

## Δίκτυα Υπολογιστών I (6ο εξάμηνο)

## Experimental Virtual Lab

### Εργασία δικτυακού προγραμματισμού : Java serial communications programming

Η εργασία που ζητείται να εκπονηθεί στο μάθημα Δίκτυα Υπολογιστών I αποτελεί μία εφαρμογή δικτυακού προγραμματισμού (network programming). Η εργασία στοχεύει **(α)** στην ανάπτυξη πειραματικής δικτυακής εφαρμογής με τη γλώσσα προγραμματισμού Java, **(β)** την εξοικείωση με μηχανισμούς ασύγχρονης σειριακής (ή σειραϊκής) επικοινωνίας (asynchronous serial communications) υπολογιστών στην πράξη και **(γ)** τη συλλογή στατιστικών μετρήσεων τιμών ορισμένων παραμέτρων που συμβάλλουν μαζί με άλλες στη διαμόρφωση της ποιότητας της επικοινωνίας πάνω από πραγματικά φυσικά κανάλια επικοινωνίας υπολογιστών.

Η εφαρμογή μπορεί να αναπτυχθεί αυτοτελώς σε προσωπικό υπολογιστή του εργαστηρίου του Τομέα ή άλλου χώρου. Ζητείται όμως ορισμένες μετρήσεις να πραγματοποιηθούν σε συνεργασία με τον σερβερ Ιθάκη του πειραματικού εικονικού εργαστηρίου (experimental virtual lab) που αναπτύσσεται στα πλαίσια των μαθημάτων Δίκτυα Υπολογιστών I και II. Η διεύθυνση <http://ithaki.eng.auth.gr/netlab/index.html> αποτελεί το σημείο αναφοράς των δραστηριοτήτων που ακολουθούν στην εργασία αυτή.

### Δημιουργία περιβάλλοντος προγραμματισμού σε Java

Για την εργασία απαιτείται η διάθεση του περιβάλλοντος προγραμματισμού Java Development Kit JDK το οποίο παρέχεται δωρεάν από τη διεύθυνση <http://java.sun.com/javase/downloads/index.jsp>. Η εξοικείωση με το περιβάλλον αυτό μπορεί να διευκολυνθεί με το ηλεκτρονικό εκπαιδευτικό υλικό που διατίθεται επίσης δωρεάν από τη διεύθυνση <http://java.sun.com/docs/books/tutorial/index.html>. Πλήρης τεκμηρίωση σε ηλεκτρονική μορφή για την Java μπορεί να βρεθεί στη διεύθυνση <http://java.sun.com/javase/6/docs/index.html>. Συμπληρωματικά τα βιβλία «Learning Java» και «Java I/O» των εκδόσεων O'Reilly Associates αποτελούν χρήσιμα εκπαιδευτικά βοηθήματα ενώ το βιβλίο «Java in a Nutshell : A Quick Reference Guide» επίσης των εκδόσεων O'Reilly Associates αποτελεί μία συνοπτική αλλά εξαιρετικά περιεκτική αναφορά στη γλώσσα προγραμματισμού Java και τις βασικές βιβλιοθήκες της.

### Εργαλεία προγραμματισμού σε Java

Κάθε εφαρμογή σε Java απαιτεί **(α)** τη δημιουργία κώδικα σε πηγαία μορφή (source code), **(β)** τη μεταγλώττισή του σε κώδικα εικονικής μηχανής Java (Java virtual machine JVM code) και **(γ)** την εκτέλεσή του με τον διερμηνευτή Java (Java interpreter). Στα πλαίσια της εργασίας αυτής προτείνεται για το στάδιο (α) η χρήση του επεξεργαστή κειμένου notepad<sup>1</sup> ενώ για τα στάδια (β) και (γ) η χρήση των εργαλείων javac και java αντίστοιχα μέσα από παράθυρο γραμμών εντολών (command line window). Τα

<sup>1</sup> Εναλλακτικά μπορεί να χρησιμοποιηθεί το εργαλείο Programmer's Notepad που διατίθεται δωρεάν από τη διεύθυνση <http://www.pnotepad.org> ή κάποιο από τα ενοποιημένα περιβάλλοντα ανάπτυξης Java IDEs (integrated development environments) NetBeans ή Eclipse που διατίθενται επίσης δωρεάν από τις διευθύνσεις <http://www.netbeans.org> και <http://www.eclipse.org> αντίστοιχα.

εργαλεία `javac` και `java` περιλαμβάνονται στην εγκατάσταση του JDK που αναφέρθηκε παραπάνω. Το `notepad` είναι διαθέσιμο σε οποιονδήποτε υπολογιστή Win/98/Me/2K/XP/Vista/7/8/10.

### *Δικτυακές εφαρμογές σε Java*

Η εργασία επικεντρώνεται στην εξοικείωση με τους μηχανισμούς ασύγχρονης επικοινωνίας υπολογιστών μέσω σειριακής θύρας (`asynchronous serial port`) ενός υπολογιστή σε συνδυασμό με διάταξη διαμορφωτή / αποδιαμορφωτή (`modulator / demodulator`), το γνωστό modem που είναι απαραίτητο για την επικοινωνία υπολογιστών μέσα από πρακτικά κάθε φυσικό κανάλι επικοινωνίας που παρέχεται από ένα δίκτυο υπολογιστών.

Η κλασική θύρα σειριακής επικοινωνίας RS232 υπηρέτησε την επικοινωνία υπολογιστών για δεκαετίες. Σήμερα η θύρα αυτή έχει αντικατασταθεί από τη νεότερη θύρα USB πρακτικά σε όλους τους υπολογιστές σύγχρονης τεχνολογίας (γιατί ?) ωστόσο παραμένει ακόμη ως ένα από τα κύρια μέσα σειριακής επικοινωνίας σε πολλούς σύγχρονους μικροεπεξεργαστές όπως για παράδειγμα στον `Amtel AVR` ή το `Arduino`. Ταυτόχρονα οι συσκευές modem ITU-V στενής ζώνης των 4 KHz του αναλογικού τηλεφωνικού δικτύου (`POTS plain old telephone system`) έχουν πρακτικά εκλείψει (γιατί ?) και αυτές. Επίσης τα κινητά τηλέφωνα με ενσωματωμένο μόντεμ GMSK τεχνολογίας CSD (`Circuit Switched Data`) έχουν και αυτά σχεδόν εκλείψει (γιατί ?) στον βαθμό που τα νέα `smart phones` ενσωματώνουν modem ευρείας ζώνης νεότερης τεχνολογίας πχ 2x20 MHz 256 QAM.

Όμως οι μηχανισμοί σειριακής επικοινωνίας παραμένουν κεντρικό ζήτημα στην επικοινωνία υπολογιστών, η δε διαχείρισή τους σε περιβάλλον θορύβου και με συνθήκες υψηλού φορτίου αποκτούν ιδιαίτερη βαρύτητα στη σημερινή τεχνολογική πραγματικότητα. Για τους λόγους αυτούς και με στόχο τη διευκόλυνση της εκπόνησης της εργασίας του μαθήματος Δίκτυα Υπολογιστών Ι, ο σέρβερ Ιθάκη του εικονικού εργαστηρίου θέτει στη διάθεση των φοιτητών/τριών μηχανισμούς Java οι οποίοι δημιουργούν εικονικά τερματικά (`virtual terminals`) αξιοποιώντας τα αναπόφευκτα (γιατί ?) ευρυζωνικά μόντεμ (`broadband modem`) τα οποία σήμερα βρίσκονται ενσωματωμένα με τη μία ή την άλλη μορφή σε κάθε συσκευή που συνδέεται καθοιονδήποτε τρόπο, ενσύρματα ή ασύρματα, στο Ιντερνετ.

Σημειώνεται ότι, για τους ενδιαφερόμενους/νες φοιτητές/τριες, το Παράρτημα Α παρουσιάζει στοιχεία σειριακής επικοινωνίας υπολογιστών λειτουργικού συστήματος Windows με μικρο-υπολογιστές, για παράδειγμα τύπου `Arduino`, είτε μέσω φυσικής θύρας RS232 είτε μέσω εικονικής θύρας RS232 που δημιουργείται με τη βοήθεια προσαρμογέα USB σε RS232 (`USB-to-RS232 converter`).

