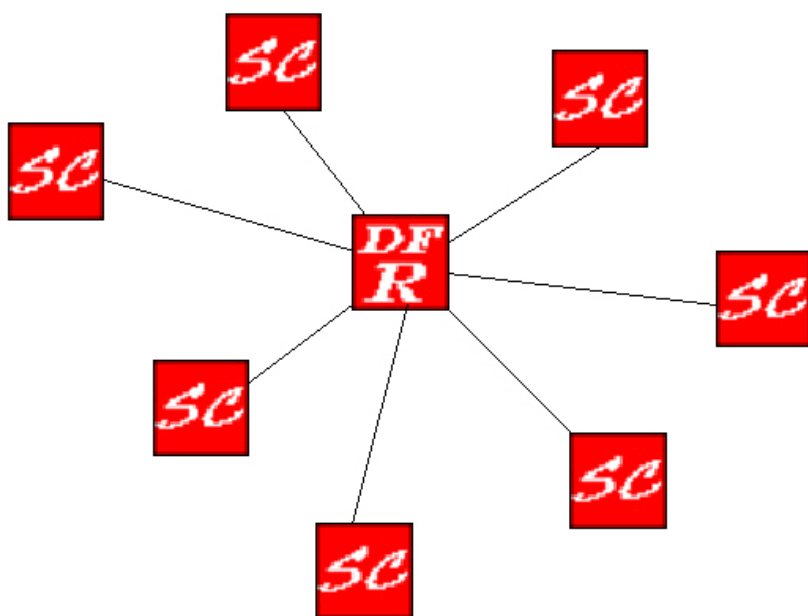




ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
& ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΜΕΤΑΔΟΣΗΣ ΠΛΗΡΟΦΟΡΙΑΣ
& ΤΕΧΝΟΛΟΓΙΑΣ ΥΛΙΚΩΝ

Ανάπτυξη Επεκτάσιμης Δικτυακής Πλατφόρμας για Συλλογή Δεδομένων από Δίκτυα Αισθητήρων Έξυπνων Αυτοκινήτων σε Πραγματικό Χρόνο

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ



ΕΜΜΑΝΟΥΗΛ ΚΟΥΚΟΥΜΙΔΗΣ

ΕΠΙΒΛΕΠΟΝΤΕΣ:

Ν. ΟΥΖΟΥΝΟΓΛΟΥ, Καθηγητής ΕΜΠ
Δρ. Α. ΑΜΔΙΤΗΣ, Ερευνητής Β' ΕΠΙΣΕΥ

ΑΘΗΝΑ 2006



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
& ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΜΕΤΑΔΟΣΗΣ ΠΛΗΡΟΦΟΡΙΑΣ
& ΤΕΧΝΟΛΟΓΙΑΣ ΥΛΙΚΩΝ

Ανάπτυξη Επεκτάσιμης Δικτυακής Πλατφόρμας για Συλλογή Δεδομένων από Δίκτυα Αισθητήρων Έξυπνων Αυτοκινήτων σε Πραγματικό Χρόνο

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΕΠΙΒΛΕΠΟΝΤΕΣ: Ν. ΟΥΖΟΥΝΟΓΛΟΥ, Καθηγητής ΕΜΠ
 Δρ. Α. ΑΜΔΙΤΗΣ, Ερευνητής Β' ΕΠΙΣΕΥ

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή:

.....
 Νικόλαος Ουζούνογλου
 Καθηγητής ΕΜΠ

.....
 Δήμητρα Κακλαμάνη
 Αν. Καθηγήτρια ΕΜΠ

.....
 Παναγιώτης Φράγκος
 Καθηγητής ΕΜΠ

Αθήνα, 2006

.....
Εμμανουήλ Γ. Κουκουμίδης

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © Εμμανουήλ Γ. Κουκουμίδης, 2006
Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΡΟΛΟΓΟΣ – PREFACE	6
ΚΕΦΑΛΑΙΟ 1: ΕΙΣΑΓΩΓΗ	
1.1 Γενικά – Αντικείμενο της εργασίας	11
1.2 Σύζευξη Δεδομένων – Ορισμός	14
1.3 Εφαρμογές Σύζευξης Δεδομένων	15
1.3.1 Στρατιωτικές εφαρμογές	15
1.3.2 Πολιτικές εφαρμογές	16
ΚΕΦΑΛΑΙΟ 2: ΕΓΧΕΙΡΙΔΙΟ ΧΡΗΣΤΗ	
2.1 Γενικά	19
2.2 Η εφαρμογή Sensor Client	19
2.3 Η εφαρμογή Data Fusion Reactor	36
2.4 Η εφαρμογή Can Logger	50
2.5 Βελτίωση της επίδοσης του συστήματος	50
ΚΕΦΑΛΑΙΟ 3: ΑΡΧΕΙΑ LOG	
3.1 Δομή των αρχείων log για συσκευές GPS	53
3.2 Δομή των αρχείων log για συσκευές radar	54
3.3 Συγχρονισμένη αναπαραγωγή των αρχείων log	56
3.4 Δημιουργία τεχνητών πειραμάτων	57
3.5 Λήψη συγχρονισμένων αρχείων log	57
ΚΕΦΑΛΑΙΟ 4: ΕΓΧΕΙΡΙΔΙΟ ΠΡΟΓΡΑΜΜΑΤΙΣΤΗ	
4.1 Πλατφορμική Όψη του συστήματος DF	60
4.2 Διαγράμματα κλάσεων	60
4.3 Επέκταση της πλατφόρμας για επιπλέον συσκευές	63
4.4 Υλοποίηση αλγορίθμων σύζευξης δεδομένων στον DFR.	64
ΚΕΦΑΛΑΙΟ 5: ΠΡΟΤΑΣΕΙΣ ΓΙΑ ΒΕΛΤΙΩΣΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ	65
ΒΙΒΛΙΟΓΡΑΦΙΑ	68

ΠΡΟΛΟΓΟΣ

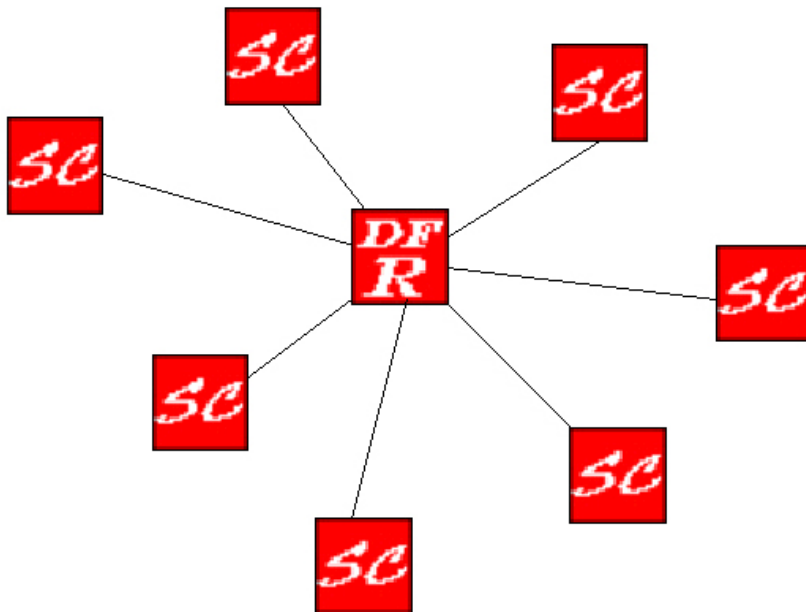
Η παρούσα διπλωματική εργασία εκπονήθηκε στη σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών σε συνεργασία με την ομάδα I-SENSE του ΕΠΙΣΕΥ. Το θέμα της αφορά στο επιστημονικό πεδίο της σύζευξης δεδομένων σε δίκτυα αισθητήρων. Πιο συγκεκριμένα, όπως λέει και ο τίτλος της διπλωματικής αυτής, αναπτύχθηκε σε Visual C++ .NET μια επεκτάσιμη πλατφόρμα για τη συλλογή δεδομένων και ανάπτυξη αλγορίθμων σύντηξης δεδομένων πραγματικού χρόνου σε δίκτυα αισθητήρων έξυπνων αυτοκινήτων. Η πλατφόρμα αυτή στην παρούσα της έκδοση (1.0) υποστηρίζει μόνο συσκευές GPS και radar αλλά μπορεί σχετικά εύκολα να επεκταθεί και για άλλες. Με την πλατφόρμα αυτή αναπτύσσεται ένα ασσύμετρο δίκτυο όπου οι διάφοροι κόμβοι, χρησιμοποιώντας κατάλληλους οδηγούς, συλλέγουν τα δεδομένα από τις συσκευές με τις οποίες είναι εξοπλισμένες τα οχήματα και τα προωθούν μέσω του πρωτοκόλλου IP σε ένα κεντρικό κόμβο που εκτελεί τον αλγόριθμο για την επεξεργασία τους (σύντηξη δεδομένων).

Επίσης, εκτός από την λειτουργία «πραγματικού χρόνου», προσφέρει και δυνατότητα για μετέπειτα αναπαραγωγή του πειράματος, είτε ολόκληρου είτε ενός μέρους του, αντλώντας τα δεδομένα από τα αντίστοιχα αρχεία log που δημιουργήθηκαν κατά τη διάρκεια του πραγματικού πειράματος. Αυτό επιτυγχάνεται αναπαράγοντας τα αρχεία log από έναν υπολογιστή (ή και πολλούς διαφορετικούς) με τρόπο συγχρονισμένο, σύμφωνα δηλαδή με τον τρόπο με τον οποίο αποκτήθηκαν. Αυτό σημαίνει ότι αν κάποιο συγκεκριμένο δεδομένο, π.χ. από το GPS του οχήματος X το πήραμε τη χρονική στιγμή t , ενώ ένα άλλο συγκεκριμένο δεδομένο από το radar του οχήματος Y τη χρονική στιγμή $t+t_0$, τότε πάλι όταν μετέπειτα αναπαράγουμε τα αντίστοιχα αρχεία log τα δεδομένα θα τα πάρουμε με το ένα να έχει την ίδια χρονική καθυστέρηση t_0 ως προς το άλλο. Έτσι, μπορούμε να αναπαράγουμε όλο το πείραμα όπως ακριβώς συνέβη με κατάλληλη μετάθεση όλων των γεγονότων/δεδομένων στο χρόνο.

Αυτό βέβαια επιτρέπει και την δημιουργία μη πραγματικών, τεχνητών πειραμάτων. Για παράδειγμα μπορούμε να δοκιμάσουμε ένα αλγόριθμο για αποφυγή (μετωπικών) συγκρούσεων διαθέτοντας ένα μόνο όχημα με το οποίο διαγράφουμε ένα κύκλο της πολυτεχνειούπολης προς τη μια κατεύθυνση, και έπειτα (ή και μετά από αυθαίρετα μεγάλο χρονικό διάστημα) προς την άλλη κατεύθυνση. Μετά το τέλος λήψης των αρχείων log, αλλάζουμε το timestamp ενός από τα δύο κατάλληλα, ώστε

το σύστημα να νομίζει ότι τα είχαμε πάρει ταυτόχρονα, και να «αναπαράξει» το επιθυμητό πλασματικό πείραμα.

Η πλατφόρμα αυτή, όπως αναφέρθηκε, αποτελείται από οντότητες δύο διαφορετικών τύπων καθώς αναπτύσσει ασύμμετρο δίκτυο τύπου client-server και όχι P2P όπως συνηθίζεται σε πολλά άλλα δίκτυα αισθητήρων. Ο λόγος είναι ότι μας ενδιαφέρει μόνο ένας οικουμενικός κόμβος (υπολογιστής) να συλλέγει όλα τα δεδομένα ώστε να μπορεί να κάνει τη σύντηξη ενώ όλοι οι άλλοι είναι επιφορτισμένοι με το απλώς να τα παίρνουνε από τις συσκευές και να τα προωθούνε σε αυτόν. Έτσι το σύστημα DF (data fusion), όπως φαίνεται και στο Σχήμα 1, συντίθεται από έναν μοναδικό, καθολικό server (DFR - Data Fusion Server) και πολλαπλούς (θεωρητικά απεριόριστους) clients (SC - Sensor Client). Οι δύο αυτές οντότητες είναι τελείως ανεξάρτητες και επικοινωνούνε μεταξύ τους, για ανταλλαγή δεδομένων, με τον μηχανισμό των sockets (Winsock).



Σχήμα 1: Ασύμμετρο δίκτυο που αναπτύσσεται από την πλατφόρμα σύντηξης δεδομένων. Κάθε SC δημιουργεί μια σύνδεση με τον DFR του συστήματος και του προωθεί τα δεδομένα. Ο DFR τα συλλέγει και τρέχει πάνω σε αυτά τον αλγόριθμο για τη σύντηξη τους.

Το σύστημα στην παρούσα έκδοση του υποστηρίζει μόνο NMEA συμβατές GPS συσκευές (μέσω παραγματικής ή εικονικής σειριακής θύρας) και το LRR-SGU

radar 2^{ης} γενιάς της Bosch (μεσω θύρας CAN). Παρόλα αυτά έχει υλοποιηθεί με τέτοιο τρόπο ώστε να μπορεί να επεκταθεί εύκολα και για άλλες συσκευές.

Η πλατφόρμα αυτή δοκιμάστηκε με επιτυχία στο πειραματικό όχημα που διαθέτει η σχολή Μηχανολόγων Μηχανικών και το I-SENSE, μαζί με άλλα βοηθητικά συμβατικά οχήματα για τη δημιουργία ενός ασύρματου δικτύου αισθητήρων. Επίσης επιδείχτηκε από την ομάδα I-SENSE σε ευρωπαϊκά ερευνητικά προγράμματα στα οποία πανεπιστήμια και εταιρείες (Bosch, Fiat, Volvo, Daimler Chrysler κτλ.) έδειξαν σημαντικό ενδιαφέρον. Έτσι, η πλατφόρμα αυτή έχει ήδη ξεκινήσει να επεκτείνεται από ερευνητές του I-SENSE ώστε να αποκτήσει επιπλέον λειτουργικότητα και να χρησιμοποιείται για ανάπτυξη αλγορίθμων σύντηξης δεδομένων.

Στο σημείο αυτό κρίνω σκόπιμο να ευχαριστήσω τον κ. Ουζούνογλου Νικόλαο, Καθηγητή του Ε.Μ.Π. καθώς και κ. Αμδίτη Άγγελο Ερευνητή Β' ΕΠΙΣΕΥ που ήταν υπεύθυνοι για την επίβλεψη της διπλωματικής μου εργασίας. Επίσης θέλω να ευχαριστήσω θερμά τον Δρ. Ηλεκτρολόγο Μηχανικό Άρη Πολυχρονόπουλο καθώς είχε τον κύριο ρόλο στην καθοδήγηση της διπλωματικής αυτής.

Τέλος κρίνω σκόπιμο να ευχαριστήσω για την πολύτιμη βοήθειά τους σε διάφορα τεχνικά θέματα τον Ηλεκτρολόγο Μηχανικό Κωνσταντίνο Κοντόπουλο και τους μεταπτυχιακούς φοιτητές Μανόλη Τσόγκα και Νικόλαο Φλούδα.

Αθήνα, 2006.

Εμμανουήλ Κουκουμίδης

PREFACE

This diploma thesis was carried out in the school of Electrical and Computer Engineering of NTUA in collaboration with the I-SENSE group of ICCS. Its subject is on the field of data fusion in sensor networks. More specifically, as stated in the title of the thesis, an expandable platform was developed for the collection of data and the development of real time data fusion algorithms in intelligent vehicles networks. This platform in its current version (1.0) supports only GPS and radar devices, but can very easily be expanded for many more types of devices. In this platform an asymmetric network is developed in which the various nodes, using the proper device drivers, collect data from the vehicles onboard devices and forward them over IP to a central node that is running the processing (data fusion) algorithm.

Furthermore, apart from the real time operation, it offers also the possibility for later reproduction of the (full or partial) experiment by reading in the data from the corresponding log files that were created during the real experiment. This is achieved by reproducing the log files by either one single computer or many distinct ones in a synchronized way that respects the timing with which each individual piece of data was initially recorded. This means that if a particular piece of data, for example from the GPS of vehicle X was received at time t , whereas another piece of data from the radar of vehicle Y at time $t+t_0$, then when we later reproduce the corresponding log files we will get the data with the later having the proper t_0 delay in comparison to the former. In this way we can reproduce the whole experiment exactly as it happened by shifting all events/data in future appropriately.

This of course also enables the creation of non-real, fake experiments. For example we can test an algorithm for head-on collisions avoidance by having just one vehicle, with which we first go around the campus in one direction and then at a later time in future towards the opposite one. After receiving these two log files, we change the timestamp of one of them, so that our system/platform thinks that they were received simultaneously and “reproduces” the desired fake experiment.

This platform, as already mentioned, consists of entities of two different types as an asymmetric client-server network is developed and not a P2P¹ one. The reason for that is that we are interested in giving only one unique node the privilege of

¹ P2P networks are generally very common in sensor networks.

receiving all the data so that it can fuse them, whereas all the other nodes are just assigned the task of getting the data from their attached device and forwarding it to this unique node. Therefore the DF (Data Fusion) system consists of a unique global server (DFR - Data Fusion Server) and many (theoretically infinite) clients (SCs – Sensor Clients). These two entities are totally independent and communicate with each other, for the exchange of data, using the sockets mechanism (Winsock).

