



ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

Ανάπτυξη Λογισμικού για Δίκτυα και Τηλεπικοινωνίες

ΕΡΓΑΣΙΑ Νο 1

Σύστημα Παρακολούθησης σε PC/Laptop

Κουτσάκης Ηλίας
Σιαμούρης Αναστάσιος

Κεφάλαιο 1

α) Κλάσεις του προγράμματος και η χρησιμότητά τους

Ακολουθήθηκε κατά το δυνατόν το αντικειμενοστραφές μοντέλο της Java και χρησιμοποιήσαμε όσες κλάσεις θεωρήσαμε ότι συντελούν σε ξεχωριστές λειτουργίες από τις υπόλοιπες.

MainThread.java

Η κλάση αυτή είναι η βασική κλάση του προγράμματος και το βασικό νήμα που ξεκινάει να τρέχει. Κάνει, σύμφωνα με τις καταστάσεις Markov, τον έλεγχο για δημιουργία καινούριων thread ανάλογα με τα interfaces που βρίσκει και κλείσιμο των interfaces (μαζί με το ανάλογο interrupt των threads). Επίσης διαβάζει το property file και δημιουργεί το shutdown hook.

ShutdownClass.java

Εκτελεί το shutdown hook και δίνει σήμα με μεταβλητή να κλείσουν όλα τα threads.

Κλάσεις των Interfaces – Access Points

Interface.java

Είναι η κλάση που αντιστοιχεί στα interfaces και έχει όλες τις setter – getter μεθόδους που χρειάζονται. Λειτουργεί και αυτή σύμφωνα με Markov και κάνει έλεγχο για τις όποιες αλλαγές καλώντας μεθόδους του parser ifconfig από την αντίστοιχη κλάση που περιγράφεται αργότερα. Επίσης δημιουργεί έναν κλώνο του εαυτού της στον οποίο δίνει τα παλιά της στοιχεία IPIN τον καινούριο έλεγχο και ανάλογα με το αποτέλεσμα λειτουργεί και η sleep().

InterfaceWireless.java

Αντίστοιχη της παραπάνω αλλά για τα wireless και μάλιστα παίρνει αρκετές από τις μεθόδους και τα attributes μέσω κληρονομικότητας (κάνει extend την Interface). Διαφοροποιείται στο ότι ελέγχει και για Access Points όταν τις δίνουμε τις κατάλληλες συνθήκες.

AccessPoint.java

Η κλάση που αποθηκεύουμε τις πληροφορίες των Access Points, έχει μόνο setters – getters.

Κλάσεις διαχείρισης δεδομένων

DataLists.java

Η κλάση περιέχει τις λίστες των Access Points, Interfaces και Wireless Interfaces και μεθόδους για την διαχείρισή τους. Την στέλνουμε σαν όρισμα όπου χρειάζεται να διαχειριστούμε είτε τα αντικείμενα (σαν λίστα) είτε τις λίστες.

DataPrinter.java

Ο “αθροιστής”. Τυπώνει τα ανάλογα μηνύματα από τα ορίσματα που στέλνουμε στις συναρτήσεις του (έχουμε την ίδια συνάρτηση με overload).

Κλάσεις των parsers

ParserIfconfig.java

Έχει όλες τις μεθόδους που χρειάζεται να εκτελέσουν process της εντολής ifconfig.

ParserIwconfig.java

Έχει όλες τις μεθόδους που χρειάζεται να εκτελέσουν process της εντολής iwconfig.

ParserIwgetid.java

Έχει όλες τις μεθόδους που χρειάζεται να εκτελέσουν process της εντολής iwgetid.

ParserIwlist.java

Έχει όλες τις μεθόδους που χρειάζεται να εκτελέσουν process της εντολής iwlist.