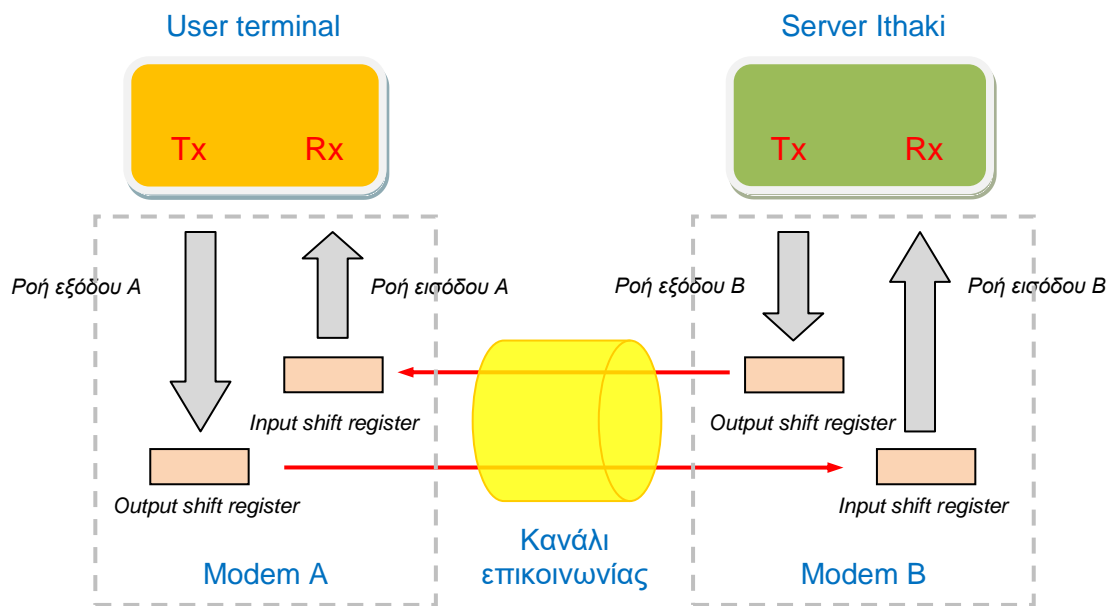
### *Το εικονικό μόντεμ Ithaki-Smart-Modem και η επικοινωνία με το εικονικό εργαστήριο*

Η κλάση αντικειμένων `Modem.class` που διατίθεται από τον σέρβερ Ιθάκη του εικονικού εργαστηρίου επιτρέπει τη σειριακή επικοινωνία ενός προσωπικού υπολογιστή με ένα τοπικό εικονικό μόντεμ (`local virtual modem`) και δι' αυτού με τον σέρβερ Ιθάκη

πάνω από μια τυπική σύνδεση Ιντερνετ, όχι απλή επιλεγόμενη (dialup) σύνδεση τηλεφωνικού δικτύου. Το εικονικό αυτό μόντεμ μπορεί να κατασκευαστεί μέσω του μηχανισμού δημιουργίας νέων αντικειμένων μιας δικτυακής εφαρμογής Java πχ

```
Modem modem = new Modem(speed) ;
```

όπου speed δηλώνει την επιθυμητή ταχύτητα<sup>2</sup> λειτουργίας του εικονικού μόντεμ από 1 Kbps έως 80 Kbps. Το αντικείμενο modem του παραπάνω παραδείγματος διαθέτει με τη σειρά του μεθόδους ανάγνωσης ή εγγραφής, modem.read()<sup>3</sup> ή modem.write(int x) ή modem.write(byte[] x) από ή προς ροές (byte streams) εισόδου ή εξόδου οι οποίες δημιουργούνται αυτομάτως κατά το στάδιο δημιουργίας του αντικειμένου<sup>4</sup>.



<sup>2</sup> Η τιμή της πραγματικής ταχύτητας με την οποία γίνεται η κλήση είναι το πλησιέστερο ακέραιο πολλαπλάσιο του 1 Kbps για επιθυμητή ταχύτητα μικρότερη από 8 Kbps και το πλησιέστερο ακέραιο πολλαπλάσιο των 8 Kbps για επιθυμητή ταχύτητα έως 80 Kbps. Ποιες μπορεί να είναι οι πραγματικές ταχύτητες για επιθυμητές ταχύτητες μικρότερες του 1 Kbps και μεγαλύτερες των 80 Kbps ? Σημειώνεται ότι η ταχύτητα λειτουργίας παραμένει σταθερή όταν καλείται η μέθοδος modem.fixed(true). Σε κάθε άλλη περίπτωση η ταχύτητα παρουσιάζει τυχαίες μεταβολές που μειώνουν την αρχική ταχύτητα speed κατά τυχαία χρονικά διαστήματα προσομοιώνοντας με τον τρόπο αυτόν τις χρονομεταβολές ενός πραγματικού τηλεφωνικού καναλιού επικοινωνίας υπολογιστών.

<sup>3</sup> Η μέθοδος read αναστέλλει την εκτέλεση του προγράμματος (blocking operation) μέχρι να εμφανιστεί ένα τουλάχιστον byte στη ροή εισόδου ή ανιχνευτεί το τέλος της ροής ή παρουσιαστεί κάποια εξαίρεση πχ τέλος χρόνου αναμονής για άφιξη του επόμενου byte (timeout exception).

<sup>4</sup> Επικοινωνικά οι εντολές modem.speed(int speed), modem.setTimeout(int timeout), modem.open(String destination), modem.close(), modem.getInputStream() και modem.getOutputStream() επιτρέπουν τον ορισμό της ταχύτητας, την απόλυση της κλήσης μετά από περίοδο αδρανείας, την ενεργοποίηση του modem, την απενεργοποίηση του modem, τη δημιουργία ροής εισόδου και τη δημιουργία ροής εξόδου αντίστοιχα μέσα από εφαρμογή Java.

Σημειώνεται ότι το άλλο «άκρο» των ροών αυτών συσχετίζεται αυτομάτως πάντοτε με ένα παρόμοιο εικονικό μόντεμ (remote virtual modem) που λειτουργεί στον σέρβερ Ιθάκη όπως φαίνεται στο σχήμα που ακολουθεί.

Σχετικά με τη διαχείριση των ροών εισόδου και εξόδου διευκρινίζεται ότι η μέθοδος `read()` επιστρέφει την τιμή `-1` ενώ η μέθοδος `write()` επιστρέφει την τιμή `false` όταν οι ροές εισόδου ή εξόδου αντίστοιχα δεν είναι διαθέσιμες όπως συμβαίνει, για παράδειγμα σε περιπτώσεις διακοπής της επικοινωνίας με τον σέρβερ Ιθάκη του εικονικού εργαστηρίου.

Οι παραπάνω μέθοδοι σε συνδυασμό με τις αντίστοιχες ροές αποτελούν τους θεμελιώδεις μηχανισμούς επικοινωνίας μιας δικτυακής εφαρμογής (λογισμικό, software) με εικονικό modem και δι' αυτού με τη γραμμή επικοινωνίας (υλικό, hardware) μιας σύγχρονης σύνδεσης Ιντερνετ.

Ας σημειωθεί ότι ένας σύντομος έλεγχος και μια σειρά απλών δοκιμών της επικοινωνίας με τον σέρβερ πάνω από οποιαδήποτε σύνδεση Ιντερνετ μπορεί να πραγματοποιηθεί χωρίς απαραίτητα να απαιτείται εκ των προτέρων η ανάπτυξη κάποιας ειδικής δικτυακής εφαρμογής. Πράγματι, η εντολή

```
java Modem <CR>
```

σε παράθυρο γραμμών εντολών CLW (Command Line Window) δημιουργεί αυτομάτως ένα εικονικό modem και ταυτόχρονα ενεργοποιεί ένα εικονικό τερματικό διά του οποίου επιτρέπεται η εισαγωγή εντολών AT. Το γεγονός αυτό καθιστά δυνατό τον έλεγχο ή τη δοκιμαστική λειτουργία της σύνδεσης με τον σέρβερ Ιθάκη του εικονικού εργαστηρίου.

Ο μηχανισμός διεπαφής (interface) του εικονικού modem αποκρίνεται επιπλέον σε ένα περιορισμένο ρεπερτόριο εντολών AT δια των οποίων γίνεται ο χειρισμός του modem πχ

```
AT<CR>
```

```
ATI<CR>
```

```
ATS<CR>
```

```
ATS=<speed>
```

```
ATH1<CR>
```

```
ATH0<CR>
```

Ειδικά για την κλήση του σερβερ του εικονικού εργαστηρίου από οποιοδήποτε τερματικό συνδεδεμένο στο Ιντερνετ διατίθεται η εντολή

```
ATD2310ITHAKI<CR>
```

Διευκρινίζεται ότι η ενεργοποίηση του τοπικού εικονικού modem μέσω του εικονικού τερματικού ως αποτέλεσμα της εντολής

```
java Modem <CR>
```

θέτει το modem σε command mode. Το γεγονός αυτό σημαίνει ότι το τοπικό modem αποκρίνεται μόνο σε εντολές AT. Στη συνέχεια το modem αυτό μπορεί να τεθεί σε data mode όταν η εκτέλεση της εντολής ATD2310ITHAKI ενεργοποιήσει επιτυχώς τη σύνδεση του τοπικού modem με το μακρινό modem. Με τη σειρά του το γεγονός αυτό σημαίνει ότι οποιαδήποτε ακολουθία χαρακτήρων ή bytes που αποστέλλεται στο τοπικό modem διαβιβάζεται ως έχει στο μακρινό modem και δι' αυτού στον σερβερ Ιθάκη.