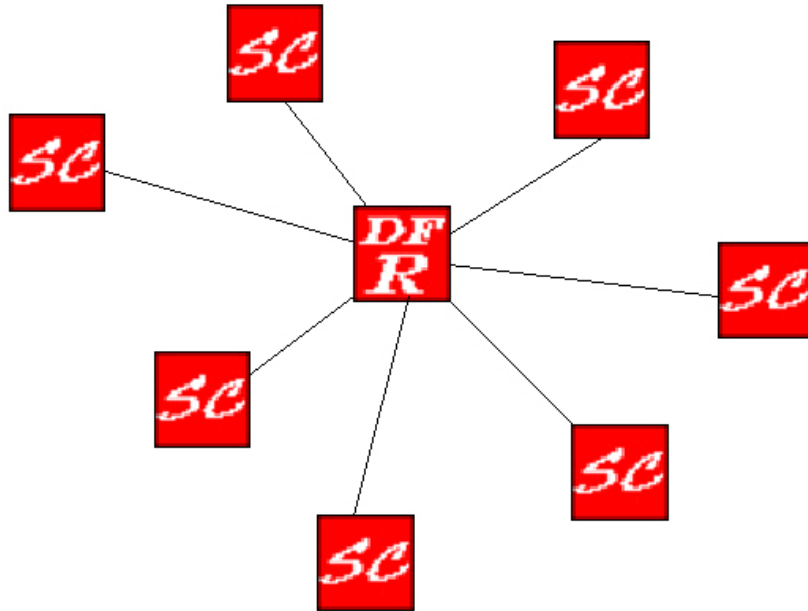


Figure 1.1: Assymmetric network that is developed by the data fusion platform. Every SC creates a connection with the systems' DFR and forwards to it its data. The DFR collects them and operates on them by running a data fusion algorithm.

The system in each current version supports only NMEA-compliant GPS devices (via real or virtual serial port) and the LRR-SGU 2nd generation radar of Bosch (via CAN port). Nevertheless, it has been developed in such a way that facilitates the integration of more devices.

This platform (Version 1.0) was tested successfully in a testbed consisting of the experimental vehicle of the school of Mechanical Engineers and I-SENSE and other standard vehicles for the creation of a wireless vehicular sensor network. It was also presented by the I-SENSE group in European research programs, in which universities and companies (Bosch, Fiat, Volvo, Daimler Chrysler e.t.c.) showed great

interest. As a result the platform is now being extended from I-SENSE researchers and used for the testing of data fusion algorithms.

At this point, I would like to thank NTUA Prof. Nikolaos Uzunoglu as well as the I-SENSE Researcher-B Dr. Angelos Amditis who were the supervisors of my thesis. Furthermore, I would like to thank very much the Electrical Engineer Dr. Aris Polychronopoulos as he had the main role in the guidance of this thesis.

Last but not least I would like to thank for their useful help in various technical issues the Electrical Engineer Kostis Contopoulos and the post-graduate PhD students Manolis Tsogas and Nikolaos Floudas.

Athens, 2006.
Emmanouil Koukoumidis

ΚΕΦΑΛΑΙΟ 1: ΕΙΣΑΓΩΓΗ

1.1 Γενικά – Αντικείμενο της εργασίας

Το βασικό αντικείμενο της διπλωματικής εργασίας αυτής είναι κυρίως η υλοποίηση μιας δικτυακής πλατφόρμας, η οποία να επιτρέπει την ανάπτυξη αλγορίθμων για σύντηξη δεδομένων σε δίκτυα αισθητήρων.

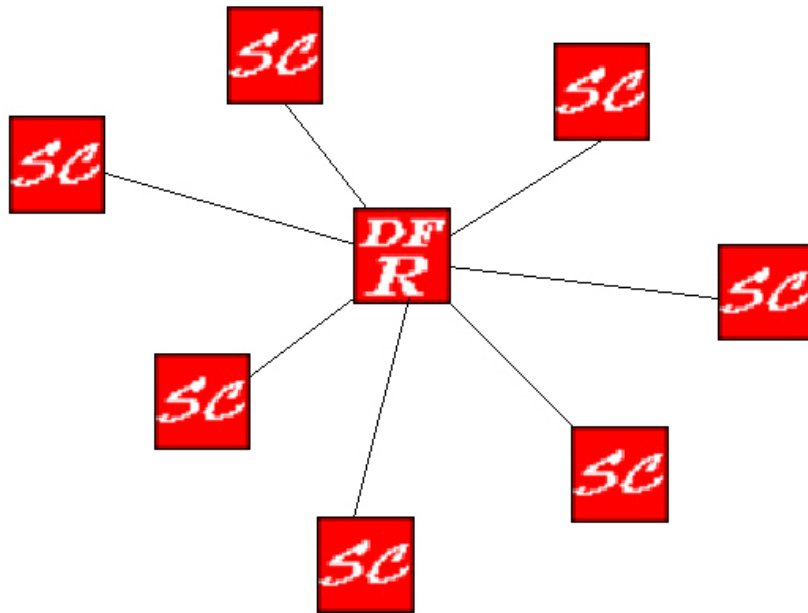
Η πλατφόρμα αυτή χρησιμοποιώντας κατάλληλους οδηγούς συλλέγει τα δεδομένα από τις συσκευές με τις οποίες είναι εξοπλισμένες τα οχήματα και τα προωθεί μέσω του πρωτοκόλλου IP σε ένα κεντρικό κόμβο που εκτελεί τον αλγόριθμο για την επεξεργασία τους (σύντηξη δεδομένων).

Επίσης, εκτός από την λειτουργία «πραγματικού χρόνου», προσφέρει και δυνατότητα για μετέπειτα αναπαραγωγή του πειράματος, είτε ολόκληρου είτε ενός μέρους του, αντλώντας τα δεδομένα από τα αντίστοιχα αρχεία log που δημιουργήθηκαν κατά τη διάρκεια του πραγματικού πειράματος. Αυτό επιτυγχάνεται αναπαράγοντας τα αρχεία log από έναν υπολογιστή (ή και πολλούς διαφορετικούς) με τρόπο συγχρονισμένο, σύμφωνα δηλαδή με τον τρόπο με τον οποίο αποκτήθηκαν. Αυτό σημαίνει ότι αν κάποιο συγκεκριμένο δεδομένο, π.χ. από το GPS του οχήματος X το πήραμε τη χρονική στιγμή t , ενώ ένα άλλο συγκεκριμένο δεδομένο από το radar του οχήματος Y τη χρονική στιγμή $t+t_0$, τότε πάλι όταν μετέπειτα αναπαράγουμε τα αντίστοιχα αρχεία log τα δεδομένα θα τα πάρουμε με το ένα να έχει την ίδια χρονική καθυστέρηση t_0 ως προς το άλλο. Έτσι, μπορούμε να αναπαράγουμε όλο το πείραμα όπως ακριβώς συνέβη με κατάλληλη μετάθεση όλων των γεγονότων/δεδομένων στο χρόνο.

Αυτό βέβαια επιτρέπει και την δημιουργία μη πραγματικών, τεχνητών πειραμάτων. Για παράδειγμα μπορούμε να δοκιμάσουμε ένα αλγόριθμο για αποφυγή (μετωπικών) συγκρούσεων διαθέτοντας ένα μόνο όχημα με το οποίο διαγράφουμε ένα κύκλο της πολυτεχνειούπολης προς τη μια κατεύθυνση, και έπειτα (ή και μετά από αυθαίρετα μεγάλο χρονικό διάστημα) προς την άλλη κατεύθυνση. Μετά το τέλος λήψης των αρχείων log, αλλάζουμε το timestamp ενός από τα δύο κατάλληλα, ώστε το σύστημα να νομίζει ότι τα είχαμε πάρει ταυτόχρονα, και να «αναπαράξει» το επιθυμητό πλασματικό πείραμα.

Η πλατφόρμα αυτή, όπως αναφέρθηκε, αποτελείται από οντότητες δύο διαφορετικών τύπων καθώς αναπτύσσει ασύμμετρο δίκτυο τύπου client-server και όχι

P2P όπως συνηθίζεται σε πολλά άλλα δίκτυα αισθητήρων. Ο λόγος είναι ότι μας ενδιαφέρει μόνο ένας οικουμενικός κόμβος (υπολογιστής) να συλλέγει όλα τα δεδομένα ώστε να μπορεί να κάνει τη σύντηξη ενώ όλοι οι άλλοι είναι επιφορτισμένοι με το απλώς να τα παίρνουνε από τις συσκευές και να τα προωθούνε σε αυτόν. Έτσι το σύστημα DF (data fusion), όπως φαίνεται και στο Σχήμα 1.1, συντίθεται από έναν μοναδικό, καθολικό server (DFR - Data Fusion Server) και πολλαπλούς (θεωρητικά απεριόριστους) clients (SC - Sensor Client). Οι δύο αυτές οντότητες είναι τελείως ανεξάρτητες και επικοινωνούνε μεταξύ τους, για ανταλλαγή δεδομένων, με τον μηχανισμό των sockets (Winsock).



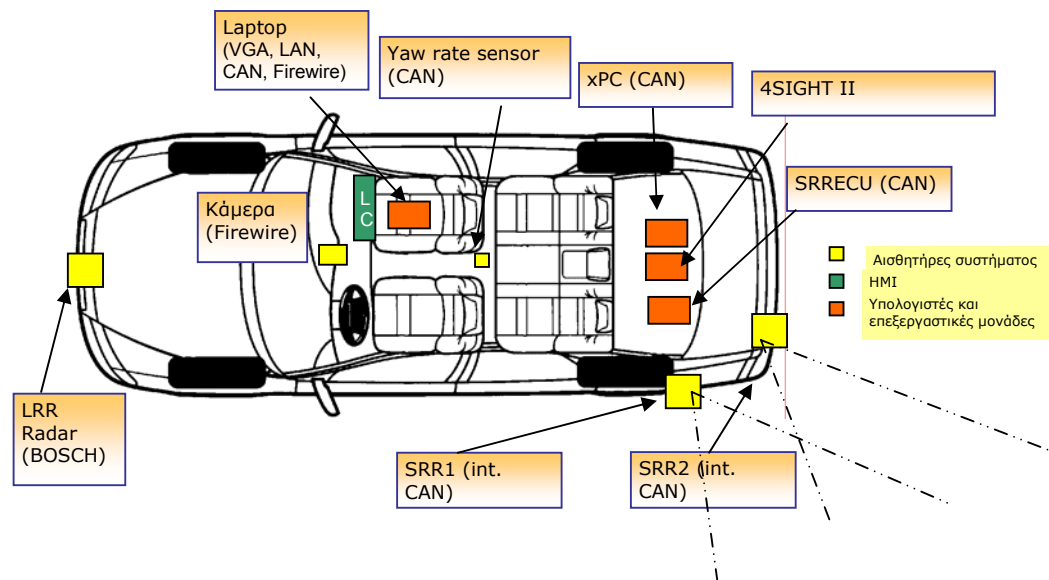
Σχήμα 1.1: Ασύμμετρο δίκτυο που αναπτύσσεται από την πλατφόρμα σύντηξης δεδομένων. Κάθε SC δημιουργεί μια σύνδεση με τον DFR του συστήματος και του προωθεί τα δεδομένα. Ο DFR τα συλλέγει και τρέχει πάνω σε αυτά τον αλγόριθμο για τη σύντηξη τους.

Το σύστημα στην παρούσα έκδοση του υποστηρίζει μόνο NMEA συμβατές GPS συσκευές (μέσω παραγματικής ή εικονικής σειριακής θύρας) και το LRR-SGU radar 2^{ης} γενιάς της Bosch (μέσω θύρας CAN). Παρόλα αυτά έχει υλοποιηθεί με τέτοιο τρόπο ώστε να μπορεί να επεκταθεί εύκολα και για άλλες συσκευές.

Στη συνέχεια αυτού του κεφαλαίου υπάρχει ο ορισμός της Σύζευξης Δεδομένων (Data Fusion) καθώς οι σημαντικότερες εφαρμογές DF με έμφαση στην

περίπτωση του συστήματος υποστήριξης οδήγησης, που εντάσσεται στα κύρια ερευνητικά ενδιαφέροντα της ομάδας I-SENSE του ΕΠΙΣΕΥ μέσα στα πλαίσια της οποίας εκπονήθηκε η διπλωματική αυτή. Στα επόμενα κεφάλαια ακολουθούν ένα εγχειρίδιο χρήστη για την επεξήγηση των παρεχόμενων λειτουργιών του συστήματος και μετέπειτα ένα εγχειρίδιο προγραμματιστή όπου επεξηγούνται χρήσιμες πληροφορίες για την υλοποίηση, που θα μπορούσαν να βοηθήσουν στην μετέπειτα επέκταση του, ώστε να υποστηρίζει και άλλες συσκευές (πχ. κάμερα) ή να αποκτήσει επιπλέον λειτουργικότητα .

Η πλατφόρμα αυτή υλοποιήθηκε σε Visual C++ .NET στο IDE του Visual Studio .NET 2005 Professional Edition.



Σχήμα 1.2: Ερευνητικό όχημα του I-SENSE.

1.2 Σύζευξη Δεδομένων – Ορισμός

Ο σκοπός της Σύζευξης Δεδομένων (DF, Data Fusion) από πολλαπλούς αισθητήρες είναι ο συνδυασμός αυτών ώστε να εξαχθούν συμπεράσματα τα οποία ενδεχομένως δεν θα ήταν δυνατόν να εξαχθούν από μόνο έναν αισθητήρα ή θα εξαγόταν με σαφώς μικρότερη ακρίβεια. Οι εφαρμογές του DF περιλαμβάνουν στρατιωτικές και πολιτικές εφαρμογές. Τα δεδομένα από διαφορετικούς αισθητήρες και τύπους αισθητήρων συνδυάζονται κάνοντας χρήση τεχνικών από διάφορους τομείς όπως: επεξεργασία σήματος, στατιστική, τεχνητή νοημοσύνη, αναγνώριση προτύπου, γνωστική ψυχολογία και θεωρία πληροφορίας. Τα δεδομένα εισόδου από τους αισθητήρες μπορεί να περιλαμβάνουν παραμετρικά δεδομένα θέσης, όπως δεδομένα γωνιών (π.χ. αζιμούθιο, elevation και συντεταγμένες εικόνας), πληροφορίες απόστασης ή ταχύτητας και δεδομένα που σχετίζονται με την ταυτότητα του αντικειμένου (π.χ. είτε πραγματική αναγνώριση ταυτότητας από έναν αισθητήρα είτε παραμετρικά δεδομένα που μπορούν να συσχετιστούν με ταυτότητα, όπως η ενεργός διατομή ραντάρ ή φασματικά δεδομένα).

Οι διαδικασίες DF διακρίνονται σε επιμέρους κατηγορίες ανάλογα με τον τρόπο επεξεργασίας των δεδομένων μεταξύ των περιφερειακών αισθητήρων και του κεντρικού επιπέδου επεξεργασίας. Τα συστήματα DF διακρίνονται, με βάση τον τρόπο που είναι συνδεδεμένοι οι αισθητήρες μεταξύ τους και τον τρόπο που γίνεται η βασική επεξεργασία, σε συστήματα DF κεντρικού επιπέδου (central level fusion) που αποτελούν και το τύπο συστήματος που υλοποιήθηκε σε αυτή τη διπλωματική, σε συστήματα DF επιπέδου αισθητήρα (sensor level fusion) και σε συστήματα Υβριδικής Σύζευξης (Hybrid Fusion).

Η συνεισφορά της Σύζευξης Δεδομένων είναι εξαιρετικά σημαντική τόσο στην περίπτωση της σύζευξης αισθητήρων διαφορετικού τύπου, ιδίου τύπου ή αισθητήρων που βρίσκονται σε απομακρυσμένες μεταξύ τους περιοχές. Σε κάθε περίπτωση ένα σύστημα DF εκμεταλλεύεται πλήρως τα αποτελέσματα όλων των αισθητήρων που το αποτελούν ελαχιστοποιώντας τις αδυναμίες που ενδέχεται να προκύψουν σε έναν ή περισσότερους αισθητήρες του συστήματος, αφού οι υπόλοιποι αισθητήρες θα συμβάλλουν στη σωστή ενημέρωση του συστήματος, μιας και θα ήταν μάλλον απίθανο η αδυναμία παρατήρησης να συμβεί ταυτόχρονα στο σύνολο των αισθητήρων. Οι παράγοντες που ενδέχεται να δημιουργήσουν αδυναμία παρατήρησης

σε έναν αισθητήρα είναι κακή επίδοση υπό συγκεκριμένες συνθήκες, ύπαρξη αντιμέτρων, κακή επικοινωνία μεταξύ των αισθητήρων και του κεντρικού επιπέδου, βλάβες κλπ. Ένα σύστημα DF πολλαπλών αισθητήρων εξασφαλίζει μεγάλη αξιοπιστία σε τέτοιου είδους προβλήματα, λόγω ακριβώς της ύπαρξης πολλών αισθητήρων.

Τα συστήματα Σύζευξης Δεδομένων όπως θα φανεί στην επόμενη παράγραφο (1.3) έχουν ένα ευρύ πεδίο εφαρμογών.

1.3 Εφαρμογές Σύζευξης Δεδομένων

Οι εφαρμογές του DF εκτείνονται σε ένα ευρύ πεδίο περιλαμβάνοντας στρατιωτικές και μη στρατιωτικές εφαρμογές. Οι στρατιωτικές εφαρμογές περιλαμβάνουν επιτήρηση ωκεανών, άμυνα αέρος-αέρος και εδάφους-αέρος, αναγνώριση πεδίου μάχης, επιτήρηση και απόκτηση στόχου και στρατηγική προειδοποίηση και άμυνα. Οι μη στρατιωτικές εφαρμογές περιλαμβάνουν εφαρμογή νόμου, αυτοματοποιημένη παρακολούθηση (monitoring) εγκαταστάσεων, ιατρική διάγνωση, ρομποτική, μετεωρολογία και αεροναυτικές εφαρμογές.