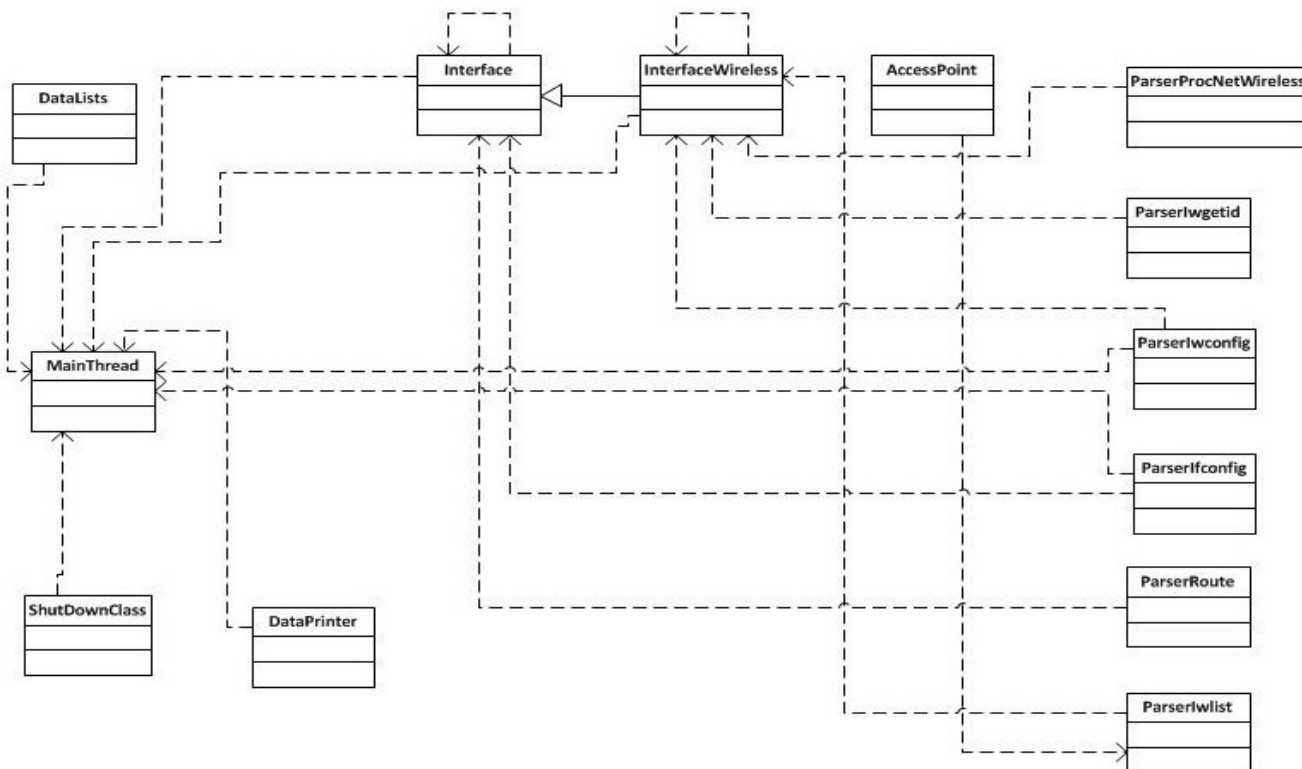
ParserProcNetWireless.java

Έχει όλες τις μεθόδους που χρειάζεται να εκτελέσουν process της εντολής /proc/net/wireless.

ParserRoute.java

Έχει όλες τις μεθόδους που χρειάζεται να εκτελέσουν process της εντολής route -n.

Ακολουθεί ένα διάγραμμα των κλάσεων με τις μεθόδους τους και ένα διάγραμμα της αλληλεπίδρασης των κλάσεων. Η αλληλεπίδραση δεν είναι εντελώς αναλυτική γιατί θα μπερδευόταν το σχήμα πολύ, αλλά έγινε κατά το δυνατόν ξεκάθαρα και τονίζοντας τα σημαντικά στοιχεία. (Σε περίπτωση που δεν φαίνεται καθαρά περιλαμβάνεται και ένα pdf με την εικόνα σε πλήρη ανάλυση.)



β) Μέθοδοι των κλάσεων

Αρχικά παρουσιάζουμε τον πίνακα με τις κλάσεις και τις μεθόδους των κλάσεων.

