

## ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ



ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ

Εδυνικόν και Καποδιστριακόν  
Πανεπιστήμιον Αδηνών

— ΙΔΡΥΘΕΝ ΤΟ 1837 —



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

Υπεύθυνος καθηγητής: Nancy Alonisioti

Εαρινό εξάμηνο 2023

Κωνσταντίνος - Μάριος Σγούρας 1115202000178

Βασίλειος Φούτρης 1115202000231

# **Πίνακας Περιεχομένων**

<b>Πίνακας Περιεχομένων.....</b>	<b>2</b>
<b>Εισαγωγή.....</b>	<b>3</b>
<b>Mininet-Wifi.....</b>	<b>3</b>
Περιγραφή σχεδιασμού και προσομοίωσης ( Outdoor ).....	3
Σφάλματα δικτύου ( Outdoor ).....	4
Περιγραφή σχεδιασμού και προσομοίωσης ( Indoor ).....	7
Σφάλματα δικτύου ( Indoor ).....	7
<b>Thingsboard.....</b>	<b>9</b>
Δημιουργία Συσκευών ( Outdoor ).....	9
Δημιουργία Συσκευών ( Indoor ).....	10
Δημιουργία συστήματος προσομοίωσης.....	10
Δημιουργία Dashboard ( Outdoor ).....	11
Δημιουργία Dashboard ( Indoor ).....	11
<b>Mockups.....</b>	<b>12</b>
Mockup Outdoor Σεναρίου.....	12
Mockup Indoor Σεναρίου.....	13

# **Εισαγωγή**

Σκοπός μας στην εργασία ήταν η προσομοίωση ενός σφάλματος και η εύρεση λύσης σε δύο δίκτυα (ένα εσωτερικό και ένα εξωτερικό). Η εφαρμογή που μας ενδιαφέρει να μείνει ενεργή παρά το σφάλμα είναι μια εφαρμογή πλοιήγησης προς την κοντινότερη έξοδο. Για τη προσομοίωση του δικτύου και την οπτική παρουσίαση του δικτυακού κομματιού της εργασίας χρησιμοποιήσαμε το mininet-wifi ένα extension του mininet για ασύρματα δίκτυα wifi. Εκεί παρουσιάσαμε μέσα από σενάρια τη συμπεριφορά του δικτύου κατά τη διάρκεια μιας καταστροφής. Για την λειτουργία των αισθητήρων και την ανίχνευση σφαλμάτων χρησιμοποιήσαμε τη πλατφόρμα Thingsboard. Τέλος δημιουργήσαμε τα mockups της εφαρμογής που παρουσιάζουν τα σενάρια σφάλματος στα δίκτυα.

## **Mininet-Wifi**

### **Περιγραφή σχεδιασμού και προσομοίωσης ( Outdoor )**

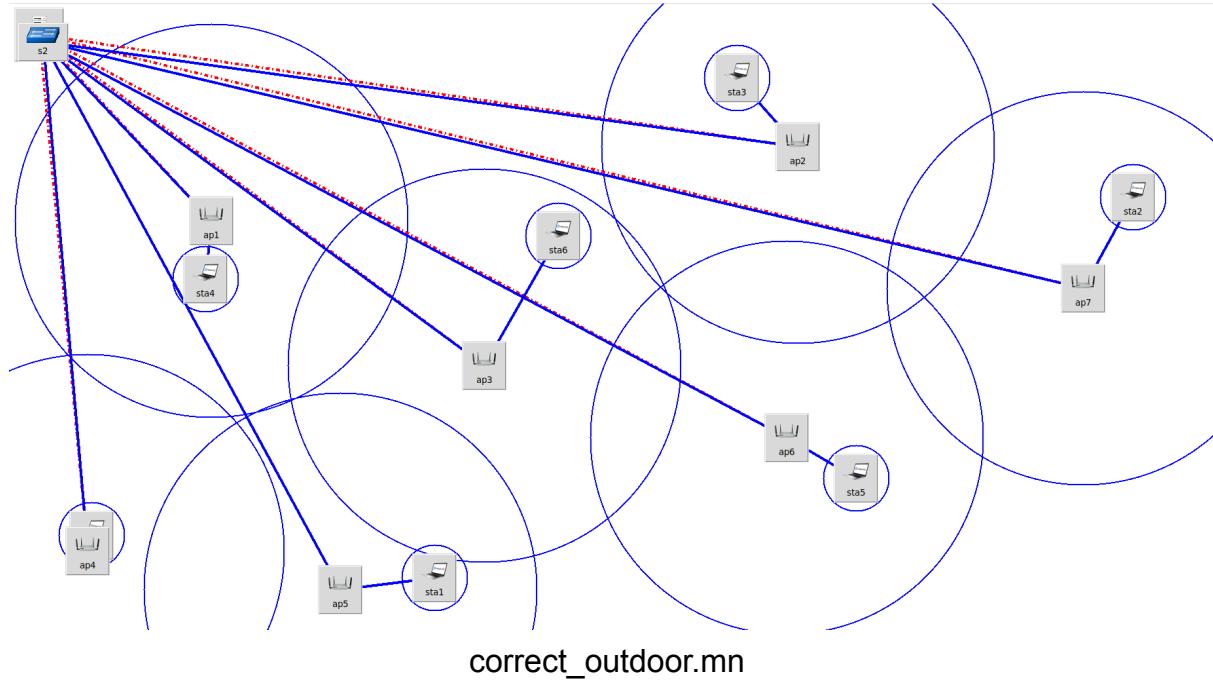
Χρησιμοποιήσαμε το mininet-wifi για τον σχεδιασμό και την προσομοίωση του δικτύου μας. Για το σχεδιασμό χρησιμοποιούμε το UI εργαλείο αν και πειραματιστήκαμε με δικό μας κώδικα για να προσπεράσουμε κάποιος περιορισμούς του εργαλείου. Αρχικά δημιουργήσαμε τις 7 κεραίες χρησιμοποιώντας access points. Άλλάξαμε τα ip addresses και τα βάλαμε σταθερά στους αριθμούς από 10.0.0.93-10.0.0.99. Το IP base του συστήματος ήταν το 10.0.0.0/8 άρα χρησιμοποιήσαμε 10 από τις τελευταίες διευθύνσεις και τις κάναμε σταθερές. Η τοποθέτηση των κεραιών βασίστηκε στο παράδειγμα που δόθηκε και στον εξωτερικό χώρο της πανεπιστημιούπολης. Για αυτό τον λόγο θέσαμε το εύρος των κεραιών στα 300 μέτρα. Καταλήξαμε σε αυτό το νούμερο και λόγο των θέσεων των access points και παίρνοντας υπόψη και την πραγματική δυνατότητα 5G κεραιών. Όλα τα access points συνδέονται μεταξύ του ενσύρματα μέσω ενός legacy switch. Ύστερα από αναζήτηση μας καταλάβαμε ότι παρότι υπάρχουν δίκτυα με ασύρματες συνδέσεις στο εσωτερικό τους (mesh συστήματα) τα περισσότερα πραγματικά συστήματα συνδέουν τις κεραίες μεταξύ τους ενσύρματα. Επίσης δοκιμάσαμε να δημιουργήσουμε ενα mesh δίκτυο στο mininet-wifi καθώς δίνει τη δυνατότητα για την αλλαγή λειτουργικότητας μια σύνδεσης σε mesh. Όμως η προσπάθεια μας δεν λειτούργησε λόγο περιορισμού του gui εργαλείου (το python αρχείο που γινόταν export δεν ήταν σωστό). Στη συνέχεια ενώσαμε όλες τις κεραίες μας με τον sdc controller μας (θα μπορούσε να ήταν χρήσιμος σε μια πιθανή διεπαφή με το thingsboard). Τέλος προσθέσαμε τα 7 μηχανήματα μας (ένα για κάθε κεραία για τους σκοπούς της όμορφης παρουσίασης τους). Βάλαμε το εύρος τους στα 50 μέτρα όπως είναι το εύρος ενός συμβατικού μηχανήματος. Συνδέσαμε κάθε συσκευή με μια κεραία ενσύρματα. Αυτό γνωρίζουμε ότι είναι πρακτικό λάθος αλλα κυρίως έγινε για