1.3.1 Στρατιωτικές εφαρμογές

Η επιτήρηση ωκεανών έχει ως σκοπό τον εντοπισμό, την ιχνηλασία και την αναγνώριση στόχων, γεγονότων και δραστηριοτήτων. Παραδείγματα εχθρικών δραστηριοτήτων ενδιαφέροντος είναι η εξαπόλυση μιας τορπίλης από ένα υποβρύχιο, η έξοδος από την επιφάνεια της θάλασσας ενός υποθαλάσσια εκτοξευμένου πυραύλου και η επικοινωνία ενός υποβρυχίου με άλλα οχήματα. Η επιτήρηση ωκεανών περιλαμβάνει υποθαλάσσιους, επιφανειακούς και εναέριους στόχους σε μια τεράστια έκταση. Τα παρατηρούμενα δεδομένα εκτείνονται σε ολόκληρο το φυσικό πεδίο, από ακουστικά δεδομένα που αποκτώνται από πλωτήρες ανίχνευσης ήχων (sonobuoys) έως πυρηνικά και μη πυρηνικά σωματίδια.

Η άμυνα αέρος-αέρος και εδάφους-αέρος αποσκοπεί στον εντοπισμό, στην ιχνηλασία και στην αναγνώριση αεροσκαφών σε περιοχές όπου ενδέχεται να υπάρχει απόκρυψη ή ανάπτυξη οπλικών συστημάτων. Τα πρωτεύοντα φαινόμενα παρατήρησης περιλαμβάνουν ηλεκτρομαγνητική ακτινοβολία (υπέρυθρες, ορατές και ράδιο - συχνότητες) χρησιμοποιώντας ενεργητικούς ή παθητικούς αισθητήρες όπως

ραντάρ, δέκτες ESM (Electronic Support Measures), κάμερες IR, ραντάρ laser και ηλεκτρο-οπτικούς αισθητήρες (TV). Οι χώροι επιτήρησης κυμαίνονται από εκατοντάδες κυβικά μίλια για στρατηγικές εφαρμογές όπως η εναέρια άμυνα μιας χώρας και σε ακτίνα μερικών μιλίων για την άμυνα ενός μόνο αεροσκάφους. Μια σχετική μη στρατιωτική εφαρμογή είναι η αναγνώριση των εισερχομένων αεροσκαφών σε ένα εμπορικό αεροδρόμιο.

Η αναγνώριση πεδίου μάχης στοχεύει στην ανίχνευση και την αναγνώριση πιθανών στόχων εδάφους (μεταφορείς, πυροβόλα ή πομπούς) ώστε να γίνουν γνωστές οι δυνατότητες, η τακτική και η στρατηγική του εχθρού. Η ηλεκτρομαγνητική ακτινοβολία είναι αυτό που κυρίως παρατηρείται, συμπεριλαμβανομένων της εκπομπής IR από εχθρικές μηχανές, τις εκπομπές RF από επικοινωνιακούς ραδιοπομπούς, ραδιόφαρους κατεύθυνσης (beacons), ραντάρ, τις φωτογραφίες της περιοχής ενδιαφέροντος και τους δέκτες ESM. Η αναγνώριση πεδίου μάχης έχει ως σκοπό τη γνώση της σύνθεσης της εχθρικής παράταξης μάχης και τον καθορισμό της σημασίας της και του σκοπού της. Η περιοχή ενδιαφέροντος κυμαίνεται από δεκάδες ως εκατοντάδες τετραγωνικά μίλια και τείνει να αυξάνεται με τη δημιουργία σύγχρονων όπλων.

Η αποστολή της στρατηγικής προειδοποίησης και άμυνας είναι διπλή: (1) να ανιχνεύσει ενδείξεις επικείμενων πολεμικών ενεργειών και στην χειρότερη περίπτωση, (2) ακολουθεί η ανίχνευση και ιχνηλασία βαλλιστικών πυραύλων και κεφαλών. Απαιτείται ένα εκτεταμένο σύστημα αισθητήρων με δορυφόρους και αεροσκάφη. Οι κυρίως παρατηρήσεις είναι η ανίχνευση ηλεκτρομαγνητικής ακτινοβολίας, πυρηνικής ακτινοβολίας κλπ. Η ανίχνευση στρατηγικών ενδείξεων και προειδοποιήσεων περιλαμβάνει παρατηρήσεις εχθρικών στρατιωτικών κινήσεων όπως επικοινωνίες, διασπορά δυνάμεων, κατάσταση συναγερμού και ακόμα μη στρατιωτικές πολιτικές δραστηριότητες.

1.3.2 Πολιτικές εφαρμογές

Η επιβολή του νόμου απαιτεί εφαρμογές παρόμοιες με τη στρατιωτική αναγνώριση και επιτήρηση. Για παράδειγμα η καταστολή των ναρκωτικών ενδέχεται να περιλαμβάνει περιπολία συνόρων. Η χρήση αντίμετρων από τους κακοποιούς και η προσπάθεια αποφυγής τους είναι ένα ακόμη κοινό σημείο με τις στρατιωτικές εφαρμογές.

Επίσης συστήματα DF χρησιμοποιούνται για την επιτήρηση της γης ώστε να αναγνωριστούν και να παρακολουθηθούν σοδειές, ο καιρός, τα αποθέματα ορυκτών, οι περιβαλλοντικές συνθήκες και τυχόν καταστροφικές απειλές (όπως πετρελαιοκηλίδες και διαρροές ραδιενέργειας). Ειδικοί αισθητήρες όπως τα ραντάρ συνθετικού παραθύρου (SAR) επιτρέπουν ενεργή επιτήρηση, ενώ παθητικοί αισθητήρες μπορούν να παρακολουθήσουν το ορατό και το IR φάσμα.

Συστήματα DF έχουν επίσης αναπτυχθεί για την παρακολούθηση και τον έλεγχο σύνθετων εγκαταστάσεων και κατασκευαστικών διαδικασιών. Συγκεκριμένα πολύπλοκα συστήματα, όπως πυρηνικά εργοστάσια και σύγχρονα αεροσκάφη, απαιτούν παρακολούθηση πέρα από αυτή που μπορεί να ασκήσει ένας άνθρωπος. Για το λόγο αυτό χρειάζεται ημιαυτόματη παρακολούθηση (monitoring) ώστε να διασφαλιστεί ότι το σύστημα συνεχίζει να λειτουργεί σωστά. Δεδομένα από πολλαπλούς αισθητήρες ελέγχονται ώστε να καταγραφεί η «υγεία» ενός συστήματος. Επίσης, έχει αναπτυχθεί ένας αριθμός από συστήματα DF για αυτοματοποιημένη διάγνωση βλάβης.

Άλλα παραδείγματα DF είναι οι τεχνικές που αναπτύχθηκαν για την ιατρική διάγνωση. Σε ένα βασικό επίπεδο, για να διαγνωστούν κοινές ασθένειες χρειάζεται η χρήση αφής, οπτικής εξέτασης, ήχου και τα συμπτώματα που περιγράφει ο ίδιος ο ασθενής. Πιο περίπλοκα προβλήματα ενδέχεται να περιλαμβάνουν δεδομένα πολλαπλών αισθητήρων (π.χ. ακτινογραφίες, μαγνητικές τομογραφίες, χημικά και βιολογικά τεστ, υπέρηχους). Έχουν γίνει πειράματα ώστε να αναπτυχθούν έμπειρα συστήματα υπολογιστών που θα μιμούνται τη διαδικασία της ιατρικής διάγνωσης.

Ένα άλλο παράδειγμα εφαρμογών DF είναι η ενσωμάτωση δεδομένων πολλαπλών αισθητήρων για ρομποτικές εφαρμογές. Βιομηχανικά ρομπότ χρησιμοποιούν τεχνικές αναγνώρισης σχήματος και εξαγωγής συμπερασμάτων ώστε να αναγνωρίζουν τρισδιάστατα αντικείμενα, να καθορίζουν τον προσανατολισμό τους και να οδηγούν προσαρτήματα τους ώστε να χειρίζονται αντικείμενα.

Οι τομείς της μετεωρολογίας και της αεροναυτικής είναι ακόμα δύο πεδία εφαρμογής συστημάτων DF λόγω της πολυπλοκότητας των φαινομένων που παρατηρούνται στους τομείς αυτούς και της αξιοπιστίας που προσφέρουν τα συστήματα DF.

Τέλος μια σημαντική εφαρμογή που μπορούν να βρουν ακόμα τα συστήματα σύντηξης δεδομένων, είναι η ενεργητική ασφάλεια οχημάτων. Ένα υπολογιστικό σύστημα με το οποίο μπορεί να είναι εξοπλισμένο ένα όχημα μπορεί να «αισθάνεται»

το περιβάλλον του με χρήση κατάλληλων αισθητήρων, ή και να ανταλλάσει πληροφορίες με άλλα γειτονικά συστήματα άλλων οχημάτων ώστε να μπορεί να πληροφορεί τα άλλα για ενδεχόμενους κινδύνους (όπως πχ. λακούβες ή αυξημένη ολισθηρότητα) καθώς και για αποφυγή σύγκρουσης τους. Επιπλέον με συνδιασμό δεδομένων από το ιστορικό μετακινήσεων του συγκεκριμένου οχήματος αλλά και από αυτά άλλων οχημάτων που είναι πρόθυμα να τα μοιραστούνε, και με χρήση τεχνικών data mining μπορεί να γίνει εκτίμηση της κυκλοφοριακής συμφόρησης ώστε το σύστημα να προτείνει εναλλακτικές διαδρομές.

Αυτά τα παραδείγματα δεν εξαντλούν το πεδίο των προβλημάτων για τα οποία οι τεχνικές DF είναι εφαρμόσιμες. Παρόλα αυτά παρέχουν μια ένδειξη της μεγάλης έκτασης τέτοιου είδους εφαρμογών.

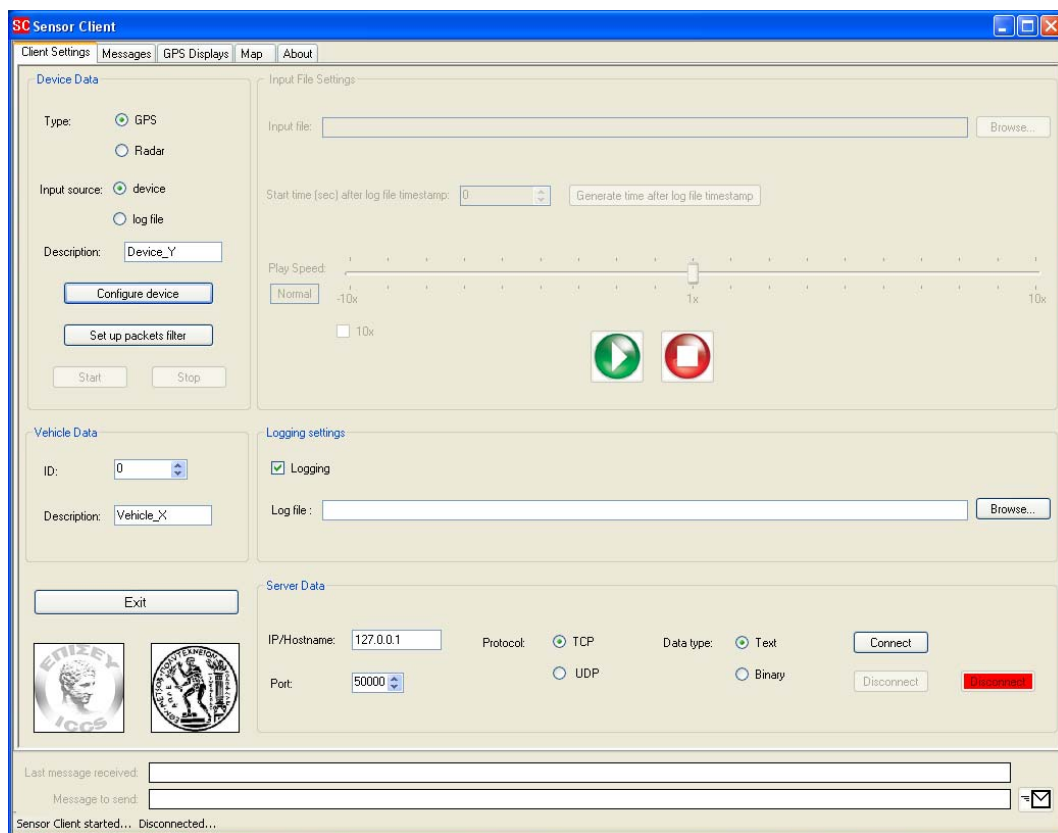
ΚΕΦΑΛΑΙΟ 2: ΕΓΧΕΙΡΙΔΙΟ ΧΡΗΣΤΗ

2.1 Γενικά

Στο κεφάλαιο παρουσιάζονται οι λειτουργίες που παρέχει το σύστημα. Πρώτα παρουσιάζεται η εφαρμογή Sensor Client και στη συνέχεια η εφαρμογή Data Fusion Reactor.

2.2 Η εφαρμογή Sensor Client

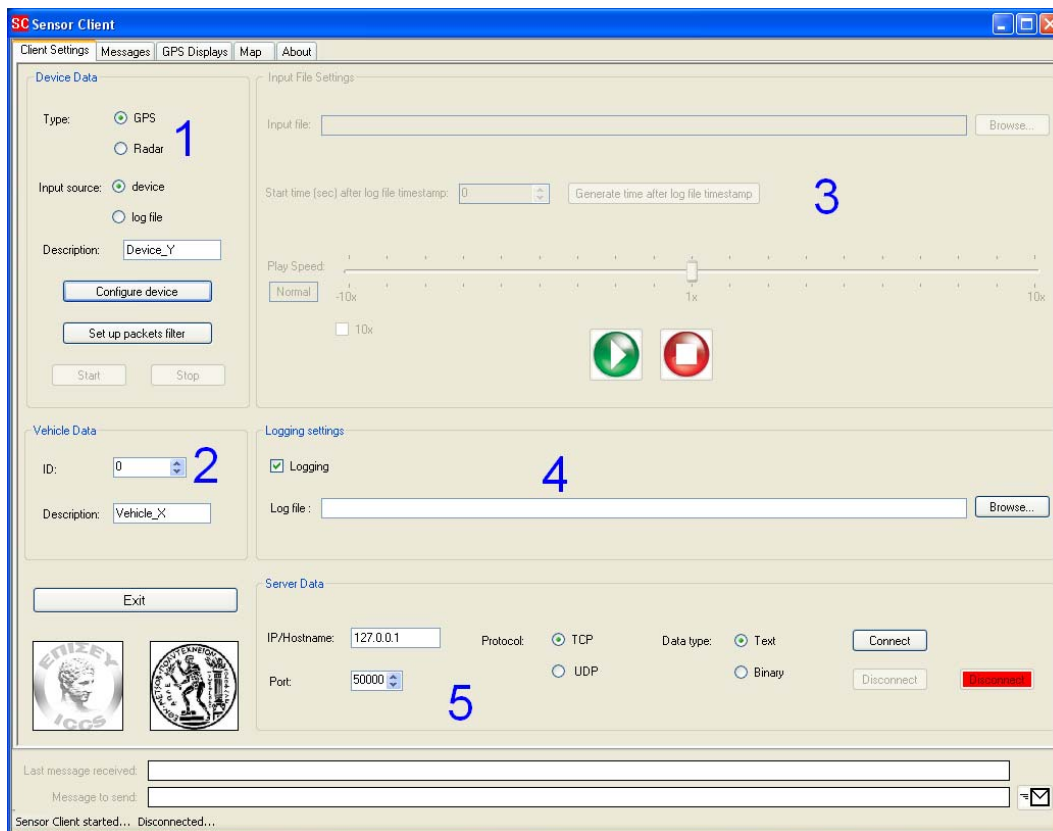
Η εφαρμογή Sensor Client είναι η client εφαρμογή που παίρνει τα δεδομένα από τις συσκευές ή από αντίστοιχα αρχεία log και τα προωθεί στο server (Data Fusion Server). Η εφαρμογή αυτή περιέχει 5 οθόνες:



Σχήμα 2.1: Οθόνη Client Settings του Sensor Client.

1) Client Settings

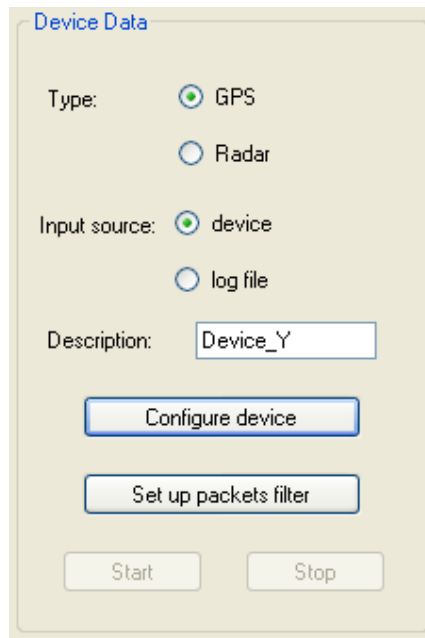
Στην οθόνη αυτή (Σχήμα 2.1) περιέχονται όλες οι βασικές ρυθμίσεις της εφαρμογής που αφορούν στον τύπο της συσκευής, τον τρόπο λήψης δεδομένων. Στο Σχήμα 2.2 φαίνεται η οθόνη διαιρεμένη σε αριθμημένα κομμάτια με βάση της οργάνωση της σε groupboxes.