<div><div><div>InterfaceWireless</div><div><div><div>-public boolean isWireless()</div><div>-public String getBaseStationMAC()</div><div>-public String getBaseStationESSID()</div><div>-public String getChannel()</div><div>-public String getAccessPointMode()</div><div>-public String getSignalLevel()</div><div>-public String getLinkQuality()</div><div>-public String getTxPower()</div><div>-public String getNoiseLevel()</div><div>-public String getMissedBeacon()</div><div>-public void setBaseStationMAC(String baseStationMAC)</div><div>-public void setBaseStationESSID(String baseStationESSID)</div><div>-public void setChannel(String channel)</div><div>-public void setAccessPointMode(String accessPointSituation)</div><div>-public void setSignalLevel(String signalLevel)</div><div>-public void setLinkQuality(String linkQuality)</div><div>-public void setTxPower(String txPower)</div><div>-public void setNoiseLevel(String noiseLevel)</div><div>-public void setMissedBeacon(String missedBeacon)</div><div>-public void setFlags(boolean checkAp, boolean checkSelf)</div><div>-public void run()</div><div>-public void copyConstructor(InterfaceWireless oldInterface)</div><div>-private boolean compareTo(InterfaceWireless newInterface, InterfaceWireless oldInterface)</div></div></div></div></div>		<div><div><div>Interface</div><div><div><div>-public boolean isWireless()</div><div>-public String getName()</div><div>-public String getMAC()</div><div>-public String getIP()</div><div>-public String getMask()</div><div>-public String getNetworkAddress()</div><div>-public String getBcastAddr()</div><div>-public String getDefaultGateway()</div><div>-public String getMTU()</div><div>-public String getPacketErrorRate()</div><div>-public String getBroadcastRate()</div><div>-public String getConsumedGauge()</div><div>-public void setName(String name)</div><div>-public void setMAC(String mac)</div><div>-public void setIP(String ip)</div><div>-public void setMask(String mask)</div><div>-public void setNetworkAddress(String networkAddress)</div><div>-public void setBcastAddr(String bcastAddr)</div><div>-public void setDefaultGateway(String defaultGateway)</div><div>-public void setMTU(String mtu)</div><div>-public void setPacketErrorRate(String packetsErrorsRate)</div><div>-public void setBroadcastRate(String broadcastRate)</div><div>-public void setConsumedGauge(String consumedGauge)</div><div>-public void setAlive(boolean checkAlive)</div><div>-public void run()</div><div>-private void copyConstructor(Interface oldInterface)</div><div>-private boolean compareTo(Interface newInterface, Interface oldInterface)</div></div></div></div></div>	
<div><div><div>ParserIfconfig</div><div><div><div>-public void getIfconfigProperties(Interface myInterface)</div><div>-public void setNetAddr(Interface myInterface)</div><div>-public ArrayList<String> findNames()</div><div>-private Double findBroadcastRate(int rxBytesInt, int txBytesInt, int curTime, String name)</div></div></div></div></div>		<div><div><div>ParserIwconfig</div><div><div><div>-public void setIwconfigProperties(InterfaceWireless myInterface)</div><div>-public boolean iwConfigCheck(String name)</div></div></div></div></div>	
<div><div><div>ParserIwgetid</div><div><div><div>-public void setChannelProperties(InterfaceWireless myInterface)</div></div></div></div></div>	<div><div><div>ShutDownClass</div><div><div><div>-public void run()</div></div></div></div></div>		
<div><div><div>DataPrinter</div><div><div><div>-public void setAlive(boolean checkAlive)</div><div>-public void printData(Interface myInterface, boolean changesFound)</div><div>-public void printData(InterfaceWireless myInterface, boolean changesFound)</div><div>-public void printData(String name, ArrayList<AccessPoint> apList, boolean changesFound)</div><div>-public void run()</div></div></div></div></div>		<div><div><div>ParserProcNetWireless</div><div><div><div>-public void setNoiseAndMissedBeacon(InterfaceWireless myInterface)</div></div></div></div></div>	
		<div><div><div>ParserIwlist</div><div><div><div>-public void getAccessPointProperties(String name, ArrayList list)</div></div></div></div></div>	
<div><div><div>AccessPoint</div><div><div><div>-public String getMAC()</div><div>-public String getESSID()</div><div>-public String getChannel()</div><div>-public String getMode()</div><div>-public String getSignalLevel()</div><div>-public void setMAC(String mac)</div><div>-public void setESSID(String essid)</div><div>-public void setChannel(String channel)</div><div>-public void setMode(String mode)</div><div>-public void setSignalLevel(String signalLevel)</div></div></div></div></div>		<div><div><div>DataLists</div><div><div><div>-public Interface getMyInterfaces(int i)</div><div>-public InterfaceWireless getMyWirelessInterfaces(int i)</div><div>-public int getMyInterfacesSize()</div><div>-public int getMyWirelessInterfacesSize()</div><div>-public int getMyAccessPointsSize()</div><div>-public void addMyInterfaces(Interface myInterface)</div><div>-public void addMyWirelessInterfaces(InterfaceWireless myInterface)</div><div>-public void removeMyInterfaces(int i)</div><div>-public void removeMyWirelessInterfaces(int i)</div></div></div></div></div>	
		<div><div><div>MainThread</div><div><div><div>-public void run()</div><div>-public void readPropertyFile(int T, int X, int k, int c)</div></div></div></div></div>	
		<div><div><div>ParserRoute</div><div><div><div>-public void getRouteProperties(Interface myInterface)</div></div></div></div></div>	

