Πανεπιστήμιο Αθηνών
Τμήμα Πληροφορικής και Τηλεπικοινωνιών
Διαχείριση Δικτύων – Network Management
Εαρινό εξάμηνο
2022 – 2023

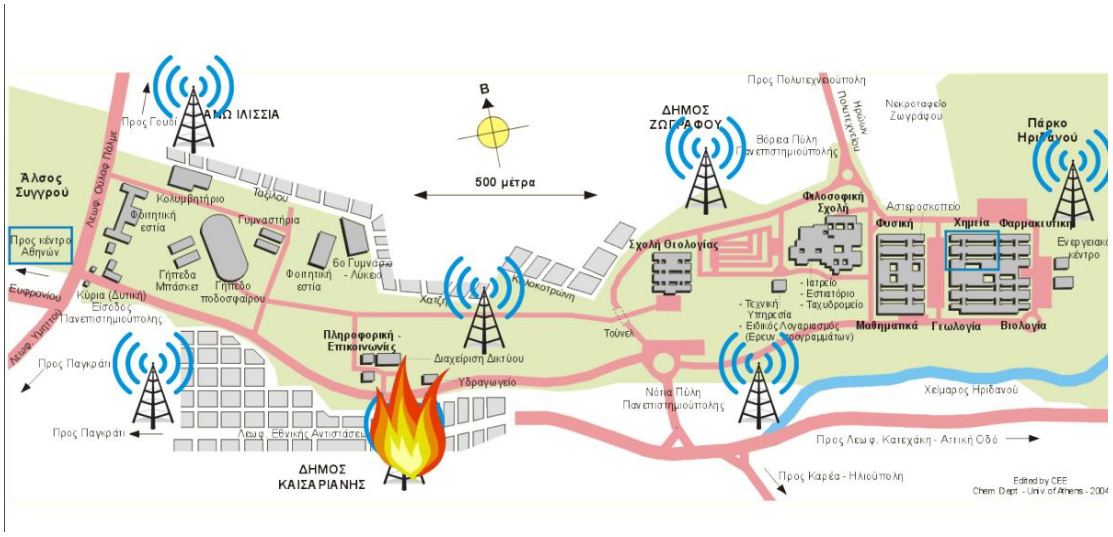
Περιεχόμενα

MININET – WIFI

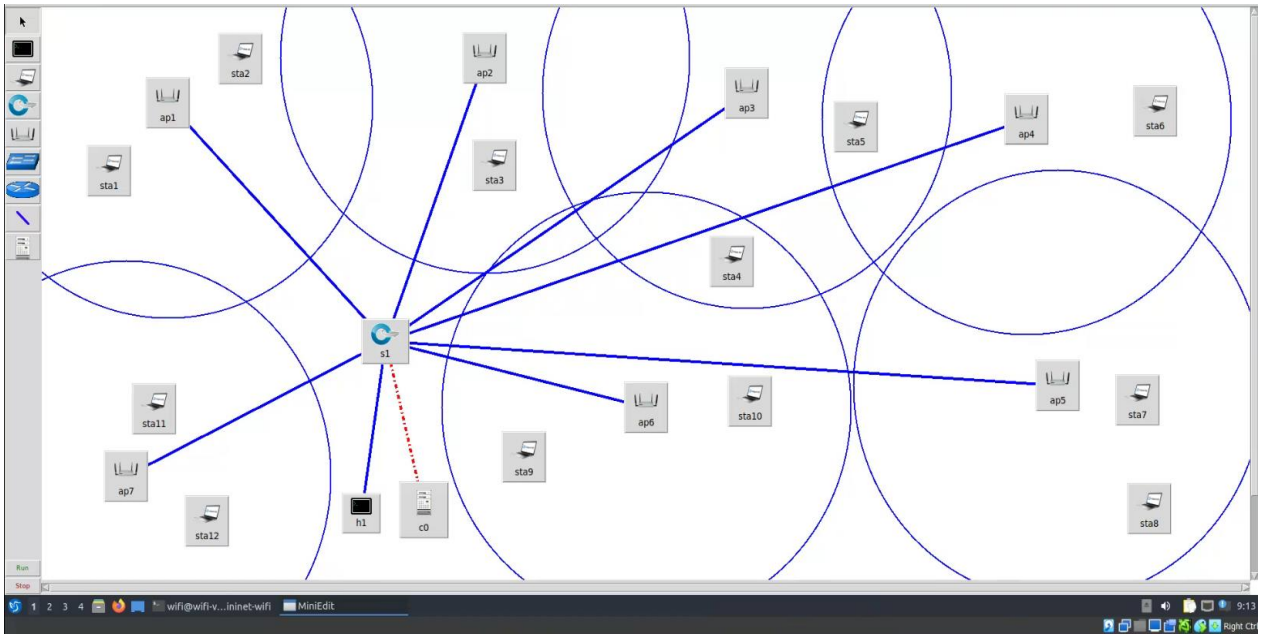
Outdoors Scenario

Για το σενάριο του εξωτερικού περιβάλλοντος καλούμαστε να προσομοιώσουμε την λειτουργία του δικτύου που απεικονίζεται στην παρακάτω εικόνα. Για το σενάριο αυτό, χρησιμοποιούμε 12 stations, 7 access points,1 controller, 1 switch, 1 host.

Χρησιμοποιήθηκε η τοπολογία του εξωτερικού χώρου.



Ακολουθώντας, την παραπάνω εικόνα και το δίκτυο που απεικονίζεται σε αυτήν, δημιουργήσαμε την παρακάτω τοπολογία με την βοήθεια του miniedit.

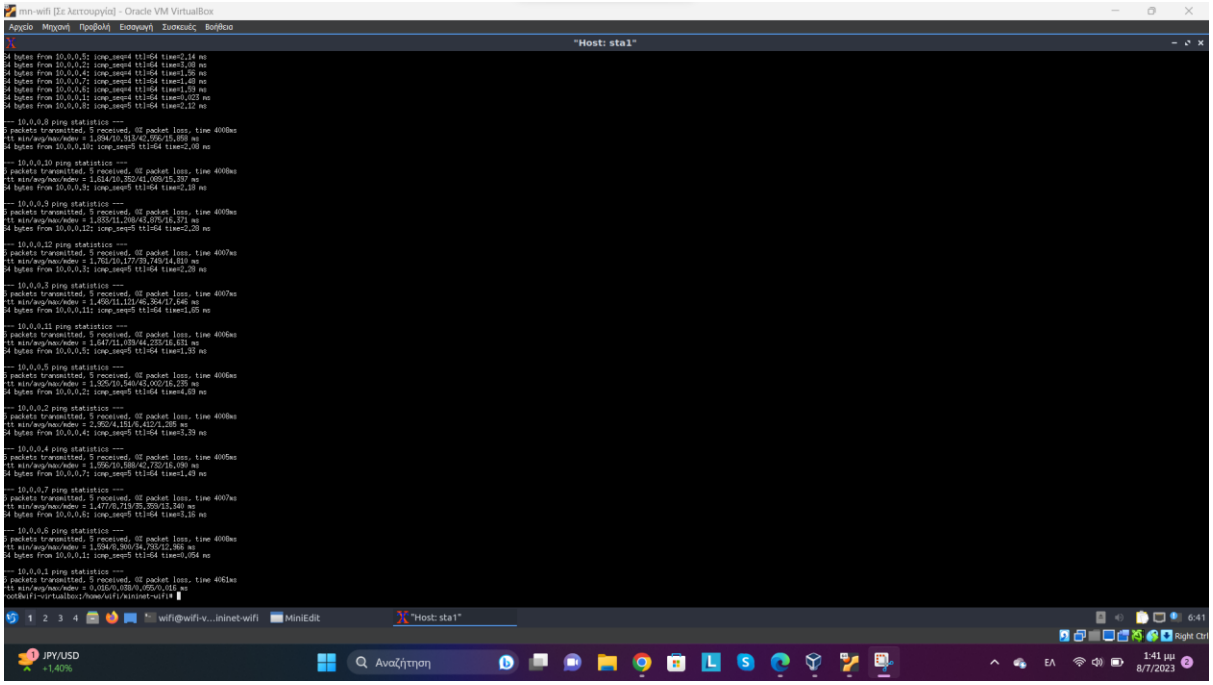


Για να είμαστε σίγουροι πως οι σταθμοί επικοινωνούν κανονικά μεταξύ τους φτιάξαμε το παρακάτω script σε γλώσσα bashscript. Στο τμήμα κώδικα που ακολουθεί έχουμε εισάγει τις διευθύνσεις των σταθμών με τις οποίες θέλουμε να επικοινωνεί ο τρέχων σταθμός. Κάνει ping με όλους τους σταθμούς και εμφανίζει συνολικά πόσα πακέτα στάλθηκαν και πόσα λήφθηκαν με επιτυχία.