Σχήμα 2.2: Οθόνη Client Settings του Sensor Client διαιρεμένη με βάση τα groupboxes.

1.1 Device Data

Το groupbox αυτό (Σχήμα 2.3) περιέχει τις ρυθμίσεις για τον τύπο δεδομένων που θα διαβαστούν, GPS ή radar, καθώς και για την πηγή από όπου θα πάρουμε τα δεδομένα αυτά, από την ίδια τη συσκευή ή από αρχεία log.

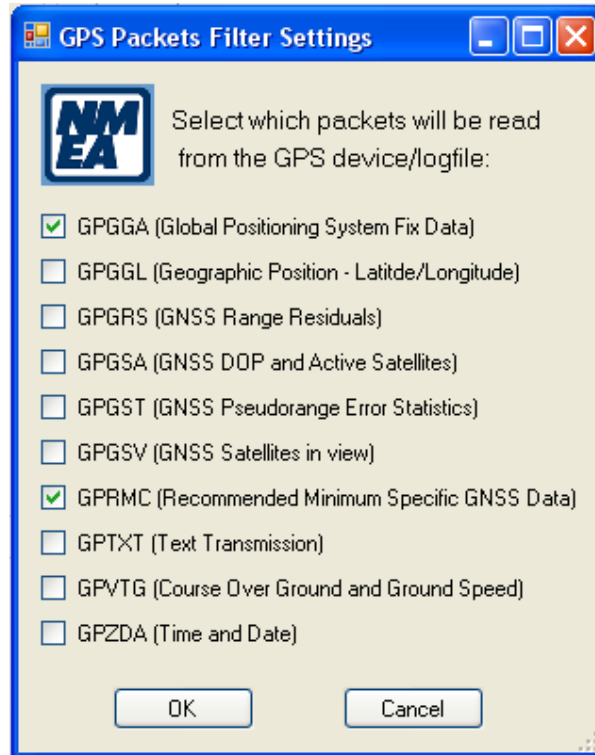


Σχήμα 2.3: Sensor Client::Device Data

Στην παρούσα της έκδοση η εφαρμογή μπορεί να διαβάσει δεδομένα στην περίπτωση του radar μόνο από αρχεία log. Ανάγνωση δεδομένων απευθείας από συσκευή radar μπορεί να γίνει μόνο από την εφαρμογή Can Logger που έχει υλοποιηθεί από το I-SENSE. Στην εφαρμογή αυτή έχουν γίνει μόνο κάποιες πολύ μικρές προσθήκες ώστε να προωθεί τα δεδομένα στο DFR.

Αν ο χρήστης επιλέξει “GPS” τότε ενεργοποιείται το κουμπί “Set up packets Filter”.

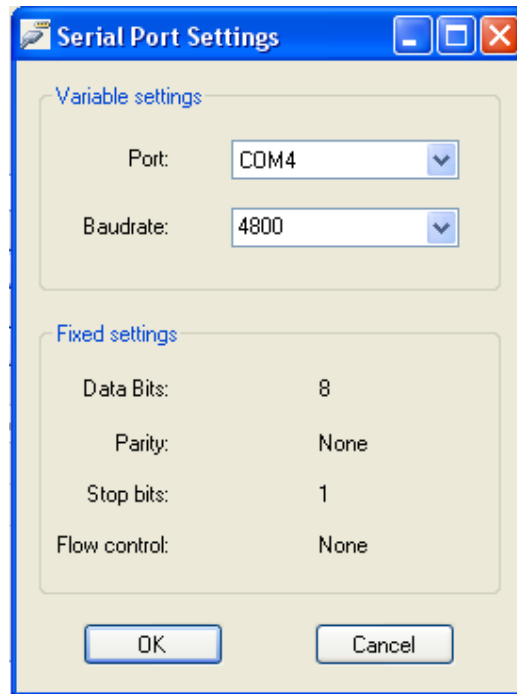
Πατώντας το κουμπί “Set up packets filter” εμφανίζεται το παράθυρο “GPS Packets Filter Settings” του Σχήματος 2.4. Στο παράθυρο αυτό ο χρήστης επιλέγει τα πακέτα του πρωτοκόλλου NMEA που θέλει να διαβαστούν είτε από τη συσκευή είτε από το αρχείο log. Όλα τα υπολοιπα απορρίπτονται. Αν ο χρήστης κλείσει το παράθυρο πατώντας “OK” τότε το σύστημα κρατάει τις αλλαγές που έκανε και αν αυτός ξανανοίξει αργότερα το παράθυρο αυτό, τότε οι προηγούμενες ρυθμίσεις που είχε κάνει θα είναι από default επιλεγμένες. Το αντίθετο συμβαίνει αν το κλείσει πατώντας “Cancel” καθώς το σύστημα κρατάει τις προηγούμενες ρυθμίσεις και αγνοεί τις νέες.



Σχήμα 2.4: Sensor Client::GPS Packets Filter Settings

Αν ο χρήστης επιλέξει GPS “device” τότε ενεργοποιείται το κουμπί “Configure Device”.

Πατώντας το κουμπί “Configure Device” εμφανίζεται το παράθυρο “Serial Port Settings” του Σχήματος 2.4. Στο παράθυρο αυτό ο χρήστης επιλέγει τη θύρα (από τη λίστα θυρών που έχουν βρεθεί στο παρόν μηχάνημα) στην οποία έχει συνδεθεί το GPS, καθώς και το ρυθμό μετάδοσης (Baudrate) στον οποίο έχει ρυθμιστεί να λειτουργεί. Αν ο χρήστης κλείσει το παράθυρο πατώντας “OK” τότε το σύστημα κρατάει τις αλλαγές που έκανε, και αν αυτός το ξαναοίξει αργότερα, τότε οι προηγούμενες ρυθμίσεις που είχε κάνει, θα είναι από default επιλεγμένες. Το αντίθετο συμβαίνει αν το κλείσει πατώντας “Cancel” καθώς το σύστημα κρατάει τις προηγούμενες ρυθμίσεις και αγνοεί τις νέες. Επίσης με το κλείσιμο του παραθύρου ενεργοποιείται και το κουμπί “Start”. Με το κουμπί αυτό ξεκινάει η λήψη δεδομένων από τη συσκευή και ενεργοποιείται το κουμπί “Stop” με το οποίο σταματάει.



Σχήμα 2.5: Sensor Client::Serial Port Settings

Από την άλλη αν ο χρήστης επιλέξει “log file”, είτε για GPS είτε για radar δεδομένα, τότε ενεργοποιούνται οι επιλογές του groupbox 3 “Input File Settings” που φαίνεται στο Σχήμα 2.6.

1.2 Input File Settings

Στο groupbox αυτό, όπως λέει και το όνομα του, περιέχονται ρυθμίσεις για το αρχείο (log) εισόδου. Το αρχείο αυτό μπορεί να καθοριστεί πατώντας το “Browse” με το οποίο ανοίγει ένα “Open File Dialog” ή πληκτρολογώντας απευθείας το path (relative ή absolute) του επιθυμητού αρχείου. Με το slide bar δίπλα από το label “Play Speed” καθορίζεται η ταχύτητα αναπαραγωγής του αρχείου. Άμα η τιμή είναι θετική πχ. “2” αυτό σημαίνει ότι το αρχείο log θα αναπαραχθεί δυο φορές πιο γρήγορα, ενώ αντιθέτως, άμα είναι αρνητική πχ “-2”, τότε θα αναπαραχθεί δύο φορές πιο αργά. Με το “10x” checkbox πολλαπλασιάζονται οι τιμές αυτές επί 10. Η αναπαραγωγή του αρχείου ξεκινάει και σταματάει με το πράσινο και το κόκκινο κουμπί αντίστοιχα. Αν ταχύτητα που έχει οριστεί είναι πολύ μεγάλη (ή όχι και ιδιαίτερα μεγάλη, αλλά ο υπολογιστής εκτελεί και άλλες διεργασίες παράλληλα) και

το σύστημα δεν προλαβαίνει να το αναπαράξει με αυτή την ταχύτητα, τότε βγάζει κατάλληλο μήνυμα για να ειδοποιήσει το χρήστη (Σχήμα 2.7).

Τα υπόλοιπα αφορούν στη συγχρονισμένη αναπαραγωγή των αρχείων log και θα εξηγηθούν στην αντίστοιχη παράγραφο.

1.3 Logging Settings

Ενώ ο χρήστης διαβάζει δεδομένα από τη συσκευή (ή ακόμη και από το αρχείο) μπορεί να τα καταγράψει σε ένα αρχείο log. Αυτό γίνεται φροντίζοντας να είναι τικαρισμένη η επιλογή “Logging” του Logging settings” groupbox #4 (Σχήμα 2.8).

Πατώντας το κουμπί “Browse” εμφανίζεται ένα “Save File Dialog” (Σχήμα 2.10) με το οποίο ο χρήστης μπορεί να καθορίσει το directory στο οποίο θα αποθηκευτεί το αρχείο log. Το όνομα του αρχείου όπως φαίνεται και στο σχήμα 2.10 έχει αυτόματα τη μορφή:

Log_File_Device_Y_Vehicle_X_12.22.06_00.35.31

Όπου:

Device_Y: είναι το string που έχουμε εισάγει στο αντίστοιχο πεδίο του groupbox #1

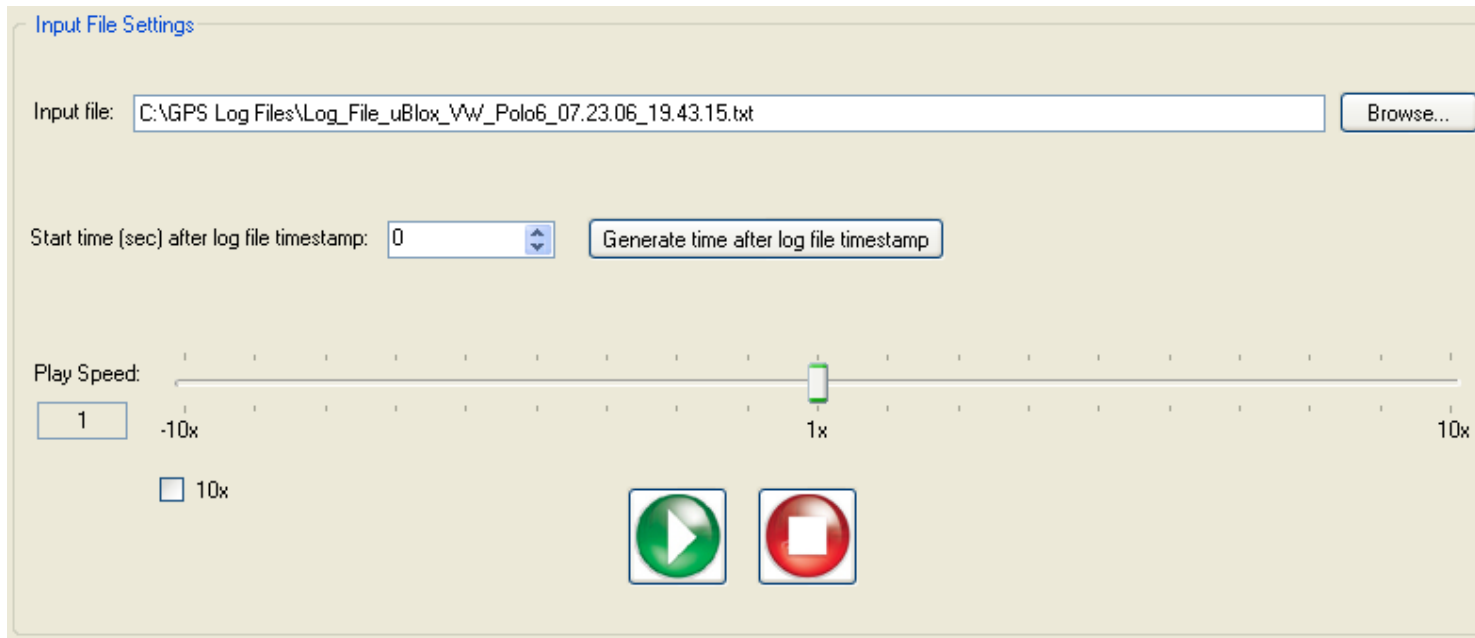
“Device data”. Δηλαδή είναι μια φιλική περιγραφή που εισάγει ο χρήστης για τη συσκευή για σκοπούς απομνημόνευσης

Vehicle_X: είναι το string που έχουμε εισάγει στο αντίστοιχο πεδίο του groupbox #2

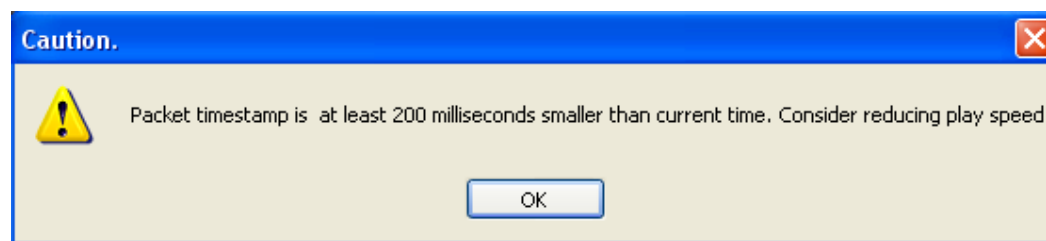
“Vehicle data”. Δηλαδή είναι μια φιλική περιγραφή που εισάγει ο χρήστης για το όχημα για σκοπούς απομνημόνευσης.

12.22.06: ημερομηνία

00.35.31: ώρα



Σχήμα 2.6: Sensor Client::Input File Settings



Σχήμα 2.7: Μήνυμα ότι το σύστημα δεν «προλαβαίνει» να αναπαράξει το αρχείο log τόσο γρήγορα που όρισε ο χρήστης.

Logging settings

Logging

Log file :

Σχήμα 2.8.: Sensor Client::Logging Settings

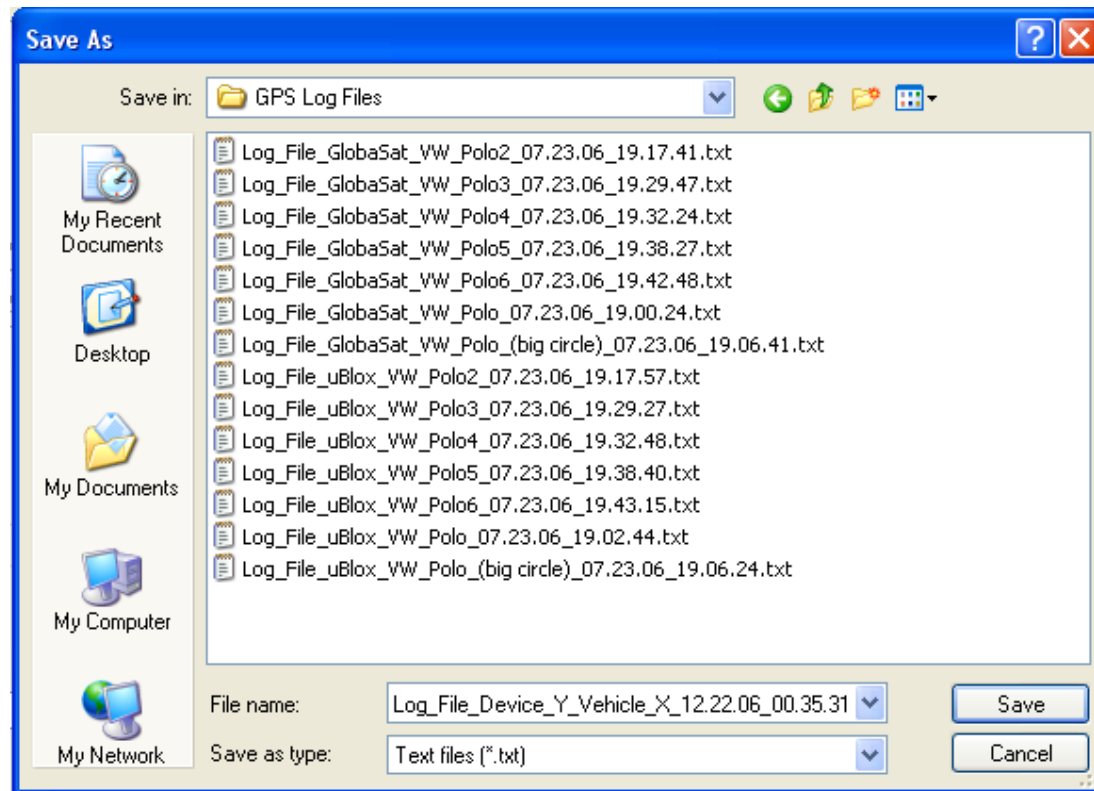
IP/Hostname:

Server Data

IP/Hostname: Protocol: TCP UDP Data type: Text Binary

Port:

Σχήμα 2.9.: Sensor Client::Server Data



Σχήμα 2.10.: Sensor Client::Save File Dialog

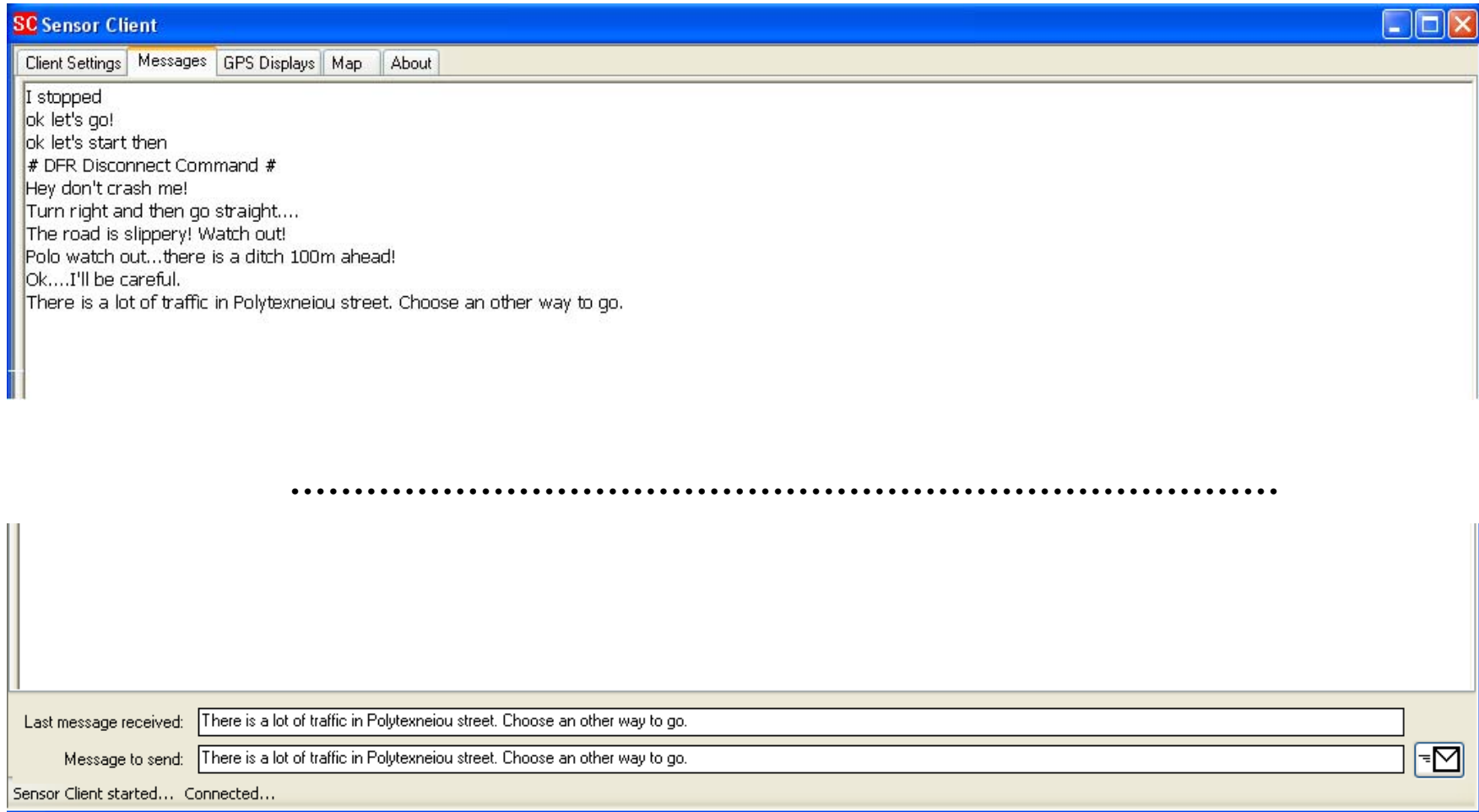
1.4 Server Data

Στο “Server Data” groupbox #5 (Σχήμα 2.9) περιέχονται οι βασικές ρυθμίσεις για τη δημιουργία σύνδεσης με το Data Fusion Server. Για παράδειγμα η θύρα στην οποία ακούει (το default είναι 50000) το IP του (127.0.0.1 ή 147.102.5.78) ή το όνομα του (πχ. “rontus”) με το οποίο βρίσκεται στη συνέχεια το IP που του έχει ανατεθεί στο δίκτυο. Επίσης καθορίζεται ο τρόπος μετάδοσης των δεδομένων πάνω στο κανάλι επικοινωνίας να είναι είτε stream communication(TCP) είτε datagram communication (UDP), που στην παρούσα έκδοση δεν υποστηρίζεται. Τέλος καθορίζεται και η μορφή των δεδομένων που θα στέλνονται: χαρακτήρες ή binary. Στην παρούσα έκδοση του προγράμματος έχει υλοποιηθεί μόνο ένα character protocol για την επικοινωνία του SC με τον DFR το οποίο είναι σαν το NMEA, απλώς επεκτεταμένο ώστε να στέλνει και πακέτα radar, μηνύματα (chat) καθώς και άλλες χρήσιμες πληροφορίες (πχ. αίτηση αποσύνδεσης, αναγνωριστικό πακέτο οχήματος-συσκευής κτλ.).

Η σύνδεση με τον DFR γίνεται πατώντας το κουμπί “Connect” και είναι επιτυχής εφόσον ο DFR έχει ξεκινήσει ένα connections listener που ακούει στο συγκεκριμένο αυτό συνδιασμό IP:port. Επίσης κατά πάσα πιθανότητα θα πρέπει να ρυθμιστεί και το firewall ώστε να μην μπλοκάρει είτε τους SCs που προσπαθούν να συνδεθούν, είτε τον DFR που προσπαθεί να ακούσει για συνδέσεις. Η αποσύνδεση γίνεται με το (λευκό) κουμπί “Disconnect”. Ένας SC για να αποσυνδεθεί κάνει αίτηση αποσύνδεσης στον DFR και αποσυνδέεται τελικά μόνο όταν πάρει πίσω μια επιβεβαίωση. Από την άλλη όμως αν για κάποιο λόγο ο DFR δεν είναι προσβάσιμος (πχ το δίκτυο έχει πέσει, ο DFR ή ο SC είναι εκτός του εύρους λήψης του ασύρματου δικτύου) τότε ο SC δεν θα καταφέρει να αποσυνδεθεί ποτέ με αυτόν τον τρόπο. Για αυτό το λόγο υπάρχει το κόκκινο κουμπί “Disconnect”. Στην περίπτωση αυτή ο SC απουσυνδέεται κλείνοντας τη σύνδεση με τον DFR. Ο τρόπος αυτός πρέπει να χρησιμοποιείται μόνο αν δεν πετυχαίνει ο άλλος καθώς ένα thread του SC επιχειρεί να κλείσει τη σύνδεση απότομα ενώ κάποιο άλλο thread του περιμένει (είναι μπλοκαρισμένο σε εντολή receive του Winsock) για μηνύματα/εντολές από τον DFR μέσω της ίδιας σύνδεσης.

1.5 Vehicle Data

Ο DFR διακρίνει μεταξύ τους τις συνδεδεμένες συσκευές με βάση το τύπο τους (κάθε τύπος συσκευής έχει δικό του αναγνωριστικό ID) καθώς και με τον κωδικό (ID) του οχήματος στο οποίο ανήκουν. Το τελευταίο καθορίζεται στο groupbox “Vehicle



Σχήμα 2.11: Sensor Client::Messages

Data” #2. Οι φιλικές περιγραφές (descriptions) που υπάρχουν τόσο για τη συσκευή (“Device Data” groupbox) όσο και για το όχημα (“Vehicle Data” groupbox) είναι μόνο για να βοηθήνε το χρήστη, χρησιμοποιώντας μνημονικά ονόματα, να τα ξεχωρίζει μεταξύ τους, καθώς και τα αντίστοιχα αρχεία log.

2) Messages

Στην οθόνη αυτή (Σχήμα 2.11) φαίνονται όλα τα μηνύματα που έχουν στείλει οι SCs ή ο ίδιος ο DFR. Όταν ένας SC στείλει ένα μήνυμα (chat packet) στον DFR τότε αυτός το προωθεί (broadcast) σε όλους τους συνδεδεμένους SCs ανεξαιρέτως (συμπεριλαμβανομένου και αυτού που αρχικά έστειλε το μήνυμα). Το ίδιο συμβαίνει και με τα μηνύματα που στέλνει ο ίδιος ο DFR, προωθούνται σε όλους. Έτσι όταν ένας SC στείλει ένα μήνυμα, του έρχεται μετά και σε αυτόν πίσω, και όταν το δει στο “Last message received:” ξέρει ότι το έχει πάρει ο DFR και ότι έχει αρχίσει να το προωθεί στους άλλους συνδεδεμένους SCs.

3) GPS Displays

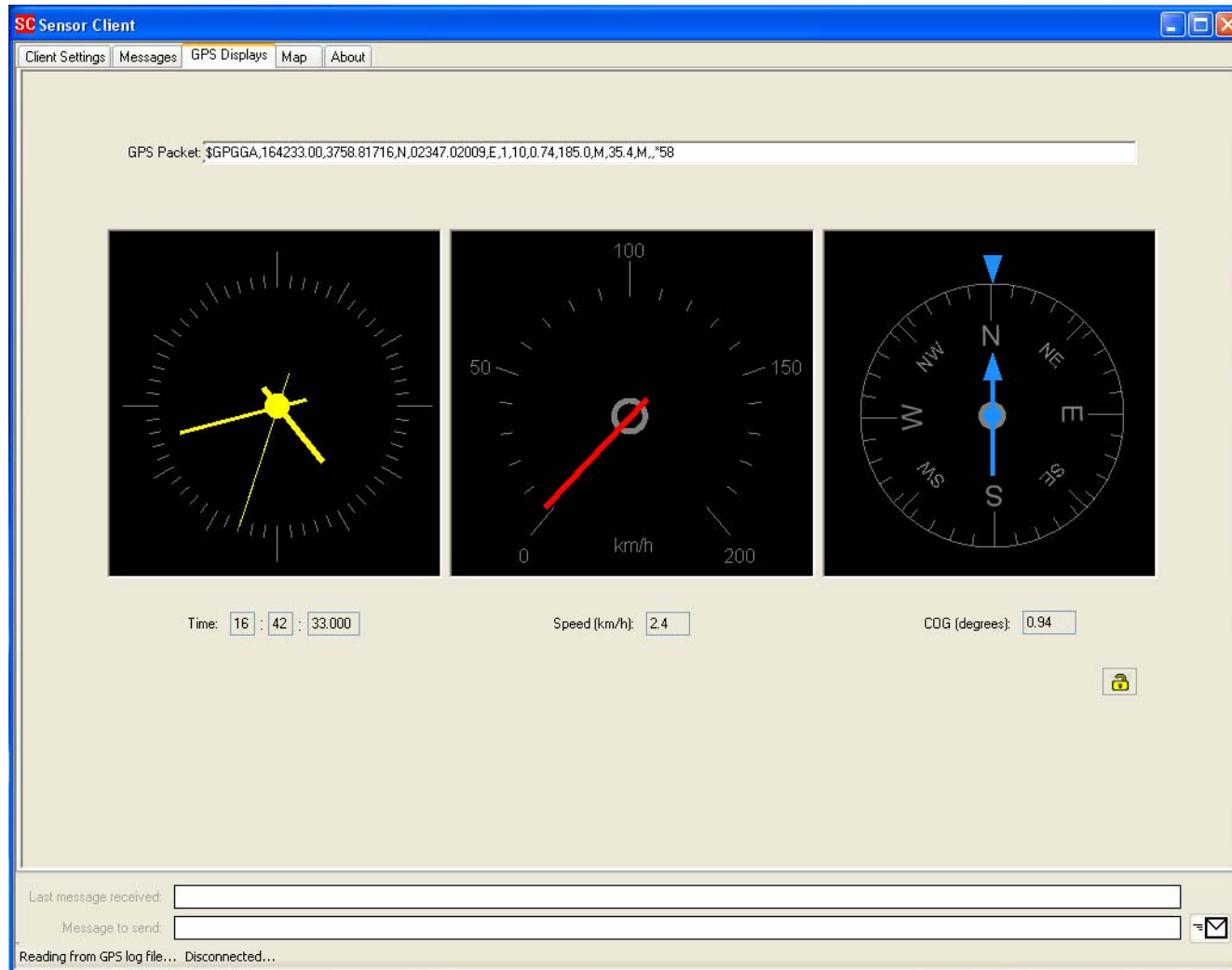
Στην οθόνη αυτή (Σχήμα 2.12) φαίνονται τα πακέτα του GPS που διαβάζονται καθώς και κάποια βασικά όργανα όπως ένα ταχύμετρο, πυξίδα και ρολόι. Το ρολόι μπορεί να είναι ένας ιδιαίτερα χρήσιμος και γρήγορος εποπτικός τρόπος για να επιβεβαιώσουμε ότι δεν κάναμε κάποιο λάθος χειρισμού και ότι τα αρχεία log όντως αναπαράγονται συγχρονισμένα.

4) Map

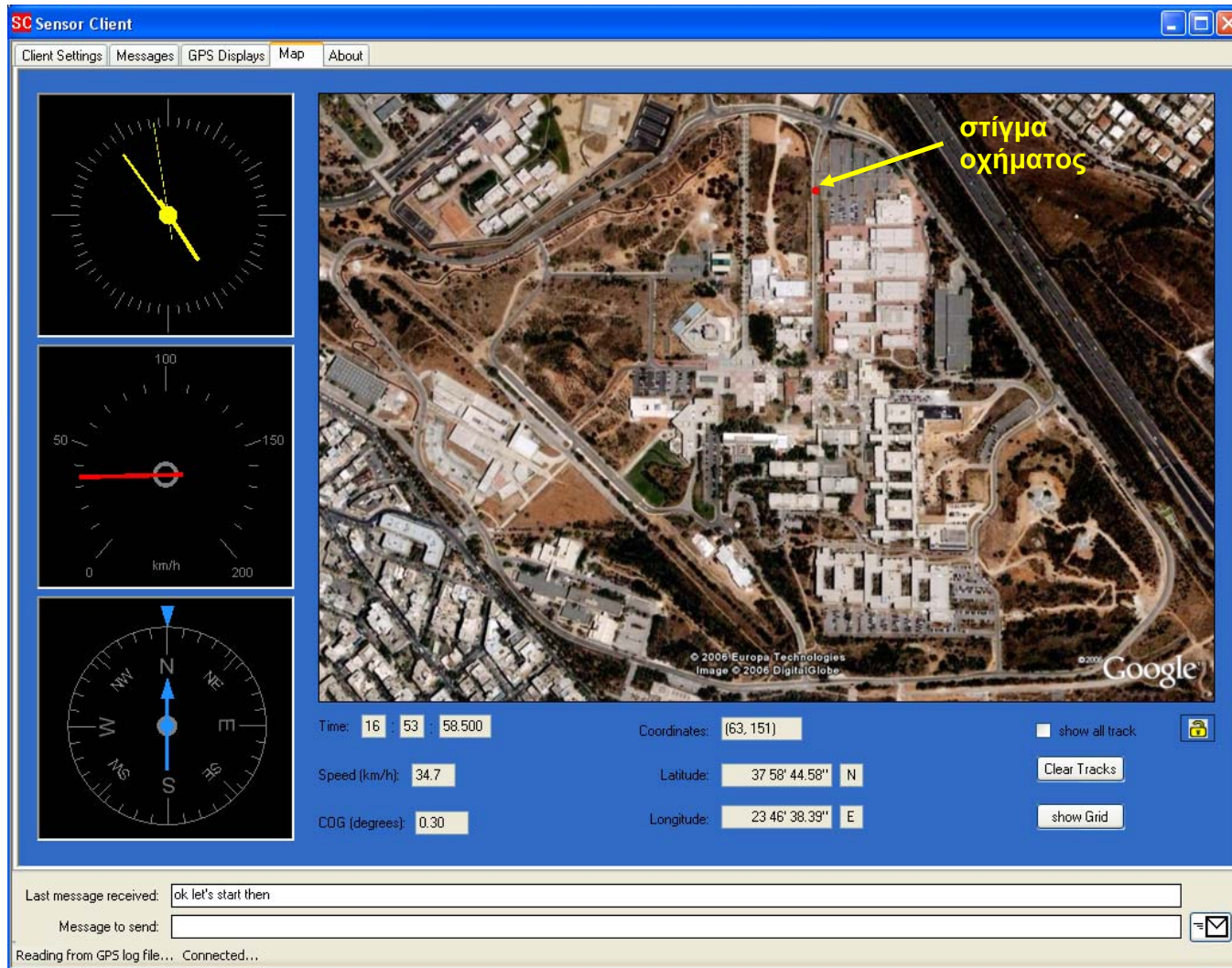
Στην οθόνη αυτή (Σχήμα 2.13) εμφανίζονται διάφορες πληροφορίες που παίρνουμε από το GPS, τα όργανα που υπάρχουν και στην οθόνη GPS Displays αλλά υπο κλίμακα, καθώς και το στίγμα που παίρνουμε από το GPS πάνω στο χάρτη του πολυτεχνείου.