Η απόλυση της κλήσης (handset hang-up) δηλαδή η διακοπή της επικοινωνίας με το μακρινό modem γίνεται με την εντολή

```
+++ATH0<CR>
```

όπου η ακολουθία +++ λειτουργεί ως escape sequence που μεταφέρει την λειτουργία του τοπικού modem από data mode σε command mode.

*Η εφαρμογή στο εικονικό εργαστήριο του μαθήματος*

Ζητείται η ανάπτυξη εφαρμογής Java η οποία **(α)** επικοινωνεί με τον server του εργαστηρίου μέσω ενός από τα οκτώ εικονικά modem που αποκρίνονται στον «αριθμό κλήσης» 2310-ITHAKI και **(β)** επιτρέπει στατιστικές μετρήσεις τιμών ορισμένων παραμέτρων της επικοινωνίας αυτής.

Τα χαρακτηριστικά της εφαρμογής και οι ζητούμενες μετρήσεις περιγράφονται στη συνέχεια.

[1] Όπως εμφανίζεται στα σχήματα που ακολουθούν, με την επιλογή της εργασίας Java serial communications programming από τον κατάλογο επιλογών που παρέχεται στη διεύθυνση <http://ithaki.eng.auth.gr/netlab/index.html> ο server αποδέχεται τον/την προσερχόμενο/νη φοιτητή/τρια για εργασία διάρκειας δύο ωρών (session) προσδιορίζοντας ένα σύνολο παραμέτρων απαραίτητων για την εκτέλεση της εργασίας και την ανάπτυξη της ζητούμενης εφαρμογής userApplication.

[2] Μεταξύ της επιλογής της εργασίας και της ενεργοποίησης των παραμέτρων της εφαρμογής userApplication που θα ακολουθήσει μεσολαβεί χρόνος έως και 60 δευτερόλεπτα. Μετά το πέρας του συνολικού χρόνου εργασίας των δύο ωρών και εάν είναι επιθυμητή η συνέχεια, απαιτείται νέα επιλογή της ίδιας εργασίας για να δοθούν νέες τιμές παραμέτρων.

[3] Η εφαρμογή userApplication που ζητείται να κατασκευαστεί θα πρέπει κατ' αρχή να παρέχει τη δυνατότητα λήψης ενός μεγάλου αριθμού πακέτων echoPacket διαδοχικά από τον server μετά από την αποστολή κάθε φορά από την εφαρμογή προς τον

server του κωδικού `echo_request_code`. Ο κωδικός έχει τη μορφή `EXXXX`<sup>5</sup> και δίνεται από τον σερβερ Ιθάκη κατά την έναρξη της εργασίας στο εικονικό εργαστήριο. Τα πακέτα `echoPacket` που αποστέλλονται από τον server είναι της μορφής :

```
PSTART DD-MM-YYYY HH-MM-SS PC PSTOP
```

όπου `DD-MM-YYYY` η ημερομηνία και `HH-MM-SS` ο χρόνος αποστολής του πακέτου και `PC` η τρέχουσα τιμή ενός μετρητή (packet counter) ο οποίος απαριθμεί τα πακέτα `echoPacket` με `modulo(100)`.

[4] Η χρήση του κωδικού `image_request_code` με τη μορφή `MXXXX`<sup>5</sup> επιτρέπει την αποστολή του τρέχοντος `frame`<sup>6</sup> από τον `videoCoder` που φιλοξενείται στη διεύθυνση <http://ithaki.eng.auth.gr/netlab/video.html> και παρέχει `real-time`<sup>7</sup> video με την κίνηση της Εγνατίας Οδού στο τμήμα εμπρός από την Πολυτεχνική Σχολή. Η αποστολή περιλαμβάνει 8 έως 80 kilobytes περίπου, ανάλογα με τις συνθήκες του φωτισμού, τη δραστηριότητα στο οπτικό πεδίο της κάμερας και τις διαστάσεις του `frame` που ζητείται. Τα bytes αυτά αποθηκεύονται κατά τη λήψη τους σε `binary` αρχείο<sup>8</sup> με επέκταση ονόματος `.jpeg` ή `.jpg` επιτρέπουν την απεικόνιση του τρέχοντος `frame` στο τερματικό με τη βοήθεια οποιουδήποτε εργαλείου απεικόνισης εικόνων π.χ. Paint.

Η αποστολή του κωδικού `image_request_code` για τη λήψη εικόνας μπορεί να συνοδεύεται από την ένδειξη `CAM=FIX` ή `CAM=PTZ` η οποία προσδιορίζει στην είσοδο του κωδικοποιητή το σήμα από αντίστοιχη κάμερα που λειτουργεί online με τον server του εικονικού εργαστηρίου. Το γεγονός αυτό έχει ως αποτέλεσμα τη λήψη μέσα από την εφαρμογή `userApplication` εικόνας από τη συγκεκριμένη κάμερα. Σε απουσία της παραμέτρου `CAM` ο server θεωρεί αυτομάτως την παράμετρο `CAM=FIX` (default value).

Σημειώνεται ότι η πρώτη κάμερα παράγει εικόνες με ανάλυση 640x480 pixels και σταθερή (fixed) γωνία λήψης ενώ η δεύτερη παράγει εικόνες με ανάλυση 320x240 pixels και

---

<sup>5</sup> Σε εφαρμογές μέσω του virtual modem του σερβερ Ιθάκη, οι κωδικοί `request_code` θα πρέπει να συνοδεύονται από τον χαρακτήρα `\r` ή ASCII 13 (code delimiter).

<sup>6</sup> Κάθε `frame` αποστέλλεται με τη μορφή αρχείου `.jpeg`. Η διαχείριση τέτοιων αρχείων (απεικόνιση, από-θήκευση, μετασχηματισμός format, κ.α.) δεν περιλαμβάνεται στις απαιτήσεις της εργασίας αυτής. Ωστόσο σημειώνεται, με σκοπό τη διευκόλυνση της εφαρμογής κατά τη λήψη των εικόνων, ότι ένα `binary` αρχείο `jpeg` αρχίζει με τα bytes `0xFF 0xD8` (start of image delimiter) και τελειώνει με τα bytes `0xFF 0xD9` (end of image delimiter).

<sup>7</sup> Το ψηφιακό σήμα `video` κωδικοποιείται σύμφωνα με μία ad-hoc έκδοχή της τεχνικής Motion JPEG (MPEG) σε συνδυασμό με έναν απλό μηχανισμό (hack it!) ελέγχου ροής (flow control) που προσαρμόζεται στις συνθήκες του εκάστοτε διαύλου επικοινωνίας. Σε κάθε `frame` αναγράφεται η μέση εκτίμηση των πλέον πρόσφατων τιμών (moving average) του ρυθμού των `frames` (fps) καθώς και της ταχύτητας μετάδοσης (bps). Ποιες είναι οι εκτιμήσεις των παραμέτρων αυτών που λαμβάνετε στο τερματικό σας ?

<sup>8</sup> Ας σημειωθεί ότι σε εφαρμογές με το εικονικό modem, για λόγους ασφαλείας, λειτουργεί μηχανισμός `timeout` ως προς τη διάρκεια της σύνδεσης, γεγονός που σε συνδυασμό με σχετικά χαμηλή ταχύτητα μετάδοσης και σχετικά μεγάλο αρχείο `jpeg` οδηγεί σε πρόωγη διακοπή της μετάδοσης εικόνας.

οπτική γωνία η οποία αλλάζει σε τυχαία χρονικά διαστήματα αν εκείνη τη χρονική περίοδο η κάμερα βρίσκεται υπό τηλε-έλεγχο (pan-tilt-zoom control) από επισκέπτη του εικονικού εργαστηρίου μέσα από διεπαφή (user interface) που διατίθεται στο μενού επιλογών του server Ιθάκη ή απευθείας από τη διεύθυνση του videoCoder του εργαστηρίου <http://ithaki.eng.auth.gr/netlab/video.html>.