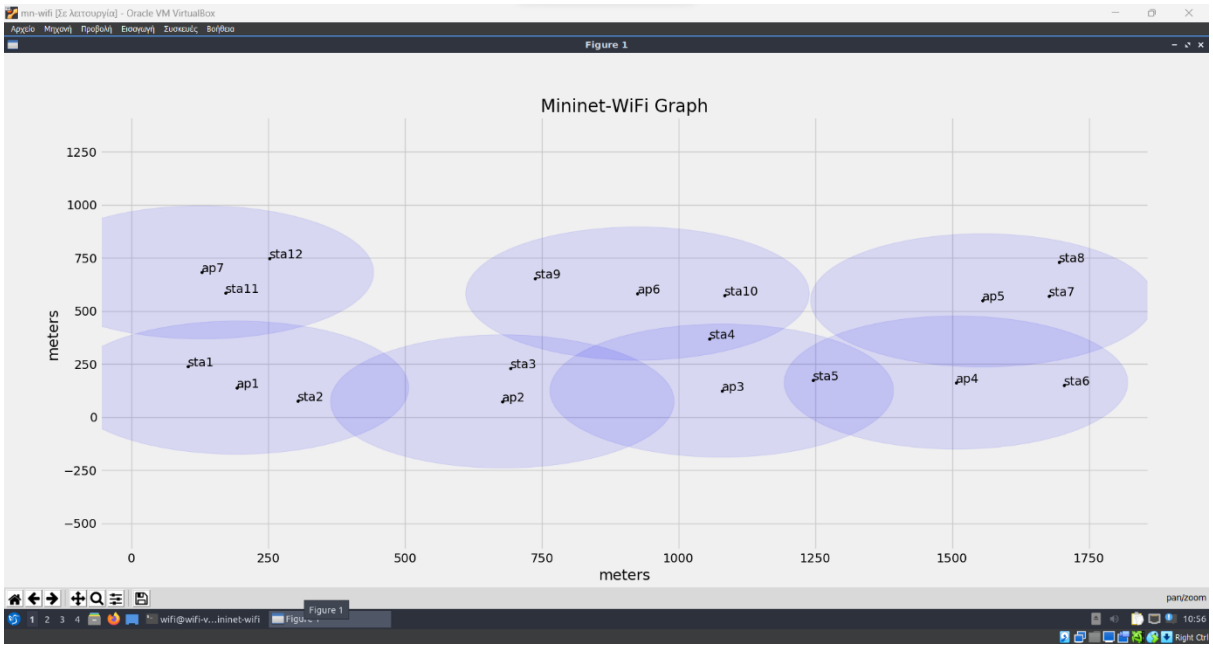
C: > Users > USER > Downloads > $ ping_all.sh

```
#!/bin/bash

# List of IP addresses of all stations except station 1
ip_addresses=("10.0.0.1" "10.0.0.2" "10.0.0.3" "10.0.0.4" "10.0.0.5" "10.0.0.6" "10.0.0.7"
              "10.0.0.8" "10.0.0.9" "10.0.0.10" "10.0.0.11" "10.0.0.12")

# Send ping requests to each station
for ip in "${ip_addresses[@]}"; do
    ping -c 5 $ip &
done
```

Στην παρακάτω εικόνα δίνεται ένα screenshot από την διαδικασία μετάδοσης και λήψης πακέτων καθώς και τα μηνύματα τα οποία λαμβάνονται από κάθε σταθμό με επιτυχία.



Στην επόμενη εικόνα απεικονίζεται το figure για την τοπολογία του εξωτερικού σεναρίου:



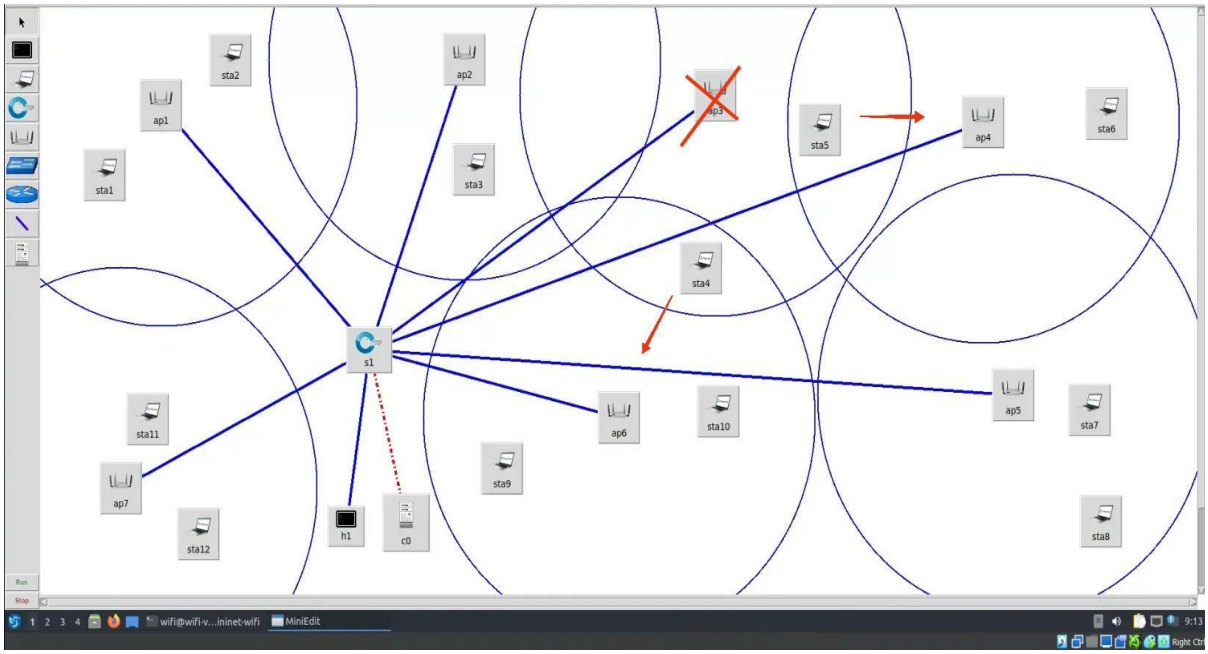
Επιπλέον, καλούμαστε να προσομοιώσουμε ένα σενάριο σχετικά με την συμπεριφορά των υπολοίπων σταθμών σε περίπτωση που μια κεραία σταματήσει να λειτουργεί. Για το συγκεκριμένο σενάριο, επιλέξαμε την εξής λειτουργικότητα:

Σε περίπτωση που μία κεραία (access point) σταματήσει να λειτουργεί, τότε η λογική είναι πως οι σταθμοί οι οποίοι ανήκουν στην συγκεκριμένη κεραία θα πρέπει να συνδεθούν στην εμβέλεια της πιο κοντινής κεραίας.

Ακολουθεί το επόμενο σενάριο:

Ας υποθέσουμε ότι η κεραία ap3 σταματάει να λειτουργεί. Επομένως, είναι αναμενόμενο να επηρεαστεί η λειτουργία των σταθμών sta4 και sta5, οι οποίοι ανήκουν στην εμβέλεια της κεραίας ap3. Στόχος, είναι με την παύση λειτουργίας της κεραίας ap3, οι σταθμοί sta4 και sta5 να συνδεθούν με τις αμέσως κοντινότερες κεραίες. Στον παρακάτω πίνακα φαίνεται η νέα αναδιαμόρφωση των σταθμών sta4 και sta5.