Ο πίνακας αυτός είναι και στο pdf που περιλαμβάνουμε. Αναλυτικά τώρα οι μέθοδοι:

MainThread.java

- `public static void main(String[] args)` : Η main μέθοδος του προγράμματος.
- `public void readPropertyFile(int T, int k, int X, int c)` : Ανοίγει και διαβάζει το property file και κάνει assign τις μεταβλητές που παίρνει από εκεί.
- `public void run()` : Εκτελεί το νήμα της MainThread όπως περιγράψαμε στις κλάσεις προηγουμένως.

Interface.java

- `public boolean isWireless()` : Μας λέει αν το εν λόγω interface είναι wireless ή όχι.
- `public void setAlive(boolean checkAlive)` : Η μέθοδος που κάνει το interrupt (με μεταβλητή που στέλνεται πό το shutdown hook).
- `private void copyConstructor(Interface oldInterface)` : Δημιουργεί έναν κλώνο του interface για την σύγκριση στην επόμενη επανάληψη.
- `private boolean compareTo(Interface newInterface, Interface oldInterface)` : Κάνει την σύγκριση για να βρει αλλαγές και να αλλάξει τον sleepTime σύμφωνα με Markov.
- `public void run()` : Τρέχει το thread του interface.

InterfaceWireless.java

- `public boolean isWireless()` : Μας λέει αν το εν λόγω interface είναι wireless ή όχι.
- `public void setAlive(boolean checkAlive)` : Η μέθοδος που κάνει το interrupt (με μεταβλητή που στέλνεται πό το shutdown hook).
- `private void copyConstructor(Interface oldInterface)` : Δημιουργεί έναν κλώνο του interface για την σύγκριση στην επόμενη επανάληψη.
- `private boolean compareTo(Interface newInterface, Interface oldInterface)` : Κάνει την σύγκριση για να βρει αλλαγές και να αλλάξει τον sleepTime σύμφωνα με Markov.
- `public void run()` : Τρέχει το thread του interface.

ΣΗΜΕΙΩΣΗ 1: Οι μέθοδοι έχουν τα ίδια ονόματα αλλά δεν κληρονομούνται καθώς διαφέρουν αρκετά στην υλοποίηση.

ΣΗΜΕΙΩΣΗ 2: Κάθε μία από τις παραπάνω κλάσεις περιλαμβάνει πολλές μεθόδους set – get οι οποίες δεν αναφέρονται εδώ γιατί θα έπιαναν πάρα πολύ χώρο, και δεν υπάρχει και κάτι ουσιαστικό να αναφερθεί. Απλά να σημειωθεί ότι όλες είναι κοινές και στα δύο είναι protected στο Interface και κληρονομούνται στο InterfaceWireless. Οι υπόλοιπες που χρειάζονται μόνο στο InterfaceWireless υλοποιούνται στην κλάση του.

AccessPoint.java

- Και εδώ έχουμε μόνο set – get μεθόδους οπότε δεν αναφέρονται αναλυτικά. Για πλήρη οδηγό, είναι όλες γραμμένες στο class diagram που έχουμε παραθέσει από πάνω.

DataLists.java

- public Interface getMyInterfaces(int I) : Επιστρέφει το Interface σύμφωνα με το I που του δίνουμε.
- public InterfaceWireless getMyWirelessInterfaces(int i) : Όμοια με πάνω αλλά για wireless.
- public int getMyInterfacesSize() : Επιστρέφει το μέγεθος της λίστας. Αντίστοιχες συναρτήσεις έχουμε για wireless και access points.
- public void addMyInterfaces(Interface myInterface) : Προσθέτει interface στη λίστα. Αντίστοιχα για wireless.
- public void removeMyInterfaces(int I) : Αφαιρεί interface από τη λίστα. Αντίστοιχα για wireless.

DataPrinter.java

- public void printData(Interface myInterface, boolean changesFound) : Εμφανίζει τα μηνύματα που θέλουμε για απλό interface.
- public void printData(InterfaceWireless myInterface, boolean changesFound) : Εμφανίζει τα μηνύματα για wireless.
- public void printData(String name, ArrayList<AccessPoint> apList, boolean changesFound) : Εμφανίζει τα μηνύματα για τα Access Points.
- public void run() : Ξεκινά το thread του αθροιστή.

