Διαχείριση Δικτύων

Network Management in Mission Critical Cases

Ελένη Φεσλιάν 1115202000204

Ιωάννης Ζάννες Βιδάλης 1115202000022

README

Περιγραφή του κώδικα Mininet-Wifi

Οι κώδικες της εργασίας μας περιλαμβάνουν ένα σενάριο στη γλώσσα προγραμματισμού Python, που δημιουργεί μια τοπολογία δικτύου χρησιμοποιώντας το Mininet-Wifi. Το δίκτυο αποτελείται από πολλαπλά σημεία πρόσβασης και κεντρικούς σταθμούς συνδεδεμένους σε ένα ασύρματο περιβάλλον.

Σενάριο Outdoor – Πριν τη φωτιά

```
ap1 = net.addAccessPoint('ap1', cls=OVSKernelAP,
ssid='ap1-ssid', channel='1', mode='g',
position='-130,60,0', range=30)
```

Στο συγκεκριμένο σενάριο δημιουργήσαμε 7 Access Points (APs) με τη χρήση της συνάρτησης **addAccessPoint()** όπως φαίνεται παραπάνω. Το πρώτο όρισμα της είναι το όνομα του AP, το δεύτερο όρισμα (cls=OVSKernelAP) υποδεικνύει την κατηγορία ή τον τύπο του σημείου πρόσβασης που χρησιμοποιείται, συγκεκριμένα ένα OVSKernelAP. Στη συνέχεια, με τον όρο SSID, δηλαδή Service Set Identifier, ορίζουμε το σημείο πρόσβασης που θα συνδεθούν οι χρήστες σε «ap1-ssid», με το τέταρτο όρισμα να είναι το κανάλι που καθορίζει τη ζώνη συχνοτήτων στην οποία λειτουργεί το ασύρματο δίκτυο, και το πέμπτο όρισμα να δείχνει το ασύρματο πρότυπο που χρησιμοποιείται, στη συγκεκριμένη περίπτωση το «g». Τέλος, ορίζουμε τη θέση του κάθε Access Point στο χάρτη δίνοντας συντεταγμένες σε δύο διαστάσεις και κάθε AP να έχει εμβέλεια 30 μέτρα (range=30).

```
h1 = net.addHost('h1', cls=Host, ip='10.0.0.1', defaultRoute=None)
sta1 = net.addStation('sta1', ip='10.0.0.1', position='-130,70,0', range=15)
```

Σε κάθε Access Point συνδέουμε ένα base station και σε κάθε base station συνδέουμε έναν host. Όπως φαίνεται παραπάνω, το πρώτο όρισμα της συνάρτησης **addHost()** είναι το όνομα που δίνουμε στο host, το δεύτερο δηλώνει την κατηγορία ή τον τύπο της συσκευής που διαμορφώνεται, η οποία είναι ένας κεντρικός υπολογιστής. Το τρίτο όρισμα είναι η διεύθυνση IP και το τελευταίο καθορίζει την προεπιλεγμένη πύλη ή δρομολογητή για τον κεντρικό υπολογιστή. Σε αυτή την περίπτωση καθορίζεται η τιμή 'None'.

Με τη συνάρτηση **addStation()** δημιουργούμε τους απαραίτητους σταθμούς βάσης, και όπως παραπάνω το πρώτο όρισμα καθορίζει το όνομα του base station, το δεύτερο την IP, το τρίτο τις συντεταγμένες του σε δύο διαστάσεις και το τελευταίο το εύρος κάλυψης σε 15 μέτρα (range=15).

Πιο συγκεκριμένα, σε όλα τα base station έχουμε συνδέσει μόνο ένα host εξαιρουμένου του “station 6” που έχουμε συνδέσει 10 hosts ώστε να προσομοιώσουμε καλύτερα το σενάριο που μας δίνεται.

```
net.addLink(sta1, ap1)
net.addLink(h1, sta1)
```

Για να συνδέσουμε AP με base station και το host με τα stations χρησιμοποιούμε τη συνάρτηση **addLink(),** όπως φαίνεται δίπλα.

