



Εθνικό και Καποδιστριακό  
Πανεπιστήμιο Αθηνών  
National and Kapodistrian  
University of Athens

# Network Management in Mission Critical Cases – PPDR & Smart evacuation Guide



Research Associate - Panagiotis Kontopoulos

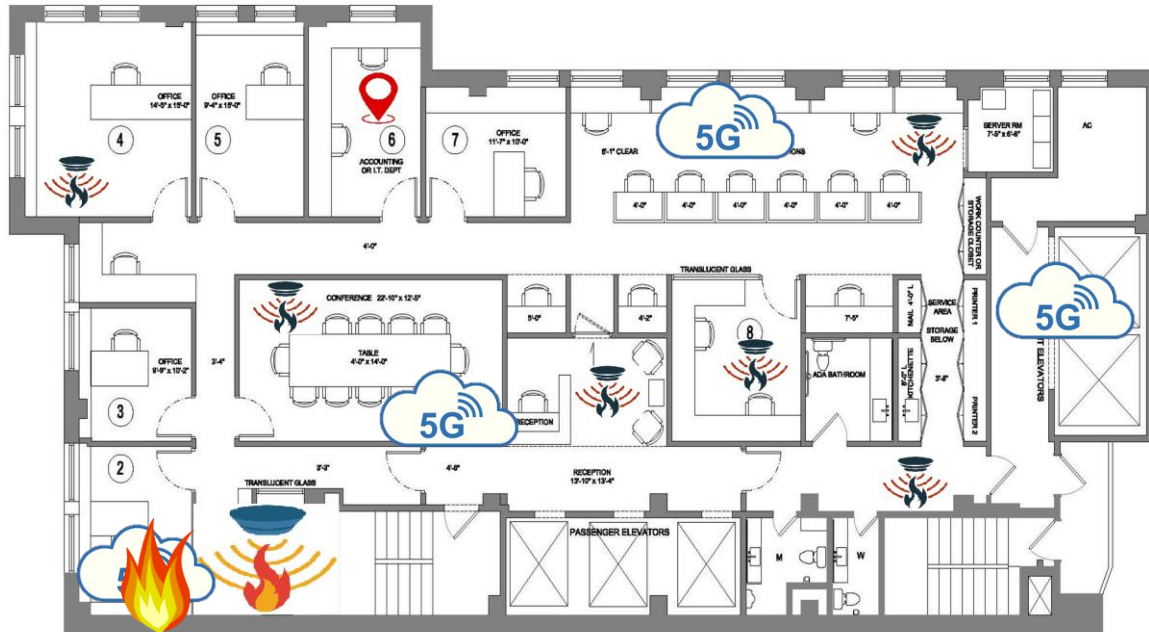
Dept. Informatics and Telecommunications  
National Kapodistrian University of Athens