την καλύτερη παρουσίαση της εργασίας. Με τη διαγραφή των συνδέσεων η λειτουργικότητα που θα εξηγήσουμε στη συνέχεια δεν αλλάζει. Επόμενο βήμα ήταν η προσομοίωση του δικτύου. Η προσομοίωση περιλαμβάνει την έναρξη του δικτύου και την εκτέλεση της εντολής `pingallfull` όπου ελέγχει τη σύνδεση μεταξύ όλων των τερματικών στο δίκτυο. Μία επιτυχής εκτέλεση της προσομοίωσης που περιλαμβάνει τις καθυστερήσεις μεταξύ κόμβων και τις επιτυχής συνδέσεις είναι οι εξής (Outdoor):

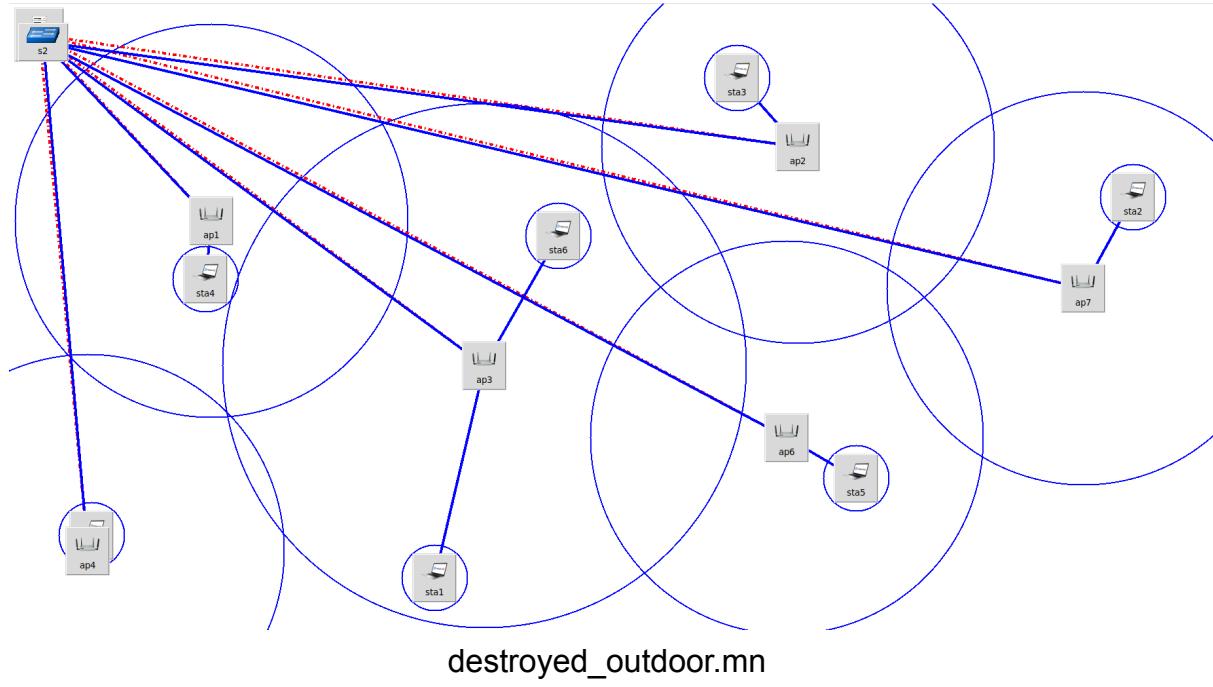
```
sta1 -> sta2sta3sta4sta5sta6sta7
sta2 -> sta1sta3sta4sta5sta6sta7
sta3 -> sta1sta2sta4sta5sta6sta7
sta4 -> sta1sta2sta3sta5sta6sta7
sta5 -> sta1sta2sta3sta4sta6sta7
sta6 -> sta1sta2sta3sta4sta5sta7
sta7 -> sta1sta2sta3sta4sta5sta6
*** Results:
sta1->sta2: 1/1, rtt min/avg/max/mdev 8.291/8.291/8.291/0.000 ms
sta1->sta3: 1/1, rtt min/avg/max/mdev 7.336/7.336/7.336/0.000 ms
sta1->sta4: 1/1, rtt min/avg/max/mdev 6.480/6.480/6.480/0.000 ms
sta1->sta5: 1/1, rtt min/avg/max/mdev 6.882/6.882/6.882/0.000 ms
sta1->sta6: 1/1, rtt min/avg/max/mdev 3.294/3.294/3.294/0.000 ms
sta1->sta7: 1/1, rtt min/avg/max/mdev 6.810/6.810/6.810/0.000 ms
sta2->sta1: 1/1, rtt min/avg/max/mdev 3.383/3.383/3.383/0.000 ms
sta2->sta3: 1/1, rtt min/avg/max/mdev 6.147/6.147/6.147/0.000 ms
sta2->sta4: 1/1, rtt min/avg/max/mdev 5.795/5.795/5.795/0.000 ms
sta2->sta5: 1/1, rtt min/avg/max/mdev 5.793/5.793/5.793/0.000 ms
sta2->sta6: 1/1, rtt min/avg/max/mdev 5.740/5.740/5.740/0.000 ms
sta2->sta7: 1/1, rtt min/avg/max/mdev 5.056/5.056/5.056/0.000 ms
sta3->sta1: 1/1, rtt min/avg/max/mdev 3.743/3.743/3.743/0.000 ms
sta3->sta2: 1/1, rtt min/avg/max/mdev 2.755/2.755/2.755/0.000 ms
sta3->sta4: 1/1, rtt min/avg/max/mdev 5.349/5.349/5.349/0.000 ms
sta3->sta5: 1/1, rtt min/avg/max/mdev 6.557/6.557/6.557/0.000 ms
sta3->sta6: 1/1, rtt min/avg/max/mdev 5.267/5.267/5.267/0.000 ms
sta3->sta7: 1/1, rtt min/avg/max/mdev 5.054/5.054/5.054/0.000 ms
sta4->sta1: 1/1, rtt min/avg/max/mdev 2.305/2.305/2.305/0.000 ms
sta4->sta2: 1/1, rtt min/avg/max/mdev 2.354/2.354/2.354/0.000 ms
sta4->sta3: 1/1, rtt min/avg/max/mdev 2.201/2.201/2.201/0.000 ms
sta4->sta5: 1/1, rtt min/avg/max/mdev 4.721/4.721/4.721/0.000 ms
sta4->sta6: 1/1, rtt min/avg/max/mdev 5.361/5.361/5.361/0.000 ms
sta4->sta7: 1/1, rtt min/avg/max/mdev 5.328/5.328/5.328/0.000 ms