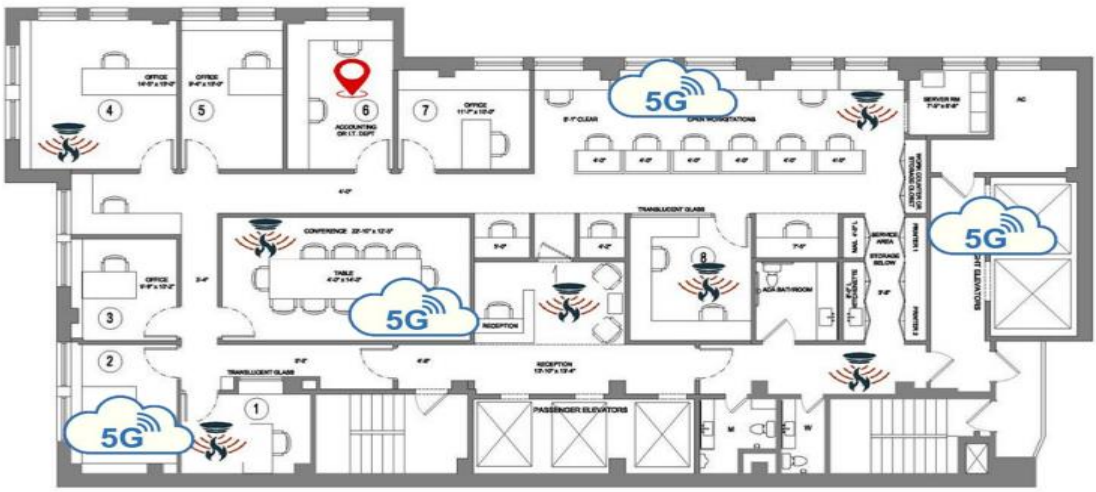
Σταθμοί	Νέες κεραίες στις οποίες θα συνδεθούν
sta4	ap6
sta5	ap4



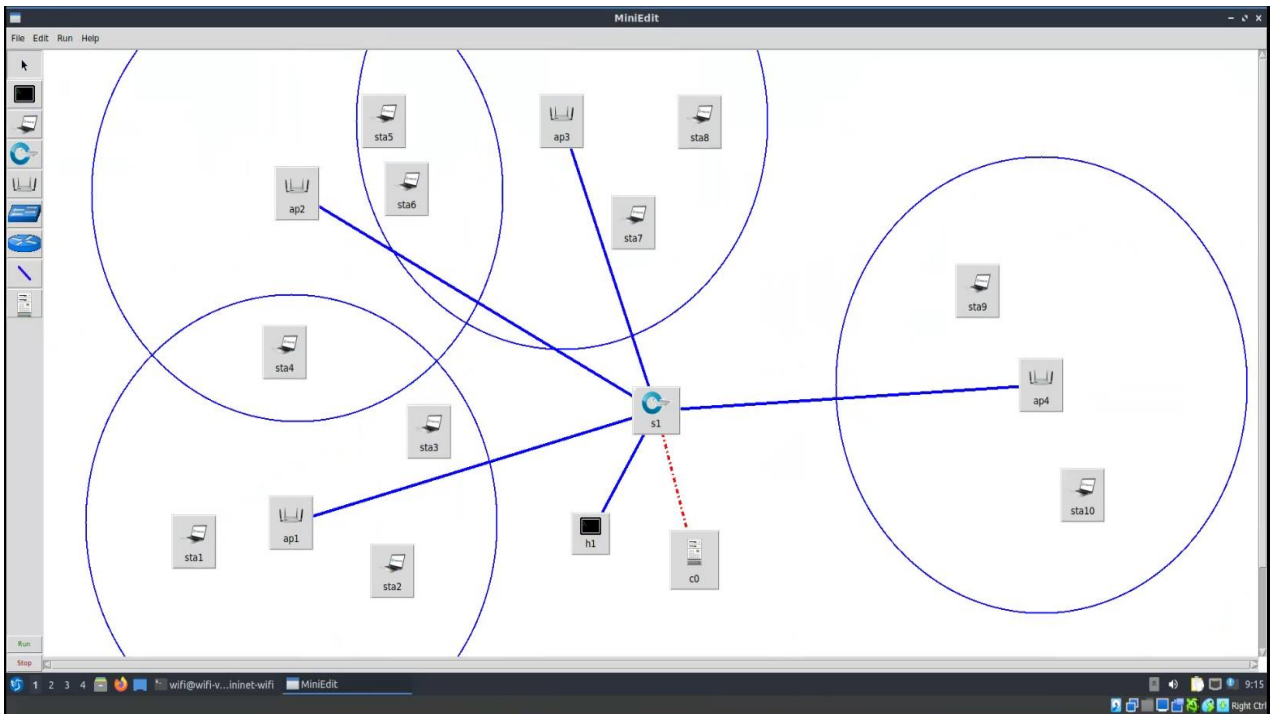
Indoors Scenario

Για το σενάριο του εσωτερικού περιβάλλοντος καλούμαστε να προσομοιώσουμε την λειτουργία του δικτύου που απεικονίζεται στην παρακάτω εικόνα. Για το σενάριο αυτό, χρησιμοποιούμε 10 stations, 4 access points,1 controller, 1 switch, 1 host.

Η τοπολογία στην οποία βασιστήκαμε είναι η παρακάτω:



Ακολουθώντας, την παραπάνω εικόνα και το δίκτυο που απεικονίζεται σε αυτήν, δημιουργήσαμε την παρακάτω τοπολογία με την βοήθεια του MiniEdit.

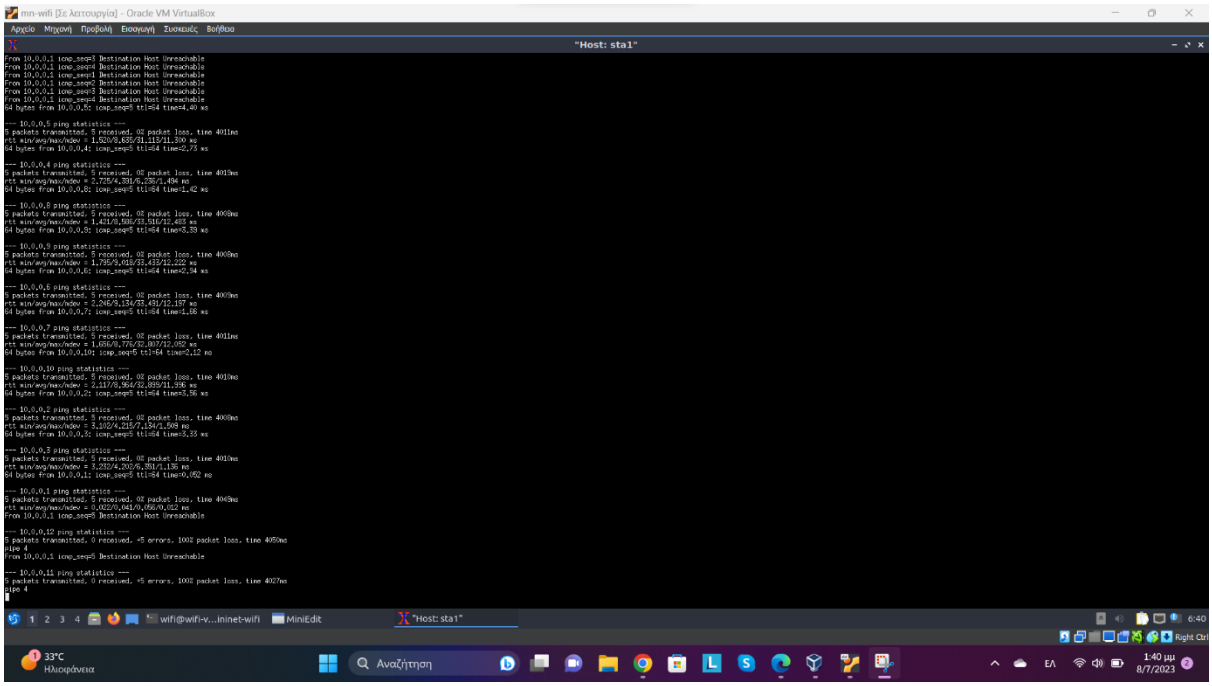


Όπως και στο προηγούμενο σενάριο, για να είμαστε σίγουροι πως οι σταθμοί επικοινωνούν κανονικά μεταξύ τους φτιάξαμε το παρακάτω script σε γλώσσα bashscript. Στο τμήμα κώδικα που ακολουθεί έχουμε εισάγει τις διευθύνσεις των σταθμών με τις οποίες θέλουμε να επικοινωνεί ο τρέχων σταθμός. Κάνει ping με όλους τους σταθμούς και εμφανίζει συνολικά πόσα πακέτα στάλθηκαν και πόσα λήφθηκαν με επιτυχία.