Η παράμετρος CAM=PTZ μπορεί να συνοδεύεται και από την παράμετρο DIR=X όπου X ένδειξη που προσδιορίζει την επιθυμητή κατεύθυνση μεταβολής της γωνίας λήψης της κάμερας σύμφωνα με το ρεπερτόριο επιλογών L αριστερά, U πάνω, R δεξιά και D κάτω. Επιπλέον οι ενδείξεις M απομνημόνευση και C κέντρο επιτρέπουν, η μεν πρώτη την απομνημόνευση της τρέχουσας γωνίας λήψης της κάμερας, η δε δεύτερη την επαναφορά της τελευταίας απομνημονευμένης γωνίας λήψης. Ας σημειωθεί ότι ο χρόνος απόκρισης της κάμερας στις επιλογές αυτές είναι μερικά δευτερόλεπτα επειδή μεσολαβεί μηχανισμός σερβοκινητήρα για την αλλαγή του προσανατολισμού της κάμερας.

Ειδικότερα, όταν χρησιμοποιείται το εικονικό μόντεμ *Ithaki-Smart-Broadband-Modem*, η παράμετρος CAM μπορεί επίσης να έχει τη μορφή CAM=XX όπου XX αριθμός που προσδιορίζει τη θέση μιας κάμερας του Συστήματος Τηλεματικής<sup>9</sup> της Περιφερειακής Οδού Θεσσαλονίκης <http://ringroad.eng.auth.gr/ringroad/index.html>. Στην περίπτωση αυτήν μπορεί να προστεθεί και η παράμετρος SIZE=L (large) ή SIZE=S (small) ανάλογα με το επιθυμητό μέγεθος της εικόνας.

[5] Ως αφορμή για περαιτέρω ενασχόληση με σύγχρονες εφαρμογές και υπηρεσίες προστιθέμενης αξίας όπως αυτές που αντιλαμβάνονται τη θέση του τερματικού (location awareness services), η χρήση του κωδικού gps\_request\_code με τη μορφή PXXX επιτρέπει την αποστολή από τον σερβερ Ιθάκη ροής ιχνών συστήματος προσδιορισμού θέσης GPS (Global Positioning System) σύμφωνα με το σειριακό πρωτόκολλο NMEA<sup>10</sup>. Το πρωτόκολλο αυτό προβλέπει έναν αριθμό αυτοτελών ενοτήτων στοιχείων (GPS sentences) που μεταφέρουν πληροφορίες θέσης, ταχύτητας, ώρας, ταυτότητας δορυφόρων, εκτίμησης σφάλματος κλπ με τη μορφή ASCII χαρακτήρων. Κάθε ενότητα αρχίζει με μία ειδική ακολουθία χαρακτήρων που προσδιορίζει ως επικεφαλίδα (header) την ταυτότητα της ενότητας με μοναδικό τρόπο<sup>11</sup>. Όλες οι ενότητες καταλήγουν σε ζεύγος δύο χαρακτήρων <CR> και <LF> το οποίο λειτουργεί ως οριοθέτης (delimiter) διαδοχικών ενοτήτων GPS. Ένα τυπικό παράδειγμα ενότητας στοιχείων θέσης GPS έχει ως ακολούθως

\$GPGGA,045208.000,4037.6331,N,02257.5633,E,1,07,1.5,57.8,M,36.1,M,,0000\*6D

<sup>9</sup> Επειδή το Σύστημα Τηλεματικής της Περιφερειακής Οδού Θεσσαλονίκης δεν καλύπτεται με συστηματική τεχνική υποστήριξη πεδίου, ορισμένες κάμερες παραμένουν εκτός λειτουργίας για μεγάλο χρονικό διάστημα.

<sup>10</sup> National Marine Electronics Association <http://www.nmea.org>

<sup>11</sup> SiRF NMEA Reference Manual <http://ithaki.eng.auth.gr/netlab/sirf-nmea-reference-manual.pdf>



όπου \$GPGGA είναι η επικεφαλίδα, 04:52:08 η ώρα λήψης του ίχνους, 40° 37.6331' το γεωγραφικό πλάτος βόρειο και 22° 57.5633' το γεωγραφικό μήκος ανατολικό, 07 ο αριθμός των ενεργών δορυφόρων και 57.8μ το ύψος της θέσης από την επιφάνεια της θάλασσας.

Σημειώνεται ότι τα ίχνη GPS που παρέχει ο σερβερ Ιθάκη προέρχονται είτε από την πόλη της Θεσσαλονίκης με τυχαία επιλογή από προ-αποθηκευμένες διαδρομές είτε από οπουδήποτε αλλού αν ο σερβερ εκείνη τη στιγμή λαμβάνει ίχνη από εν εξελίξει διαδρομή σε πραγματικό χρόνο. Ειδικότερα στην πρώτη περίπτωση, αν η παράμετρος R=XPPPLL συνοδεύει τον κωδικό gps\_request\_code ο σερβερ Ιθάκη παραθέτει LL ίχνη GPS από την προ-αποθηκευμένη διαδρομή X με αρχή το ίχνος PPPP. Με σκοπό τη διευκόλυνση στη λήψη των ενδείξεων GPS, ο σερβερ Ιθάκη προσθέτει στην αρχή της ροής GPS το μήνυμα START ITHAKI GPS TRACKING\r\n και στο πέρας της ροής το μήνυμα STOP ITHAKI GPS TRACKING\r\n. Τέλος η παράμετρος T=AABBΓΓΔΔΕΕΖΖ, η οποία μπορεί να επαναληφθεί έως και εννέα φορές μετά τον κωδικό gps\_request\_code, επιτρέπει την απεικόνιση ίχνους με γεωγραφικό μήκος AA° BB' ΓΓ' και γεωγραφικό πλάτος ΔΔ° ΕΕ' ΖΖ' σε εικόνα .jpeg από Google Maps που επιστρέφει ο σέρβερ Ιθάκη.

[6] Η εργασία στο εικονικό εργαστήριο ενδιαφέρεται επίσης για την εξοικείωση με τη διαχείριση των σφαλμάτων μετάδοσης που προκαλεί ο θόρυβος στο κανάλι επικοινωνίας. Επειδή τα modems που λειτουργούν στο εικονικό εργαστήριο δεν έχουν απενεργοποιημένους τους εσωτερικούς μηχανισμούς διαχείρισης σφαλμάτων, ο server προκαλεί ο ίδιος ψευδοτυχαία σφάλματα, πριν τα πακέτα πληροφορίας εισέλθουν στο κανάλι επικοινωνίας. Η εισαγωγή ψευδοτυχαίου θορύβου γίνεται σε δύο περιπτώσεις μετάδοσης πληροφορίας όπως αναφέρεται στη συνέχεια.

[7] Στην πρώτη περίπτωση, ο κωδικός image\_request\_code με τη μορφή GXXXX<sup>5</sup> επιτρέπει την εισαγωγή ψευδοτυχαίων σφαλμάτων κατά την αποστολή του τρέχοντος frame από τον videoCoder του εργαστηρίου. Τα σφάλματα εισάγονται στο δεύτερο ήμισυ της παραγόμενης εικόνας έτσι ώστε να είναι δυνατή η «οπτικοποίησή» τους κατά την απεικόνιση του αρχείου .jpeg στο τερματικό.

[8] Στη δεύτερη περίπτωση τα σφάλματα εισάγονται σε μεμονωμένα πακέτα με τυχαίο τρόπο. Η εφαρμογή userApplication καλείται να αντιμετωπίσει τα σφάλματα αυτά με κατάλληλους μηχανισμούς ARQ (automatic repeat request). Για την ανάπτυξη πειραματικών μηχανισμών ARQ χρησιμοποιούνται οι κωδικοί QXXXX<sup>5</sup> και RXXXX<sup>5</sup>. Ο πρώτος αποστέλλεται από την εφαρμογή userApplication προς τον server ως ένδειξη ACK(positive acknowledgement) και ταυτόχρονα ως αίτημα για την αποστολή του επόμενου νέου πακέτου ενώ ο δεύτερος ως ένδειξη NACK(negative acknowledgement) και ταυτόχρονα ως αίτημα για επανάληψη της εκπομπής (repeat request) του ιδίου πακέτου. Τα πακέτα που αποστέλλει ο server έχουν την ακόλουθη δομή :

```
PSTART DD-MM-YYYY HH-MM-SS PC <XXXXXXXXXXXXXXXXXX> FCS PSTOP
```



όπου <XXXX> είναι κρυπτογραφημένη ακολουθία 16 χαρακτήρων η οποία προέρχεται από ψευτο-τυχαία θέση στο κείμενο με τα σχεδιαζόμενα θέματα εξετάσεων στο μάθημα Δίκτυα Υπολογιστών Ι για την εξεταστική περίοδο που ακολουθεί ☺ Ο server του εικονικού εργαστηρίου εισάγει σφάλματα σε ψευτο-τυχαίες θέσεις κατά μήκος της ακολουθίας αυτής. Το πεδίο FCS (frame check sequence) παρέχει με τη μορφή 3-ψηφίου αριθμού στο 10-δικό σύστημα το αποτέλεσμα της εφαρμογής του λογικού τελεστή XOR διαδοχικά στους 16 κρυπτογραφημένους χαρακτήρες που περιλαμβάνονται στην παραπάνω ακολουθία. Κατά τη λήψη του πακέτου, η εφαρμογή ελέγχει τα πεδία XXXX και FCS (error detection) και ανάλογα απαιτεί ή όχι την επανεκπομπή (automatic retransmission request) του ιδίου πακέτου (error correction) με τη βοήθεια των ενδείξεων NACK ή ACK αντίστοιχα.