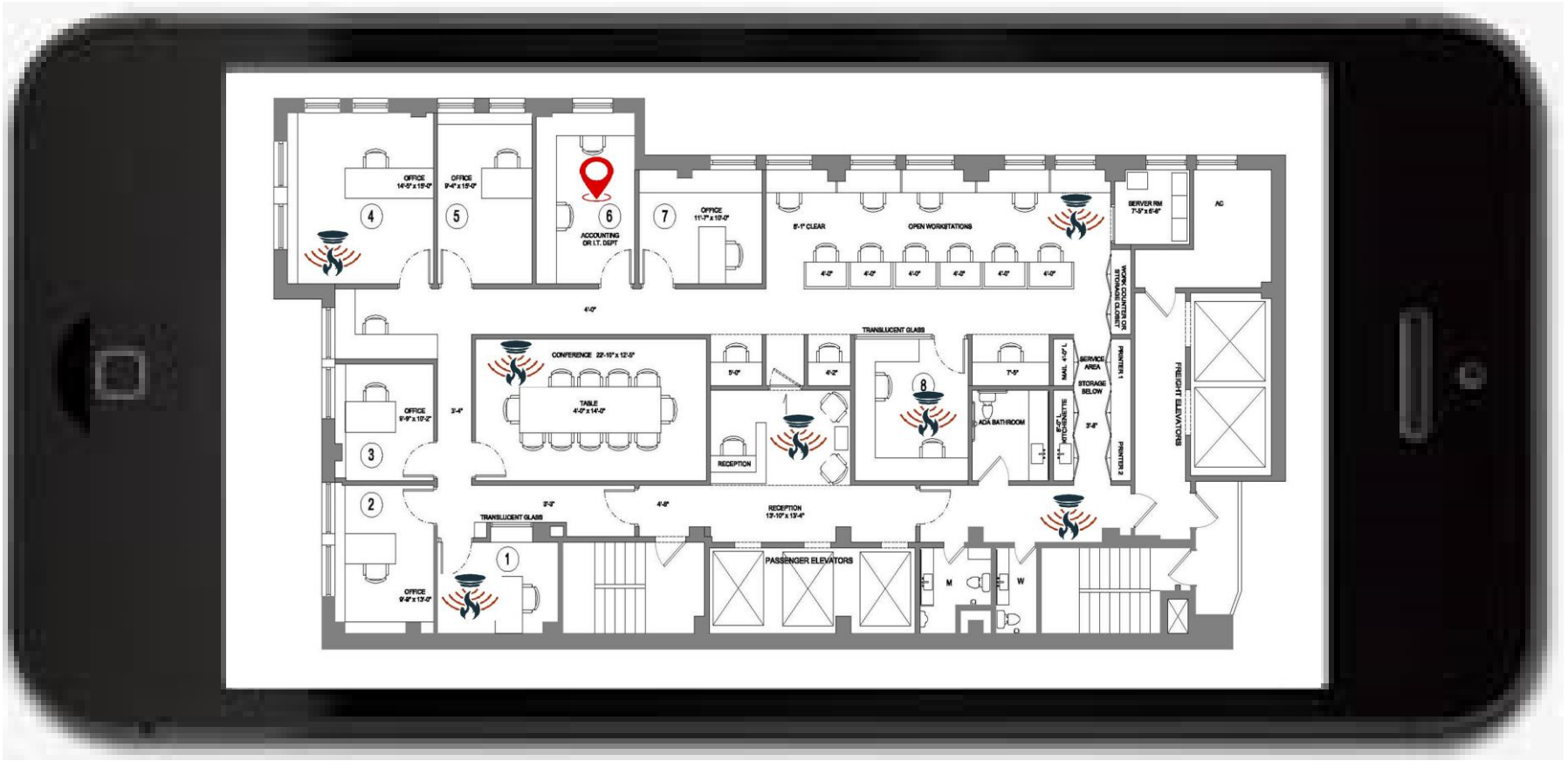




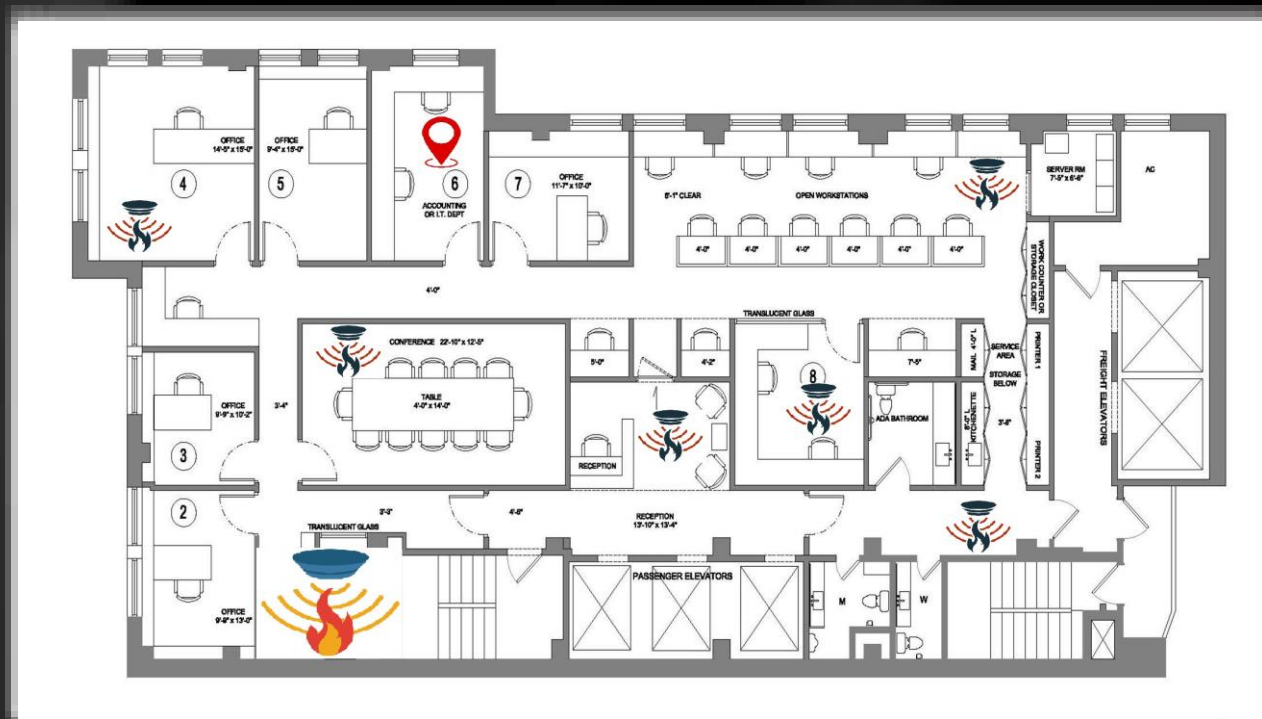
# Indoor



# User Interface



# Identification of fire incident



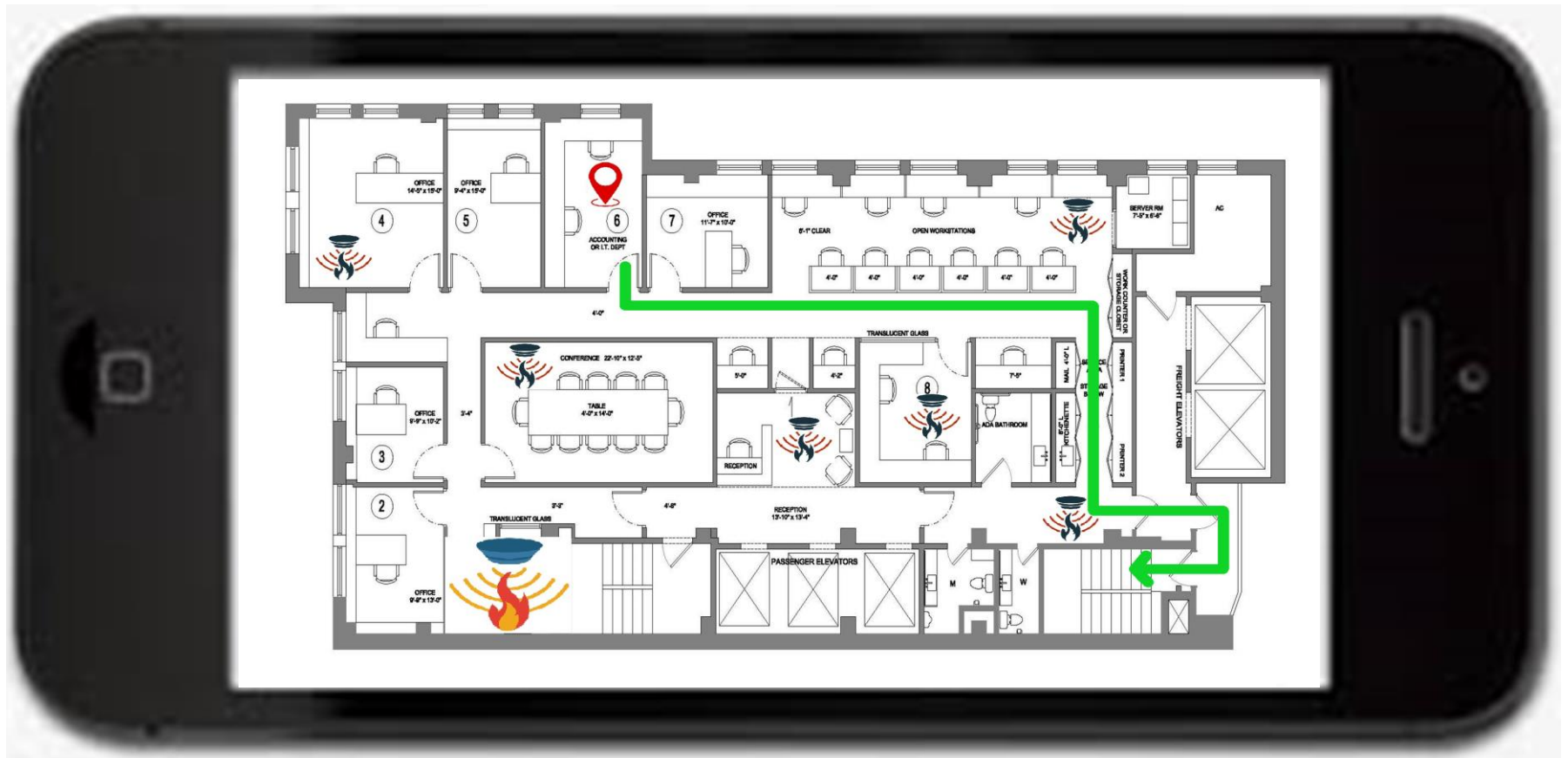


# Audio and Visual Alarm

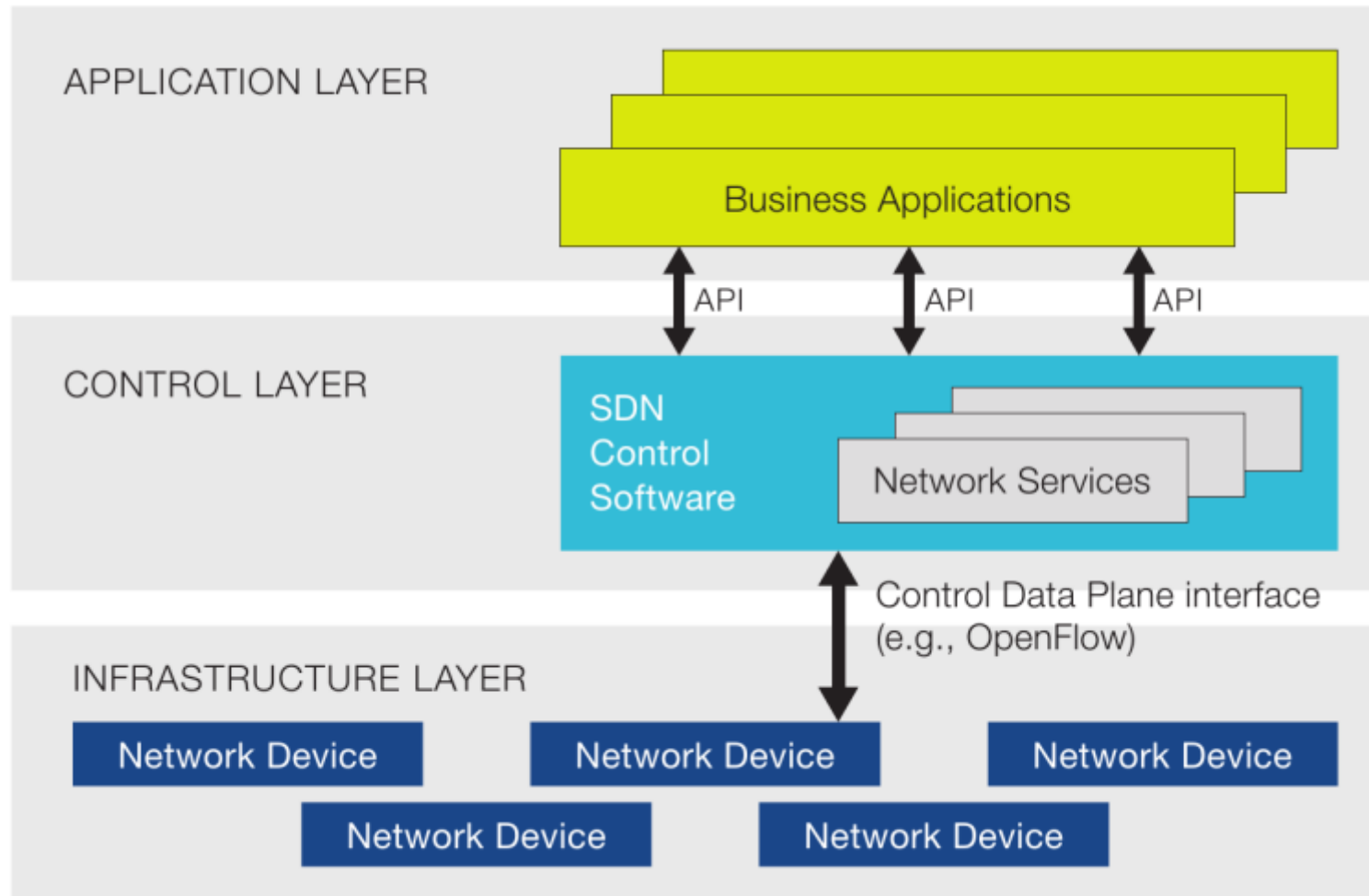




# Escape route identification – smart navigation

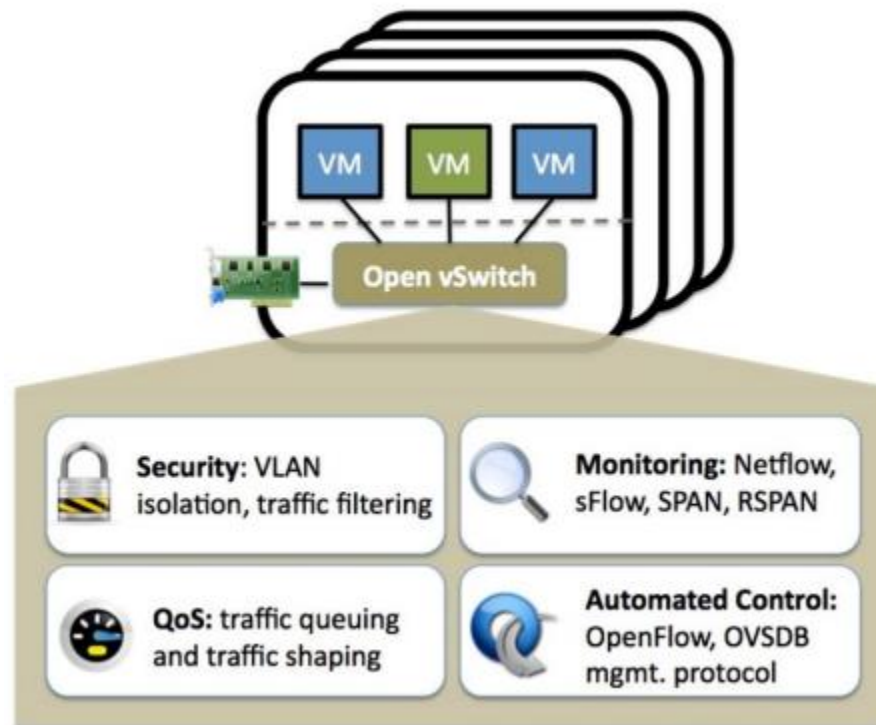


# Some Background - SDN



# Some Background - OVS

- Open vSwitch is a production quality, multilayer virtual switch licensed under the open source Apache2.0 license.
- Network automation through programmatic extensions



# OVS – Basic commands

- `ovs-vsctl` : Used for configuring the `ovs-vswitchd` configuration database (known as `ovs-db`)
- `ovs-ofctl` : A command line tool for monitoring and administering OpenFlow switches
- `ovs-dpctl` : Used to administer Open vSwitch datapaths `ovs-appctl` : Used for querying and controlling Open vSwitch daemons
- `ovs-ofctl dump-flows sw` : List the flows assigned to the bridge
- `ovs-ofctl show sw` : connects to the switch and prints out port state and OF capabilities
- `ovs-ofctl del-flows sw` : Deletes all flows from the switch
- `ovs-ofctl add-flow sw in_port=1,actions=output:2` : Adds a flow (rule) – port-based.
- `ovs-ofctl mod-flows switch in_port=1,actions=drop` : Modifies a flow entry