ParserIfconfig.java

- public void getIfconfigProperties(Interface myInterface) : Παίρνει τα properties που μπορούμε να πάρουμε από την εντολή ifconfig.
- public void setNetAddr(Interface myInterface) : Βρίσκει και κάνει set την Network Address.
- public ArrayList<String> findNames() : Αυτή η μέθοδος καλείται μόνο από την main για να πάρει ΜΟΝΟ τα ονόματα και να δει αν υπάρχουν καινούρια interfaces ή αν έχουν εξαφανιστεί παλιά interfaces.
- private Double findBroadcastRate(int rxBytesInt, int txBytesInt, int curTime, String name) : Βρίσκει το broadcast rate σύμφωνα με τις οδηγίες.

ParserIwconfig.java

- `public void setIwconfigProperties(InterfaceWireless myInterface)` : Παίρνει τα properties που μπορούμε να πάρουμε από την εντολή `iwconfig`.
- `public boolean iwConfigCheck(String name)` : Ελέγχει αν το interface είναι wireless ή όχι, καλείται από την `main`.

ParserIwgetid.java

- `public void setChannelProperties(InterfaceWireless myInterface)` : Βρίσκει το channel που είναι συνδεδεμένο το wireless.

ParserIwlist.java

- `public void getAccessPointProperties(String name, ArrayList list)` : Παίρνει τις πληροφορίες για τα access points και τα δημιουργεί αν χρειαστεί.

ParserProcNetWireless.java

- `public void setNoiseAndMissedBeacon(InterfaceWireless myInterface)` : Παίρνει τον θόρυβο και τα missed beacons πό την εντολή `/proc/net/wireless`.

ParserRoute.java

- `public void getRouteProperties(Interface myInterface)` : Παίρνουμε την default gateway.

ShutDownClass.java

- `public void run()` : Εκτελεί το shut down hook και κλείνει όλα τα νήματα.

Στον φάκελο συμπεριλαμβάνουμε και ένα Javadoc με όλες τις κλάσεις και μεθόδους.

Κεφάλαιο 2

α) Δοκιμές και screenshots

Οι δοκιμές έγιναν σε laptop με λειτουργικό Ubuntu Linux 10.04 με Java 7. Δυστυχώς δεν είχαμε δεύτερο ασύρματο interface για τις δοκιμές. Παραθέτονται τα screenshots που δείχνουν τη λειτουργία του προγράμματος μας.

Ο υπολογιστής είχε μόνιμα τα εξής interfaces: eth0, lo, wlan0. Οπότε όλα τα screenshots παρουσιάζουν τις λειτουργίες πάνω σε αυτά τα interfaces. Λόγω έλλειψης χώρου θα αναλύσουμε κάποια κομμάτια από τα screenshots, τα οποία όμως έχουν αποσταλεί ολόκληρα στον ανάλογο φάκελο.

```
*** Properties File Found! ***
Only 1 Wireless Interface found. Resuming checks with this interface only.
Current Thread Number : 3
Total Interfaces Number : 3 Wireless Interfaces Number : 1
\\*** Main thread will sleep for 5000
Interface Name is eth0
    MAC      : 00:90:f5:80:3e:af
    IP       : Not Found
    Bcast Address : Not Found
    Default Gateway : Not Found
    Mask     : Not Found
    MTU      : 1500
    Network Addr : Not Found
    Packet Error Rate : Not Found
    Brcast Rate : 0.0
\\*** eth0 will sleep for 5000
Interface Name is lo
    MAC      : Not Found
    IP       : 127.0.0.1
    Bcast Address : Not Found
    Default Gateway : Not Found
    Mask     : 255.0.0.0
    MTU      : 16436
    Network Addr : 127.0.0.0
    Packet Error Rate : Not Found
    Brcast Rate : 0.0
```