#### *Συλλογή μετρήσεων και παρουσίαση αποτελεσμάτων*

**[Α]** Ζητείται η παρουσίαση αποτελεσμάτων τουλάχιστον από δύο συνόδους (sessions) με τον server του εικονικού εργαστηρίου που απέχουν μεταξύ τους τουλάχιστον 48 ώρες.

**[Β]** Από κάθε σύνοδο ζητείται η παρουσίαση **(i)** ενός τουλάχιστον γραφήματος G1 το οποίο εμφανίζει, για χρονική διάρκεια τουλάχιστον 4 λεπτών, το χρόνο απόκρισης του συστήματος σε milliseconds για κάθε πακέτο που έχει αποσταλεί στη διάρκεια αυτήν, **(ii)** δύο τουλάχιστον εικόνων E1 και E2 από τον videoCoder του εργαστηρίου, μία χωρίς σφάλματα και μία με σφάλματα μετάδοσης, **(iii)** μία τουλάχιστον εικόνα M1 με τουλάχιστον τέσσερα ίχνη GPS τα οποία απέχουν το καθένα από το επόμενο του τουλάχιστον τέσσερα δευτερόλεπτα όπως προκύπτουν από αντίστοιχες ενότητες \$GPXXXX ελεύθερης επιλογής από τη διαδρομή X=1 και **(iv)** ενός τουλάχιστον γραφήματος G2 που εμφανίζει για χρονική διάρκεια τουλάχιστον 4 λεπτών το χρόνο απόκρισης του συστήματος σε milliseconds για κάθε πακέτο που λαμβάνει το τερματικό επιτυχώς με τη βοήθεια του μηχανισμού ARQ σε συνθήκες ψευδοτυχαίων σφαλμάτων.

**[Γ]** Με βάση τις μετρήσεις στο σημείο (iv) της παραγράφου [Β], ζητείται **(α)** η εκτίμησή σας για την κατανομή πιθανότητας του αριθμού επανεκπομπών που καταγράψατε και **(β)** επίσης η εκτίμησή σας για την πιθανότητα σφάλματος που επικράτησε στα πειράματά σας. Η παρουσίαση του (α) θα γίνει με τη μορφή ενός γραφήματος G3 καταλλήλου τύπου (πχ bar chart) ενώ του (β) με την παράθεση του (απλού) υπολογισμού της τιμής της παραμέτρου BER (bit error rate).

**[Δ]** Τα παραπάνω αποτελέσματα ζητείται να συνοδεύονται από **(α)** σύντομα σχόλια ή παρατηρήσεις στις μετρήσεις και τα διαγράμματα που παρουσιάζετε και **(β)** μία μικρή βιβλιογραφική τεχνική αναφορά σε μηχανισμούς και πρωτόκολλα λειτουργίας των διατάξεων modem που χρησιμοποιούνται σε παρόμοιες εφαρμογές.

**[Ε]** Τα αποτελέσματα μαζί με τα σχόλιά σας, την τεχνική αναφορά και τον πηγαίο κώδικα της εφαρμογής ζητείται να υποβληθούν με τη μορφή αρχείων PDF και τις εξής απα-

ραίτητα προδιαγραφές μορφοποίησης : **(i)** τα γραφήματα G1, G2 και G3, οι εικόνες E1, E2 και M1 καθώς και ο υπολογισμός της τιμής BER από την πρώτη σύνοδο θα αποτελούν ένα διακριτό αρχείο PDF με όνομα *session1.pdf* ενώ τα ίδια στοιχεία από τη δεύτερη σύνοδο θα αποτελούν ένα δεύτερο διακριτό αρχείο με όνομα *session2.pdf*, **(ii)** στον τίτλο κάθε γραφήματος θα φαίνονται απαραίτητα η ημέρα και η ώρα που έγιναν οι μετρήσεις μαζί με τον κωδικό {echo, image, gps, arq}\_request\_code που χρησιμοποιήθηκε στη διάρκεια των μετρήσεων αυτών, **(iii)** τα κείμενα των σημείων (α) και (β) της παραγράφου [Δ] θα αποτελούν ένα τρίτο διακριτό αρχείο με όνομα *report.pdf*, **(iv)** ο πηγαίος κώδικας της εφαρμογής *userApplication* θα αποτελεί ένα τέταρτο διακριτό αρχείο με όνομα *source.pdf* και **(v)** όλες οι σελίδες των αρχείων *session1.pdf*, *session2.pdf*, *report.pdf* και *source.pdf* θα φέρουν κεφαλίδα με την ένδειξη *Δίκτυα Υπολογιστών I* και το ονοματεπώνυμο με το ΑΕΜ του/της φοιτητή/τριας που εκπόνησε<sup>12</sup> την εργασία.

Τα παραπάνω αρχεία PDF θα υποβληθούν μέσω της επιλογής μεταφόρτωσης αρχείων (*file upload*) του σέρβερ Ιθάκη στον κοινό χώρο εναπόθεσης εργασιών (*dropbox*<sup>13</sup>) του εικονικού εργαστηρίου.

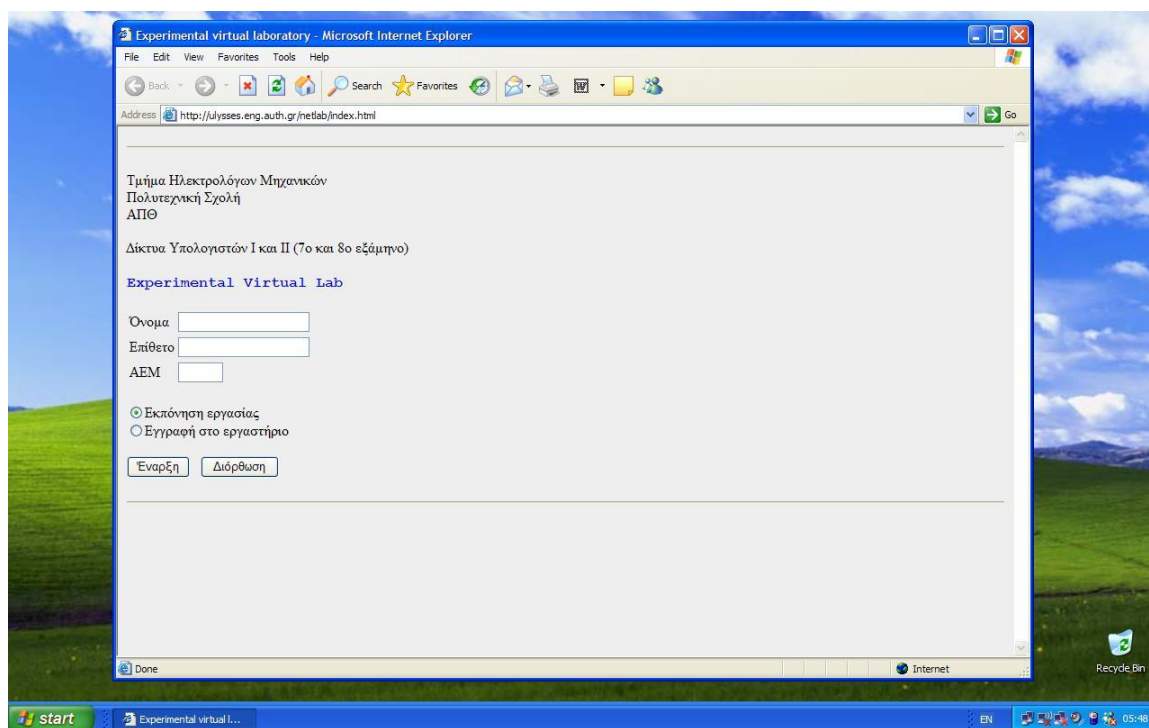
**[Z]** Η διεύθυνση επικοινωνίας με τον διδάσκοντα για το εικονικό εργαστήριο και το μάθημα Δίκτυα Υπολογιστών I (7<sup>ο</sup> εξάμηνο) είναι [mitrakos@vergina.eng.auth.gr](mailto:mitrakos@vergina.eng.auth.gr)