# Mininet-WiFi

- A fork of Mininet
- Emulator for Software Defined Networking
- Simple to create topologies and run network scenarios
- <https://mininet-wifi.github.io/>
- Virtualized Access Points
- Emulates mobility & supports various mobility models
- Remote controllers are also supported

# Mininet-WiFi – Installation

- Option 1 – VM install:
  - [Download the Mininet-WiFi VM image](#)
  - Download and install a virtualization system. We recommend VirtualBox (free, GPL) because it is free and works on OS X, Windows, and Linux (though it's slightly slower than VMware in our tests.) You can also use Qemu for any platform, VMware Workstation for Windows or Linux, VMware Fusion for Mac, or KVM (free, GPL) for Linux.
- Option 2 - Native Installation from Source (Debian):
  - `git clone git://github.com/intrig-unicamp/mininet-wifi`
  - `cd mininet-wifi`
  - `sudo util/install.sh -WInfv`

Everything regarding the installation process is available: <https://mininet-wifi.github.io/get-started/>

# Mininet-WiFi - Basic Classes

- **addHost():** adds a host to a topology and returns the
  - host name
- **addStation():** adds a station to a topology and returns the station name
- **addAccessPoint():** adds an access point to a topology and returns the access point name
- **addLink():** adds a bidirectional link to a topology