Οι επιπλέον λειτουργίες που παρέχονται είναι οι εξής:

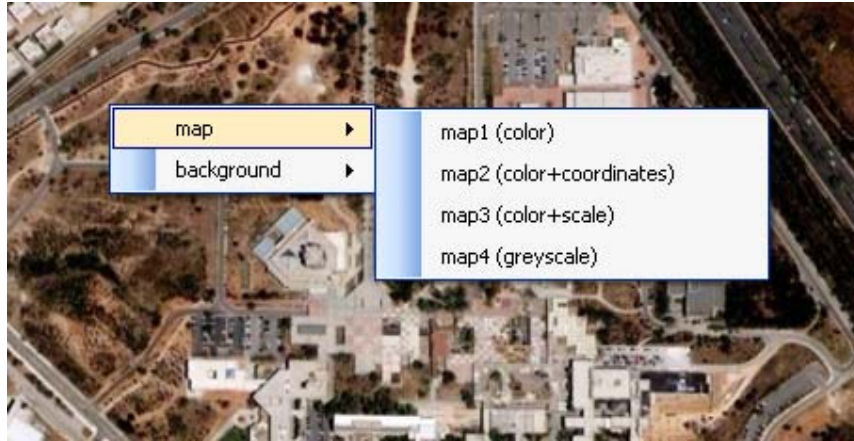
- Ο χρήστης μπορεί να βλέπει ολόκληρη την τροχιά του οχήματος και όχι μόνο το τρέχον στίγμα του, αν τικάρει το checkbox “show all track”.
- Ο χάρτης μπορεί να καθαριστεί από προηγούμενα στίγματα πατώντας το κουμπί “Clear Tracks”.
- Ο χάρτης και όλες οι απεικονιζόμενες πληροφορίες μπορούν να «παγωθούν» πατώντας πάνω στο κουμπί που απεικονίζει την κλειδαριά. Το ίδιο ισχύει και για το αντίστοιχο κουμπί που υπάρχει στην οθόνη GPS Displays. Επίσης με αυτό τον τρόπο,



Σχήμα 2.12: Sensor Client::GPS Displays



Σχήμα 2.13: Sensor Client::Map



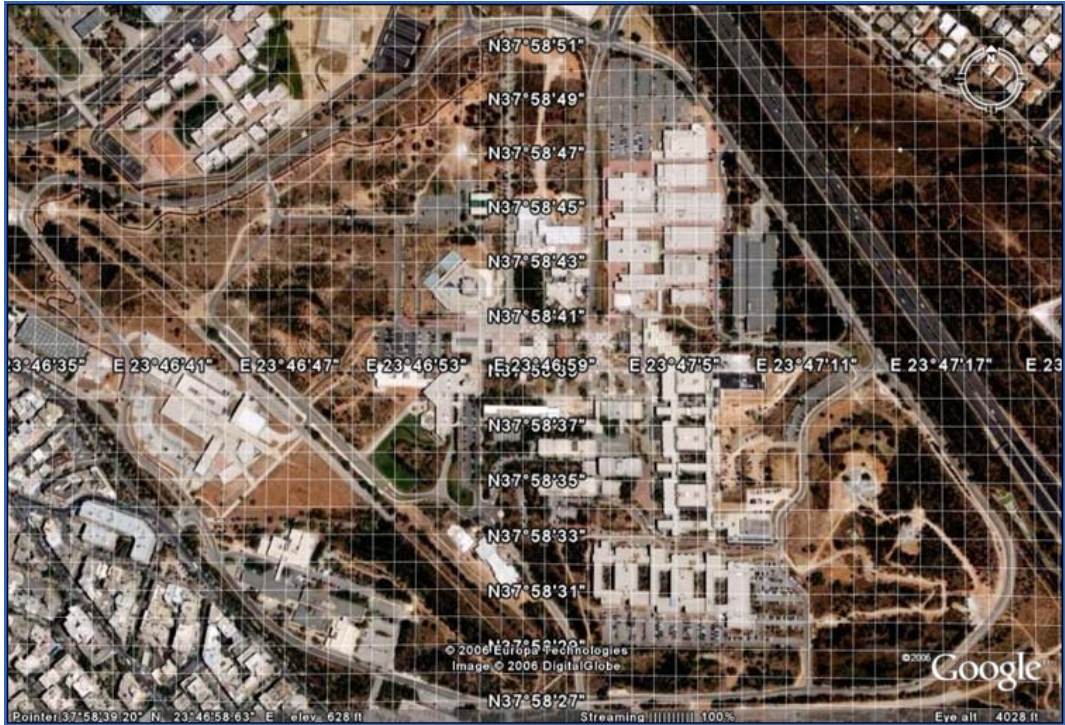
Σχήμα 2.14: Sensor Client::Map. Επιλογές που εμφανίζονται κάνοντας δεξί κλικ πάνω στο χάρτη

αν δεν μας νοιάζει η ανανέωση του GUI, μπορούμε να εξοικονομήσουμε processing resources. Αυτό είναι ιδιαίτερα χρήσιμο στην περίπτωση που στον ίδιο υπολογιστή τρέχουν περισσότερες από μία εφαρμογές (πχ. περισσότεροι από ένας SCs και πιθανώς και ο DFR του συστήματος).

- Κάνοντας δεξί κλικ πάνω στο χάρτη εμφανίζονται διάφορες επιλογές (Σχήμα 2.14) για αλλαγή του χάρτη και για επιλογή ενός μονόχρωμου φόντου.

Αν επιλέξουμε το “map2 (color+coordinates) τότε εμφανίζεται ένας άλλος χάρτης με συντεταγμένες γεωγραφικού πλάτους και μήκους (Σχήμα 2.15). Ο χάρτης αυτός είναι χρήσιμος και για να φανεί το λάθος που γίνεται στην απεικόνιση του στίγματος πάνω στο χάρτη. Για να φανεί αυτό πρέπει να πατήσουμε το κουμπί “show Grid” οπότε εμφανίζεται ένα πλέγμα από σημεία που αντιστοιχεί στα σημεία που διασταυρώνονται οι γραμμές γεωγραφικού πλάτους με της γραμμές γεωγραφικού μήκους της φωτογραφίας από το Google. Το σφάλμα σε μια συγκεκριμένη θέση καθορίζεται από την απόκλιση ενός σημείου του πλέγματος από το αντίστοιχο σημείο διασταύρωσης και διαφέρει απο σημείο σε σημείο του χάρτη (Σχήμα 2.16).

5) About : Πληροφορίες για την ανάπτυξη της εφαρμογής.



Σχήμα 2.15: Sensor Client::Map. Ο χάρτης που εμφανίζεται αν επιλέξουμε map2 (color+coordinates).



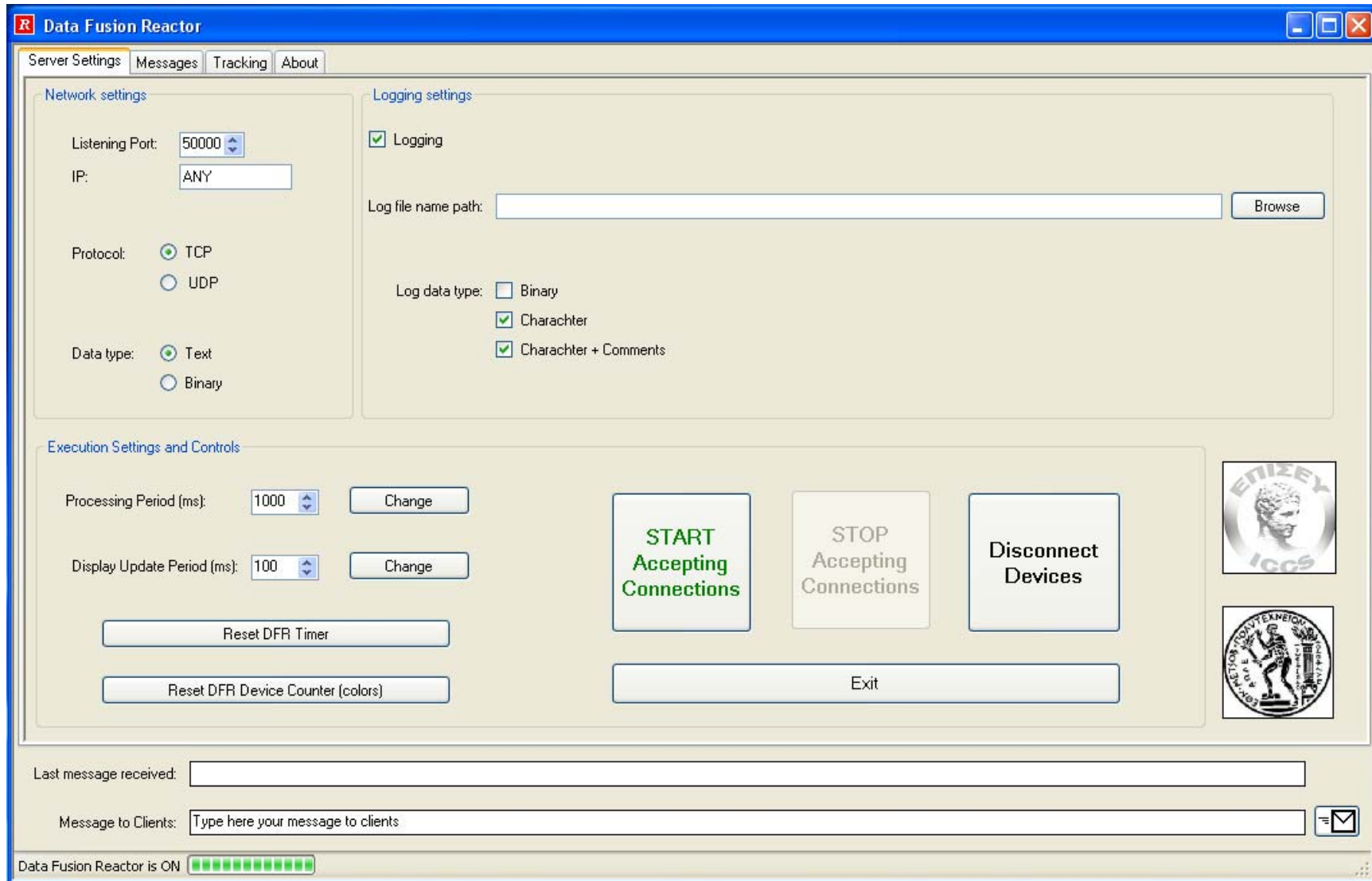
Σχήμα 2.16: Sensor Client::Map. Ο χάρτης που εμφανίζεται αν επιλέξουμε map2 (color+coordinates) και πατήσουμε το κουμπί “show Grid”.



Σχήμα 2.17: Τα σφάλματα απεικόνισης δείχνονται με βέλη με κατεύθυνση από αυτό που υπολογίζει το πρόγραμμα (κουκίδα) προς το πραγματικό σημείο της φωτογραφίας από το Google Earth (σημείο διασταύρωσης «οριζόντιων» και «κατακόρυφων» γραμμών). Η απεικόνιση θα ήταν τέλεια αν οι γραμμές γεωγραφικού πλάτους και μήκους σχημάτιζαν τέλει ορθογωνικό πλέγμα καθώς το πρόγραμμα υποθέτει αυτό για να είναι εύκολη η αντιστοίχιση από τιμές γεωγραφικού μήκους και πλάτους σε pixels εικόνας.

2.3 Η εφαρμογή Data Fusion Reactor

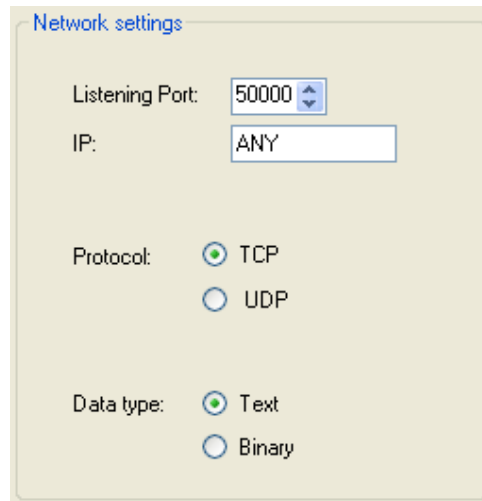
Η εφαρμογή Data Fusion Reactor (DFR) είναι η server εφαρμογή στην οποία συνδέονται όλοι οι clients (Sensor Clients) και προωθούν τα δεδομένα τους. Η εφαρμογή αυτή συγκεντρώνει όλα τα εισερχόμενα δεδομένα σε μια κοινή δομή ώστε να μπορεί στη συνέχεια ένας αλγόριθμος σύζευξης δεδομένων να τα επεξεργαστεί. Η εφαρμογή αυτή περιέχει 4 οθόνες:



Σχήμα 2.18: Data Fusion Reactor::Server Settings

2) Server Settings

Στην οθόνη αυτή (Σχήμα 2.18) περιέχονται όλες οι βασικές ρυθμίσεις της εφαρμογής που αφορούν στη δημιουργία συνδέσεων με τους SCs, την αποθήκευση των λαμβανόμενων δεδομένων σε αρχεία log, την περίοδο επεξεργασίας και ανανέωσης του GUI.



Σχήμα 2.19: Data Fusion Reactor::Network Settings

2.1 Network Settings

Το groupbox αυτό (Σχήμα 2.3) περιέχει τις ρυθμίσεις για τη δημιουργία συνδέσεων με τους SCs. Πχ. Η θύρα στην οποία ακούει για συνδέσεις ο connections listener του DFR (το default είναι 50000) και το IP του (127.0.0.1 ή 147.102.5.78) ή το όνομα του (πχ. “rontus”) με το οποίο βρίσκεται στη συνέχεια το IP που του έχει αντεθεί στο δίκτυο. Το IP είναι χρήσιμο να καθορίζεται αντί να χρησιμοποιούμε το τετριμμένο 127.0.0.1 γιατί υπάρχει περίπτωση ο υπολογιστής να έχει περισσότερες από μια κάρτες δικτύου (πχ. και ενσύρματη και ασύρματη). Έτσι στην περίπτωση αυτή καθορίζοντας το IP καθορίζουμε και ποια θέλουμε να χρησιμοποιούμε. Εισάγοντας στο πεδίο αυτό “ANY” που είναι και η προκαθορισμένη επιλογή (default) επιτρέπουμε συνδέσεις σε όλες τις κάρτες δικτύου και τα αντίστοιχα IP. Έτσι πχ. ένας υπολογιστής που τρέχει έναν DFR μπορεί να δέχεται συνδέσεις και πακέτα ταυτόχρονα και απο το κλασσικό ενσύρματο LAN αλλά και από το ασύρματο WiFi. Επίσης καθορίζεται ο τρόπος μετάδοσης των δεδομένων πάνω στο κανάλι επικοινωνίας να είναι είτε stream communication(TCP) είτε datagram communication (UDP) που στην παρούσα έκδοση δεν υποστηρίζεται. Τέλος καθορίζεται και η μορφή των δεδομένων που θα στέλνονται: χαρακτήρες ή δυαδικά (binary). Στην παρούσα έκδοση του προγράμματος έχει υλοποιηθεί μόνο ένα πρωτόκολλο χαρακτήρων (character protocol) για την επικοινωνία του SC με τον DFR το οποίο είναι σαν το NMEA απλώς επεκτεταμένο ώστε να στέλνει και πακέτα

Logging settings

Logging

Log file name path:

Log data type: Binary
 Character
 Character + Comments

Σχήμα 2.20: Data Fusion Reactor::Logging Settings

Execution Settings and Controls

Processing Period (ms):

Display Update Period (ms):

Σχήμα 2.21: Data Fusion Reactor::Execution Settings and Controls

radar, μηνύματα (chat), καθώς και άλλες χρήσιμες πληροφορίες (πχ. αίτηση αποσύνδεσης, αναγνωριστικό πακέτο οχήματος-συσκευής κτλ.).

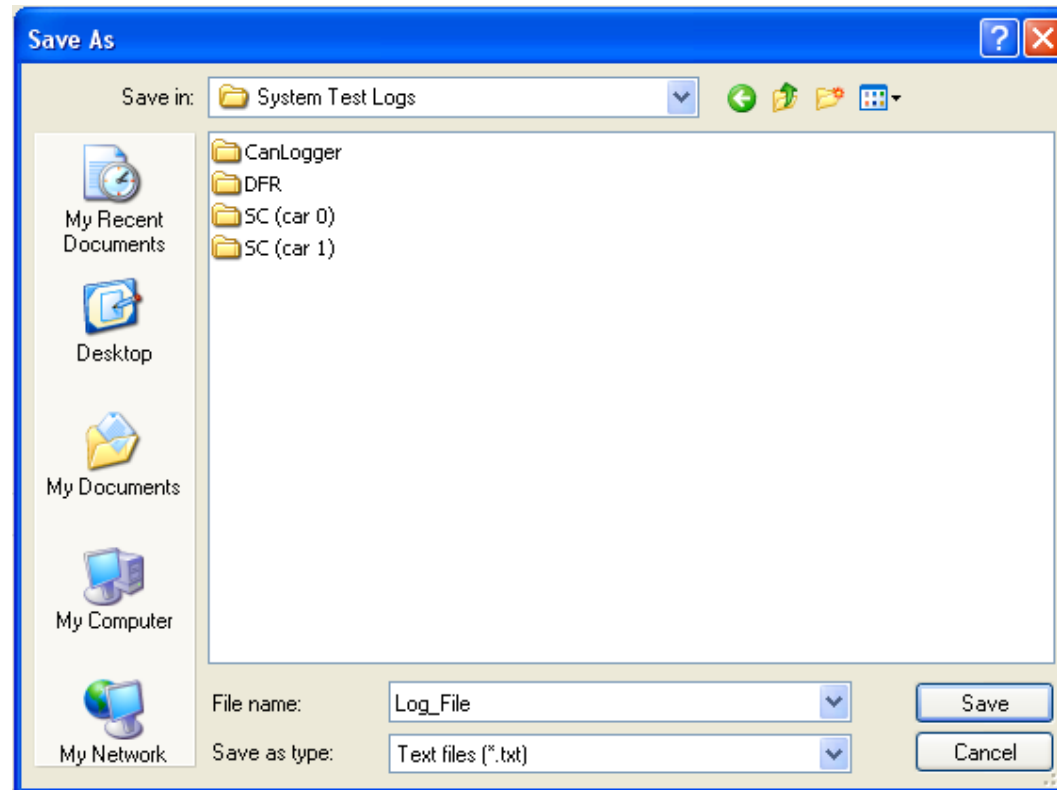
2.2 Logging Settings

Ο DFR διακρίνει μεταξύ τους τις συνδεδεμένες συσκευές με βάση το τύπο τους (κάθε τύπος συσκευής έχει δικό του αναγνωριστικό ID) καθώς και με τον κωδικό (ID) του οχήματος στο οποίο ανήκουν. Τα στοιχεία περιέχονται στο πρώτο πακέτο που στέλνει κάθε Sensor Client με το που καταφέρει να συνδεθεί με τον DFR, που είναι ένα, όπως λέγεται, αγνωριστικό πακέτο (identification packet). Το πακέτο αυτό, όπως ειπώθηκε, περιέχει όλες τις απαραίτητες πληροφορίες ώστε να ξέρει ο DFR στη συνέχεια πως (τι τύπος συσκευής είναι;) να χειριστεί τα πακέτα δεδομένων που ακολουθούν αλλά και που (τι τύπος συσκευής είναι και ποιο όχημα φέρει τη συσκευή αυτή;) να τα αποθηκεύσει στη κοινή αυτή δομή που συγκεντρώνονται όλα δεδομένα για να είναι διαθέσιμα για επεξεργασία.