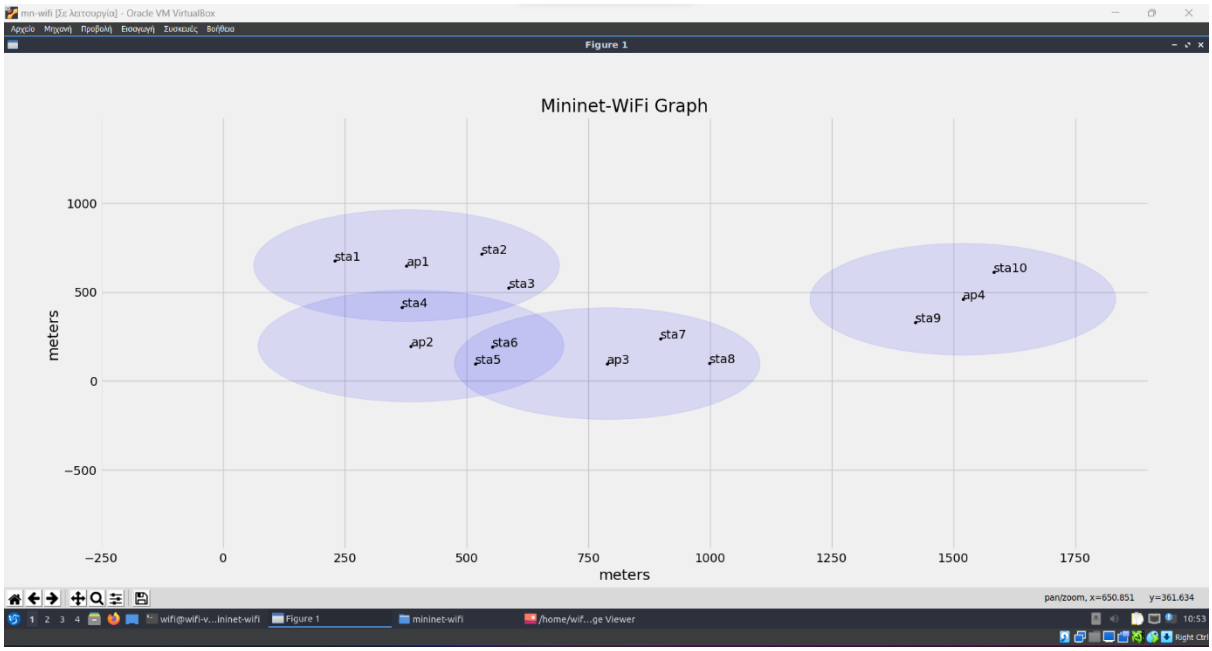
C: > Users > USER > Downloads > $ ping_all2.sh

```
#!/bin/bash

# List of IP addresses of all stations except station 1
ip_addresses=("10.0.0.1" "10.0.0.2" "10.0.0.3" "10.0.0.4" "10.0.0.5"
              "10.0.0.6" "10.0.0.7" "10.0.0.8" "10.0.0.9" "10.0.0.10")

# Send ping requests to each station
for ip in "${ip_addresses[@]}"; do
    ping -c 5 $ip &
done
```

Στην παρακάτω εικόνα δίνεται ένα screenshot από την διαδικασία μετάδοσης και λήψης πακέτων καθώς και τα μηνύματα τα οποία λαμβάνονται από κάθε σταθμό με επιτυχία.



Στην παρακάτω εικόνα, απεικονίζεται το διάγραμμα (figure) για το σενάριο indoors:



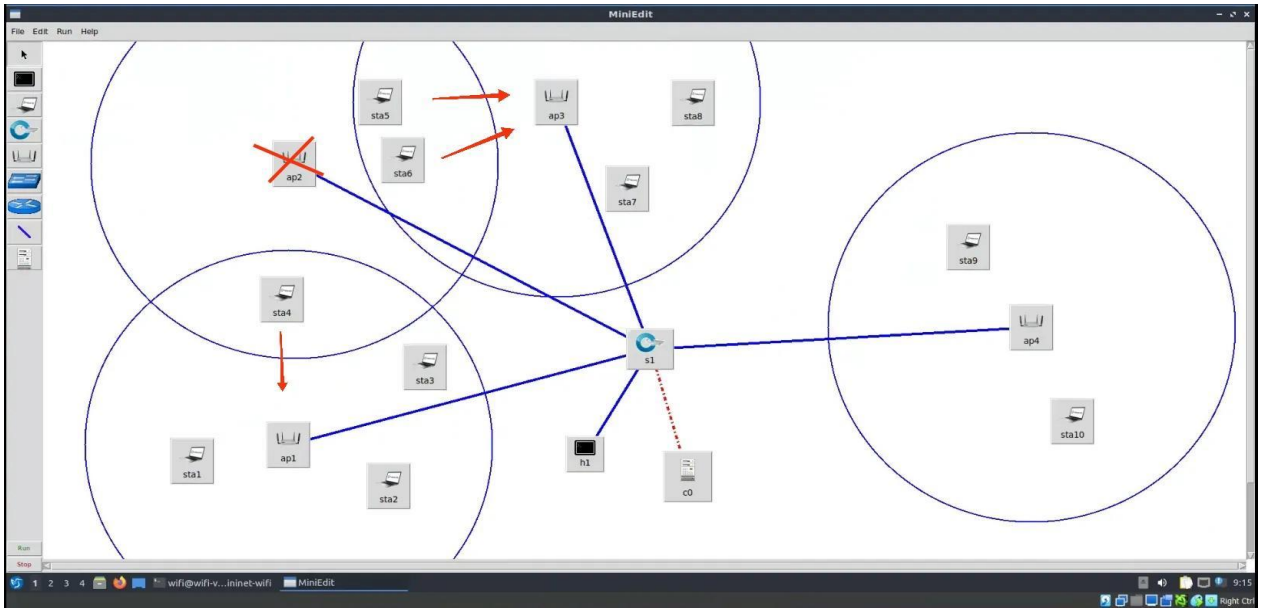
Επιπλέον, καλούμαστε να προσομοιώσουμε ένα σενάριο σχετικά με την συμπεριφορά των υπολοίπων σταθμών σε περίπτωση που μία κεραία εσωτερικού χώρου σταματήσει να λειτουργεί. Για το συγκεκριμένο σενάριο, επιλέξαμε την εξής λειτουργικότητα:

Σε περίπτωση που μία κεραία (access point) σταματήσει να λειτουργεί, τότε η λογική είναι πως οι σταθμοί οι οποίοι ανήκουν στην συγκεκριμένη κεραία θα πρέπει να συνδεθούν στην εμβέλεια της πιο κοντινής κεραίας.

