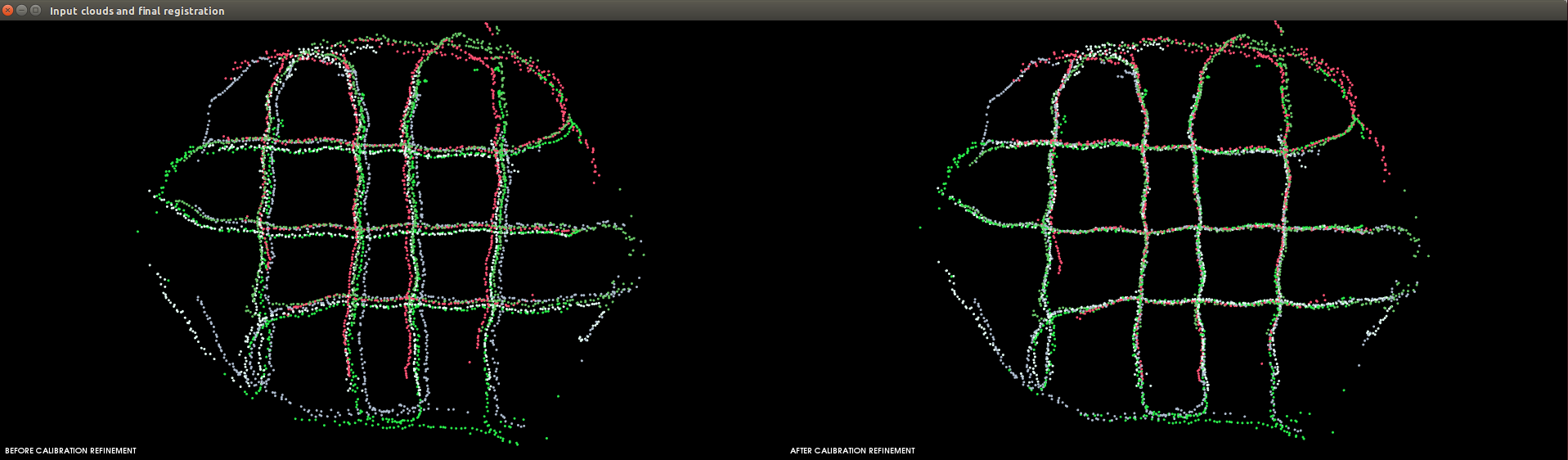
### Calibration Refinement

If, after calibration is complete, there is track splitting then calibration refinement can be performed. Details to conducting calibration refinement can be found [here](https://github.com/OpenPTrack/open_ptrack/wiki/Calibration-Refinement)[[5]](#footnote-5). The general practice for calibration refinement is:

1. Stop ./opttrack
2. Leave all of the other processes running
3. On master start:
   1. ./optcalrefine
4. A screen will appear, the Rviz screen with a black background. Walk around the tracking area in a checkerboard configuration **ensuring not to walk the same path in the same direction!** This this happens calibration refinement will need to be canceled (control+c) the started again.
5. On the master run:
   1. ./optcalsave
   2. The terminal should look like this when saving (need to update screengrab):



1. Check the tracking visualization to verify that each sensor’s track is closer together and the they are level, such as like this:



1. Press cntrl+c in the ./optcalrefine window
2. Rerun ./opttrack

### Detection and Tracking

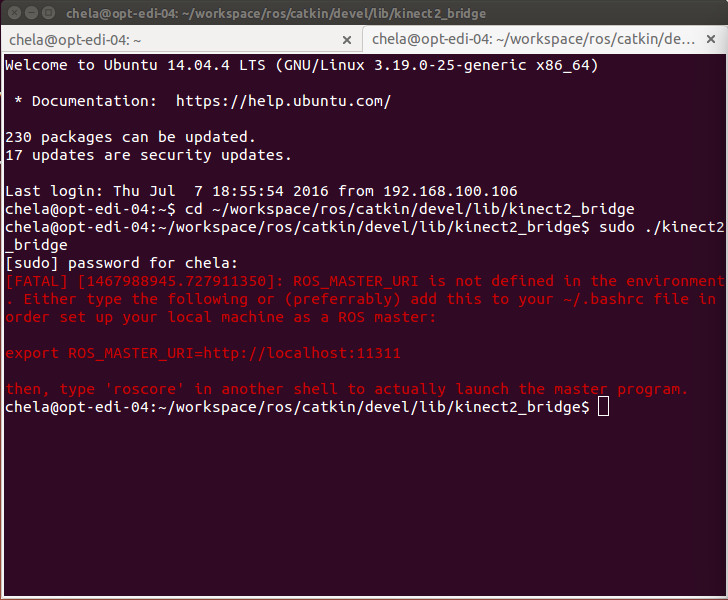
To start tracking:

1. First turn on the (2) Shuttle computers
2. Verify that all of the Kinect adapters have a [white light](#kinect_white_lights)

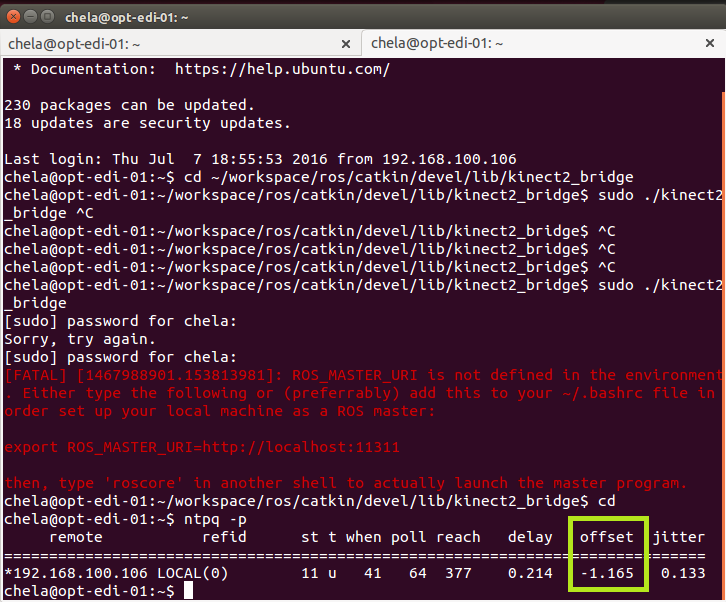
On the desktop of opt-flow-01 (master laptop), there is a file named 20170113\_OPT\_FLOW. This file contains basic information and the commands for starting detection and tracking.

1. After all of the computers are on, opt-flow-01, run:
   1. **./optterms**
2. Once the terminal windows are opened, they are organized, and it is verified all computers are operational, check the NTP offset by running on each node (this only needs to be completed after a computer has been turned off or restarted):
   1. **sudo ~/workspace/ros/catkin/devel/lib/kinect2\_bridge/./kinect2\_bridge**
   2. Then when prompted type in the computer’s password: cona
   3. Now, an error will occur, THIS IS OKAY! This is what the error looks like

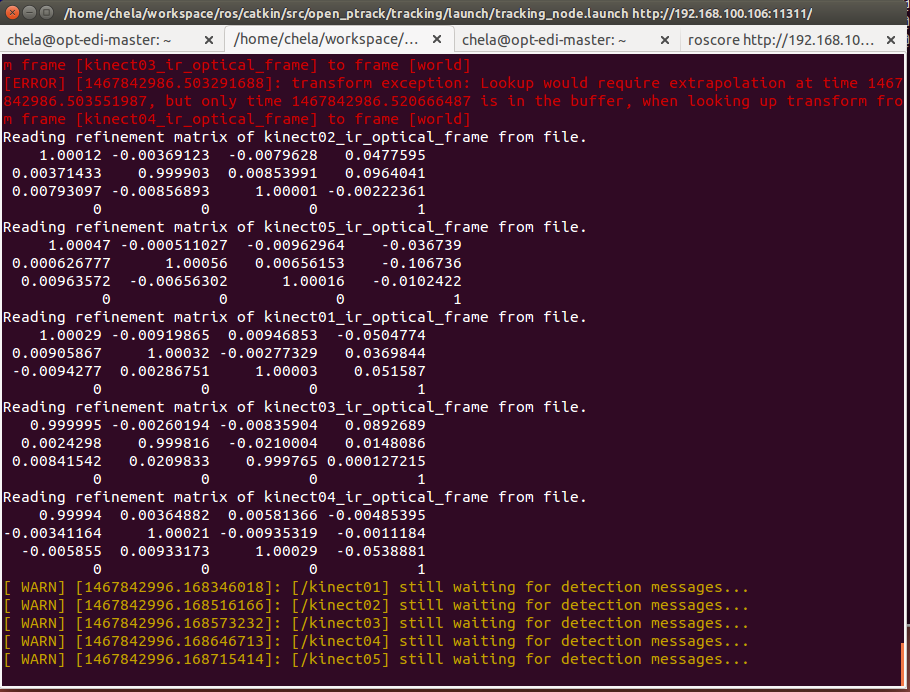
(need to update screengrab):