Αν η επιλογή “Logging” είναι ενεργοποιημένη τότε ο DFR θα κρατάει όλα τα εισερχόμενα σε διαφορετικά αρχεία log ανάλογα με τη συσκευή από την οποία προέρχονται. Πιο συγκεκριμένα πατώντας το “Browse” μπορούμε να επιλέξουμε το directory μέσα στο οποίο θα αποθηκευτούν, αλλά το όνομα τους, όπως φαίνεται και στο σχήμα 2.22, θα ξεκινάει από default με το “Log_File”. Παρόλα αυτά ο χρήστης μπορεί να το σβήσει και να γράψει ότι θέλει. Το τελικό όνομα του αρχείου θα έχει τη μορφή (Σχήμα 2.23):

<directory path>/Log_File_Device_Vehicle_Timestamp.txt

<directory path>:	Το directory που επέλεξε ο χρήστης να σώσει τα αρχεία χρησιμοποιώντας το “Save File Dialog” του παραπάνω σχήματος.
Log_File:	ή οτιδήποτε άλλο εισήγαγε ο χρήστης στο Save File Dialog ως file name.
Device:	Μια μνημονική φιλική περιγραφή που είχε εισάγει ο χρήστης στον Sensor Client για τη συσκευή.
Vehicle:	Μια μνημονική φιλική περιγραφή που είχε εισάγει ο χρήστης στον Sensor Client για το όχημα που φέρει τη συσκευή.
Timestamp	ημερομηνία και ώρα



Σχήμα 2.22: Data Fusion Reactor::Save File Dialog



Σχήμα 2.23: Data Fusion Reactor::Logging Settings

Επίσης ο χρήστης μπορεί να επιλέξει και το είδος των αρχείων log που θέλει να κρατήσει όπως π.χ. “binary” αρχεία log (δεν υποστηρίζεται στην τρέχουσα έκδοση αλλά μπορεί να υλοποιηθεί πολύ εύκολα), “text” και “text with comments”. Το “text” σημαίνει ότι τα αρχεία log είναι κλασσικά character files όπου τα πακέτα αποθηκεύονται όπως ακριβώς τα στέλνει ο Sensor Client (υπενθυμίζουμε ότι το πρωτόκολλο επικοινωνίας είναι ASCII πρωτόκολλο όπως το NMEA) με τη διαφορά ότι προστίθεται πριν από κάθε τέτοιο πακέτο και ένα timestamp το οποίο είναι ο χρόνος που έχει περάσει από τη στιγμή που ξεκίνησε ο DFR. Στην περίπτωση του “text with comments” τα λαμβανόμενα πακέτα αναλύονται ανάλογα με τον τύπο τους, και οι πληροφορίες που φέρουν αποθηκεύονται σε ευανάγνωστη μορφή μαζί με αυτά

2.3 Execution Settings and Controls

Με το “Processing Period” (Σχήμα 2.21) ο χρήστης μπορεί να καθορίσει τη περίοδο με την οποία τρέχει ο αλγόριθμος της σύζευξης δεδομένων. Μια άλλη υλοποιήσιμη τεχνική αλλά αμφίβολης χρησιμότητας θα ήταν να τρέχει ο αλγόριθμος κάθε φορά που έρχεται ένα νέο δεδομένο.

Με το “Display Update Period” ο χρήστης μπορεί να καθορίσει τη περίοδο με την οποία θα ανανεώνεται το GUI. Μια άλλη υλοποιήσιμη τεχνική αλλά αμφίβολης απόδοσης θα ήταν να ανανεώνεται κάθε φορά που έρχεται ένα νέο δεδομένο όπως γίνεται στην περίπτωση του Sensor Client. Αν το κάναμε αυτό όμως στην περίπτωση του DFR που μπορεί να παίρνει δεδομένα από πολλούς SCs ταυτόχρονα, θα καταλήγουμε να σπαταλάμε πάρα πολλά resources για την συχνή ανανέωση του GUI. Έτσι αν οι απαιτήσεις επεξεργασίας του αλγορίθμου σύζευξης είναι μεγάλες και ταυτόχρονα δε μας ενδιαφέρει και πολύ το GUI μπορούμε να του θέσουμε μια πολύ μεγάλη περίοδο ανανέωσης.

Με το κουμπί “Reset DFR Timer” ο χρήστης μπορεί να μηδενίσει τον αντίστοιχο timer που χρησιμοποιείται για να μπαίνουν τα timestamps στα αρχεία log. Αυτό είναι χρήσιμο να γίνεται πριν ξεκινήσει ο DFR να λαμβάνει τα πρώτα πακέτα και ειδικά στην περίπτωση που είχαμε ξεκινήσει την εφαρμογή πολύ ώρα πιο πριν. Από την άλλη δεν θα ήταν καλή ιδέα να το κάνουμε τη στιγμή που παίρνουμε πακέτα και έχουμε ήδη προσθέσει κάποια σε αρχεία log.

Ανάλογα με τη σειρά που συνδέονται οι συσκευές στον DFR τους ανατίθεται και ένας αύξων αριθμός, ο οποίος αντιστοιχίζεται με ένα χρώμα. Τα πρώτα 10 είναι καθορισμένα ενώ τα υπόλοιπα παράγονται με ψευδοτυχαίο τρόπο. Αν ο χρήστης πατήσει το κουμπί “Reset DFR Device Counter (colors)” τότε μηδενίζεται ο αντίστοιχος μετρητής και η ανάθεση χρωμάτων σε συσκευές ανακυκλώνεται. Σημειώνεται ότι ο μετρητής αυτός εξυπηρετεί μόνο την ανάθεση χρωμάτων στις συσκευές του GUI και όχι στη διάκριση των συσκευών μεταξύ τους για το fusion.

Για να ξεκινήσει ο DFR να δέχεται συνδέσεις πρέπει ο χρήστης να πατήσει το κουμπί “START Accepting Connections” με το οποίο δημιουργείται ο listener που ακούει στο συγκεκριμένο IP (ή σε όλα τα διαθέσιμα) και στη συγκεκριμένη θύρα. Πατώντας το κουμπί “STOP Accepting Connections” ο connections listener σταματάει να δέχεται συνδέσεις αλλά αυτό δε σημαίνει ότι σταματάει να δέχεται πακέτα από τις συσκευές που έχουν ήδη συνδεθεί. Απλώς παύει να δέχεται νέες συνδέσεις. Έτσι αν είμαστε σίγουροι ότι έχουν συνδεθεί όλες οι επιθυμητές συσκευές τότε μπορούμε να σταματήσουμε (τερματίσουμε) τον listener ώστε να εξοικονομήσουμε resources ή και να αποτρέψουμε τη σύνδεση άλλων ανεπιθύμητων συσκευών. Επίσης αφού τερματίσουμε τον listener μπορούμε να ξεκινήσουμε ένα νέο που να ακούει στο ίδιο συνδυασμό IP-port με πριν ή και σε διαφορετικό. Έτσι δεν είναι ανάγκη όλες οι συσκευές να συνδέονται με τα ίδια ζευγάρια IP-port αλλά μπορούν να χρησιμοποιούν διαφορετικά.

2) Messages

Στην οθόνη αυτή (Σχήμα 2.24) φαίνονται όλα τα μηνύματα που έχουν στείλει οι SCs ή ο ίδιος ο DFR. Όταν ένας SC στείλει ένα μήνυμα (chat packet) στον DFR τότε αυτό εμφανίζεται με το αντίστοιχο χρώμα που έχει ανατεθεί στον αποστολέα του, με βάση τον αύξων αριθμό συσκευής που αναφέραμε πιο πριν. Τα μηνύματα που προέρχονται από τον ίδιο τον DFR εμφανίζονται με γκρι χρώμα ή γκρι φόντο. Επίσης φαίνονται και οι αιτήσεις αποσύνδεσης που κάνουν οι SCs στον DFR.

3) Tracking

Με το που ξεκινήσει ο DFR η οθόνη είναι όπως φαίνεται στο σχήμα 2.25. Στο αριστερό μέρος της οθόνης αυτής φαίνονται όλες οι συνδεδεμένες συσκευές, με το δικό της χρώμα η κάθε μια, και ομαδοποιημένες ανά όχημα σε μορφή δένδρου (Σχήμα 2.26). Όταν μια νέα συσκευή συνδεθεί τότε εμφανίζεται ένα νέος κόμβος για τη συσκευή αυτή καθώς και ο αντίστοιχος κόμβος για το αντίστοιχο όχημα που φέρει τη συσκευή αν δεν υπήρχε από πριν. Το αντίθετο συμβαίνει αν αποσυνδεθεί μια συσκευή.

Όπως και στην περίπτωση του χάρτη του Sensor Client υπάρχουν οι εξής δυνατότητες:

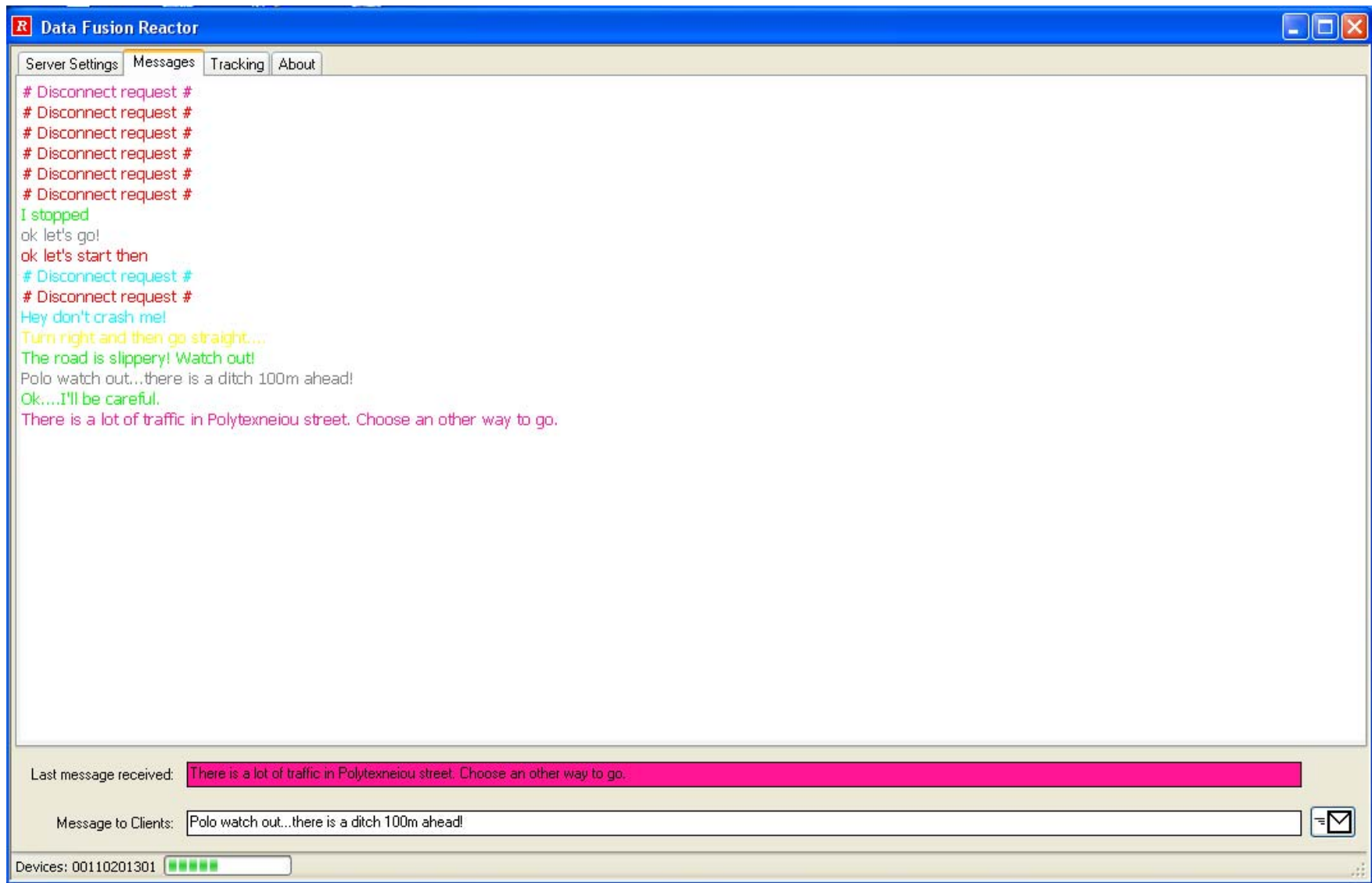
- Ο χρήστης μπορεί να βλέπει ολόκληρη την τροχιά (Σχήμα 2.28) των συνδεδεμένων οχημάτων και όχι μόνο το τρέχον στίγμα τους (Σχήμα 2.29) αν τικάρει το checkbox “show all track”.
- Ο χάρτης μπορεί να καθαριστεί από προηγούμενα στίγματα πατώντας το κουμπί “Clear Tracks”.
- Ο χάρτης και όλες οι απεικονιζόμενες πληροφορίες μπορούν να «παγωθούν» πατώντας πάνω στο κουμπί που απεικονίζει την κλειδαριά. Όπως αναφέρθηκε και πριν, με αυτό τον τρόπο αν δεν μας νοιάζει η ανανέωση του GUI μπορούμε να εξοικονομήσουμε processing resources. Αυτό είναι ιδιαίτερα χρήσιμο στην περίπτωση που στον ίδιο υπολογιστή τρέχουν περισσότερες από μία εφαρμογές (πχ. περισσότεροι από ένας SCs παράλληλα με τον DFR του συστήματος).

- Κάνοντας δεξί κλικ πάνω στο χάρτη εμφανίζονται διάφορες επιλογές (Σχήμα 2.27-2.29) για αλλαγή του χάρτη, επιλογή ενός μονόχρωμου φόντου και αλλαγή της περιόδου ανανέωσης του GUI.

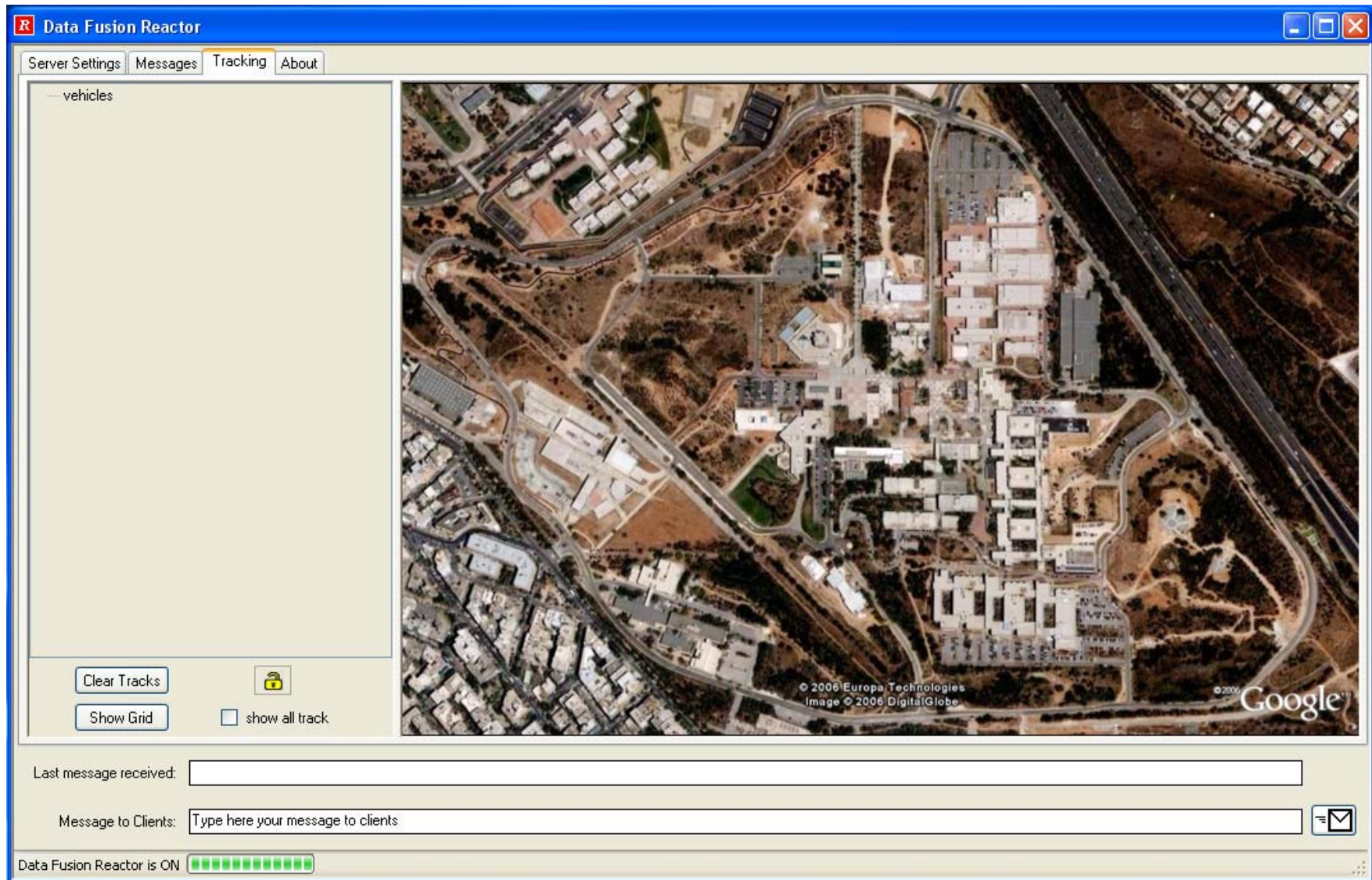
Στην δενδρική δομή στα αριστερά φαίνονται όλες οι συνδεδεμένες συσκευές με το χρώμα που έχει αντιστοιχηθεί στην κάθε μία. Η τροχιάς απεικονίζονται δεξιά στο χάρτη με το ίδιο χρώμα. Ο τύπος της συσκευή φαίνεται απο το “dev X”. Αν το X είναι 0 σημαίνει ότι έχει συνδεθεί συσκευή GPS, ενώ αν είναι 1 συσκευή radar. Στην προκειμένη περίπτωση στο όχημα 0 υπάρχουν μια συσκευή GPS και μια radar αλλά ο χρήστης έχει κάνει λάθος και στη συσκευή radar έχει δώσει την φιλική περιγραφή “uBlox_GPS” που αντιστοιχεί σε μάρκα GPS. Παρόλα αυτά όπως έχει αναφερθεί οι φιλικές περιγραφές εξυπηρετούν μόνο στο να βοηθάνε το χρήστη να ξεχωρίζει τις συσκευές και τα αντίστοιχα αρχεία log και όχι το σύστημα.