Ακολουθεί το επόμενο σενάριο:

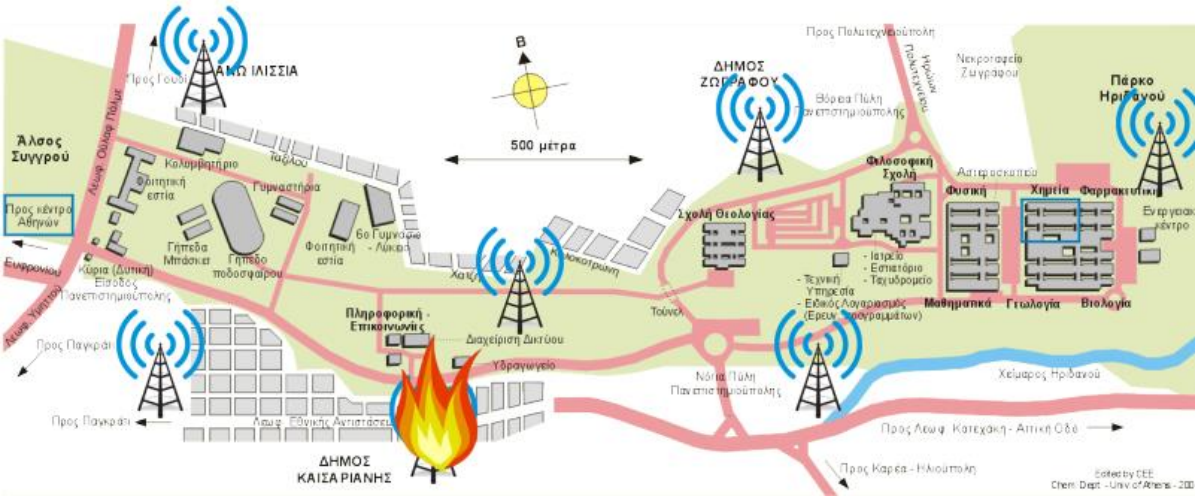
Ας υποθέσουμε ότι η κεραία ap2 σταματάει να λειτουργεί. Επομένως, είναι αναμενόμενο να επηρεαστεί η λειτουργία των σταθμών sta4, sta5 και sta6, οι οποίοι ανήκουν στην εμβέλεια της κεραίας ap2. Στόχος, είναι με την παύση λειτουργίας της κεραίας ap2, οι σταθμοί sta4, sta5 και sta6 να συνδεθούν με τις αμέσως κοντινότερες κεραίες. Στον παρακάτω πίνακα φαίνεται η νέα αναδιαμόρφωση των σταθμών sta4, sta5 και sta6.

Σταθμοί	Νέες κεραίες στις οποίες θα συνδεθούν
sta4	ap1
sta5	ap3
sta6	ap3



IOT (THINGSBOARD)

Σενάριο για Outdoors



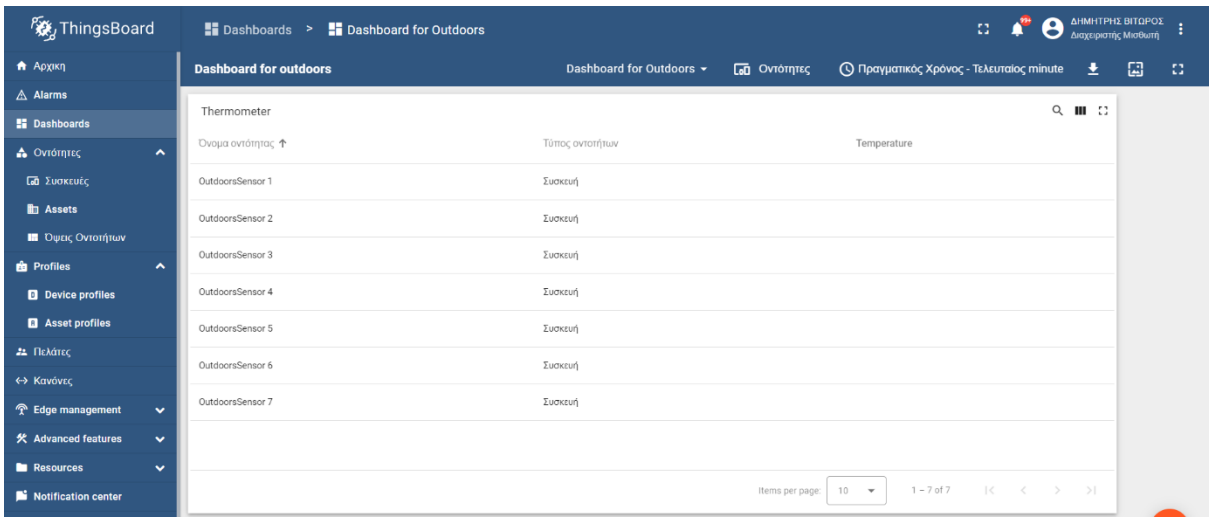
Στο συγκεκριμένο σενάριο, προσομοιώσαμε την περίπτωση που κάποιος εξωτερικός αισθητήρας φτάσει σε πολύ υψηλή θερμοκρασία. Σε

περίπτωση που ο αισθητήρας ξεπεράσει την θερμοκρασία των 58°C, ενεργοποιείται ο alarm που έχουμε δημιουργήσει για την κάθε περίπτωση. Ειδικότερα:

Δημιουργήσαμε 7 αισθητήρες ανίχνευσης θερμοκρασίας για τον εξωτερικό χώρο. Για να εισάγουμε τιμές στους αισθητήρες κάνουμε χρήση της εντολής curl μέσω του terminal. Η εντολή που εισάγουμε είναι η εξής:

```
curl -v -X POST -d "{\"temperature\":
25}"https://demo.thingsboard.io/api/v1/ABC123/telemetry --header
"Content-Type:application/json"
```

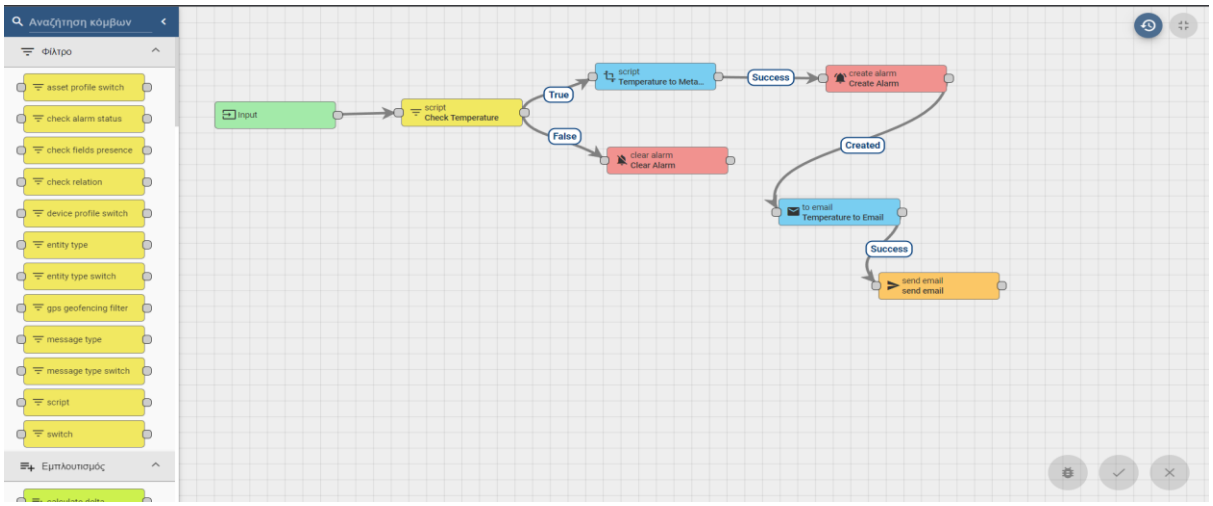
Στην παραπάνω εντολή για κάθε συσκευή θα πρέπει να βάζουμε την επιθυμητή θερμοκρασία αλλά και το $ACCESS_TOKEN.