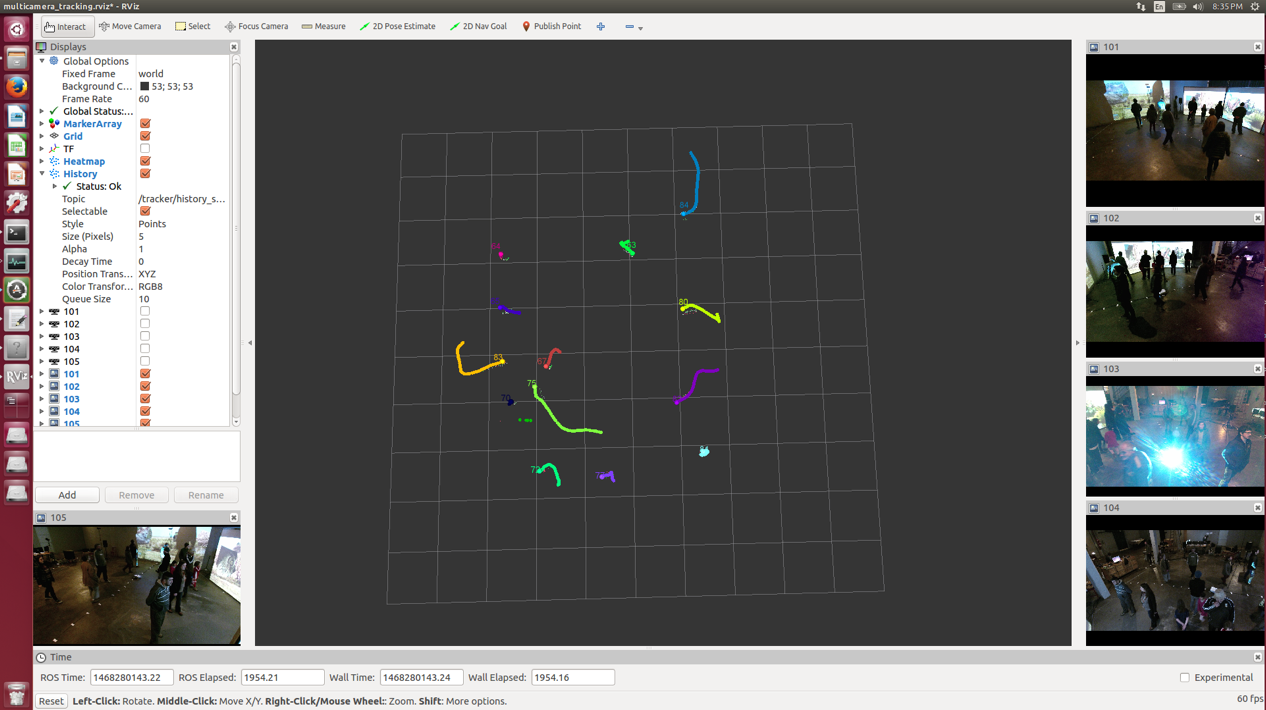
1. The verify the time sync, by running on all nodes:
   1. **ntpq –p**
      1. The output for the offset needs to be between +/- 15 which will look like this (need to update screengrab):



