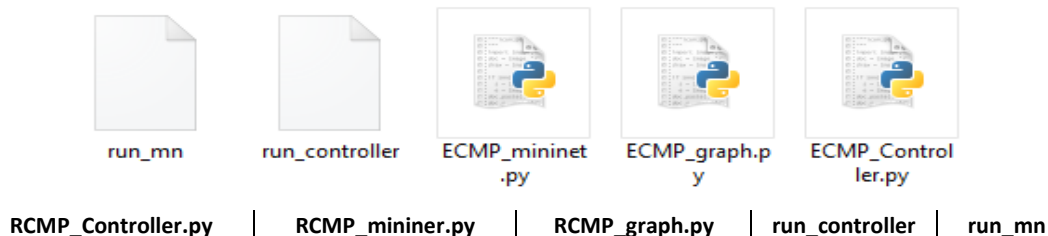


SDN - הוראות הרצה לפרויקט

שלב הכנת הפרויקט

הקבצים שצריך בשביל להריץ את הפרויקט שלי (המצורפים בקובץ zip. המטלה) :



אז ככה:

לאחר התקנת ה mininet & pox באופן הבא:

תכנס לתיקיית home שלך, ותפתח שמה את ה terminal ותריץ את הפקודה הבאה:

```
git clone git://github.com/mininet/mininet
```

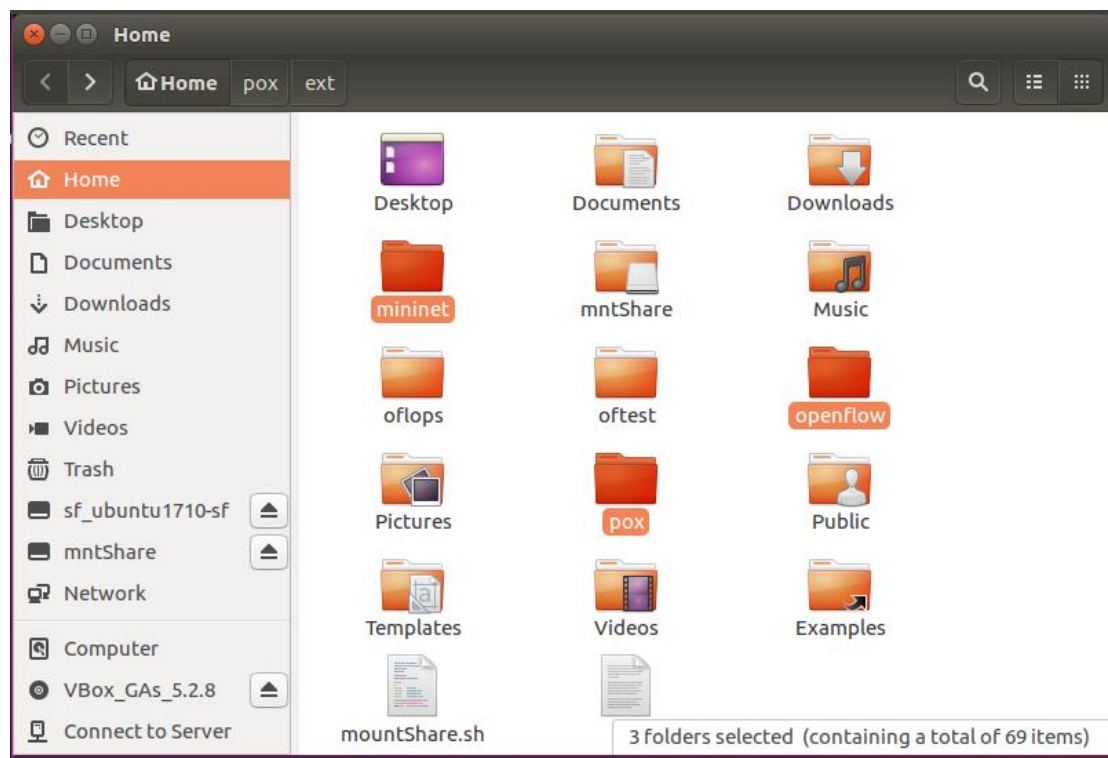
זה הוריד לך את תיקיית mininet בתיקיית ה home, לאחר ההתקנה בטרמינל הנוכחי, תקליד:

```
mininet/util/install.sh [options]
```

וכן את הפקודה: `sudo apt-get install mininet`

על מנת שיתקין את ה Path עבור mininet.