# Mininet-WiFi – Usage (1)

<https://mininet-wifi.github.io/usage/>

- Start a minimal topology and enter the CLI: `sudo mn -wifi`
- Display Mininet CLI commands: `mininet-wifi> help`
  - Display Nodes: `mininet-wifi> nodes`
  - Run a command on a station process: `mininet-wifi> sta1 ifconfig -a`
  - Run a command on a access point: `mininet-wifi> ap1 ifconfig -a`
  - Get information of the wireless network interfaces: `py sta1.wintfs`
  - Verify that you can ping from station1 to station2: `mininet-wifi> sta1 ping -c1 sta2`
    - Ping all network pairs (ex. sta1 – sta2, sta1-sta3, sta2 – sta3 etc.): `mininet-wifi> pingall`
  - Exit CLI: `mininet> exit`
  - **If Mininet crashes for some reason, clean it up: `sudo mn -c`**
  - You can create a wired link between station and access point with `cls=TCLink`, as shown below:
    - Import in your file: `from mininet.link import TCLink`
    - Use command in file: `net.addLink(sta1, ap1, cls=TCLink)`
  - To display an xterm for sta1 and sta2: `xterm sta1 sta2`
  - Start AP/Shutting AP down: `mininet-wifi> py ap1.start_()` / `mininet-wifi> py ap1.stop_()`
  - Start/ Stop Simulation: `mininet-wifi> start` / `mininet-wifi> stop`

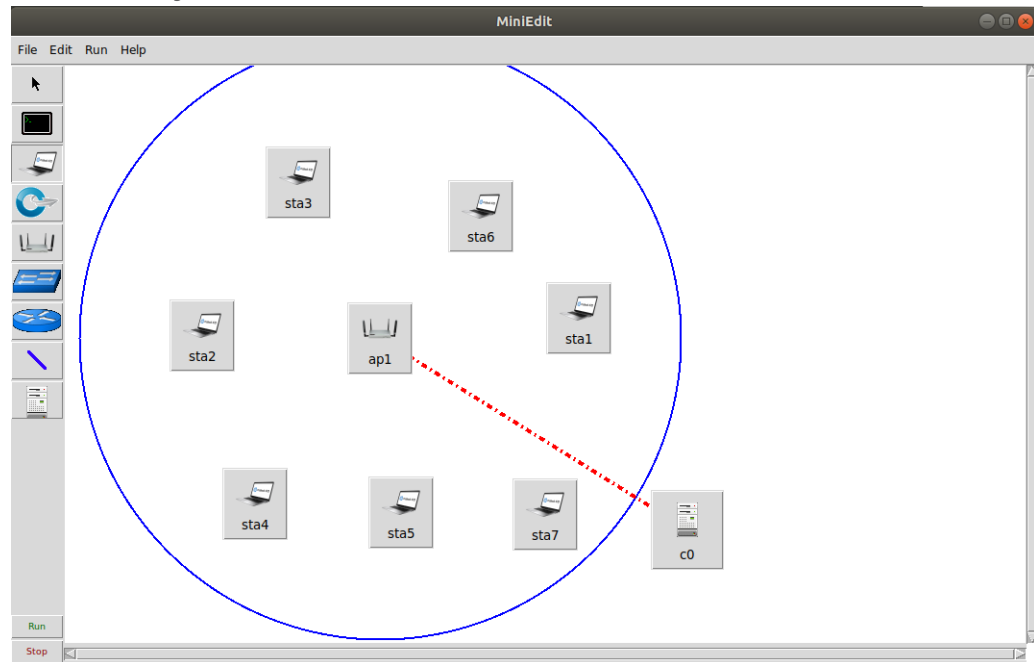


# Mininet-WiFi – Usage (2)

<https://mininet-wifi.github.io/advanced/>

<https://mininet-wifi.github.io/commands/>

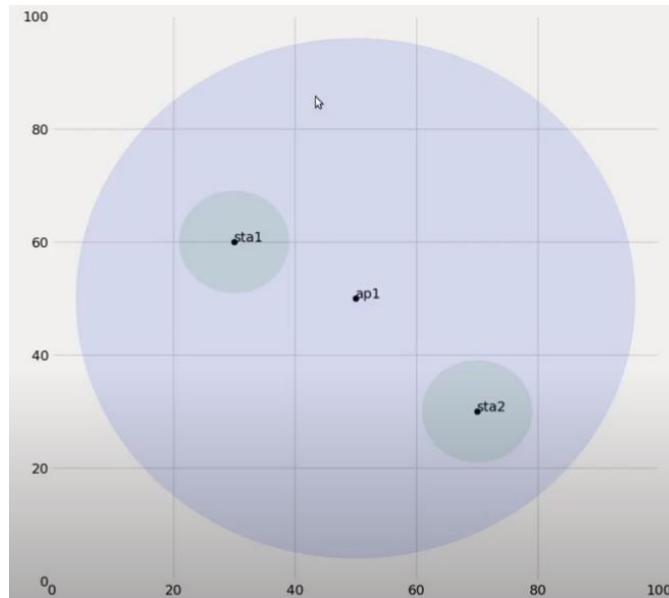
- You can use also the miniedit GUI: `sudo python examples/miniedit.py`
- [https://www.youtube.com/watch?v=j4JS4xxCrCA&list=PLccoFREVAAt\\_4nEtrkl59mjff5ZzRX8DZA&index=17](https://www.youtube.com/watch?v=j4JS4xxCrCA&list=PLccoFREVAAt_4nEtrkl59mjff5ZzRX8DZA&index=17)



# Mininet-WiFi – Usage (3)

<https://www.youtube.com/watch?v=k69t9Xkb0nU>

- Setting and getting node attributes via socket



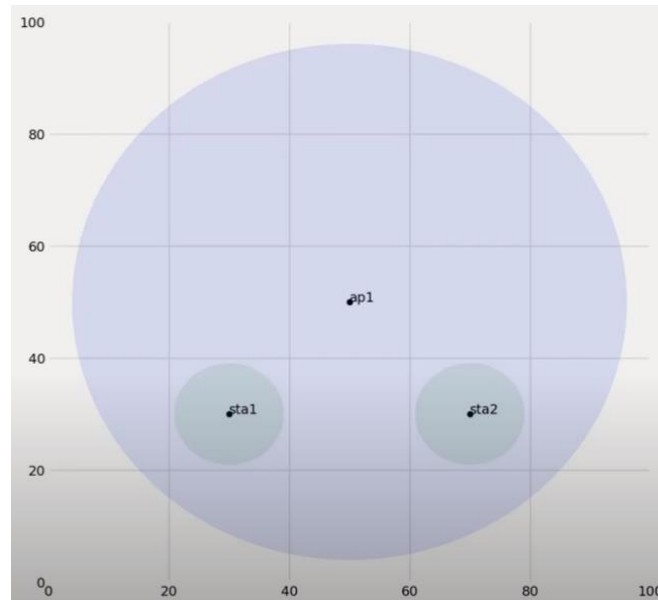
```
alpha@alpha-Inspiron-5480:~/mininet-wifi$ sudo python examples/socket_client.py
[sudo] password for alpha:
-> get.sta1.position
Received from server: [30.0, 60.0, 0.0]
-> get.sta1.rssi
Received from server: [-76.0]
-> get.sta1.range
Received from server: [9]
```