Εδώ φαίνεται η εκκίνηση του προγράμματος. Έχει διαβάσει με επιτυχία το property file, βρήκε ένα μόνο wireless (οπότε αυτό θα κάνει όλους τους ελέγχους για access points) και βάζει τα threads να κοιμηθούν αφού έχει πάρει τις πληροφορίες που πρέπει και τις έχει στείλει στον αθροιστή για εκτύπωση.


```

***lo will sleep for 5000
10 Access Points Found by wlan0 wireless interface.
  ESSID : "ThomsonB95AD4" MAC : 00:24:17:94:8C:AD
  Channel : 11 AP Mode : Master
  Signal level : -74 dBm
  ESSID : "hol - NetFaster WLAN 3" MAC : 00:05:59:45:F8:9F
  Channel : 1 AP Mode : Master
  Signal level : -75 dBm
  ESSID : "hol" MAC : 00:05:59:35:B3:25
  Channel : 6 AP Mode : Master
  Signal level : -73 dBm
  ESSID : "CONNX" MAC : 00:1D:19:8C:EF:CA
  Channel : 6 AP Mode : Master
  Signal level : -71 dBm
  ESSID : "NetFaster WLAN" MAC : D0:15:4A:19:FB:E2
  Channel : 6 AP Mode : Master
  Signal level : -72 dBm
  ESSID : "hol - NetFaster WLAN 3" MAC : 00:05:59:4A:3B:02
  Channel : 11 AP Mode : Master
  Signal level : -48 dBm
  ESSID : "\x00\x00\x00\x00\x00\x00\x00" MAC : 30:46:9A:84:24:4E
  Channel : 11 AP Mode : Master
  Signal level : -78 dBm
  ESSID : "NINA" MAC : 00:0C:E3:64:AF:EC
  Channel : 11 AP Mode : Master
  Signal level : -78 dBm
  ESSID : "0TEbc847e" MAC : 00:13:33:A2:74:C8
  Channel : 6 AP Mode : Master
  Signal level : -79 dBm
  ESSID : "conn-xa54d2b" MAC : 4C:09:B4:A5:4D:2B
  Channel : 13 AP Mode : Unknown/bug

```

Εμφανίζουμε τα Access Points, τις πληροφορίες τους και ποιά interface έκανε τον έλεγχο. Φαίνεται ότι οι πληροφορίες γίνονται parsing με τον σωστό τρόπο.

```

Interface Name is wlan0
MAC : 00:16:ea:ce:5a:12
IP : Not Found
Bcast Address : Not Found
Default Gateway : Not Found
Mask : Not Found
MTU : 1500
Network Addr : Not Found
Packet Error Rate : Not Found
Brcast Rate : 0.0
BS MAC : Not-Associated
BS ESSID : "f2\x0D\xB71X\XA3Z%\x05\x17X\xE9^\xD4\xAB\xB2\xCD\xC6\x9B\xB4T\x11\x0E\x82tA!=\xDC\x87"
Channel : 11
Access Point Mode : Managed
Link Quality : Not Found
Signal level : Not Found
Tx-Power : 15 dBm
Noise Level : 0
Missed Beacon Num : 0
\*** wlan0 will sleep for 5000
Only 1 Wireless Interface found. Resuming checks with this interface only.
Current Thread Number : 3
Total Interfaces Number : 3 Wireless Interfaces Number : 1

```

Βλέπουμε ότι εδώ, έχουμε αποσυνδέσει το wlan0 οπότε δεν βρίσκει καθόλου σχετικές με access point πληροφορίες (έχουμε αρχικοποιήσει μήνυμα για αυτές τις περιπτώσεις). Το όνομα του BS essid που βγάζει το εμφανίζει και η εντολή ifconfig από terminal οπότε δεν πρόκειται για λάθος στο parsing, αλλά δεν ξέρουμε γιατί το εμφανίζει.

Στο τέλος της εικόνας φαίνεται ένα μήνυμα που εμφανίζεται περιοδικά για τυπικό έλεγχο του αριθμού και του είδους των interfaces.

```

\*** Main thread will sleep for 4995
eth0 had no changes!
\*** eth0 will sleep for 3485
lo had no changes!
\*** lo will sleep for 3478
Only 1 Wireless Interface found. Resuming checks with this interface only.
Current Thread Number : 3
Total Interfaces Number : 3 Wireless Interfaces Number : 1

```

Εμφανίζουμε μήνυμα όταν δεν υπάρχουν αλλαγές, και εδώ φαίνεται και η αλλαγή στο sleepTime σύμφωνα με Markov. Ο αριθμός δεν είναι στρογγυλός και συγκεκριμένος γιατί υπολογίζουμε και το Δt του χρόνου εκτέλεσης, όπως ζητείται.


```

\*** Main thread will sleep for 9989
eth0 had no changes!
\*** eth0 will sleep for 6969
lo had no changes!
\*** lo will sleep for 6940
11 Access Points Found by wlan0 wireless interface.
  ESSID : "ThomsonB95AD4" MAC : 00:24:17:94:8C:AD
  Channel : 11 AP Mode : Master
  Signal level : -76 dBm

  ESSID : "hol - NetFaster WLAN 3" MAC : 00:05:59:45:F8:9F
  Channel : 1 AP Mode : Master
  Signal level : -75 dBm

  ESSID : "hol" MAC : 00:05:59:35:B3:25
  Channel : 6 AP Mode : Master
  Signal level : -76 dBm