Τέλος, για να ενεργοποιηθεί το δίκτυο χρησιμοποιούμε τις εξής εντολές:

```
#plot the network graph
net.plotGraph(max_x=250, max_y=250)

info('*** Starting network\n')
net.build()
info('*** Starting controllers\n')
for controller in net.controllers:
    controller.start()

info('*** Starting switches/APs\n')
net.get('ap1').start([])
net.get('ap2').start([])
net.get('ap3').start([])
net.get('ap4').start([])
net.get('ap5').start([])
net.get('ap6').start([])
net.get('ap7').start([])

info('*** Post configure nodes\n')

CLI(net)
```

Σενάριο Outdoor – Μετά τη φωτιά

```
ap2 = net.addAccessPoint('ap2', cls=OVSKernelAP,
ssid='ap2-ssid', channel='1', mode='g',
position='-35,-35,0', range=65)
# IS DESTROYED #ap6 = net.addAccessPoint('ap6', cls=OVSKernelAP,
# ssid='ap6-ssid', channel='1', mode='g',
# position='-60,-75,0', range=30)
ap7 = net.addAccessPoint('ap7', cls=OVSKernelAP,
ssid='ap7-ssid', channel='1', mode='g',
position='-131,-45,0', range=65)
```

Σε αυτό το σενάριο καταστρέφουμε το “AP 6” με αποτέλεσμα τα “AP 2” και “AP 7” να αυξάνουν την εμβέλειά τους στα 65 μέτρα, ώστε να καλύψουν τους χρήστες του AP που καταστράφηκε.

Οι χρήστες που ήταν συνδεδεμένοι στο “AP 6” μοιράζονται στα άλλα δύο AP, πιο συγκεκριμένα οι τέσσερις χρήστες πάνε στο “AP 2” και οι υπόλοιποι έξι μεταφέρονται στο “AP 7”.

Σενάριο Indoor – Πριν την φωτιά

Ακολουθώντας το προηγούμενο μοτίβο, δημιουργήσαμε 4 AP με εμβέλεια 8,75 μέτρα, σε κάθε Access Point συνδέουμε ένα base station, το οποίο έχει εμβέλεια 5 μέτρα, και για το δικό μας σενάριο βάλαμε έναν μόνο χρήστη να βρίσκεται στον εσωτερικό χώρο ο οποίος είναι συνδεδεμένος στο “AP 3”.

Στη συνέχεια ακολουθούμε την ίδια διαδικασία όπως στο outdoor σενάριο πριν τη φωτιά.