# Mininet-WiFi – Usage (4)

<https://www.youtube.com/watch?v=k69t9Xkb0nU>

- Moving a station to a new location

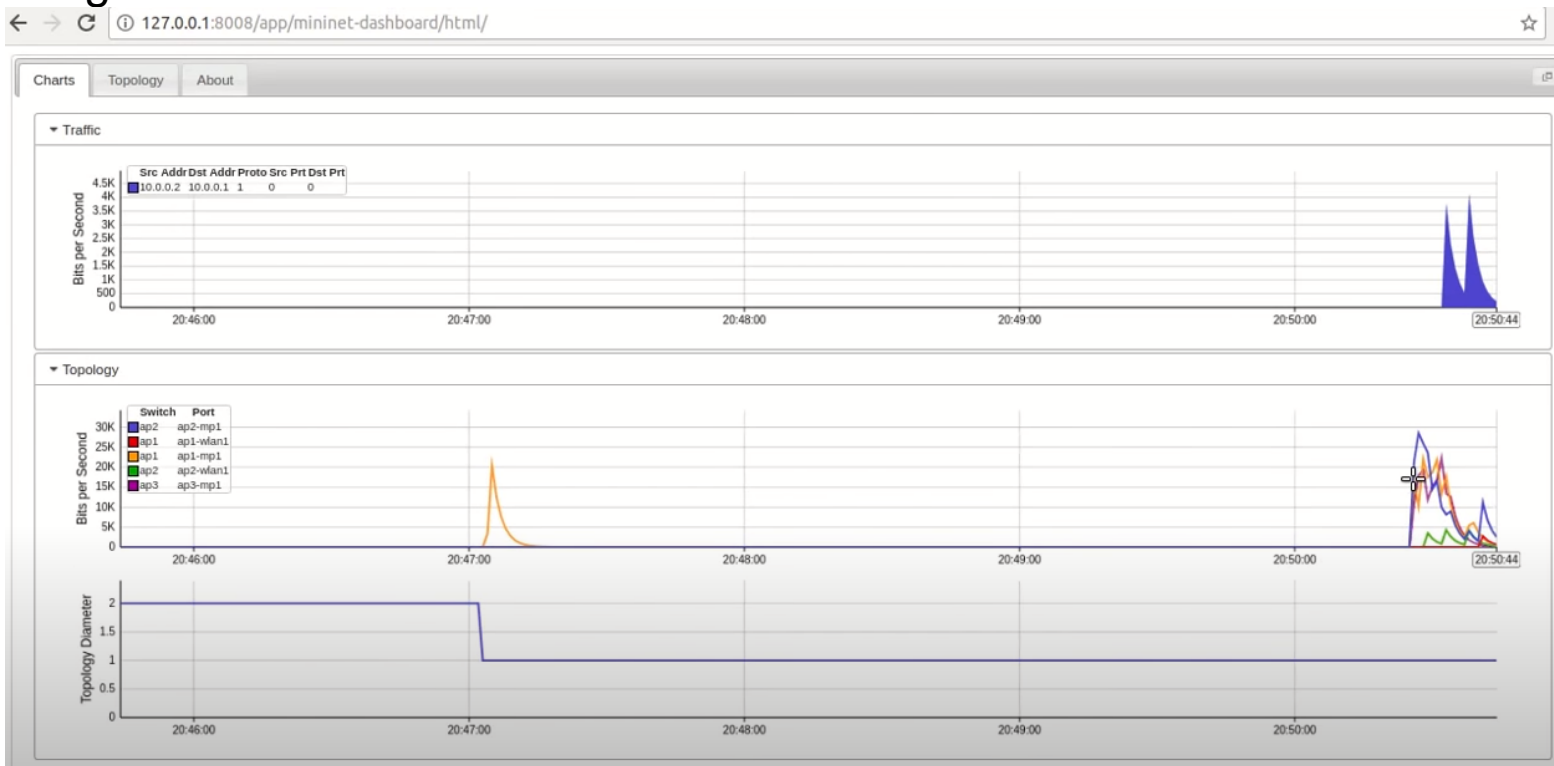


```
-> set.sta1.setPosition.30,30,0  
Received from server: command accepted!
```

# Mininet-WiFi – Usage (5)

[https://www.youtube.com/watch?v=4ccua2b26k8&list=PLccoFREVAAt\\_4nEtrkl59mjjf5ZzRX8DZA&index=18](https://www.youtube.com/watch?v=4ccua2b26k8&list=PLccoFREVAAt_4nEtrkl59mjjf5ZzRX8DZA&index=18)

- Integration between Mininet-WiFi and sflow-rt



# Mininet-WiFi – Usage (6)

[https://www.youtube.com/watch?v=xWfWSqtYA8A&list=PLccoFREVAAt\\_4nEtrkl59mjff5ZzRX8DZA&index=22](https://www.youtube.com/watch?v=xWfWSqtYA8A&list=PLccoFREVAAt_4nEtrkl59mjff5ZzRX8DZA&index=22)