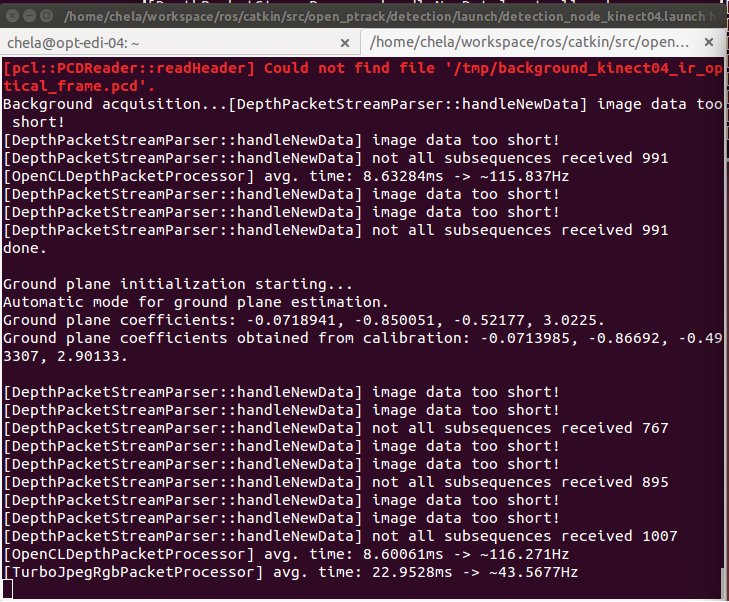
1. After both of these steps have been completed, on opt-flow-1 run:
   1. ./roscore
   2. **./opttrack**
   3. Which will look like (need to update screengrab):

****

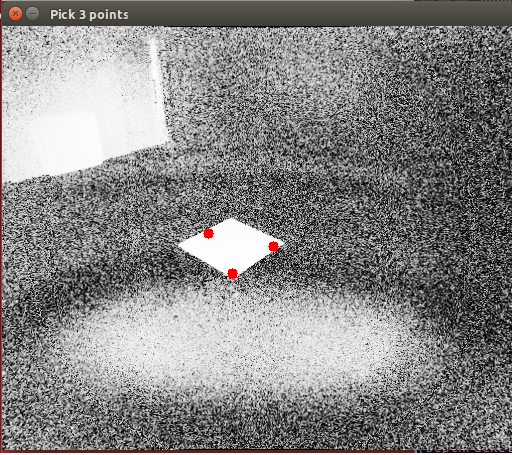
* 1. When this command is run Rviz will open (need to update screengrab):



1. When Rviz first opens, the RGB images from the (2) Kinect v2s will not be visible. Each of the sensors need to be started. To this, run on all nodes:
   1. **./optdect** (when this command is started the space should be clear. This is due to background subtraction starting which will create “dead spots” (untracked areas) where a person was standing)
   2. When the sensors have started correctly the printout in the terminal windows should look similar to this (need to update screengrab):



1. When starting each camera, the ground plane will need to be manually selected to do this first:
   1. Put a single sheet of Styrofoam (or any other flat white material) on the ground in front of the Kinect that the detection process is being initiated
   2. Then when the pick three points screen appears use the left mouse click to choose three points on the white surface which should look similar to this:



* 1. Then right click to save the image at which point the pick three points screen will automatically close.

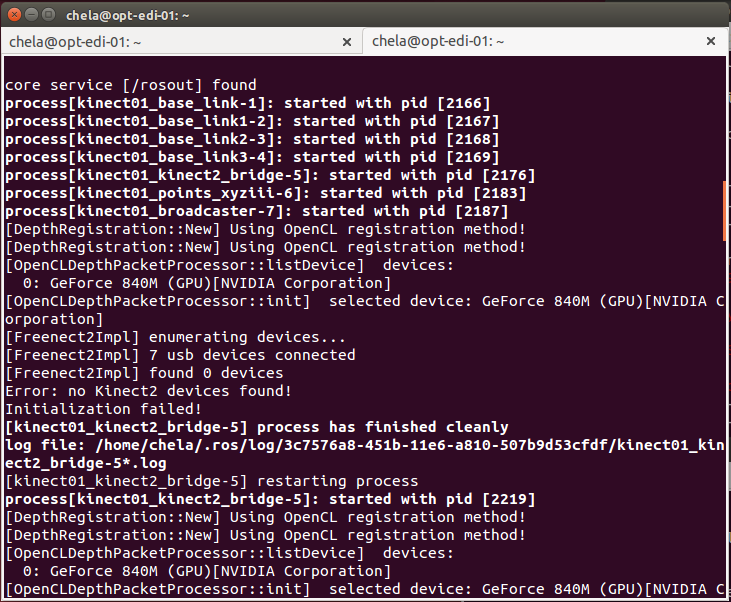
1. When testing tracking, the person walking should be a single line, if a person is multiple lines, i.e. splitting, the software troubleshooting section should be referenced.