sta5->sta1: 1/1, rtt min/avg/max/mdev 2.601/2.601/2.601/0.000 ms
sta5->sta2: 1/1, rtt min/avg/max/mdev 2.481/2.481/2.481/0.000 ms
sta5->sta3: 1/1, rtt min/avg/max/mdev 2.025/2.025/2.025/0.000 ms
sta5->sta4: 1/1, rtt min/avg/max/mdev 2.499/2.499/2.499/0.000 ms
sta5->sta6: 1/1, rtt min/avg/max/mdev 4.753/4.753/4.753/0.000 ms
sta5->sta7: 1/1, rtt min/avg/max/mdev 4.743/4.743/4.743/0.000 ms
sta6->sta1: 1/1, rtt min/avg/max/mdev 0.918/0.918/0.918/0.000 ms
sta6->sta2: 1/1, rtt min/avg/max/mdev 2.935/2.935/2.935/0.000 ms
sta6->sta3: 1/1, rtt min/avg/max/mdev 2.583/2.583/2.583/0.000 ms
sta6->sta4: 1/1, rtt min/avg/max/mdev 2.996/2.996/2.996/0.000 ms
sta6->sta5: 1/1, rtt min/avg/max/mdev 2.203/2.203/2.203/0.000 ms
sta6->sta7: 1/1, rtt min/avg/max/mdev 4.936/4.936/4.936/0.000 ms
sta7->sta1: 1/1, rtt min/avg/max/mdev 2.346/2.346/2.346/0.000 ms
sta7->sta2: 1/1, rtt min/avg/max/mdev 2.534/2.534/2.534/0.000 ms
sta7->sta3: 1/1, rtt min/avg/max/mdev 2.687/2.687/2.687/0.000 ms
sta7->sta4: 1/1, rtt min/avg/max/mdev 2.310/2.310/2.310/0.000 ms
sta7->sta5: 1/1, rtt min/avg/max/mdev 2.290/2.290/2.290/0.000 ms
sta7->sta6: 1/1, rtt min/avg/max/mdev 2.450/2.450/2.450/0.000 ms
```

## Σφάλματα δικτύου ( Outdoor )

Σκοπός μας ήταν ομως να περιγράψουμε την περίπτωση ενός σφάλματος δικτύου. Για να το δείξουμε αυτό δημιουργήσαμε δύο οντολογίες. Η πρώτη είναι αυτή που περιγράφουμε στη παραπάνω παράγραφο. Παρακάτω βλέπουμε το σχήμα:



Η δεύτερη έχει όλα τα στοιχεία της πρώτης εκτός από την 5η κεραία (ap5) ή θεωρούμε ότι είναι κατεστραμμένη (ή απλά όχι ενεργή). Πάνω σε αυτό κάνουμε κάποιες παρατηρήσεις. Αρχικα η διεύθυνση IP της δεν καταλαμβάνεται από κάποια άλλη συσκευή λόγο της στατικότητας της. Θα μπορούσε ωστόσο ο controller σε περίπτωση ανίχνευσης καταστροφής να ελευθερώνει την διεύθυνση για εξυπηρέτηση περισσότερων συσκευών (πολύ σημαντικό για τη συγκεκριμένη εφαρμογή). Δεύτερον παρατηρούμε ότι εκτελούμε μια αλλαγή στο εύρος των γειτονικών κεραιών (συγκεκριμένα της 3). Αυτή η αλλαγή συμβαίνει για την εξυπηρέτηση όλων των αποκομμένων λόγω της βλάβης συσκευών. Αυτή η αλλαγή θα έχει επίδραση στην ταχύτητα του δικτύου (συγκεκριμένα θα κάνει τις συνδέσεις σε αυτή τη κεραία πιο αργές) αλλά θα διατηρήσει τη διαθεσιμότητα της εφαρμογής που στη συγκεκριμένη περίπτωση προέχει της ταχύτητας. Παρακάτω βλέπουμε το σχήμα:



Τέλος αξίζει να τρέξουμε τη προσομοίωση μας και για τη δεύτερη τοπολογία για να δούμε ότι είναι λειτουργική.

```

sta1 -> sta2sta3sta4sta5sta6sta7
sta2 -> sta1sta3sta4sta5sta6sta7
sta3 -> sta1sta2sta4sta5sta6sta7
sta4 -> sta1sta2sta3sta5sta6sta7
sta5 -> sta1sta2sta3sta4sta6sta7
sta6 -> sta1sta2sta3sta4sta5sta7
sta7 -> sta1sta2sta3sta4sta5sta6
*** Results:
sta1->sta2: 1/1, rtt min/avg/max/mdev 16.718/16.718/16.718/0.000
sta1->sta3: 1/1, rtt min/avg/max/mdev 13.047/13.047/13.047/0.000
sta1->sta4: 1/1, rtt min/avg/max/mdev 7.702/7.702/7.702/0.000 ms
sta1->sta5: 1/1, rtt min/avg/max/mdev 7.044/7.044/7.044/0.000 ms
sta1->sta6: 1/1, rtt min/avg/max/mdev 3.383/3.383/3.383/0.000 ms
sta1->sta7: 1/1, rtt min/avg/max/mdev 5.649/5.649/5.649/0.000 ms
sta2->sta1: 1/1, rtt min/avg/max/mdev 2.410/2.410/2.410/0.000 ms
sta2->sta3: 1/1, rtt min/avg/max/mdev 4.903/4.903/4.903/0.000 ms
sta2->sta4: 1/1, rtt min/avg/max/mdev 4.627/4.627/4.627/0.000 ms
sta2->sta5: 1/1, rtt min/avg/max/mdev 4.752/4.752/4.752/0.000 ms
sta2->sta6: 1/1, rtt min/avg/max/mdev 5.356/5.356/5.356/0.000 ms
sta2->sta7: 1/1, rtt min/avg/max/mdev 4.592/4.592/4.592/0.000 ms
sta3->sta1: 1/1, rtt min/avg/max/mdev 3.354/3.354/3.354/0.000 ms
sta3->sta2: 1/1, rtt min/avg/max/mdev 2.974/2.974/2.974/0.000 ms
sta3->sta4: 1/1, rtt min/avg/max/mdev 5.397/5.397/5.397/0.000 ms
sta3->sta5: 1/1, rtt min/avg/max/mdev 5.048/5.048/5.048/0.000 ms
sta3->sta6: 1/1, rtt min/avg/max/mdev 5.096/5.096/5.096/0.000 ms
sta3->sta7: 1/1, rtt min/avg/max/mdev 4.654/4.654/4.654/0.000 ms
sta4->sta1: 1/1, rtt min/avg/max/mdev 2.593/2.593/2.593/0.000 ms
sta4->sta2: 1/1, rtt min/avg/max/mdev 3.129/3.129/3.129/0.000 ms
sta4->sta3: 1/1, rtt min/avg/max/mdev 2.197/2.197/2.197/0.000 ms
sta4->sta5: 1/1, rtt min/avg/max/mdev 5.719/5.719/5.719/0.000 ms
sta4->sta6: 1/1, rtt min/avg/max/mdev 5.275/5.275/5.275/0.000 ms
sta4->sta7: 1/1, rtt min/avg/max/mdev 4.086/4.086/4.086/0.000 ms
sta5->sta1: 1/1, rtt min/avg/max/mdev 1.945/1.945/1.945/0.000 ms
sta5->sta2: 1/1, rtt min/avg/max/mdev 2.931/2.931/2.931/0.000 ms
sta5->sta3: 1/1, rtt min/avg/max/mdev 2.612/2.612/2.612/0.000 ms
sta5->sta4: 1/1, rtt min/avg/max/mdev 2.768/2.768/2.768/0.000 ms
sta5->sta6: 1/1, rtt min/avg/max/mdev 5.256/5.256/5.256/0.000 ms
sta5->sta7: 1/1, rtt min/avg/max/mdev 4.715/4.715/4.715/0.000 ms
sta6->sta1: 1/1, rtt min/avg/max/mdev 0.761/0.761/0.761/0.000 ms
sta6->sta2: 1/1, rtt min/avg/max/mdev 3.465/3.465/3.465/0.000 ms
sta6->sta3: 1/1, rtt min/avg/max/mdev 2.421/2.421/2.421/0.000 ms
sta6->sta4: 1/1, rtt min/avg/max/mdev 2.748/2.748/2.748/0.000 ms
sta6->sta5: 1/1, rtt min/avg/max/mdev 2.817/2.817/2.817/0.000 ms
sta6->sta7: 1/1, rtt min/avg/max/mdev 4.735/4.735/4.735/0.000 ms
sta7->sta1: 1/1, rtt min/avg/max/mdev 2.700/2.700/2.700/0.000 ms
sta7->sta2: 1/1, rtt min/avg/max/mdev 2.434/2.434/2.434/0.000 ms
sta7->sta3: 1/1, rtt min/avg/max/mdev 2.470/2.470/2.470/0.000 ms
sta7->sta4: 1/1, rtt min/avg/max/mdev 2.698/2.698/2.698/0.000 ms
sta7->sta5: 1/1, rtt min/avg/max/mdev 2.574/2.574/2.574/0.000 ms
sta7->sta6: 1/1, rtt min/avg/max/mdev 2.571/2.571/2.571/0.000 ms

```

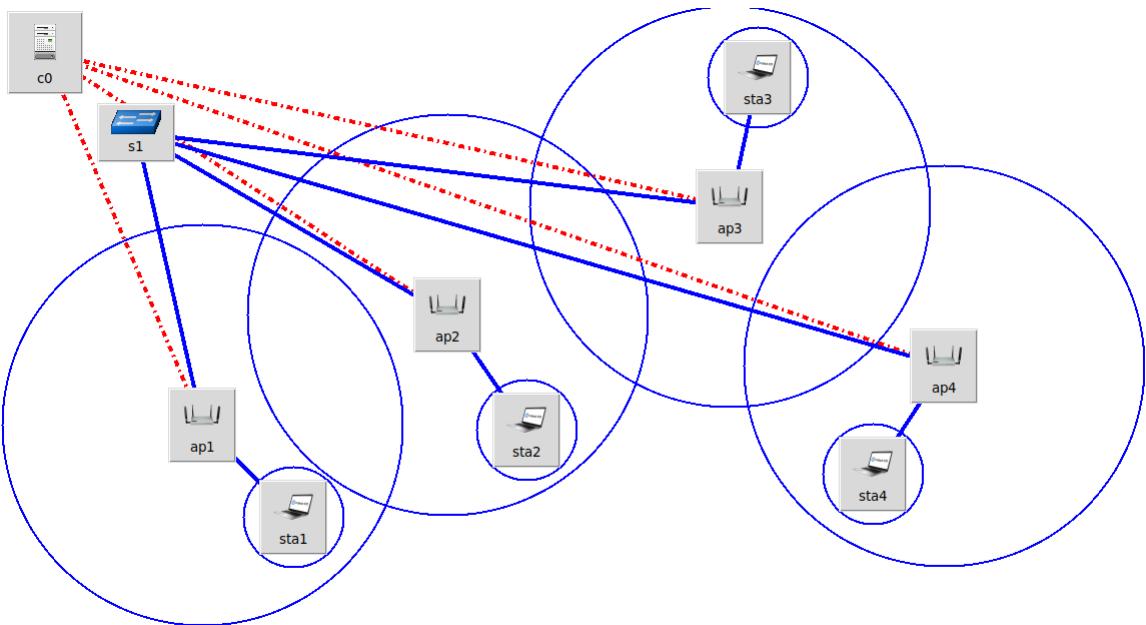
## Περιγραφή σχεδιασμού και προσομοίωσης ( Indoor )