```

Διαφορετικός αριθμός από Access Points (πριν ήταν 10 και τώρα 11), όπως είναι αναμενόμενο, και άλλα SleepTime πάλι. Σε σύγκριση με πριν, εφόσον δεν υπάρχουν αλλαγές στο lo, το sleepTime αυξάνεται.

```

Interface Name is wlan0
  MAC : 00:16:ea:ce:5a:12
  IP : 192.168.2.7
  Bcast Address : 192.168.2.255
  Default Gateway : 192.168.2.1
  Mask : 255.255.255.0
  MTU : 1500
  Network Adr : 192.168.2.0
  Packet Error Rate : Not Found
  Brcast Rate : 0.0

  BS MAC : 00:05:59:4A:3B:02
  BS ESSID : "hol - NetFaster WLAN 3"
  Channel : 11
  Access Point Mode : Managed
  Link Quality : 62/70
  Signal level : -48 dBm
  Tx-Power : 15 dBm
  Noise Level : -256
  Missed Beacon Num : 0
\*** wlan0 will sleep for 3211
lo had no changes!
\*** lo will sleep for 8897
eth0 had no changes!
\*** eth0 will sleep for 8924
Only 1 Wireless Interface found. Resuming checks with this interface only.

```

Τώρα συνδεθήκαμε στο wireless AP και βλέπουμε ότι το wlan0 πήρε κανονικά τις πληροφορίες που του αντιστοιχούν.

```
\*** Main thread will sleep for 14978
7 Access Points Found by wlan0 wireless interface.
   ESSID : "ThomsonB95AD4" MAC : 00:24:17:94:8C:AD
   Channel : 11   AP Mode : Master
   Signal level : -77 dBm
```

Το MainThread έχει φτάσει να κοιμάται για πολύ χρόνο πράγμα λογικό αφού δεν βρίσκει αλλαγές στα interfaces.

```
\*** wlan0 will sleep for 2696
^C *** Executing ShutDown Hook. ***
\*** Main Thread Interrupted!

lo had no changes!
\*** lo will sleep for 7381
eth0 had no changes!
\*** eth0 will sleep for 7407
Interface Name is wlan0
   MAC      : 00:16:ea:ce:5a:12
   IP       : 192.168.2.7
   Bcast Address : 192.168.2.255
   Default Gateway : 192.168.2.1
   Mask     : 255.255.255.0
   MTU      : 1500
   Network Adr : 192.168.2.0
   Packet Error Rate : Not Found
   Brcast Rate : 0.0

   BS MAC    : 00:05:59:4A:3B:02
   BS ESSID  : "hol - NetFaster WLAN 3"
   Channel   : 11
   Access Point Mode : Managed
   Link Quality : 64/70
   Signal level : -46 dBm
   Tx-Power  : 15 dBm
   Noise Level : -256
   Missed Beacon Num : 0
\*** wlan0 will sleep for 3108
wlan0 is finished!

lo is finished!

eth0 is finished!

Current Thread Number : 0
Total Interfaces Number : 0 Wireless Interfaces Number : 0
Data Printer and all threads closed. Program finished. Thank you!
lilykos@pc:~/Desktop$ |
```

Εδώ βλέπουμε το κλείσιμο του προγράμματος. Αμέσως μετά το ctrl-c εκτελείται το shutdown hook και κλείνει το MainThread. Δίνει σήμα σε όλα τα υπόλοιπα threads να τελειώσουν αυτό που κάνουν και στην επόμενη επανάληψη να κλείσουν. Κλείνουν ομαλά, κλείνει και ο αθροιστής και εμφανίζεται ανάλογο μήνυμα. Βλέπουμε από το μήνυμα ότι πλέον οι ίστες είναι άδειες και όλα τα threads έκλεισαν.