וזו יתקין את כל מה שצריך, כולל את תיקיית האסס (ותיקיית) בתיקיית home, זה יראה דבר כזה



לפרטים נוספים בהתקנות ה mininet והאסס יש להיכנס לאתר : <http://mininet.org/download/>

התקנות מוקדמות נוספות:

- יש להתקין את matplotlib באופן הבא:

```
$ sudo apt install python-numpy  
$ sudo apt install python-matplotlib
```

- יש להתקין את networkx באופן הבא:

```
$ pip install networkx
```

- יש להתקין את simplejson באופן הבא:

```
$ pip install simplejson
```

- במידה ואין לך את קוק יש להתקין אותו באופן הבא:

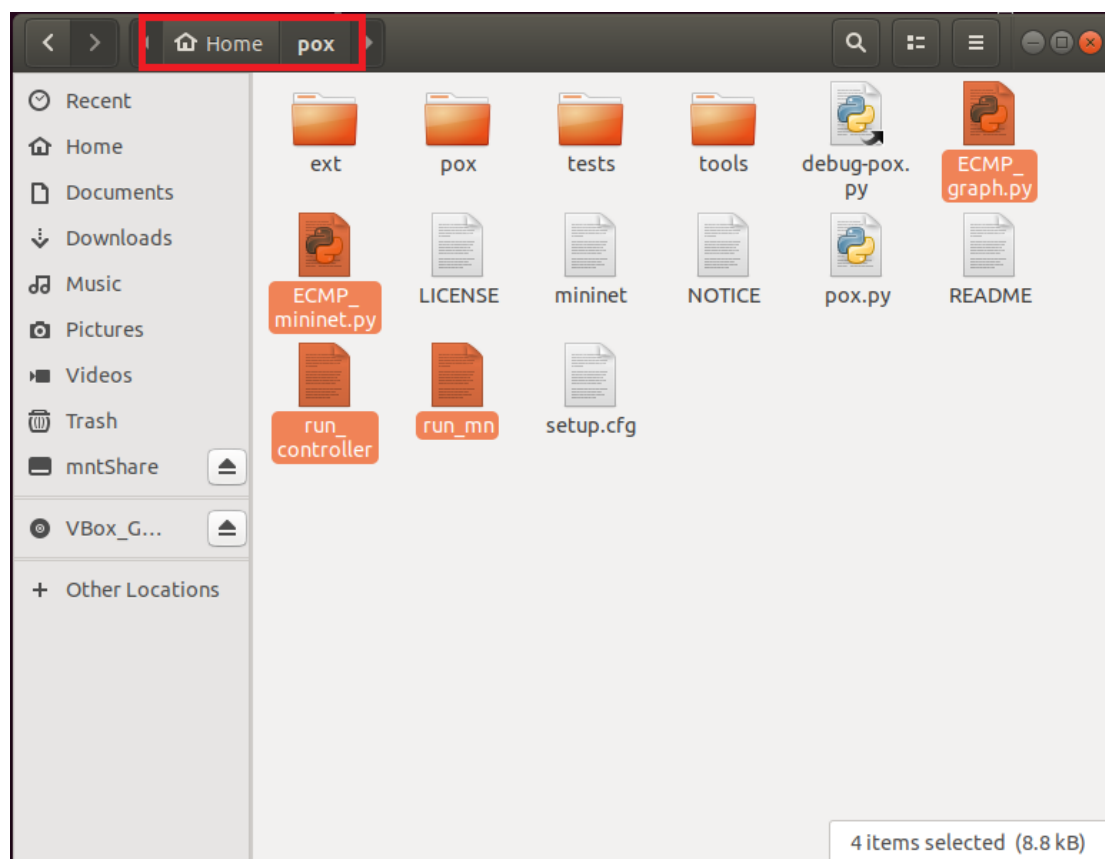
```
$ sudo apt-get install python-pip
```

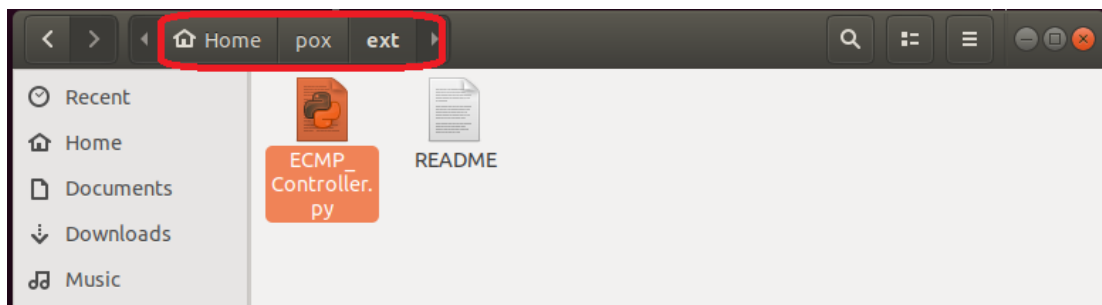
שלב הרצת הפרויקט

יש למקם את הקבצים הבאים בתיקיות הבאות:

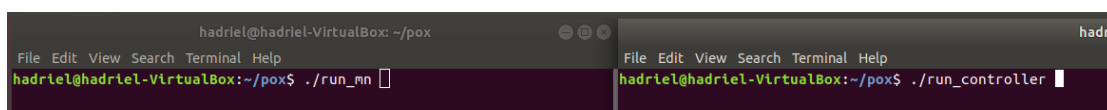
```
+ home  
| + pox  
| | run_mn  
| | run_controller  
| | mininetLoadBalancer.py  
| | + ext  
| | | SimpleLoadBalancerController.py
```

הנה תמונות מסך כדי לא להתבלבל:

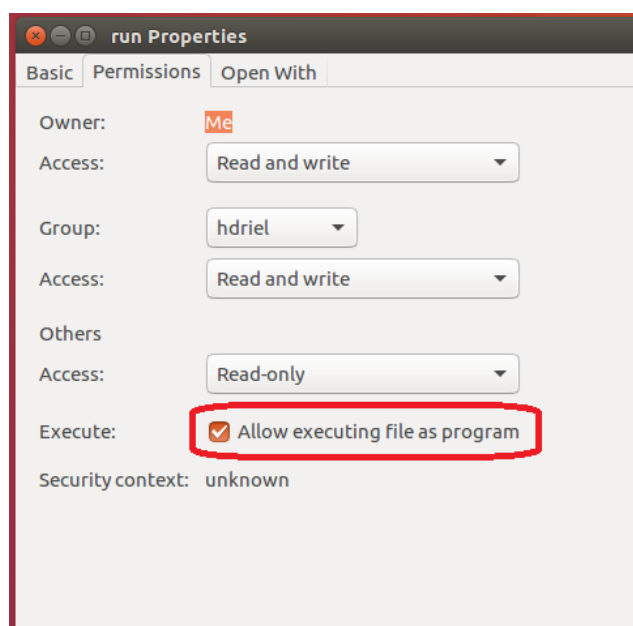




לאחר מכן יש לפתוח 2 חלונות טרמינלים נפרדים בתיקייה של pox ולהקליד:



כדי לוודא שזה עובד, תוודא בproperties>Permission>Allow executing file as program במצב check כך: (לבדוק את זה בקבצים: run_controller, run_mn)