Δουλεύοντας με τον ίδιο ακριβώς τρόπο σχεδιάσαμε και αναπαραστήσαμε το εσωτερικό δίκτυο στο mininet-wifi. Χρησιμοποιήσαμε το UI εργαλείο και ακολουθώντας την κάτοψη που μας δόθηκε δημιουργήσαμε το δίκτυο με την παραδοχή, για εμφανησιακούς λόγους, να θέσουμε την εμβέλεια των κεραιών στα 200m διότι δεν έχουμε την δυνατότητα zoom-in και zoom-out στο UI εργαλείο. Σε διαφορετική περίπτωση αν είχαμε ρυθμίσει την εμβέλεια των κεραιών 5G σε ρεαλιστική, της τάξης των 20m για εσωτερικά δίκτυα, δε θα ήταν οπτικά καλό στην εργασία καθώς θα πατούσε η μία συσκευή πάνω στην άλλη. Αντίστοιχα λοιπόν θέσαμε και 50m την εμβέλεια των συσκευών. Έπειτα με τον ίδιο τρόπο ανοίξαμε το δίκτυο και με την εντολή pingallfull ελέγχαμε τις συνδέσεις μεταξύ των τερματικων και είχαμε τις εξής επιτυχείς συνδέσεις:

```
mininet-wifi> pingallfull
*** Ping: testing ping reachability
sta1 -> sta2sta3sta4
sta2 -> sta1sta3sta4
sta3 -> sta1sta2sta4
sta4 -> sta1sta2sta3
*** Results:
sta1->sta2: 1/1, rtt min/avg/max/mdev 86.096/86.096/86.096/0.000 ms
sta1->sta3: 1/1, rtt min/avg/max/mdev 166.239/166.239/166.239/0.000 ms
sta1->sta4: 1/1, rtt min/avg/max/mdev 119.267/119.267/119.267/0.000 ms
sta2->sta1: 1/1, rtt min/avg/max/mdev 143.730/143.730/143.730/0.000 ms
sta2->sta3: 1/1, rtt min/avg/max/mdev 177.860/177.860/177.860/0.000 ms
sta2->sta4: 1/1, rtt min/avg/max/mdev 267.682/267.682/267.682/0.000 ms
sta3->sta1: 1/1, rtt min/avg/max/mdev 87.889/87.889/87.889/0.000 ms
sta3->sta2: 1/1, rtt min/avg/max/mdev 138.738/138.738/138.738/0.000 ms
sta3->sta4: 1/1, rtt min/avg/max/mdev 156.530/156.530/156.530/0.000 ms
sta4->sta1: 1/1, rtt min/avg/max/mdev 10.560/10.560/10.560/0.000 ms
sta4->sta2: 1/1, rtt min/avg/max/mdev 84.995/84.995/84.995/0.000 ms
sta4->sta3: 1/1, rtt min/avg/max/mdev 32.049/32.049/32.049/0.000 ms
mininet-wifi>
```

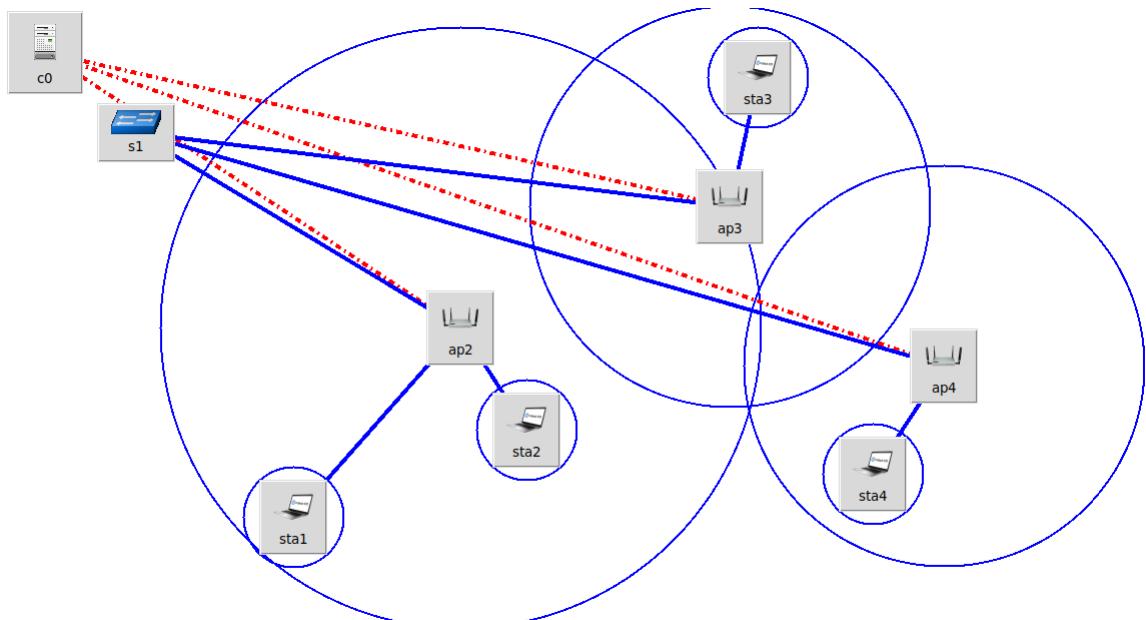
## Σφάλματα δικτύου ( Indoor )