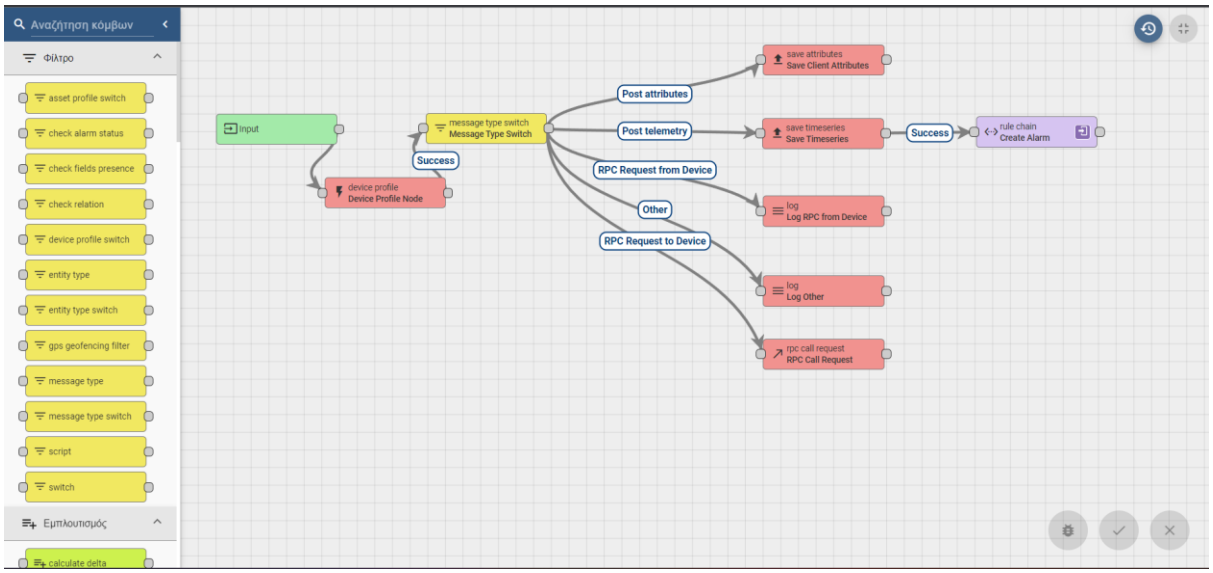


Όπως φαίνεται στην παραπάνω εικόνα δημιουργήσαμε 7 αισθητήρες εξωτερικού χώρου. Με βάση και το σενάριο που υλοποιήσαμε και στο πρώτο σκέλος της εργασίας με το Mininet, ο αισθητήρας ο οποίος θα ξεπεράσει τον επιθυμητό βαθμό θερμοκρασίας (58°C) είναι ο OutdoorsSensor 3. Οι υπόλοιποι αισθητήρες είναι ρυθμισμένοι να έχουν κατάλληλη θερμοκρασία.

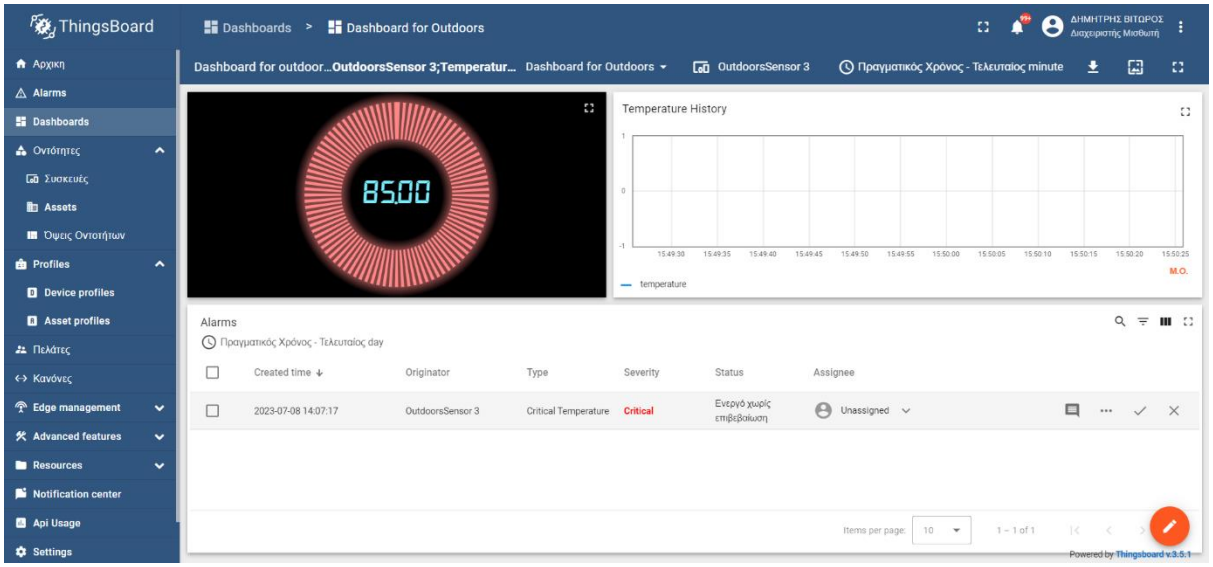
Στόχος μας είναι όταν η θερμοκρασία του αισθητήρα ξεπερνάει τους 58°C, να δίνεται μήνυμα από το alert και να ειδοποιεί τον χρήστη για το γεγονός ότι η θερμοκρασία είναι κρίσιμη (critical).

Για την δημιουργία του alarm μεταβήκαμε στους κανόνες αλυσίδας και δημιουργήσαμε μία καινούργια με το όνομα Create Alarm. Σε αυτήν την αλυσίδα στο script θέσαμε τον επιθυμητό βαθμό θερμοκρασίας που θέλουμε να έχουμε. Αν η θερμοκρασία είναι μεγαλύτερη των 58°C τότε η συνθήκη είναι αληθής και κάνουμε create alarm, διαφορετικά αν η συνθήκη είναι ψευδής τότε κάνουμε clear alarm. Επιπλέον, δημιουργήθηκε μία σύνδεση η οποία στέλνει email στον χρήστη σε περίπτωση που ξεπεραστεί η θερμοκρασία. Η αλυσίδα Create Alarm που δημιουργήσαμε ενώνεται με το Root Rule Chain με την μεταβλητή success στο Save Timeseries.





Αφού δημιουργήσαμε τους αισθητήρες, προχωρήσαμε στην δημιουργία του Dashboard for Outdoors, στο οποίο απεικονίζουμε με την βοήθεια των widgets την θερμοκρασία, το διάγραμμα με τις μεταβολές της θερμοκρασίας και τα alarms τα οποία μας εμφανίζονται σε κάθε περίπτωση. Στην παρακάτω εικόνα, το διάγραμμα εμφανίζεται με την συγκεκριμένη μορφή διότι μεταβάλαμε την θερμοκρασία πολλές φορές για να εξασφαλίσουμε μεγαλύτερο εύρος τιμών.



Στην συγκεκριμένη περίπτωση, η θερμοκρασία για τον αισθητήρα OutdoorsSensor 3 είναι 85°C > 58°C, άρα εμφανίζεται alarm με την ένδειξη Critical.