ניתן להיכנס לזה בלחיצה ימנית עם העכבר על הקובץ, ואז בתחתית האפשרויות יש properties לאחר שתריצו את הפקודה ./run. ייפתחו 2 חלונות טרמינל לפרויקט, וזה יראה דבר כזה:

```

hadriel@hadriel-VirtualBox: ~/pox
File Edit View Search Terminal Help
hadriel@hadriel-VirtualBox:~/pox$ ./run_mn
*** Init mininet Network
save random graph image to file: graph.png
save random graph json to file: graph.txt
*** Adding hosts and switches...
*** Creating links...
*** Creating network
*** Adding controller
Unable to contact the remote controller at 127.0.0.1:6633
*** Adding hosts:
h1 h2 h3 h4 h5
*** Adding switches:
s1 s2 s3 s4 s5
*** Adding links:
(h1, s3) (h2, s4) (s1, h5) (s1, s2) (s2, s3) (s2, s4) (s2, s5) (s3, h3) (s4, h4) (s4, s5)
*** Configuring hosts
h1 h2 h3 h4 h5
*** Starting network
*** Starting controller
c0
*** Starting 5 switches
s1 s2 s3 s4 s5 ...
*** Running CLI
*** Starting CLI:
mininet> h1 ping h2 -c 1
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data:
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=43.6 ms

--- 10.0.0.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/ndev = 43.625/43.625/43.625/0.000 ms
mininet> h1 ping h4 -c 1
PING 10.0.0.4 (10.0.0.4) 56(84) bytes of data:
64 bytes from 10.0.0.4: icmp_seq=1 ttl=64 time=32.5 ms

--- 10.0.0.4 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/ndev = 32.570/32.570/32.570/0.000 ms
mininet> h1 ping h4 -c 1
PING 10.0.0.4 (10.0.0.4) 56(84) bytes of data:
64 bytes from 10.0.0.4: icmp_seq=1 ttl=64 time=14.8 ms

--- 10.0.0.4 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/ndev = 14.889/14.889/14.889/0.000 ms
mininet>

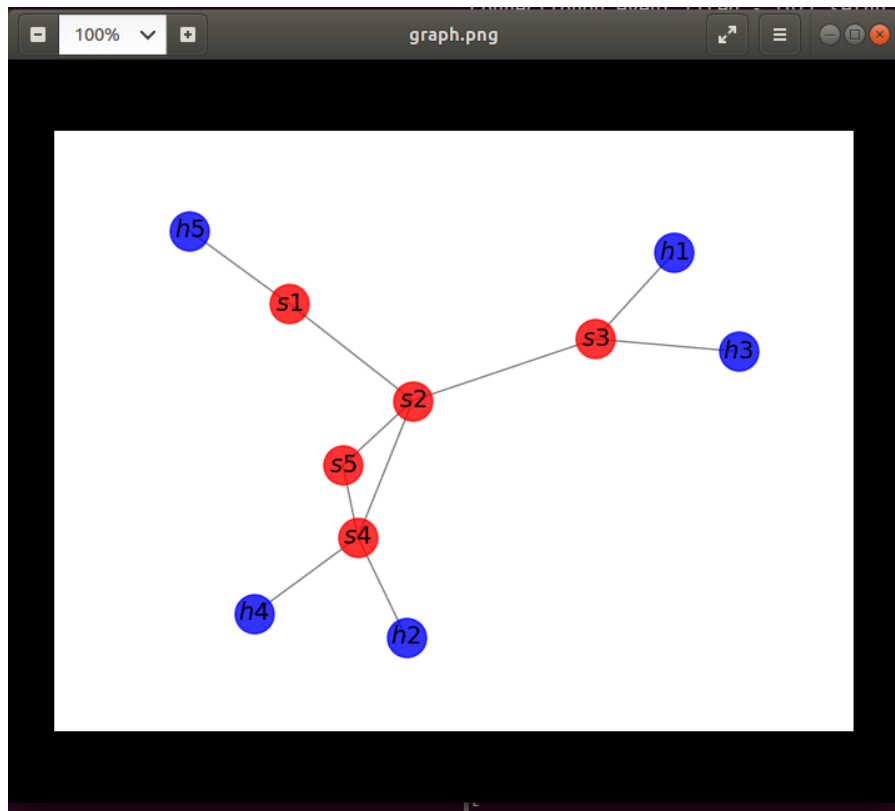
```

```

hadriel@hadriel-VirtualBox: ~/pox
File Edit View Search Terminal Help
hadriel@hadriel-VirtualBox:~/pox$ ./run_controller
POX 0.5.0 (eel) / Copyright 2011-2014 James McCauley, et al.
Custom Controller running.
ConnectionUp event fired - init setup switch s 2 !
ConnectionUp event fired - init setup switch s 4 !
ConnectionUp event fired - init setup switch s 1 !
ConnectionUp event fired - init setup switch s 3 !
ConnectionUp event fired - init setup switch s 5 !
ARP: start request connection...
From host : ip = 10.0.0.1 , mac = 00:00:00:00:00:01
To host : ip = 10.0.0.2 , mac = 00:00:00:00:00:02
-----
Random path :
[ h2 -> s4 -> s2 -> s5 -> s4 -> h2 ]
-----
ARP: start request connection...
From host : ip = 10.0.0.2 , mac = 00:00:00:00:00:02
To host : ip = 10.0.0.1 , mac = 00:00:00:00:00:01
-----
Random path :
[ h1 -> s3 -> s5 -> s2 -> s3 -> h1 ]
-----
ARP: start request connection...
From host : ip = 10.0.0.1 , mac = 00:00:00:00:00:01
To host : ip = 10.0.0.4 , mac = 00:00:00:00:00:04
-----
Random path :
[ h2 -> s4 -> s5 -> s2 -> s4 -> h4 ]
-----
ARP: start request connection...
From host : ip = 10.0.0.4 , mac = 00:00:00:00:00:04
To host : ip = 10.0.0.1 , mac = 00:00:00:00:00:01
-----
Random path :
[ h4 -> s4 -> s5 -> s2 -> s3 -> h1 ]
-----
[ h1 -> s3 -> s2 -> s4 -> h4 ]
-----
[ h4 -> s4 -> s2 -> s3 -> h1 ]
-----

```

ניתן לראות עבור הרצת הפרויקט שלנו שיש לנו שנוצר לנו קובץ בתיקיית `sx` בשם `graph.png` שהוא ציור של הגרף שלנו :



ואז כשאנחנו נכתוב את הפקודות של למשל:

\$ h1 ping h2 -c 1

\$ h2 ping h3 -c 1

נקבל במסך של הcontroller את הכיוון שהפקטה זרמה דרכו (דרך איזה מסלול) ובכל פעם זה מסלול נדומלי אחר לפי דרישות המטלה.

הגרפים נצרים בצורה רנדומלית, הנה כמה דוגמאות של גרפים רנדומליים שנוצרים:

(בכל הרצת `./run_mn` יש קריאה לקובץ `ECMP_graph.py`, שיוצר לנו את הקבצים `graph.png` המתאר את הציור של הגרף שלנו בהרצה, ואת הקובץ `graph.txt` שבו נשמר ה `json` של הנתונים לגרף שלנו כדי לטעות אותו מה `controller` ומה `mininet`)