Αντίστοιχα με την περίπτωση του εξωτερικού δικτύου αναπαραστήσαμε και την περίπτωση σφάλματος στο εσωτερικό δίκτυο καθώς και την λειτουργία για αντίδραση-αποφυγή προβλημάτων στην κάλυψη του δικτύου μας. Αρχικά έχουμε το σχήμα με την αρχική-χωρίς βλάβη κατάσταση του συστήματος μας:



indoor\_normal.mn

Στο παρακάτω σχήμα παρουσιάζουμε την περίπτωση βλάβης όπου λόγω βλάβης χάνουμε την 1η κεραία (ap1). Ομοίως με την Outdoor περίπτωση, η διεύθυνση IP της δεν καταλαμβάνεται από κάποια άλλη συσκευή λόγω στατικότητας της. Παρατηρούμε επιπλέον την επέκταση της εμβέλειας της κεραίας 2 (ap2) με σκοπό την κάλυψη της συσκευής sta1 από το δίκτυο. Όπως και στην Outdoor περίπτωση αυτή η αλλαγή θα έχει επίδραση στην ταχύτητα του δικτύου (συγκεκριμένα θα κάνει τις συνδέσεις σε αυτή τη κεραία πιο αργές) αλλά θα διατηρήσει τη διαθεσιμότητα της εφαρμογής που στη συγκεκριμένη περίπτωση προέχει της ταχύτητας.



indoor\_destroyed.mn

Στο τέλος τεστάρουμε με τον ίδιο τρόπο την λειτουργικότητα του δικτύου μας:

```
mininet-wifi> pingallfull
*** Ping: testing ping reachability
sta1 -> sta2sta3sta4
sta2 -> sta1sta3sta4
sta3 -> sta1sta2sta4
sta4 -> sta1sta2sta3
*** Results:
sta1->sta2: 1/1, rtt min/avg/max/mdev 49.465/49.465/49.465/0.000 ms
sta1->sta3: 1/1, rtt min/avg/max/mdev 192.642/192.642/192.642/0.000 ms
sta1->sta4: 1/1, rtt min/avg/max/mdev 156.226/156.226/156.226/0.000 ms
sta2->sta1: 1/1, rtt min/avg/max/mdev 11.702/11.702/11.702/0.000 ms
sta2->sta3: 1/1, rtt min/avg/max/mdev 261.484/261.484/261.484/0.000 ms
sta2->sta4: 1/1, rtt min/avg/max/mdev 179.760/179.760/179.760/0.000 ms
sta3->sta1: 1/1, rtt min/avg/max/mdev 122.597/122.597/122.597/0.000 ms
sta3->sta2: 1/1, rtt min/avg/max/mdev 95.901/95.901/95.901/0.000 ms
sta3->sta4: 1/1, rtt min/avg/max/mdev 264.987/264.987/264.987/0.000 ms
sta4->sta1: 1/1, rtt min/avg/max/mdev 81.645/81.645/81.645/0.000 ms
sta4->sta2: 1/1, rtt min/avg/max/mdev 155.209/155.209/155.209/0.000 ms
sta4->sta3: 1/1, rtt min/avg/max/mdev 133.603/133.603/133.603/0.000 ms
```

## Thingsboard

Επόμενο βασικό βήμα της εργασίας ήταν το στήσιμο του ασύρματου δικτύου αισθητήρων χρησιμοποιώντας την πλατφόρμα Thingsboard. Θα χωρίσουμε αυτή τη διαδικασία στη δημιουργία των συσκευών (Outdoor-Indoor), στη δημιουργία των Dashboard και στην δημιουργία ενός συστήματος για την προσομοίωση των τιμών των αισθητήρων.

### Δημιουργία Συσκευών (Outdoor)

Αρχικά δημιουργήσαμε τις συσκευές. Αξίζει να αναφέρουμε ότι αποφασίσαμε να θεωρήσουμε τους αισθητήρες μας ως αισθητήρες θερμοκρασίας. Αργότερα στα κριτήρια των ειδοποιήσεων αναφέρεται λεπτομερώς το γιατί. Στο menu devices δημιουργήσαμε όλες τις συσκευές με profile {thermostat}. Σε κάθε μια από αυτές προσθέσαμε δύο καινούργια server attributes: [latitude] και [longitude], βασισμένες στις πραγματικές γεωγραφικές θέσεις τους στη πανεπιστημιούπολη. Επιπλέον σε αυτό το σημείο ελέγχαμε την λειτουργικότητα της ασύρματης ενημέρωσης της τιμής της συσκευής μέσω δικτύου. Ένα ενδεικτικό request είναι το εξής:

```
curl -v -X POST -d "{\"temperature\": 25}"
localhost:8080/api/v1/vcujm35XUHs45JieG31X/telemetry --header
"Content-Type:application/json"
```

Στη συνέχεια αλλάξαμε το profile του {thermostat} για να ορίσουμε το σωστό alert προσθέσαμε ένα απλό alert το οποίο χτυπάει σε περίπτωση που η θερμοκρασία είναι πάνω από 80 βαθμούς. Η συγκεκριμένη επιλογή έγινε για την ευκολότερη προσομοίωση του συστήματος. Θα μπορούσαμε να είχαμε δημιουργήσει και ένα πιο πολύπλοκο σύστημα που εκτός από την θερμοκρασία θα έπαιρνε υπόψη και την υγρασία δηλαδή η ειδοποίηση θα ερχόταν με την αύξηση της θερμοκρασίας και την μείωση της υγρασίας. Παρακάτω φαίνεται ενδεικτικά ο κώδικας των alert της συσκευής:

Create alarm rules

Severity Major	Condition: <b>temperature</b> greater than <b>Current device</b> <b>temperatureAlarmT...</b> 
	Schedule: Active all the time 
	Details: Current temperature = \${temperature} 

Clear alarm rule

Condition: <b>temperature</b> less or equal <b>Current device</b> <b>temperatureAlarmT...</b> 
Schedule: Active all the time 
Details: Current temperature = \${temperature} 

## Δημιουργία Συσκευών (Indoor)

Στην περίπτωση του εσωτερικού δικτύου, χάριν συντομίας χρησιμοποιήσαμε τα ίδια {thermostat} που δημιουργήσαμε για τους εξωτερικούς αισθητήρες μας εφόσον έχουμε ίδιο αριθμό αισθητήρων και ως επικίνδυνη συνθήκη την θερμοκρασία να ξεπερνάει τους 80 βαθμούς κελσίου μέσα σε δωμάτιο με αισθητήρα. Στη συγκεκριμένη περίπτωση έχουμε το alert που μας ενημερώνει για τη φωτιά. Προσθέσαμε επιπλέον στα attributes τα [xPos] και [yPos] ώστε να μπορέσουμε να τα τοποθετήσουμε στην πραγματική τους θέση μέσα στα δωμάτια του κτιρίου. Η λειτουργικότητα των αισθητήρων είχε ήδη ελεγχθεί με την δημιουργία τους για τον εξωτερικό χώρο οπότε όπως θα φανεί παρακάτω, τα ελέγχαμε όλα μαζί σαν σύνολο στο σύστημα προσομοίωσης.