**[H]** Enjoy ☺

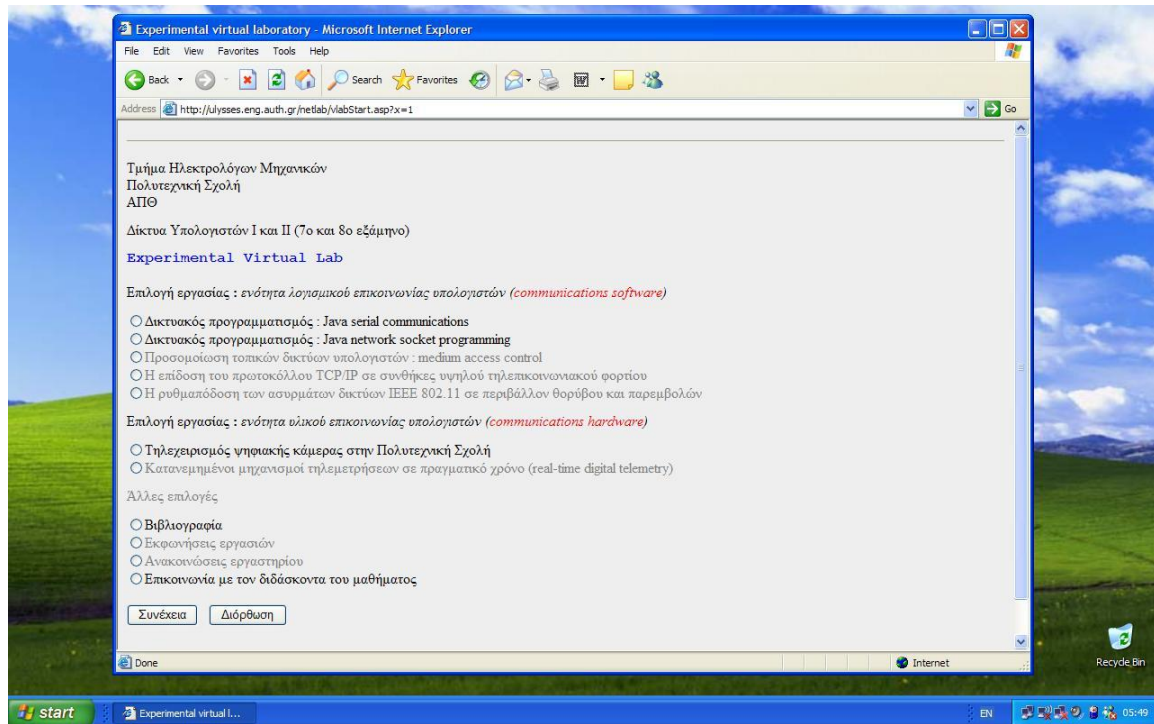
---

<sup>12</sup> Όχι το ονοματεπώνυμο και το ΑΕΜ του/της φοιτητή/τριας που υποβάλει την εργασία ☺

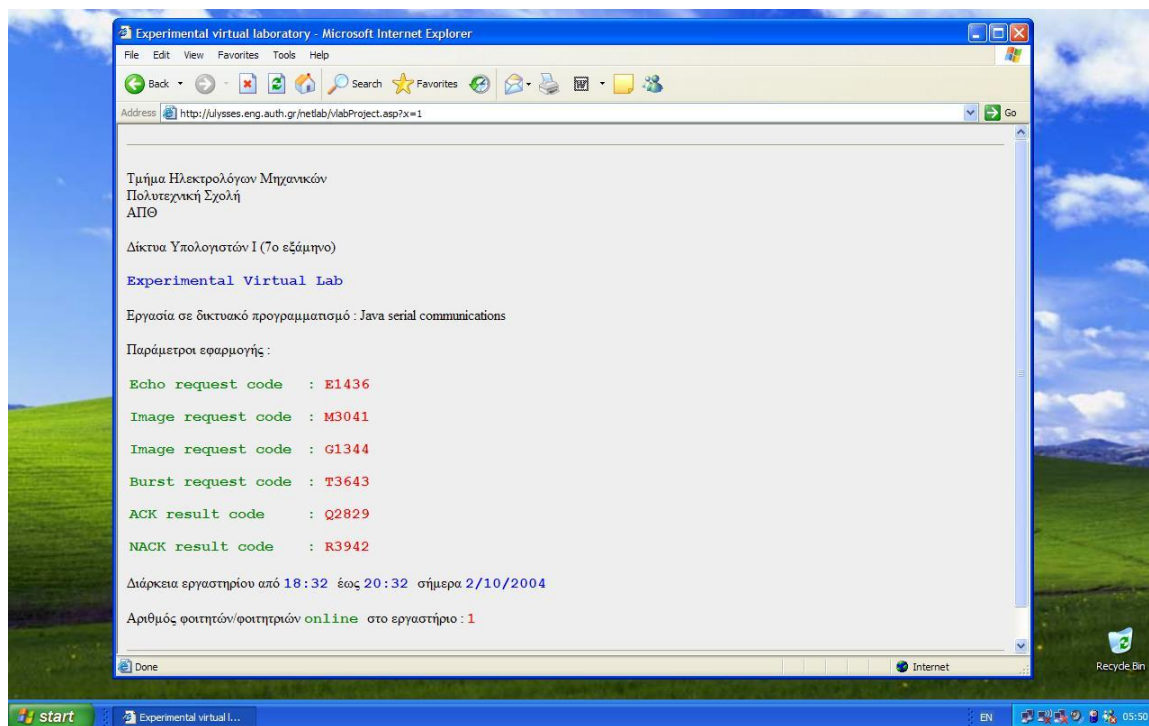
<sup>13</sup> Για λόγους ασφαλείας το μέγιστο μέγεθος κάθε αρχείου περιορίζεται στα 2 Mbytes ενώ για παρόμοιους λόγους ο μέγιστος αριθμός αρχείων περιορίζεται στα 8 αρχεία ανά ΑΕΜ.



Εικόνα 1 : Η είσοδος στο εικονικό εργαστήριο <http://ithaki.eng.auth.gr/netlab/index.html>