- Mininet-WiFi: Using 4-address for AP and client mode

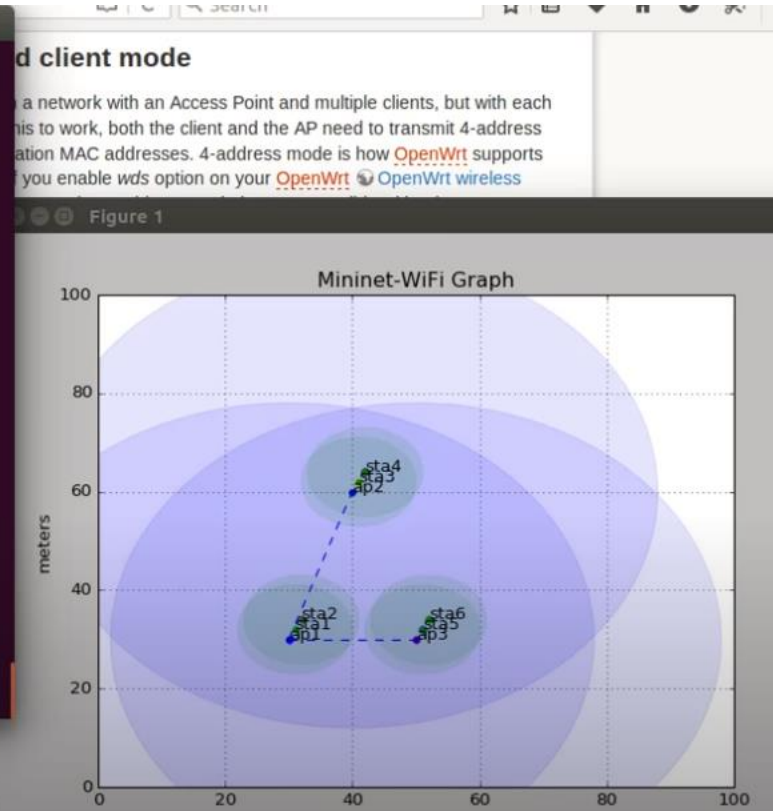
```
alpha@alpha-Inspiron: ~/mininet-wifi
sta1 *** sta6 : ('ping -c1 192.168.0.2',)
PING 192.168.0.2 (192.168.0.2) 56(84) bytes of data.
64 bytes from 192.168.0.2: icmp_seq=1 ttl=64 time=7.09 ms

--- 192.168.0.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 7.099/7.099/7.099/0.000 ms
sta2 *** sta6 : ('ping -c1 192.168.0.3',)
PING 192.168.0.3 (192.168.0.3) 56(84) bytes of data.
64 bytes from 192.168.0.3: icmp_seq=1 ttl=64 time=8.92 ms

--- 192.168.0.3 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 8.926/8.926/8.926/0.000 ms
sta3 *** sta6 : ('ping -c1 192.168.0.4',)
PING 192.168.0.4 (192.168.0.4) 56(84) bytes of data.
64 bytes from 192.168.0.4: icmp_seq=1 ttl=64 time=4.49 ms

--- 192.168.0.4 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 4.493/4.493/4.493/0.000 ms
sta4 *** sta6 : ('ping -c1 192.168.0.5',)
PING 192.168.0.5 (192.168.0.5) 56(84) bytes of data.
64 bytes from 192.168.0.5: icmp_seq=1 ttl=64 time=0.627 ms

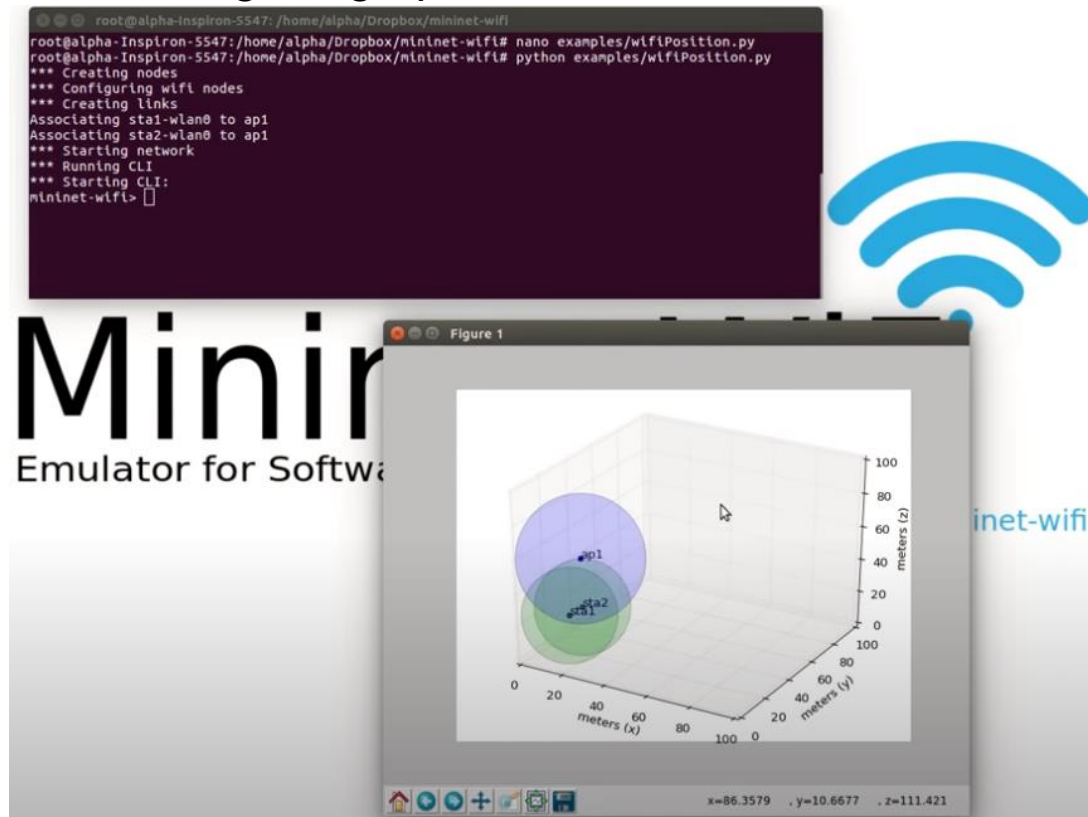
--- 192.168.0.5 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.627/0.627/0.627/0.000 ms
sta5
*** Results: 0% dropped (30/30 received)
mininet-wifi>
```



# Mininet-WiFi – Usage (7)

[https://www.youtube.com/watch?v=IMkIV0YBTss&list=PLccoFREVAAt\\_4nEtrkl59mjjf5ZzRX8DZA&index=34](https://www.youtube.com/watch?v=IMkIV0YBTss&list=PLccoFREVAAt_4nEtrkl59mjjf5ZzRX8DZA&index=34)

- Mininet-WiFi: Building 3D graphic

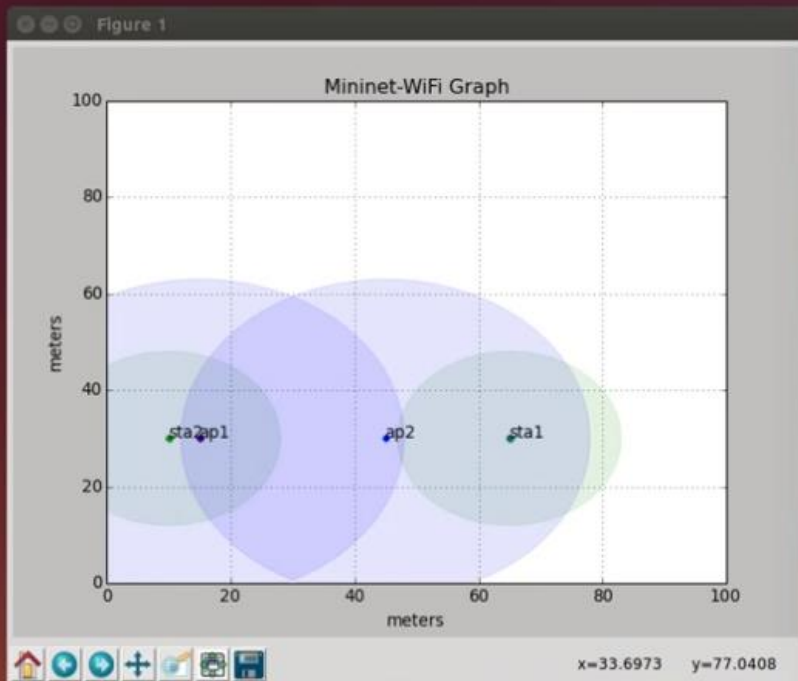




# Mininet-WiFi – Usage (8)

[https://www.youtube.com/watch?v=-Q59NNNjt9I&list=PLccoFREVAAt\\_4nEtrkl59mjff5ZzRX8DZA&index=41](https://www.youtube.com/watch?v=-Q59NNNjt9I&list=PLccoFREVAAt_4nEtrkl59mjff5ZzRX8DZA&index=41)