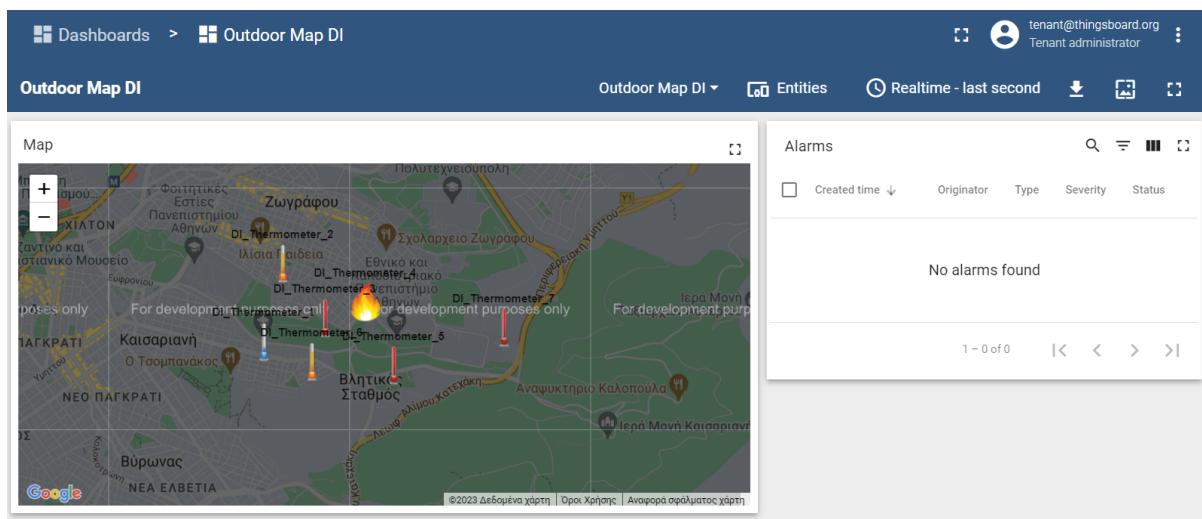
## Δημιουργία συστήματος προσομοίωσης

Για να ελέγχουμε την λειτουργικότητα των dashboard μας σχεδιάσαμε ένα απλό σύστημα προσομοίωσης (ένα bash script). Είναι ένα σύστημα το οποίο κάθε 5

δευτερόλεπτα στέλνει όσα requests όσα και οι συσκευές μας. Οι τιμές είναι τυχαίες σε ένα εύρος λογικών τιμών. Με αυτόν τον τρόπο έχουμε ένα διαδραστικό τρόπο παρουσίασης του dashboard. Ο κώδικας βρίσκεται στο αρχείο tester.sh

## Δημιουργία Dashboard (Outdoor)

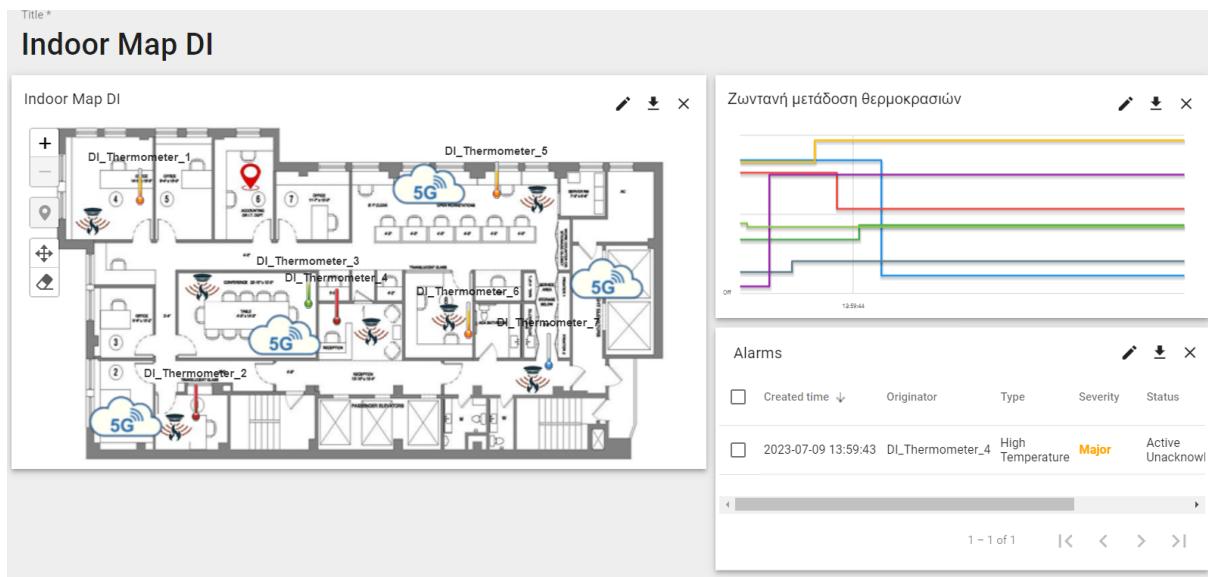
Το dashboard για το outdoor σενάριο αποτελείται από δύο widgets. Το πρώτο widget παρουσιάζει τα alerts όταν αυτά εμφανιστούν. Το δεύτερο widget είναι ένας χάρτης ο οποίος παρουσιάζει τους αισθητήρες στην τοποθεσία που ανήκουν βασισμένος στο [latitude] και το [longitude] τους. Επιπλέον τα εικονίδια που φαίνονται είναι δυναμικά και τα προγραμματίσαμε ειδικά για να αναπαριστούν ρεαλιστικά τις συνθήκες του κλίματός μας. Τέλος όταν επιλέγουμε μια από αυτές μπορούμε να δούμε σε πραγματικό χρόνο την θερμοκρασία της. Παρακάτω παρατηρούμε την τελική μορφή του dashboard:



## Δημιουργία Dashboard (Indoor)

Το dashboard για το Indoor σενάριο πυρκαγιάς επιλέξαμε να το εμπλουτίσουμε με ακόμα ένα widget διότι με αυτό, σε περίπτωση πυρκαγιάς σε εσωτερικό χώρο, θα είναι πολύ πιο ξεκάθαρη η σοβαρότητα της κατάστασης και η πρόβλεψη της εξάπλωσης της πυρκαγιάς αφού θα επηρεαστεί η θερμοκρασία σε όλο τον εσωτερικό χώρο. Ως αποτέλεσμα αυτού, σε ένα ρεαλιστικό σενάριο θα επηρεαστούν ανάλογα και διπλανοί αισθητήρες άμεσα, πράγμα που δεν θα γίνει στο εξωτερικό δίκτυο λόγω των μεγάλων αποστάσεων μεταξύ τους. Οι αισθητήρες, όπως αναφέραμε και προηγουμένως, έχουν τοποθετηθεί στους χώρους που φαίνεται στην κάτωφη ότι υπάρχουν αισθητήρες με τα attributes [xPos] και [yPos]. Στη συνέχεια τρέξαμε το bash script μας και βλέπουμε ένα παράδειγμα λειτουργίας του συστήματος μας. Την χρονική στιγμή 13:59:43 έφτασε στον αισθητήρα (DI\_Thermometer\_4) θερμοκρασία >80 βαθμούς και βλέπουμε στον πίνακα alarms ότι έχουμε ενημέρωση για πρόβλημα

από αυτόν. Στον πίνακα με τη ζωντανή μετάδοση θερμοκρασιών δεν φαίνεται η ρεαλιστική λογική που περιγράψαμε, γιατί όπως αναφέραμε το bash script φτιάχτηκε για σκοπούς δοκιμής με αποτέλεσμα να φτάνουν τυχαίες τιμές σε κάθε αισθητήρα και όχι αναλογικές με τον γειτονικό του. Παρακάτω φαίνεται το παράδειγμα που περιγράψαμε:



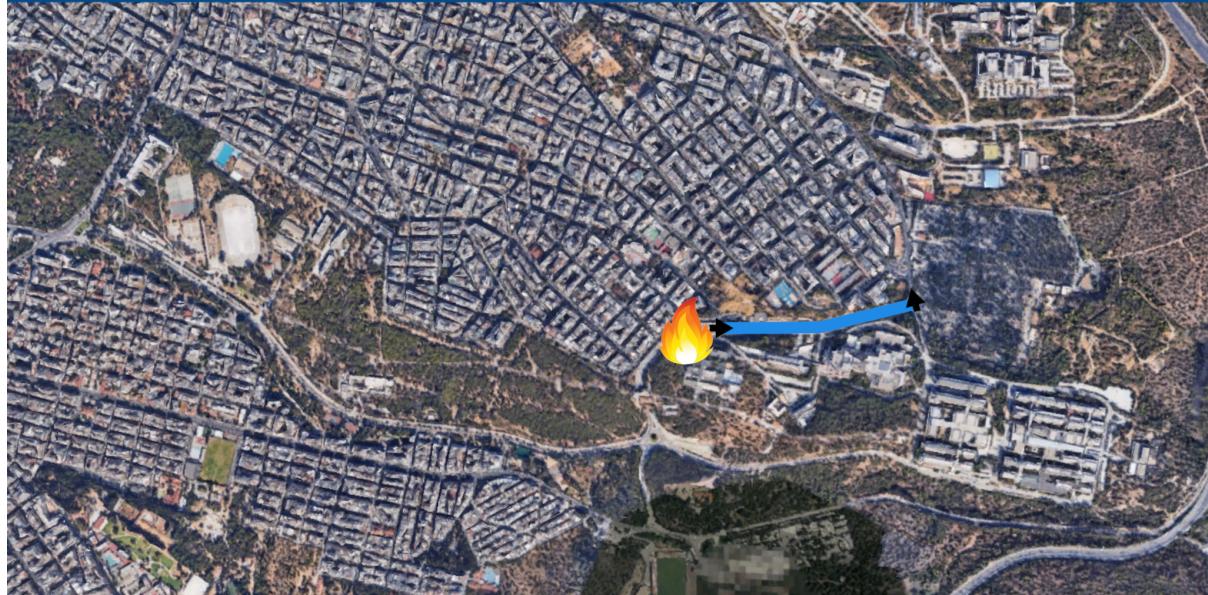
## Mockups

### Mockup Outdoor Σεναρίου

Η εφαρμογή έχει σκοπό να καθοδηγήσει το χρήστη στην κοντινότερη έξοδο σε περίπτωση κινδύνου. Στο επόμενο σενάριο παρατηρούμε την καταστροφή μιας κεραίας (κοντά στη Θεολογική σχολή) και την καθοδήγηση του χρήστη στη Βόρεια έξοδο της πανεπιστημιούπολης ορίζοντας παρουσιάζοντας το μονοπάτι που πρέπει να ακολουθήσει ο χρήστης, την κατεύθυνση που χρειάζεται να προχωρήσει καθώς και τον προσανατολισμό στον οποίο βρίσκεται.



## Κατευθυνθείτε Ανατολικά



### Mockup Indoor Σεναρίου

Η εφαρμογή όπως και στην περίπτωση εξωτερικής βλάβης έχει σκοπό να καθοδηγήσει το χρήστη στην κοντινότερη έξοδο σε περίπτωση κινδύνου. Στο επόμενο σενάριο παρατηρούμε την καταστροφή μιας κεραίας μέσα στο κτίριο και την καθοδήγηση του χρήστη στις σκάλες του κτιρίου παρουσιάζοντας το μονοπάτι που πρέπει να ακολουθήσει ο χρήστης, την κατεύθυνση που χρειάζεται να προχωρήσει καθώς και τον προσανατολισμό στον οποίο βρίσκεται.



## Κατευθεινθείτε όπως υποδυκνείται