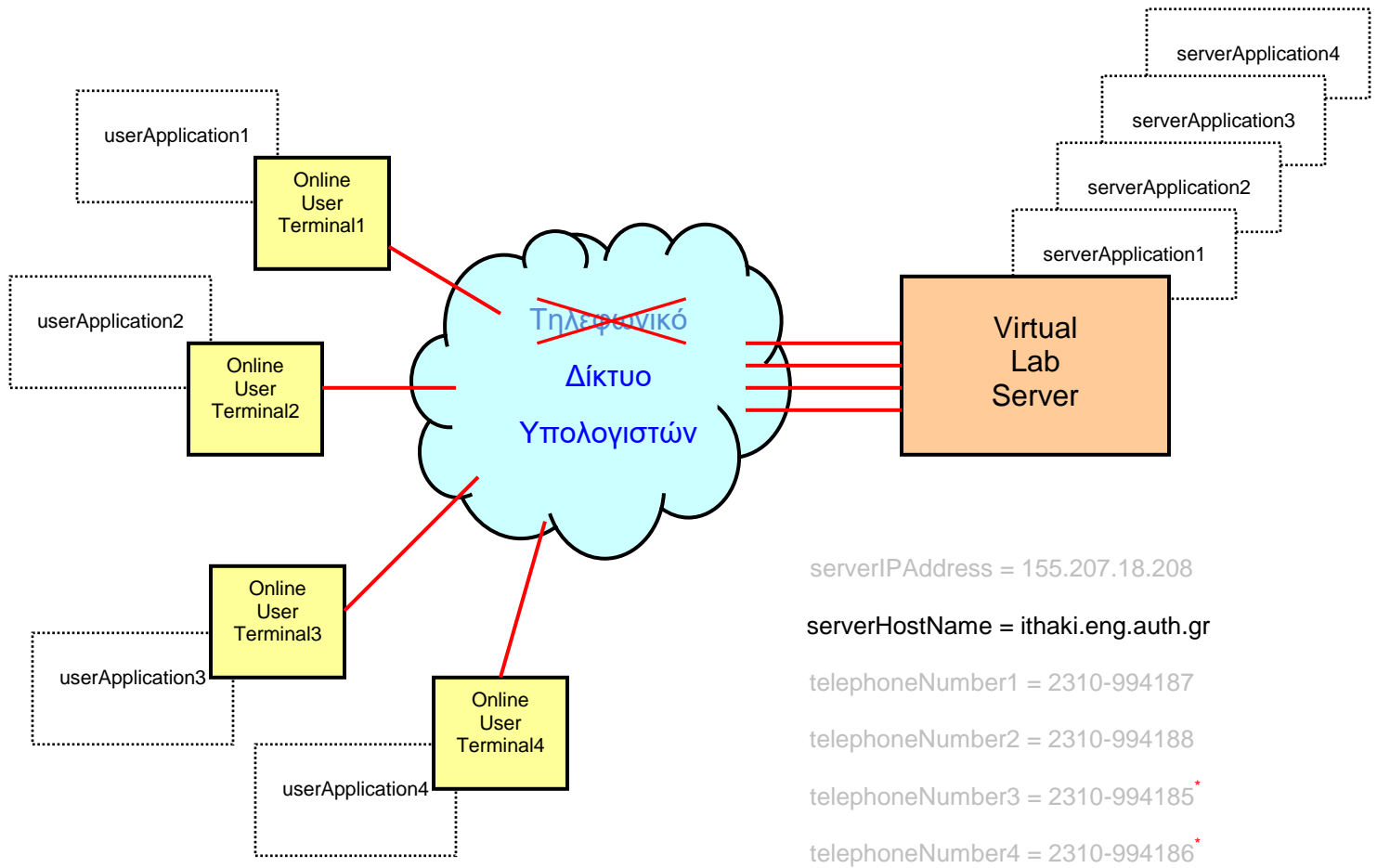


Εικόνα 2 : Η επιλογή της εργασίας προς εκπόνηση και η έναρξη μίας συνόδου (session) με τον server του εικονικού εργαστηρίου του μαθήματος



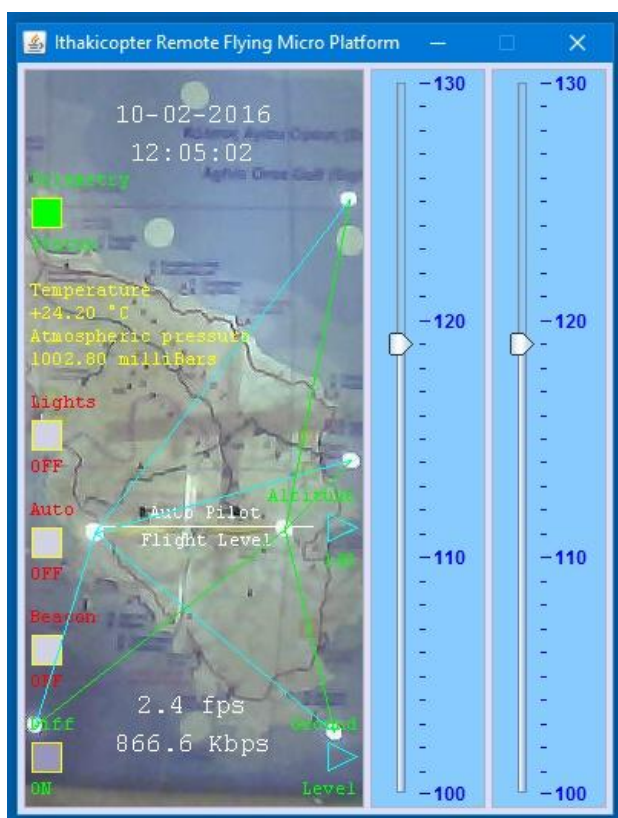
Εικόνα 3 : Ενδεικτικές τιμές των προτεινομένων παραμέτρων της εφαρμογής κατά τη διάρκεια μίας συνόδου στο εικονικό εργαστήριο

Virtual Lab URL = <http://ithaki.eng.auth.gr/netlab/index.html>



Σχήμα 1 : Συνοπτικό διάγραμμα επικοινωνιών στο εικονικό εργαστήριο

<http://ithaki.eng.auth.gr/netlab/ithakicopter.html>





## ΠΑΡΑΡΤΗΜΑ : Το λειτουργικό σύστημα Windows και η θύρα RS232

*Η βιβλιοθήκη java.comm για τη σειριακή επικοινωνία υπολογιστών*

Οι βιβλιοθήκες Java οι οποίες κατεξοχήν χρησιμοποιούνται για τη διαχείριση της θύρας RS232 σε περιβάλλον Windows είναι η `java.comm` και η `java.io`. Η πρώτη περιλαμβάνει κλάσεις (Java classes) διαχείρισης τηλεπικοινωνιακών πόρων σειριακής επικοινωνίας (serial communications resources) ενώ η δεύτερη κλάσεις διαχείρισης υπολογιστικών πόρων (computer resources).

Η βιβλιοθήκη `java.comm` δεν περιλαμβάνεται εξ αρχής στο περιβάλλον Java Development Kit JDK. Για το λόγο αυτόν πρέπει να μεταφερθεί από τον server του εικονικού εργαστηρίου μέσω της επιλογής `java_serial_communications` ή από την διεύθυνση της Sun Microsystems που εμφανίζεται στην ίδια επιλογή και στη συνέχεια να ενσωματωθεί στο περιβάλλον εργασίας JDK σύμφωνα με τις οδηγίες που εμπεριέχονται στο πακέτο `javacomm`. Για διευκόλυνση, η επιλογή `help_file` του server «οπτικοποιεί» την διαδικασία εγκατάστασης της βιβλιοθήκης `java.comm`.

Η βιβλιοθήκη `java.comm` προσφέρει σειρά διεπαφών προγραμματισμού εφαρμογών (application programming interfaces, APIs) που βοηθούν αποτελεσματικά στο χειρισμό τόσο της σειριακής θύρας του υπολογιστή όσο και του modem που ακολουθεί και δι' αυτών της επικοινωνίας με τον απομακρυσμένο υπολογιστή στο άλλο άκρο της τηλεφωνικής γραμμής. Η μέθοδος `CommPortIdentifier.getPortIdentifier()` δέχεται ως όρισμα το όνομα μίας θύρας με τη μορφή `COMx`,  $x=1, 2, 3, \dots$  την οποία εντοπίζει ως αντικείμενο (object) της γενικής κλάσης `Java CommPort`. Με τη σειρά της, η κλάση αυτή περιλαμβάνει τη μέθοδο `open()` η οποία επιτρέπει το «άνοιγμα» της σειριακής θύρας επιστρέφοντας ένα αντικείμενο της ειδικής κλάσης `SerialPort`. Στη συνέχεια, οι μέθοδοι `setSerialPortParams()` και `setFlowControlMode()` επιτρέπουν τον προγραμματισμό σειράς παραμέτρων όπως ταχύτητα επικοινωνίας σε bps, μήκος στοιχειώδους μονάδος ασύγχρονης επικοινωνίας (byte), τύπο ισοτιμίας, τύπο μηχανισμού ελέγχου ροής (flow control) κ.α. Επιπλέον οι μέθοδοι `getInputStream()` και `getOutputStream()` επιτρέπουν τη συσχέτιση της σειριακής θύρας με αντίστοιχες ροές (byte streams) εισόδου/εξόδου οι οποίες αποτελούν τελικά τους θεμελιώδεις μηχανισμούς επικοινωνίας της δικτυακής εφαρμογής (λογισμικό, software) με το modem και την τηλεφωνική γραμμή (υλικό, hardware).

Οι παραπάνω ροές εισόδου/εξόδου ως αντικείμενα του περιβάλλοντος Java κληρονομούν πλήθος μεθόδων και ιδιοτήτων από αντίστοιχες κλάσεις της βιβλιοθήκης `java.io`. Για παράδειγμα, η μέθοδος `available()` επιστρέφει ένδειξη σχετικά με το πλήθος των bytes τα οποία έχουν ενδεχομένως καταφτάσει στην είσοδο της σειριακής θύρας και είναι διαθέσιμα προς ανάγνωση από τη δικτυακή εφαρμογή. Η ανάγνωση γίνεται με τη μέθοδο `read()`. Αντίστοιχα, η μέθοδος `write()` τοποθετεί στη ροή εξόδου bytes τα οποία η δικτυακή εφαρμογή επιθυμεί να αποστείλει στην έξοδο της σειριακής θύρας.