Σενάριο για Indoors

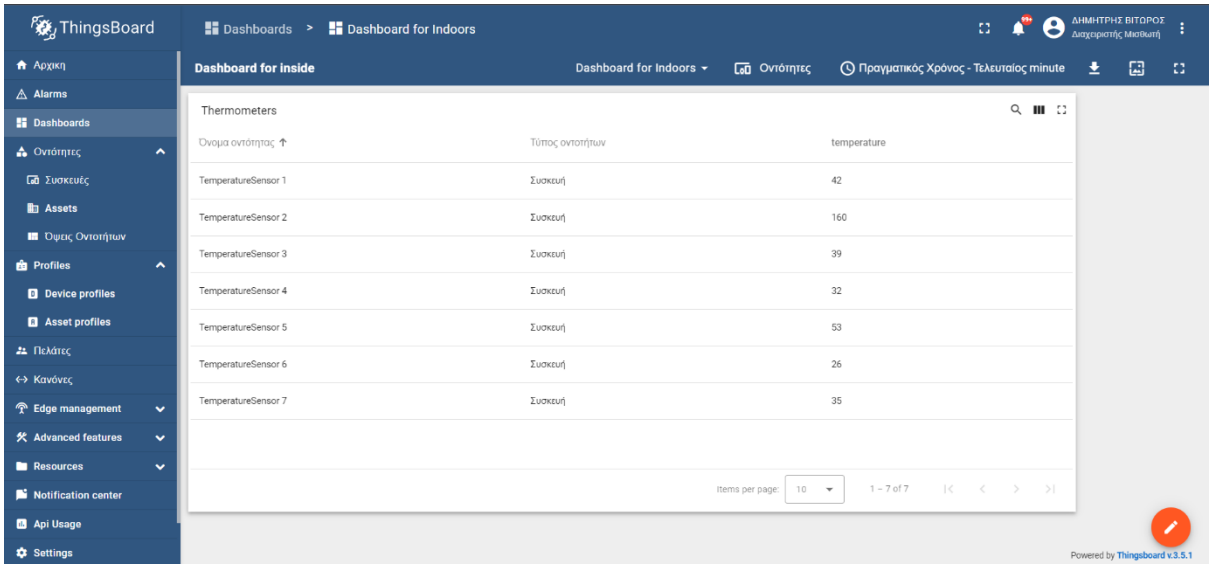
Στο συγκεκριμένο σενάριο, προσομοιώσαμε την περίπτωση που κάποιος εσωτερικός αισθητήρας φτάσει σε πολύ υψηλή θερμοκρασία. Σε περίπτωση που ο αισθητήρας ξεπεράσει την θερμοκρασία των 58°C, ενεργοποιείται ο alarm που έχουμε δημιουργήσει για την κάθε περίπτωση.

Ειδικότερα:

Δημιουργήσαμε 7 αισθητήρες ανίχνευσης θερμοκρασίας για τον εσωτερικό χώρο. Για να εισάγουμε τιμές στους αισθητήρες κάνουμε χρήση της εντολής curl μέσω του terminal. Η εντολή που εισάγουμε είναι η εξής:

```
curl -v -X POST -d "{\"temperature\":
25}"https://demo.thingsboard.io/api/v1/ABC123/telemetry --header
"Content-Type:application/json"
```

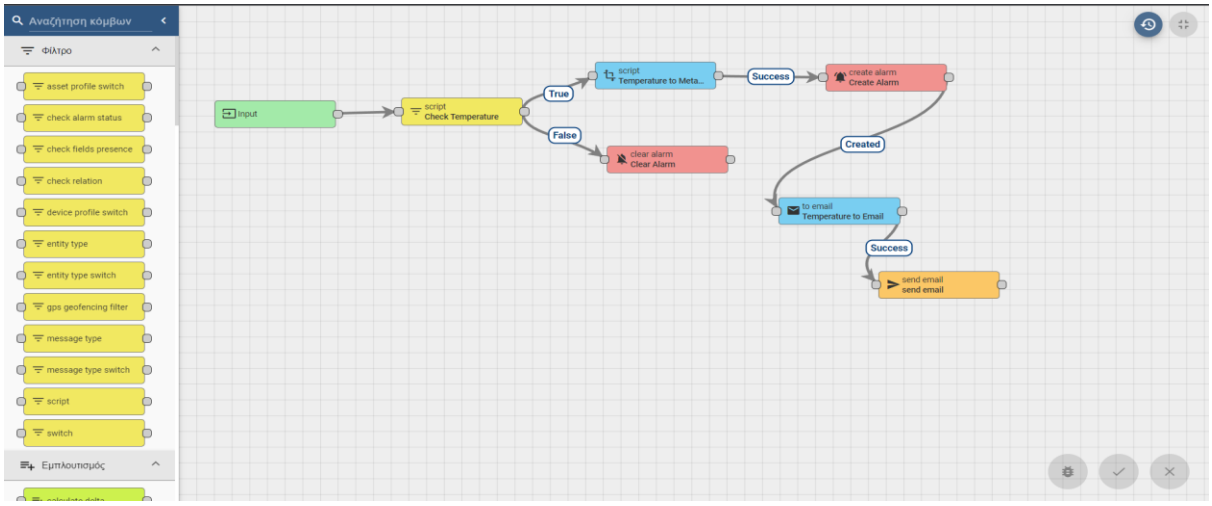
Στην παραπάνω εντολή για κάθε συσκευή θα πρέπει να βάζουμε την επιθυμητή θερμοκρασία και στην θέση του “ABC123” θα βάζουμε το $ACCESS_TOKEN της κάθε συσκευής.

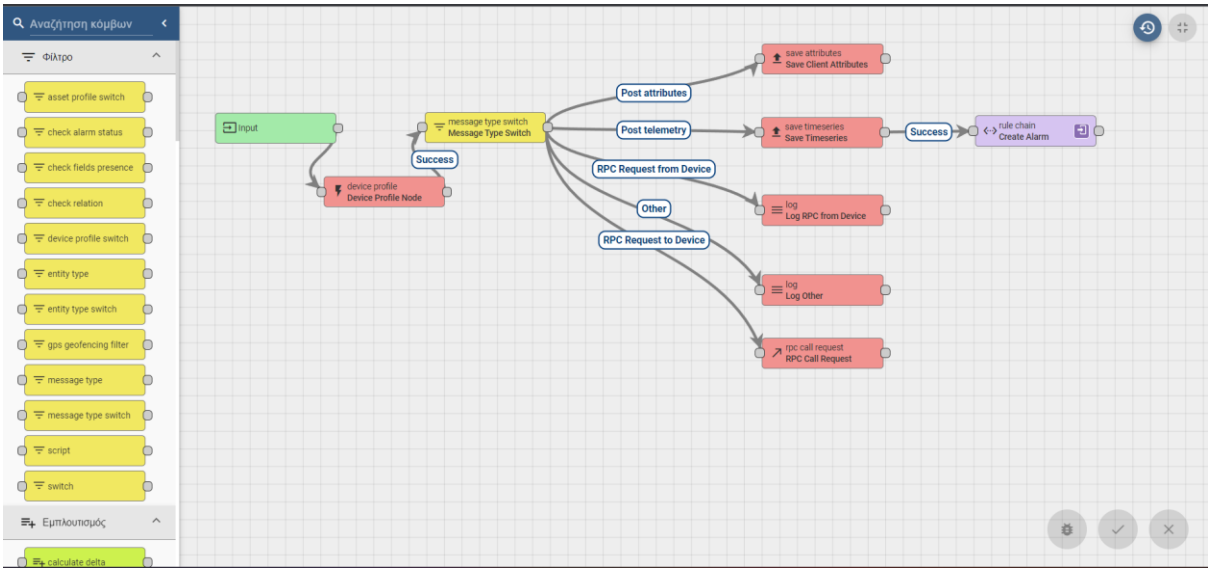


Όπως φαίνεται στην εικόνα παραπάνω δημιουργήσαμε 7 αισθητήρες. Με βάση και το σενάριο που υλοποιήσαμε και στο πρώτο σκέλος της εργασίας με το Mininet, ο αισθητήρας ο οποίος θα ξεπεράσει τον επιθυμητό βαθμό θερμοκρασίας (58°C) είναι ο TemperatureSensor 2. Οι υπόλοιποι αισθητήρες έχουν κατάλληλη θερμοκρασία.