- Mininet-WiFi: Working at runtime



```
alpha@alpha-Inspiron-5547: ~
mininet-wifi> py sta1.moveStationTo('75,30,0')
mininet-wifi> sta1 iwconfig
sta1-wlan0 IEEE 802.11abgn ESSID:"ap-ssid2"
Mode:Managed Frequency:2.412 GHz Access Point: 02:00:00:00:03:00
Bit Rate:1 Mb/s Tx-Power=20 dBm
Retry short limit:7 RTS thr:off Fragment thr:off
Encryption key:off
Power Management:off
Link Quality=70/70 Signal level=-30 dBm
Rx invalid nwid:0 Rx invalid crypt:0 Rx invalid frag:0
Tx excessive retries:0 Invalid misc:0 Missed beacon:0

lo no wireless extensions.

mininet-wifi> py sta1.moveStationTo('75,55,0')
mininet-wifi> py sta1.moveStationTo('75,65,0')
mininet-wifi> py sta1.moveStationTo('5,65,0')
mininet-wifi> sta1 iwconfig
sta1-wlan0 IEEE 802.11abgn ESSID:"ap-ssid1"
Mode:Managed Frequency:2.412 GHz Access Point: 02:00:00:00:02:00
Bit Rate:1 Mb/s Tx-Power=20 dBm
Retry short limit:7 RTS thr:off Fragment thr:off
Encryption key:off
Power Management:off
Link Quality=70/70 Signal level=-30 dBm
Rx invalid nwid:0 Rx invalid crypt:0 Rx invalid frag:0
Tx excessive retries:0 Invalid misc:0 Missed beacon:0

lo no wireless extensions.

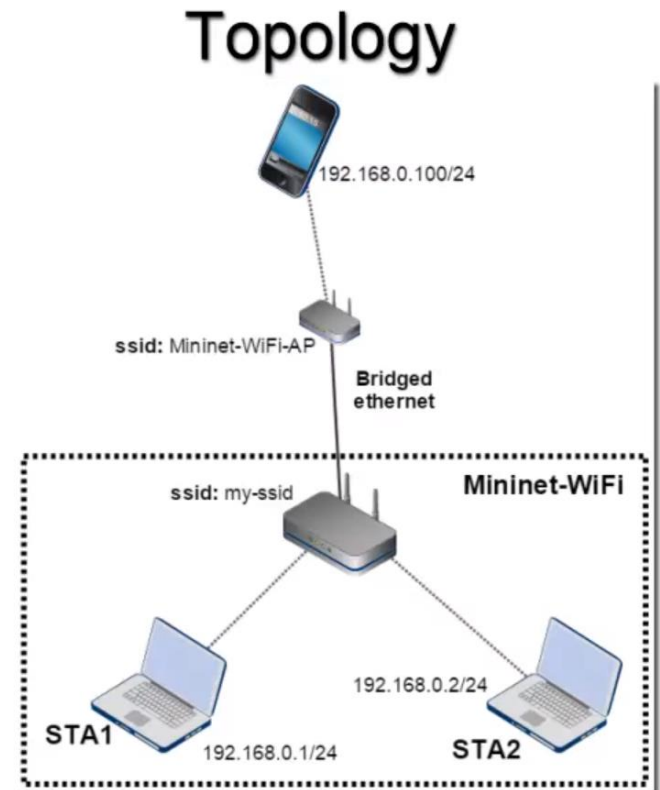
mininet-wifi> py sta1.moveStationTo('5,30,0')
mininet-wifi> py sta1.moveStationTo('65,30,0')
mininet-wifi> py sta1.moveStationTo('65,50,0')
```



# Mininet-WiFi – Usage (9)

[https://www.youtube.com/watch?v=WH6bSOKC7Lk&list=PLccoFREVAAt\\_4nEtrkl59mjff5ZzRX8DZA&index=52](https://www.youtube.com/watch?v=WH6bSOKC7Lk&list=PLccoFREVAAt_4nEtrkl59mjff5ZzRX8DZA&index=52)

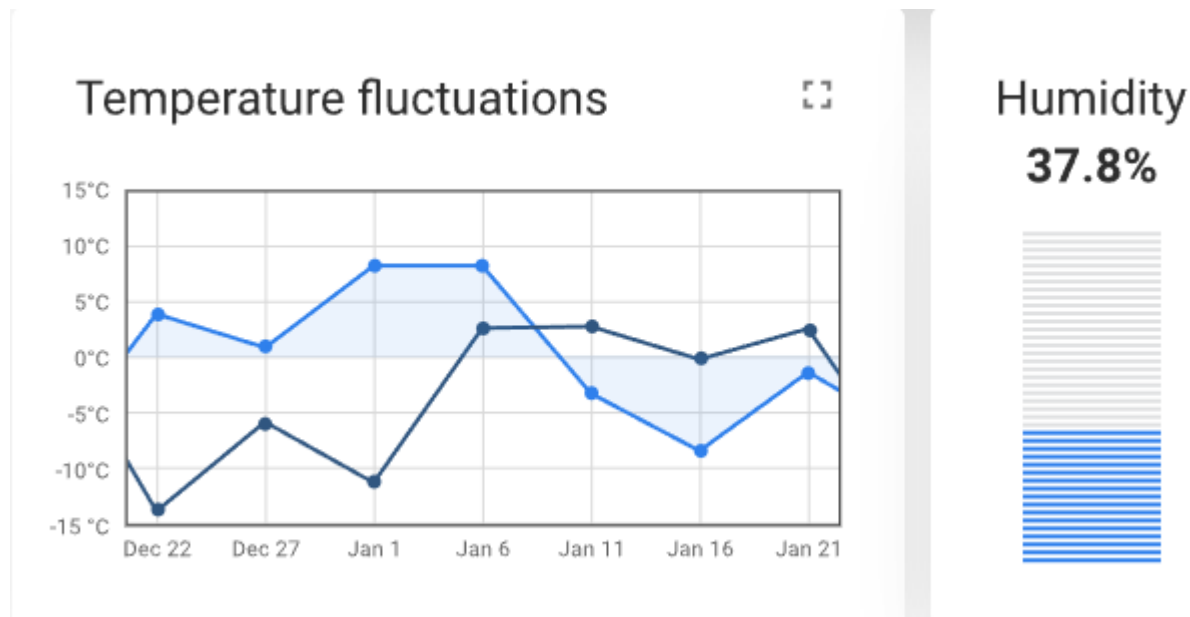
- Integration between Mininet-WiFi and Physical Wireless Interface
- Όπου το κινητό μπορεί να είναι ο server του Thingsboard με δικιά του IP και Port
- Και stations μπορούν να είναι οι clients που στέλνουν τα δεδομένα θερμοκρασίας στον server (άρα έχουν τον ρόλο των sensors)



# Thingsboard

ThingsBoard CE - <https://thingsboard.io/>

- Open-source IoT Platform
- Device management, data collection, processing and visualization for your IoT solution



# Thingsboard – Installation (1)

<https://thingsboard.io/docs/user-guide/install/installation-options/>

- Μπορείτε να επιλέξετε όποιο θέλετε (προτίμηση στις επιλογές **Ubuntu, Windows**)

## ThingsBoard installation options

ThingsBoard is designed to run and utilize on majority of hardware, from local Raspberry PI to powerful servers in the cloud



Live demo



On premise



Cloud

ubuntu



Windows

Raspberry Pi3



Maven  
Building from sources



Cluster setup



# Thingsboard – Installation (2)

<https://thingsboard.io/docs/user-guide/install/ubuntu/>

<https://thingsboard.io/docs/user-guide/install/windows/>

- Ακολουθείτε τα βήματα του συνδέσμου και επιλέγετε για την βάση και queue service τα παρακάτω

- Για την βάση:

**PostgreSQL**  
(recommended for < 5K msg/sec)

- Για το queue service:

**In Memory**  
(built-in and default)

- Μπορείτε να δοκιμάσετε και τις άλλες επιλογές αλλά οι recommended είναι οι πιο απλές



# Thingsboard – Usage (1)

<https://thingsboard.io/docs/user-guide/install/ubuntu/>

<https://thingsboard.io/docs/user-guide/install/windows/>

- Getting started guides - These guides provide quick overview of main ThingsBoard features. Designed to be completed in 15-30 minutes.
- Connect your device - Learn how to connect devices based on your connectivity technology or solution.
- Data visualization - These guides contain instructions how to configure complex ThingsBoard dashboards.
- Data processing & actions - Learn how to use ThingsBoard Rule Engine.
- IoT Data analytics - Learn how to use rule engine to perform basic analytics tasks.
- Hardware samples - Learn how to connect various hardware platforms to ThingsBoard.
- Advanced features - Learn about advanced ThingsBoard features.