Επιπλέον η βιβλιοθήκη `java.io` διαθέτει πλήθος άλλων κλάσεων για τη διαχείριση αρχείων. Για παράδειγμα η μέθοδος κατασκευής αντικειμένων `File()` επιτρέπει τη δημιουργία αντικειμένων τύπου `File` και τη συσχέτισή τους με αρχεία στο σκληρό δίσκο μέσω των ονομάτων τους όπως αυτά δίνονται από το `filesystem` του υπολογιστή. Οι κλάσεις `FileInputStream` και `FileOutputStream` παρέχουν μεθόδους για ανάγνωση ή εγγραφή bytes από/προς αρχεία δημιουργώντας αντίστοιχες ροές (byte streams) από/προς αυτά.

Μεταξύ των θεμελιωδών κλάσεων Java περιλαμβάνεται επίσης η κλάση `java.lang.System` η οποία διαθέτει τη μέθοδο `currentTimeMillis()`. Η μέθοδος αυτή επιστρέφει σε μία εφαρμογή την τρέχουσα ώρα του συστήματος σε milliseconds. Καλούμενη σε διαφορετικά σημεία μιας εφαρμογής κατά την εκτέλεσή της, η μέθοδος παρέχει την δυνατότητα χρονομέτρησης γεγονότων (events) όπως αυτά των αναχωρήσεων και αφίξεων πακέτων. Η εργασία που ζητείται να εκπονηθεί στη συνέχεια του εικονικού εργαστηρίου απαιτεί συχνά τη χρονομέτρηση τέτοιων γεγονότων.

#### *Η λειτουργία του modem στην (αναλογική) τηλεφωνική επικοινωνία υπολογιστών*

Η επικοινωνία διαμέσου της (αναλογικής) τηλεφωνικής γραμμής γίνεται μέσα από το περιορισμένο εύρος φάσματος των (τηλεφωνικών) ακουστικών συχνοτήτων, περίπου 4 KHz (voiceband communications). Για το λόγο αυτόν, η μετάδοση των σημάτων RS232 μέσα από την τηλεφωνική γραμμή απαιτεί κατάλληλη διαμόρφωση/αποδιαμόρφωση (modulation/demodulation). Η διαδικασία αυτή αποτελεί τη θεμελιώδη λειτουργία του γνωστού modem. Σήμερα το modem προσφέρει επιπλέον (i) μηχανισμούς ελέγχου σφάλματος (error control) για προστασία έναντι θορύβου κατά τη μετάδοση και (ii) συμπίεσης δεδομένων (data compression) για βελτίωση της ρυθμαπόδοσης (throughput) ενός διαύλου επικοινωνίας που λειτουργεί με δεδομένο ρυθμό μετάδοσης (channel rate).

Η επικοινωνία δύο υπολογιστών μέσω modem πάνω από μία τηλεφωνική γραμμή πραγματοποιείται αφού προηγουμένως τα δύο modem (i) καλέσουν το ένα (originating modem) το άλλο (answering modem) και (ii) αποκαταστήσουν τον μεταξύ τους διάυλο επικοινωνίας (telephone channel) προσδιορίζοντας σειρά παραμέτρων (carrier frequency, modulation type, error control protocol, etc) κατάλληλα. Η έναρξη του σταδίου (i) προκαλείται μετά από κατάλληλες εντολές (commands) που στέλνουν οι δικτυακές εφαρμογές των αντίστοιχων υπολογιστών στα συνδεδεμένα modem μέσω των σειριακών τους θυρών RS232. Το σύνολο AT-command-set αποτελεί σήμερα το πλέον διαδεδομένο βιομηχανικό πρότυπο ρεπερτορίου εντολών δια των οποίων το λογισμικό εφαρμογών επικοινωνιών μπορεί να ελέγξει ένα modem.

Η σύνταξη των εντολών αυτών είναι της μορφής :

ATXXX<CR>

όπου AT είναι η ακολουθία των χαρακτήρων 'A' και 'T' (για ιστορικούς λόγους από τη λέξη 'Attention'), XXX είναι ακολουθία χαρακτήρων που προσδιορίζουν την

εκάστοτε εντολή και <CR> είναι ο χαρακτήρας Carriage Return (θέση 13 στον κώδικα ASCII).

Οι χαρακτήρες που αποτελούν μία εντολή ελέγχου modem αποστέλλονται από την εφαρμογή στο modem μέσω της σειριακής θύρας RS232 ως μία ενότητα. Για παράδειγμα, η εντολή

```
ATDT2310994188<CR>
```

θέτει σε λειτουργία τη διαδικασία κλήσης (call setup) από ένα modem (originating) προς ένα δεύτερο (answering) το οποίο είναι συνδεδεμένο στο άκρο τηλεφωνικής γραμμής με αριθμό κλήσης 2310-994188. Σύμφωνα με την παραπάνω εντολή, η κλήση ('D'ial) γίνεται με τη διαδικασία αποστολής τόνων ('T'one) ακολουθώντας το πρότυπο DTMF (dual tone multiple frequencies). Αποτέλεσμα επιτυχούς εκτέλεσης της εντολής αυτής είναι η παραγωγή της ένδειξης

```
RING<CR><LF>
```

από το άλλο modem (answering) σε κάθε κωδωνισμό (ringing tone) και η αποστολή της προς τον αντίστοιχο υπολογιστή διαμέσου της σειριακής θύρας RS232, όπου <LF> είναι ο χαρακτήρας Line Feed (θέση 10 στον κώδικα ASCII). Αντιλαμβανόμενος ο υπολογιστής αυτός διαμέσου κατάλληλης εφαρμογής επικοινωνιών την ένδειξη αυτή μπορεί δια της εντολής

```
ATA<CR>
```

να ζητήσει το modem να «απαντήσει» στην κλήση. Τότε τα δύο modem εισέρχονται στο στάδιο (ii) που αναφέρθηκενωρίτερα. Η επιτυχής ολοκλήρωση του σταδίου αυτού σηματοδοτείται με την αποστολή από το κάθε modem προς τους αντίστοιχους υπολογιστές και ειδικά στις αντίστοιχες εφαρμογές των ενδείξεων

```
CONNECT XXXX <CR><LF>
```

όπου XXXX η ταχύτητα σύνδεσης σε bps.

Από το σημείο αυτό πλέον, οι δύο υπολογιστές μπορεί να αρχίσουν να επικοινωνούν σύμφωνα με το πρωτόκολλο επικοινωνίας το οποίο υλοποιούν συνεργαζόμενες οι δύο δικτυακές εφαρμογές στα δύο άκρα της τηλεφωνικής γραμμής. Εναλλακτικά, το modem που πρόκειται να δεχτεί την κλήση μπορεί να τεθεί σε κατά-σταση αναμονής (answer mode) εξ αρχής προσδιορίζοντας έναν μη μηδενικό αριθμό κωδωνισμών μετά τους οποίους το modem αποκρίνεται στην κλήση. Ο αριθμός αυτός ορίζεται σε έναν από τους ενταμιευτές (registers) που διαθέτει το modem ως εξής :

```
ATS0=2<CR>
```

όπου S0 παριστά τον ενταμιευτή '0' στον οποίο αποθηκεύεται π.χ. ο αριθμός 2

**Δίκτυα Υπολογιστών I (7ο εξάμηνο)**

Experimental Virtual Lab

```
CommPortIdentifier portId;

SerialPort asyncPort;

InputStream ip;

OutputStream op;

String message;

portId = CommPortIdentifier.getPortIdentifier("COM4");

asyncPort = (SerialPort) portId.open("vLab serial port",
    4000);
asyncPort.setSerialPortParams(57600, SerialPort.DATABITS_8,
    SerialPort.STOPBITS_1, SerialPort.PARITY_NONE);
asyncPort.setFlowControlMode(SerialPort.FLOWCONTROL_RTSCS_
    IN & SerialPort.FLOWCONTROL_RTSCS_OUT);
asyncPort.setDTR(true);
asyncPort.setRTS(false);

op = asyncPort.getOutputStream();

ip = asyncPort.getInputStream();

message="Hello there !";

op.write(message.getBytes());

ip.available();

ip.read();

ip.close();

op.close();

asyncPort.close();
```

**RS232  
9-Pin  
Female  
Connector**



Pin1 : <b>CD</b>	Pin4 : <b>DTR</b>	Pin7 : <b>RTS</b>
Pin2 : <b>TXD</b>	Pin5 : <b>GND</b>	Pin8 : <b>CTS</b>
Pin3 : <b>RXD</b>	Pin6 : <b>DSR</b>	Pin9 : <b>RI</b>