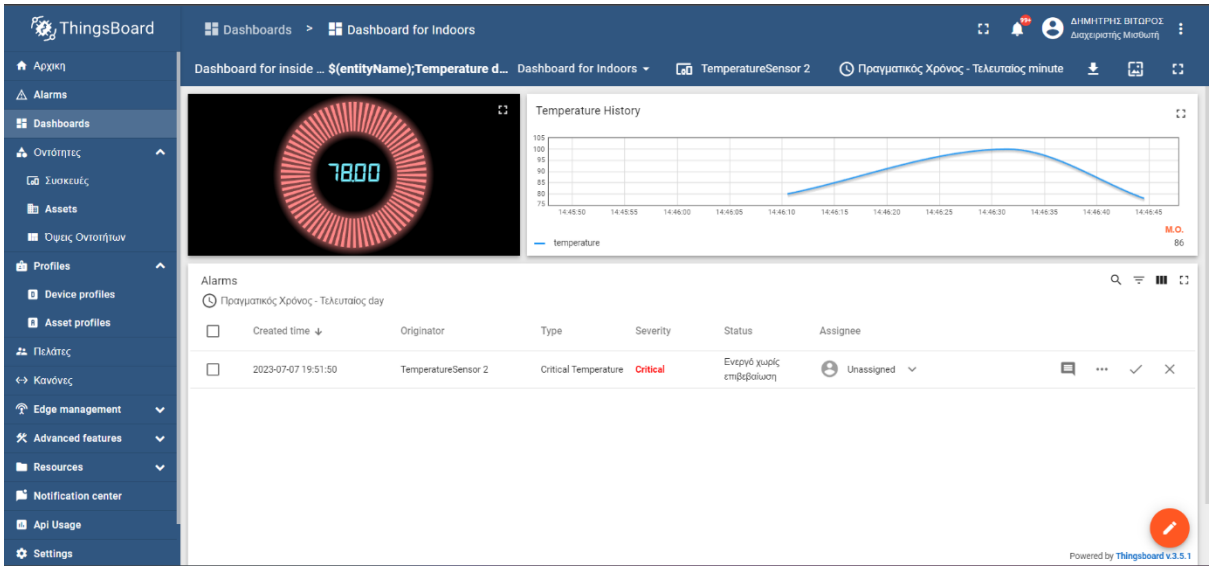
Στόχος μας είναι όταν η θερμοκρασία του αισθητήρα ξεπερνάει τους 58°C, να δίνεται μήνυμα από το alert και να ειδοποιεί τον χρήστη για το γεγονός ότι η θερμοκρασία είναι κρίσιμη.

Για την δημιουργία του alarm μεταβήκαμε στους κανόνες αλυσίδας και δημιουργήσαμε μία καινούργια με το όνομα Create Alarm. Σε αυτήν την αλυσίδα στο script θέσαμε τον επιθυμητό βαθμό θερμοκρασίας που θέλουμε να έχουμε. Αν η θερμοκρασία είναι μεγαλύτερη των 58°C τότε η συνθήκη είναι αληθής και κάνουμε create alarm, διαφορετικά αν η συνθήκη είναι ψευδής τότε κάνουμε clear alarm. Επιπλέον, δημιουργήθηκε μία σύνδεση η οποία στέλνει email στον χρήστη σε περίπτωση που ξεπεραστεί η θερμοκρασία. Η αλυσίδα Create Alarm που δημιουργήσαμε ενώνεται με το Root Rule Chain με την μεταβλητή success στο Save Timeseries.





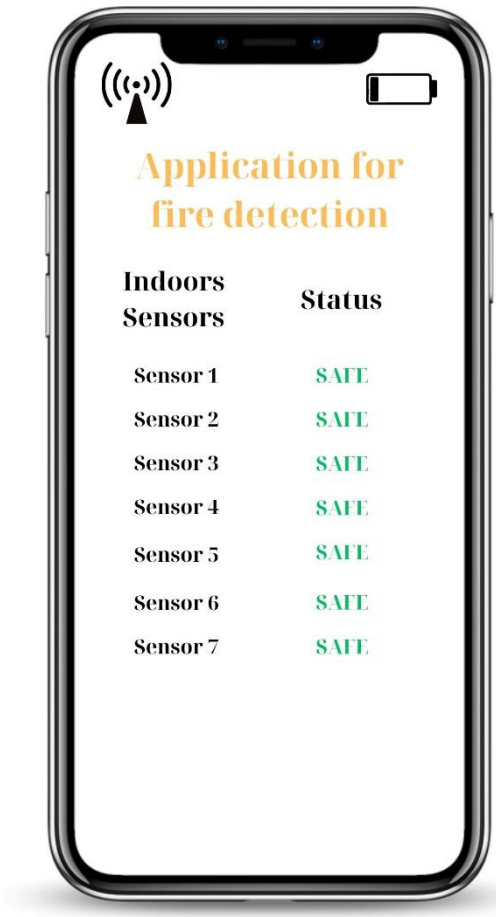
Αφού δημιουργήσαμε τους αισθητήρες, προχωρήσαμε στην δημιουργία του Dashboard for Indoors, στο οποίο απεικονίζουμε με την βοήθεια των widgets την θερμοκρασία, το διάγραμμα με τις μεταβολές της θερμοκρασίας και τα alarms τα οποία μας εμφανίζονται σε κάθε περίπτωση. Στην παρακάτω εικόνα το διάγραμμα εμφανίζεται με την συγκεκριμένη μορφή, διότι μεταβάλαμε την θερμοκρασία πολλές φορές για μεγαλύτερο εύρος τιμών.



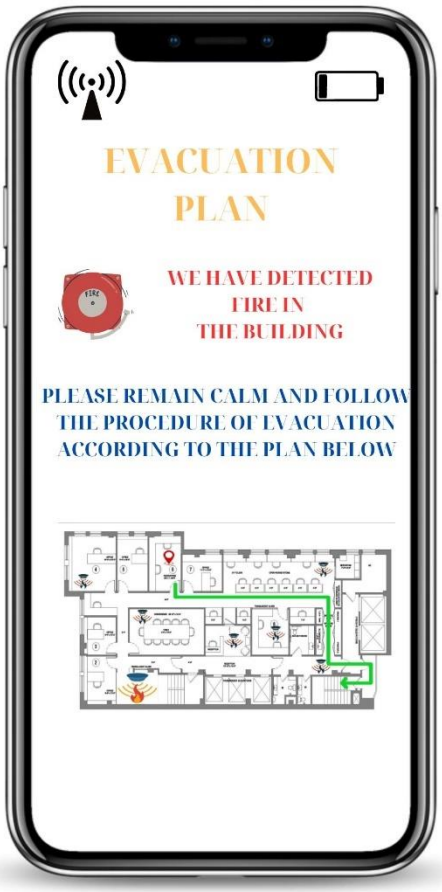
Στην συγκεκριμένη περίπτωση, η θερμοκρασία είναι 78°C > 58°C, άρα εμφανίζεται alarm με την ένδειξη Critical.

MOCK UPS ΓΙΑ INDOORS SCENARIO

Για το Indoors scenario, δημιουργήσαμε τα παρακάτω mock ups. Σε αυτά τα mock ups γίνεται προσομοίωση μιας κινητής συσκευής που θα μπορεί να διαθέτει ο κάθε χρήστης. Σε αυτήν την συσκευή, θα έχουμε συνοπτικά όλους τους sensors εσωτερικού χώρου οι οποίοι ανιχνεύουν την ύπαρξη πυρκαγιάς. Σε περίπτωση που ανιχνευτεί πυρκαγιά στον χώρο, στέλνεται ειδοποίηση στην συσκευή αναφορικά με το ποιος ανιχνευτής εντόπισε τον κίνδυνο και στην συνέχεια στέλνεται το πλάνο εκκένωσης σύμφωνα με το πρωτόκολλο. Παρακάτω φαίνονται τα mock ups που δημιουργήθηκαν:


Application for fire detection
Indoors Sensors
Status
Sensor 1
SAFE
Sensor 2
SAFE
Sensor 3
SAFE
Sensor 4
SAFE
Sensor 5
SAFE
Sensor 6
SAFE
Sensor 7
SAFE


Application for fire detection
Indoors Sensors
Status
Sensor 1
SAFE
Sensor 2
SAFE
Sensor 3
DANGER
Sensor 4
SAFE
Sensor 5
SAFE
Sensor 6
SAFE
Sensor 7
SAFE
ATTENTION PLEASE!!!
SENSOR 3 WARNS FOR FIRE
PLEASE FOR YOUR OWN SAFETY EVACUATE THE BUILDING!!!


EVACUATION PLAN
WE HAVE DETECTED FIRE IN THE BUILDING
PLEASE REMAIN CALM AND FOLLOW THE PROCEDURE OF EVACUATION ACCORDING TO THE PLAN BELOW