### Troubleshooting (Hardware & Software)

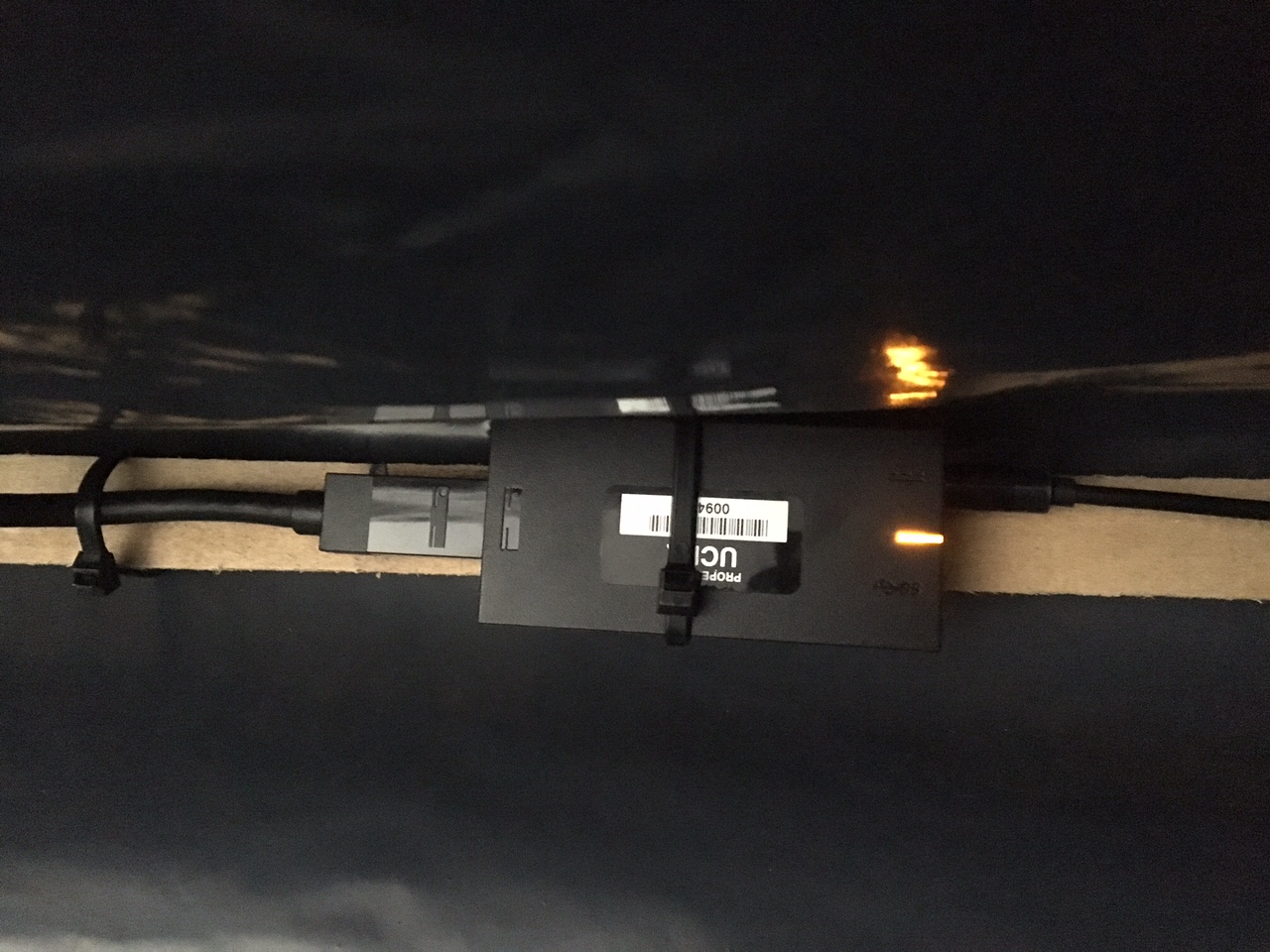
If there are trouble with software there are a few ways to troubleshoot.