# Thingsboard – Usage (2)

## Hello world

Learn how to collect IoT device data using MQTT, HTTP or CoAP and visualize it on a simple dashboard. Provides variety of sample scripts that you can run on your PC or laptop to simulate the device.

## End user IoT dashboards

Learn how to perform basic operations over Devices, Customers, and Dashboards.

## Device data management

Learn how to perform basic operations over device attributes to implement practical device management use cases.

## Getting started with Rule Engine

Learn about ThingsBoard rule engine and typical use cases you can implement. Review Hello World example and learn how-to enable filtering of incoming telemetry messages.





# Thingsboard – Usage (3)

- Για να συνδέσετε συσκευή που να στέλνει δεδομένα στο server σας προτιμάμε το HTTP API
- Σε κανονικά σενάρια προτιμούμε το MQTT, επειδή προσφέρει πολλές δυνατότητες

## Connect devices using ThingsBoard HTTP API

Learn how to connect your devices using HTTP protocol and ThingsBoard built-in payload format.

## Connect devices using ThingsBoard MQTT API

Learn how to connect your devices using MQTT protocol and ThingsBoard built-in payload format.

# Thingsboard – Usage (4)

- Παράδειγμα χρήσης JSON και API

- JSON →

```
{  
  "stringKey": "value1",  
  "booleanKey": true,  
  "doubleKey": 42.0,  
  "longKey": 73,  
  "jsonKey": {  
    "someNumber": 42,  
    "someArray": [1, 2, 3],  
    "someNestedObject": {"key": "value"}  
  }  
}
```

- API: `http(s)://host:port/api/v1/$ACCESS_TOKEN/telemetry`

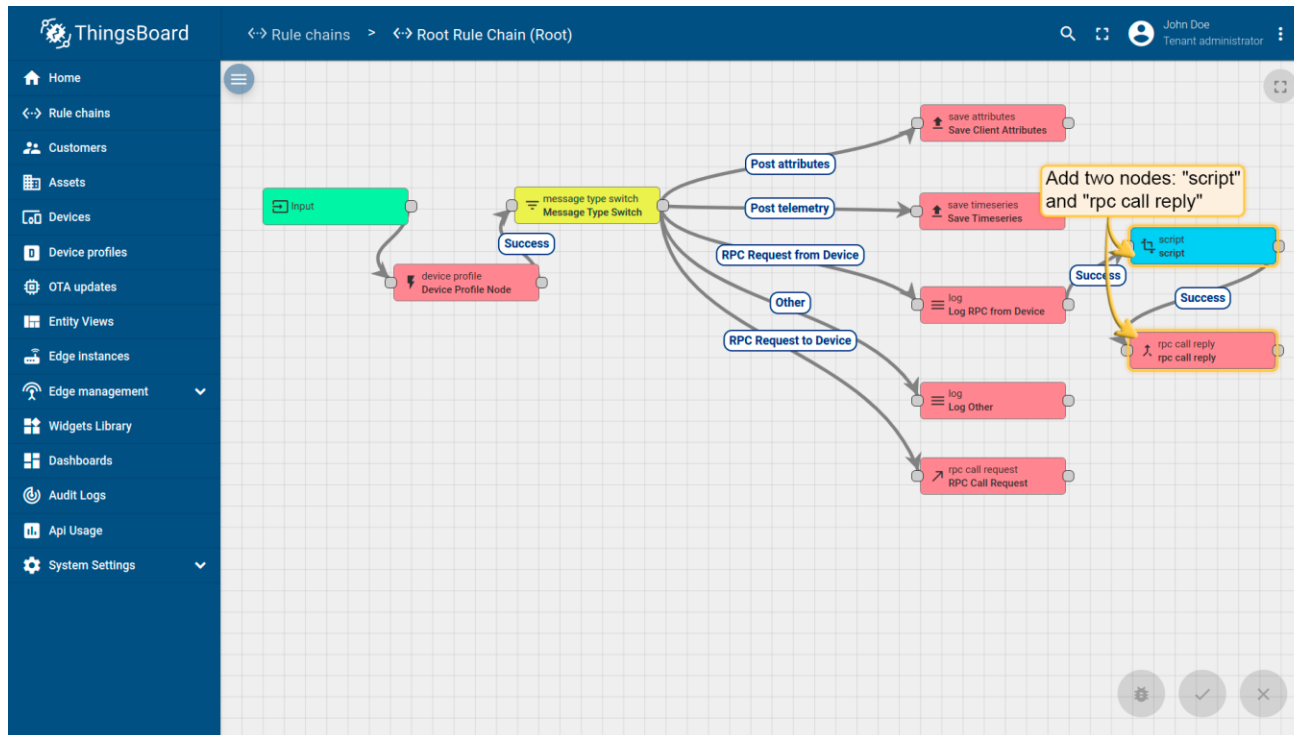
- Παράδειγμα:

- # Publish data as an object without timestamp (server-side timestamp will be used). Replace \$THINGSBOARD\_HOST\_NAME and \$ACCESS\_TOKEN with corresponding values.
- `curl -v -X POST --data '{"temperature":42,"humidity":73}'  
http://$THINGSBOARD_HOST_NAME/api/v1/$ACCESS_TOKEN/telemetry --header "Content-Type:application/json"`



# Thingsboard – Usage (4)

- Θα χρειαστεί να χρησιμοποιήσετε το GUI για την εφαρμογή κανόνων που να ανταποκρίνονται στα σενάρια
- Για παράδειγμα στους κανόνες



# Thingsboard – Usage (5)

- Παραδείγματα Dashboard που θα χρειαστείτε

The image displays two screenshots of the Thingsboard dashboard interface. The top screenshot shows the 'District A' dashboard, featuring a map of the district with two buildings marked: Building A at 3100 24th St and Building B at 2865 Harrison St. The bottom screenshot shows the 'Building A' dashboard, displaying a floor plan with three sensors: Thermostat A-1, Energy Meter A-1, and Water Meter A-1. The sensor data table is as follows:

Entity name	Energy	Temperature	Water
Energy Meter A-1	19615.07 kWh	-	-
Thermostat A-1	-	25.00 °C	-
Water Meter A-1	-	-	11652.40 gal

# Thingsboard – Usage (5)

- Βοηθητικοί σύνδεσμοι για να μπορέσετε να φτιάξετε τα σενάρια
  - [Widget configuration guide, Part 1: Basic Settings](#)
  - [Widget configuration guide, Part 2: Latest Values Map Widget](#)
  - [Widget configuration guide, Part 3: Time Series Map Widget](#)
  - [Getting started with ThingsBoard](#)
- Παραδείγματα Widget που θα χρειαστείτε