β) Περιγραφή υλοποίησης Markov

Ακολουθήσαμε όσο πιο πιστά γινόταν την εκφώνηση στην υλοποίηση της Markov. Στο τέλος κάθε επανάληψης ενός thread δίνουμε σήμα να κοιμηθεί ανάλογα με τις αλλαγές που έγιναν μέσω μιας boolean μεταβλητής. Υπάρχουν οι εξής περιπτώσεις:

- Αν έγινε κάποια αλλαγή τότε αυτόματα το sleepTime γίνεται T, όπου T ο χρόνος που δίνεται από το property file.
- Αντίθετα, αν δεν βρει αλλαγές, αλλά δεν είναι ακόμα στον αριθμό c που έχει δοθεί κοιμάται για όποιο ήταν το προηγούμενο sleepTime που κρατάμε σε τοπική μεταβλητή.
- Σε περίπτωση που φτάσει στις c επαναλήψεις (to βρίσκουμε μέσω mod) και δεν έχει βρει αλλαγή προχωράει στην επόμενη κατάσταση, η οποία είναι sleepTime + T. Έτσι δεν χρησιμοποιούμε μετρητή, ούτε εξαρτώμαστε από κάποια προηγούμενη sleepTime.

Όπως ζητείται υπολογίζουμε και το Δt μεταξύ των εκτελέσεων. Το πρόβλημα είναι ότι αν έχουμε μικρό χρόνο αναμονής και μεγάλο c, μπορεί να μη λειτουργήσει σωστά γιατί κάποιες συναρτήσεις αργούν να τελειώσουν, όπως η iwlist, και ο συνολικός χρόνος βγαίνει αρνητικός. Σαν συνθήκη παίρνουμε λοιπόν το απόλυτο του sleepTime που βρίσκουμε, για να αποφύγουμε exception. Με μεγαλύτερους αριθμούς όμως, όπως T = 5, 6 και c = 1, 2, 3 δεν παρατηρήθηκε πρόβλημα γενικά.

Κεφάλαιο 3

α) Εκτέλεση προγράμματος

Το πρόγραμμα έτρεξε και ελέγχθηκε κατά το δυνατόν για σωστή λειτουργία σε Netbeans και terminal. Η μεταγλώτιση έγινε σε Netbeans και εκτελέσαμε το jar που προκύπτει από τερματικό. Για την εκτέλεση φτιάξαμε ένα μικρό script (exe.sh) σε bash που βρίσκει το τρέχον μονοπάτι και τρέχει την εντολή “java -jar MainThread.jar” για να λειτουργήσει το εκτελέσιμο όπως ζητείται. Το script είναι στον φάκελο με το jar και θα λειτουργήσει μόνο αν και τα 2 είναι στον ίδιο φάκελο.

ΣΗΜΑΝΤΙΚΟ: Στο laptop που εργαστήκαμε είχαμε εγκαταστήσει την java στο προεπιλεγμένο μονοπάτι, όπως εμφανίζεται κατά την εγκατάσταση του sun jdk. Μόνο έτσι λειτουργεί το script μέσω της εντολής ./exe.sh και έτσι έχει δημιουργηθεί και το jar.

β) Παραδοχές – Λεπτομέρειες

Αρχικά στέλνουμε μαζί με το Netbeans project και το jar τα: readme, javadoc, screenshots, executable script και class diagram pdf. Με αυτά θεωρούμε ότι καλύπτουμε τις οδηγίες αποστολής.

Σαν παραδοχές υλοποίησης και εκτέλεσης να αναφέρουμε ότι

- Έχουμε φτιάξει τα απαραίτητα threads, δίνουμε τις πληροφορίες που πρέπει στα interfaces και στέλνουμε τις πληροφορίες για εκτύπωση στον αθροιστή (που είναι και αυτός νήμα) ανα sleepTime (διαφορετικό για κάθε νήμα αφού κάθε interface στέλνει τις δικές του πληροφορίες.
- Ο αθροιστής σύμφωνα με τις οδηγίες, δεν εκτελεί αλγόριθμο μεταβλητού sleepTime.
- Όταν δεν υπάρχουν αλλαγές στέλνονται ανάλογα μηνύματα, όπου είναι εμφανές αν γίνονται αλλαγές όχι.
- Ελέγχουμε για νέα interfaces με βάση τα ονόματα καθώς κάθε όνομα είναι μοναδικό για κάθε pc.
- Ανανεώνουμε τη λίστα με τα access points και εκτυπώνουμε όταν βρει διαφορετικό αριθμό (σύμφωνα με ότι ειπώθηκε στην λίστα του μαθήματος), αλλά τα ίδια τα access points τα αλλάζουμε κάθε φορά που γίνεται έλεγχος, το θεωρήσαμε πιο σωστό σαν λογική (να μην ξεμένουν λάθος ονόματα στην λίστα των access points).

Γενικά θεωρούμε ότι η άσκηση υλοποιεί το σύνολο αυτών που ζητούνται. Ελπίζουμε να μην έχουμε παραλείψει κάτι άθελα μας.