Σενάριο Indoor – Μετά την φωτιά

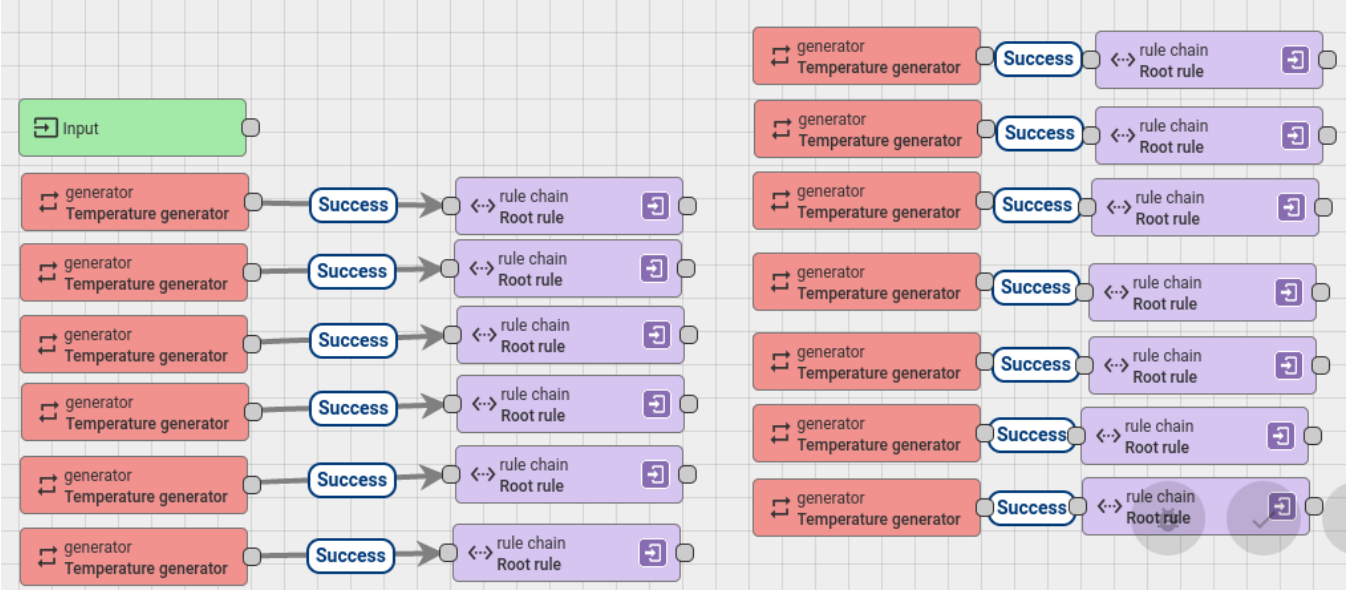
Σε αυτό το σενάριο καταστρέφουμε το “AP 1” με αποτέλεσμα το “AP 2” να αυξάνει την εμβέλειά του στα 12 μέτρα, ώστε να καλύψει τον χρήστη του AP που καταστράφηκε. Ο χρήστης δεν χρειάζεται να συνδεθεί σε κάποιο άλλο AP με βάση τη φωτογραφία της εκφώνησης που δίνεται.

Περιγραφή του κώδικα της πλατφόρμας Thingsboard

Και στα δύο σενάρια (indoor & outdoor) οι συσκευές έχουν τέσσερα πεδία: θερμοκρασία, υγρασία, πλάτος και μήκος. Πιο συγκεκριμένα, έχουμε ορίσει η θερμοκρασία κυμαίνεται στο διάστημα (25,30) βαθμούς Κελσίου, ενώ η υγρασία παίρνει τιμές στο διάστημα (45,51). Τα μεγέθη αυτά αλλάζουν αυτόματα τιμές ανά 10 δευτερόλεπτα.

Για να το υλοποιήσουμε αυτό, δημιουργήσαμε δύο Rule Chains, ένα για το κάθε ένα:

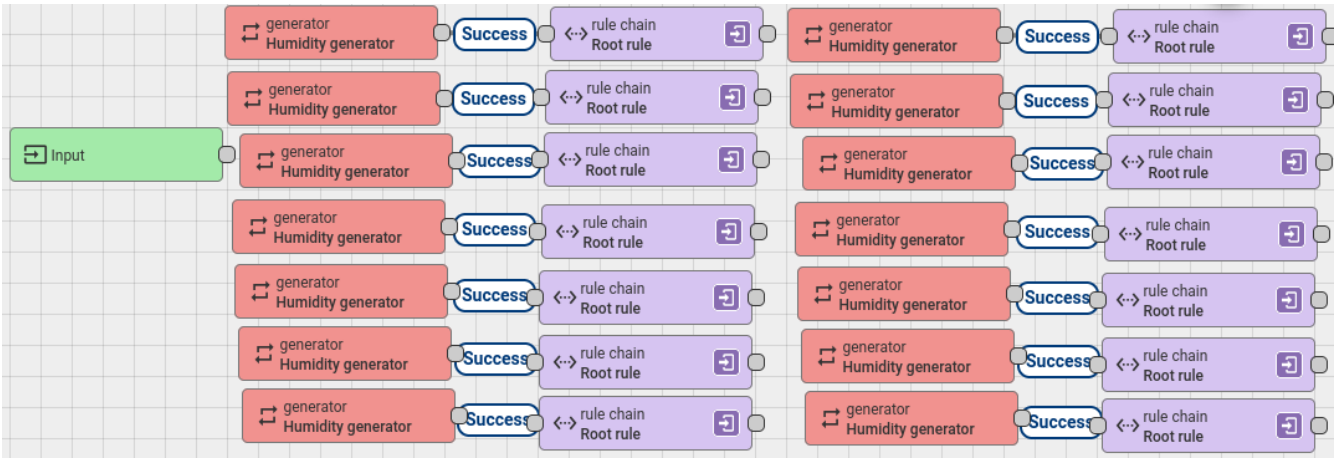
- **Temperature emulation**



Κάθε κόκκινο block αναπαριστά ένα generator με 7 συσκευές για το indoor και 7 για το outdoor σενάριο. Κάθε μία παράγει την επιθυμητή θερμοκρασία, η οποία παίρνει τυχαίες τιμές με βάση τον παρακάτω κώδικα:

```
var temperature = Math.floor((Math.random()*50+250)*10)/100;
var msg = { temperature: temperature };
var metadata = { data: 40 };
var msgType = "POST_TELEMETRY_REQUEST";

return { msg: msg, metadata: metadata, msgType: msgType };
```

- **Humidity emulation**



Κάθε κόκκινο block αναπαριστά ένα generator με 7 συσκευές για το indoor και 7 για το outdoor σενάριο. Κάθε μία παράγει την επιθυμητή υγρασία, η οποία παίρνει τυχαίες τιμές με βάση τον παρακάτω κώδικα:

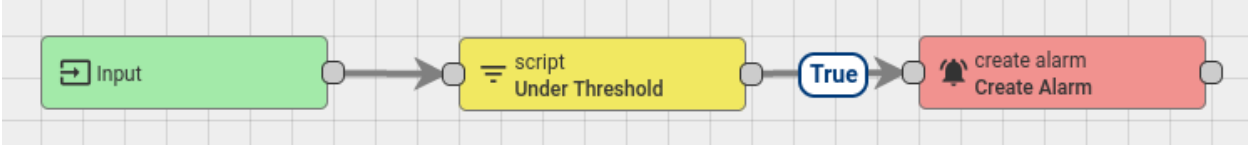
```
var humidity = Math.floor((Math.random()*60+450)*10)/100;
var msg = { humidity: humidity };
var metadata = { data: 40 };
var msgType = "POST_TELEMETRY_REQUEST";

return { msg: msg, metadata: metadata, msgType: msgType };
```

Δημιουργία Alarm

Θεωρήσαμε στο outdoor σενάριο ότι το alarm πρέπει να χτυπάει μόνο όταν η θερμοκρασία είναι άνω των 50 βαθμών Κελσίου, ενώ στο indoor ο συναγερμός να χτυπάει όχι μόνο για το προηγούμενο κριτήριο που ορίσαμε, αλλά και όταν η τιμή της υγρασίας υπερβαίνει το 75%.

Πήραμε αυτή την απόφαση, διότι στο εξωτερικό σενάριο η λειτουργία του δικτύου δεν θα επηρεαστεί από μια υψηλή τιμή υγρασίας, παρά μόνο σε περίπτωση που αυξηθεί σε ακραίο βαθμό η θερμοκρασία. Ενώ στο εσωτερικό σενάριο μια υψηλή τιμή υγρασίας μπορεί να προβεί μοιραία για τις συσκευές.

Δημιουργήσαμε το παρακάτω Rule Chain:



Το κίτρινο block αναπαριστά ένα script που επιβάλλει τον περιορισμό για το πότε θα χτυπήσει ο συναγερμός σύμφωνα με την συγκεκριμένη εντολή:

```
1  return msg.temperature > 50;
```

Η παραπάνω εντολή χρησιμοποιείται και για το indoor αλλά και για το outdoor σενάριο.

Επιπλέον, για το indoor σενάριο χρησιμοποιούμε και την εξής εντολή:

```
1  return msg.temperature > 50 || msg.humidity>75;
```