### Error: No kinect2 devices found

If when starting a Kinect, the terminal prints out “Error: No Kinect2 devices found” there are steps that can be done to resolve this (need to update screengrab):



1. Verify that the Kinect2 is connected via USB.
   1. If the connect does NOT have power, and orange light will be shown on the USB adapter, which looks like:

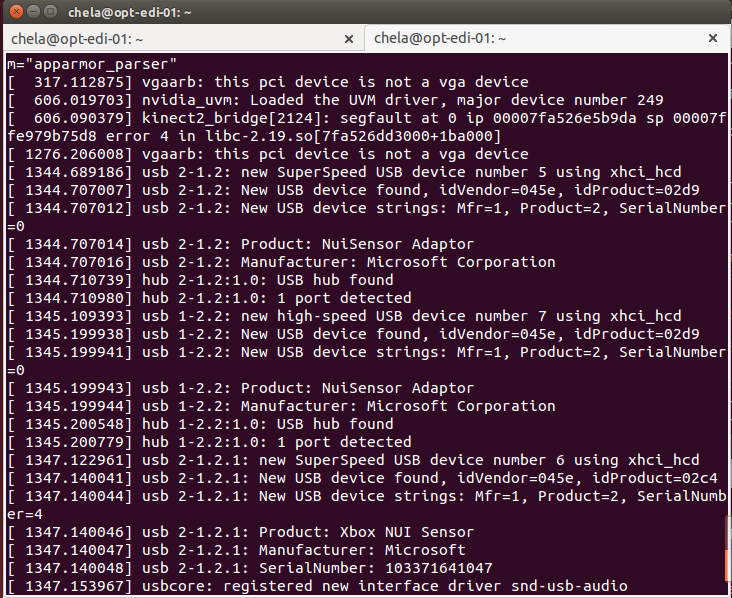


* + 1. This means the Kinect is not properly connected to the USB port. Check at the Shuttle computer that the Kinect’s USB connector is properly seated, then do the same at the USB adapter.
  1. If the USB connection is OK then it will have a white light such as this:



* 1. If there is no light, then the Kinect does not have power. Check the power sources at the USB adapter and the wall.

1. One can also verify that the Kinect is being identified by the computer by entering the command **lsusb** into a terminal window. If the Kinect is being seen properly by the computer, the terminal print out a line with “Microsoft”
   1. There is another way to check as well:
      1. First disconnect the Kinect USB from the laptop in question. Then, plug it back in.
      2. In a terminal window quickly enter the command **dmesg**. If the computer is properly recognizing the Kinect, there should be a line that stays “Xbox NUI Sensor”, and serial number (need to update screengrab):



### OpenPTrack is started, but there is no tracking

This can happen from time to time. The first thing to do is verify there are no error in the ./opttrack terminal window. An error will look similar to this (need to update screengrab):



If this happens, then:

1. Stop ./opttrack
2. Stop **all** the ./optdect functions
3. Restart ./opttrack
4. Restart all of the ./optdect
   1. Rarely, this will not fix the issue and all of the computers in the OpenPTrack network will need to be restarted

### Network Troubleshooting

### NTP Synchronization

From time to time the NTP offset will be over the +/- 15ms threshold. There are three ways of correcting this problem, to bring the NTP offset into the acceptable tolerance.

1. Wait 5 minutes and check again. Time will generally bring the system into tolerance.
2. On each computer, starting with the master, run sudo service ntp restart
   1. After this command is run on all machines, check **ntpq –p** again to check the offset
3. The third was is to run on all of the OpenPTrack computers, starting with the master:
   1. sudo service ntp stop
   2. sudo ntpd -gq
   3. sudo service ntp start
   4. Then again run **ntpq –p** to check the offset

## **Unity**

* **Installing the dependencies:** NDN-IoT (C#) <https://github.com/remap/ndn-flow/tree/master/framework/ndn_iot_dot_net#compile>
* **Installing Unity:** <https://unity3d.com/cn/get-unity/download>
* **Downloading and using the WWBlimp Unity project:** detailed information for the installation and its dependencies can be found on GitHub: <https://github.com/remap/ndn-flow/tree/master/application/unity/WWBlimp>

1. <https://github.com/OpenPTrack/open_ptrack/wik> [↑](#footnote-ref-1)
2. <https://github.com/OpenPTrack/open_ptrack/wiki/Deployment-Guide> [↑](#footnote-ref-2)
3. <https://github.com/OpenPTrack/open_ptrack/wiki/Camera-Network-Calibration> [↑](#footnote-ref-3)
4. <https://github.com/OpenPTrack/open_ptrack/wiki/Calibration-in-Practice> [↑](#footnote-ref-4)
5. <https://github.com/OpenPTrack/open_ptrack/wiki/Calibration-Refinement> [↑](#footnote-ref-5)