Για να δούμε το σφάλμα στην απεικόνιση μπορούμε να κάνουμε τη μέθοδο που περιγράφηκε στην προηγούμενη παράγραφο για την εφαρμογή Sensor Client.

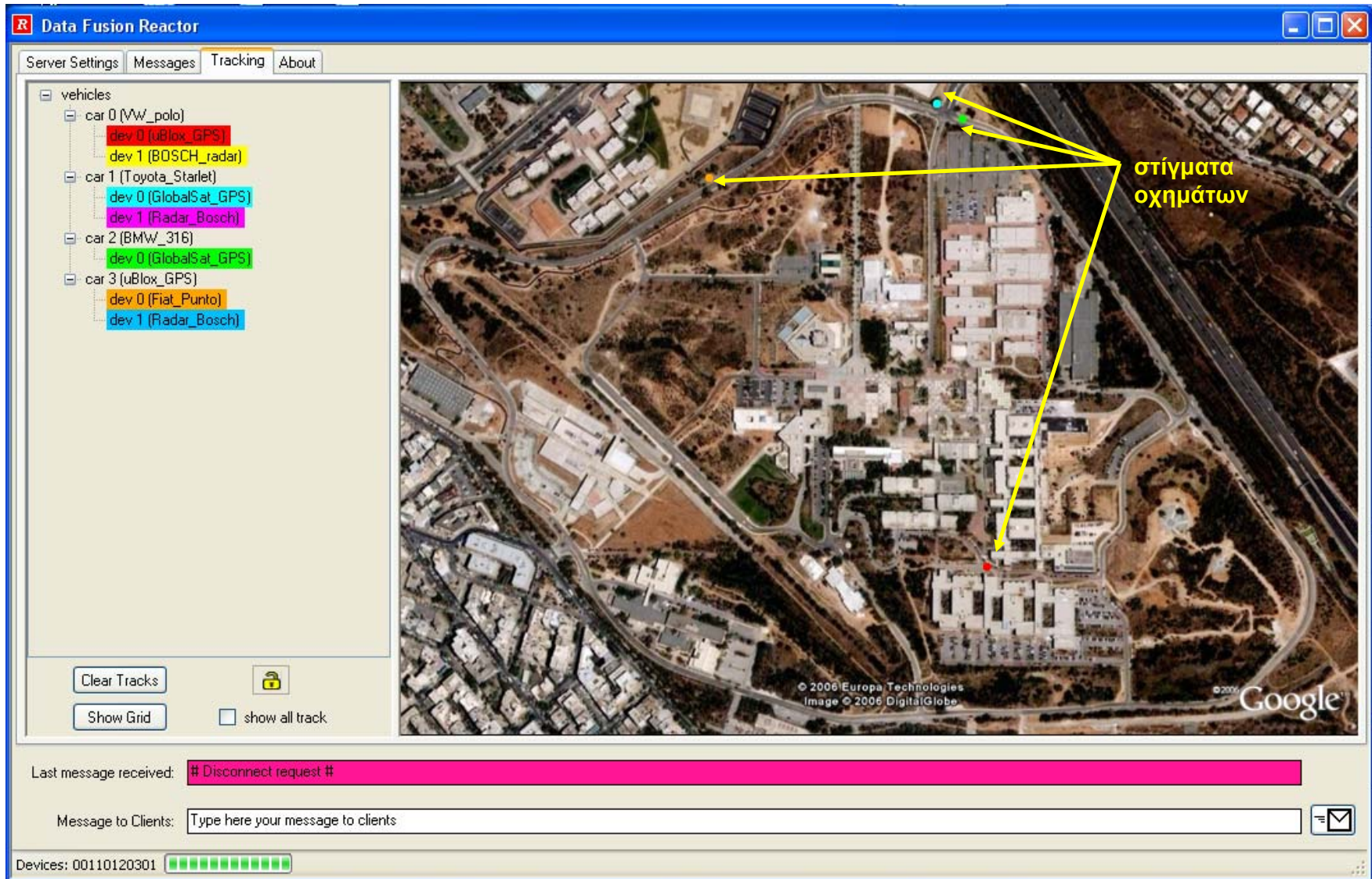
5) About : Πληροφορίες για την ανάπτυξη της εφαρμογής.



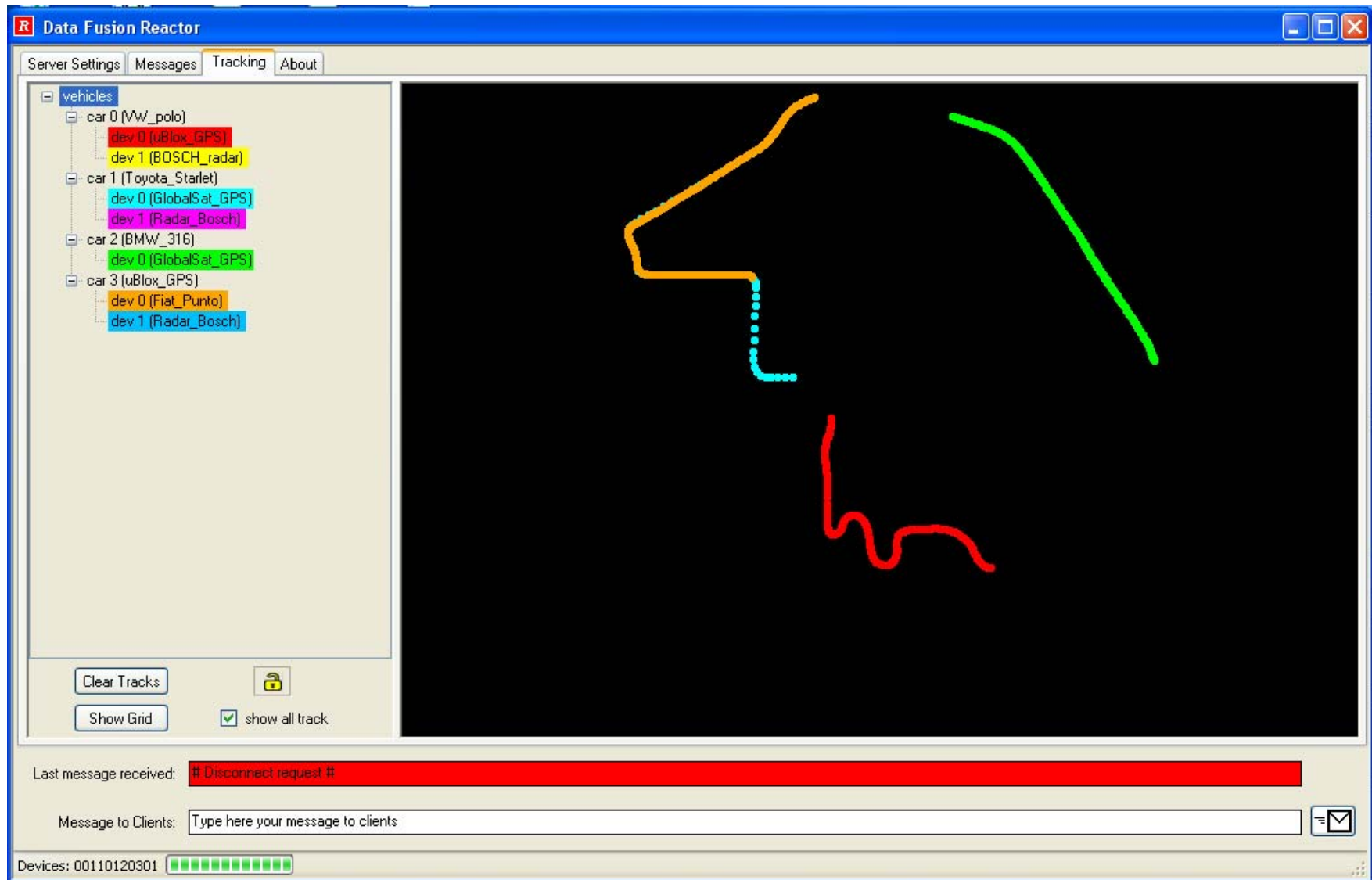
Σχήμα 2.24: Data Fusion Reactor::Messages



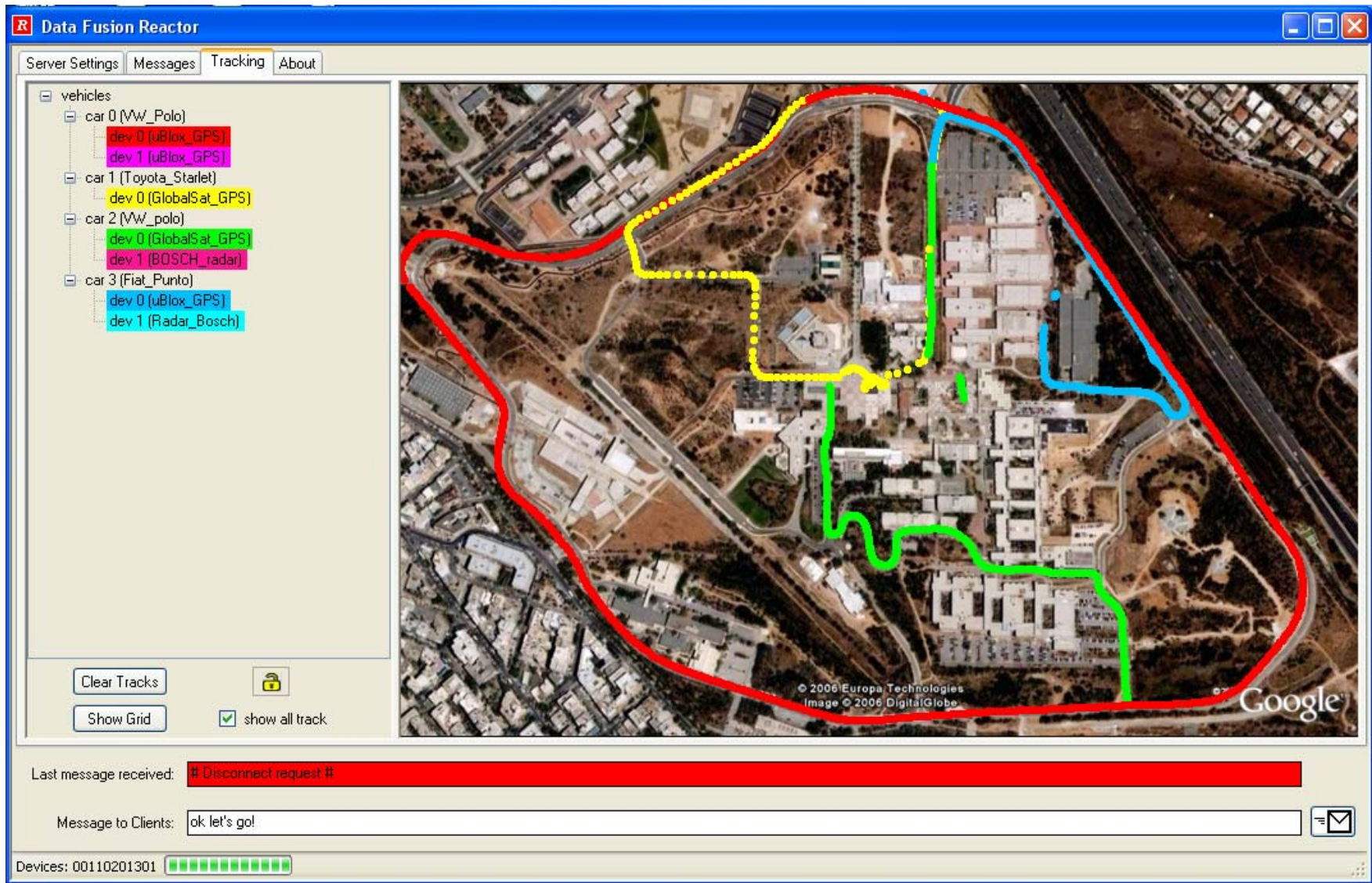
Σχήμα 2.25: Data Fusion Reactor::Tracking



Σχήμα 2.26: Data Fusion Reactor::Tracking. Στην δενδρική δομή στα αριστερά φαίνονται όλες οι συνδεδεμένες συσκευές με το χρώμα που έχει αντιστοιχηθεί στην κάθε μία. Το στίγματα απεικονίζονται δεξιά στο χάρτη με το ίδιο χρώμα.



Σχήμα 2.27: Data Fusion Reactor::Tracking. Στην δενδρική δομή στα αριστερά φαίνονται όλες οι συνδεδεμένες συσκευές με το χρώμα που έχει αντιστοιχηθεί στην κάθε μία. Η πλήρεις τροχιές έχουν σχεδιαστεί δεξιά σε μαύρο φόντο με τα αντίστοιχα χρώματα.



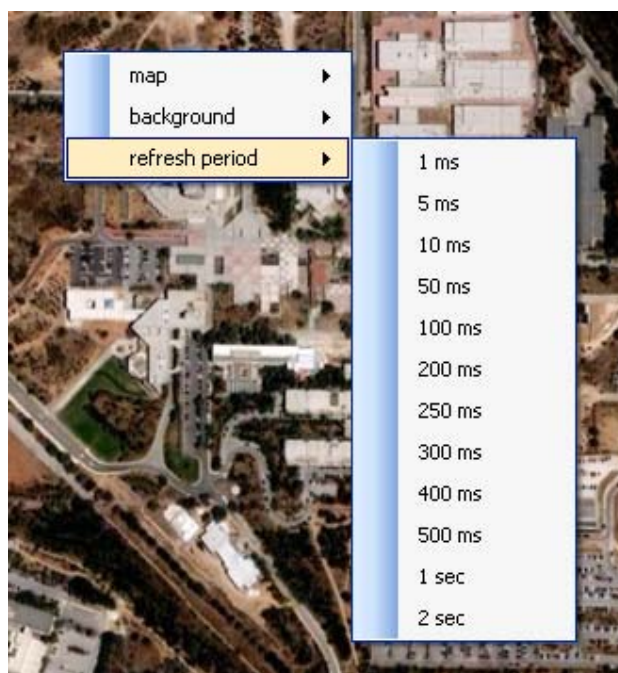
Σχήμα 2.28: Data Fusion Reactor::Tracking. Στην δενδρική δομή στα αριστερά φαίνονται όλες οι συνδεδεμένες συσκευές με το χρώμα που έχει αντιστοιχηθεί στην κάθε μία. Η πλήρεις τροχιές απεικονίζονται δεξιά στο χάρτη με το ίδιο χρώμα.



Σχήμα 2.29



Σχήμα 2.30



Σχήμα 2.31 Data Fusion Reactor::Tracking. Επιλογές που εμφανίζονται κάνοντας δεξί κλικ πάνω στο χάρτη.

2.4 Η εφαρμογή Can Logger

Όπως αναφέρθηκε και προηγουμένως η εφαρμογή Sensor Client στην παρούσα της έκδοση μπορεί να διαβάσει δεδομένα radar μόνο από αρχεία log και όχι απευθείας από τη συσκευή. Ανάγνωση δεδομένων απευθείας από συσκευή radar μπορεί να γίνει μόνο από την εφαρμογή Can Logger που έχει υλοποιηθεί από το I-SENSE. Στην εφαρμογή αυτή έχουν γίνει μόνο κάποιες πολύ μικρές προσθήκες ώστε να προωθεί τα δεδομένα στον DFR.

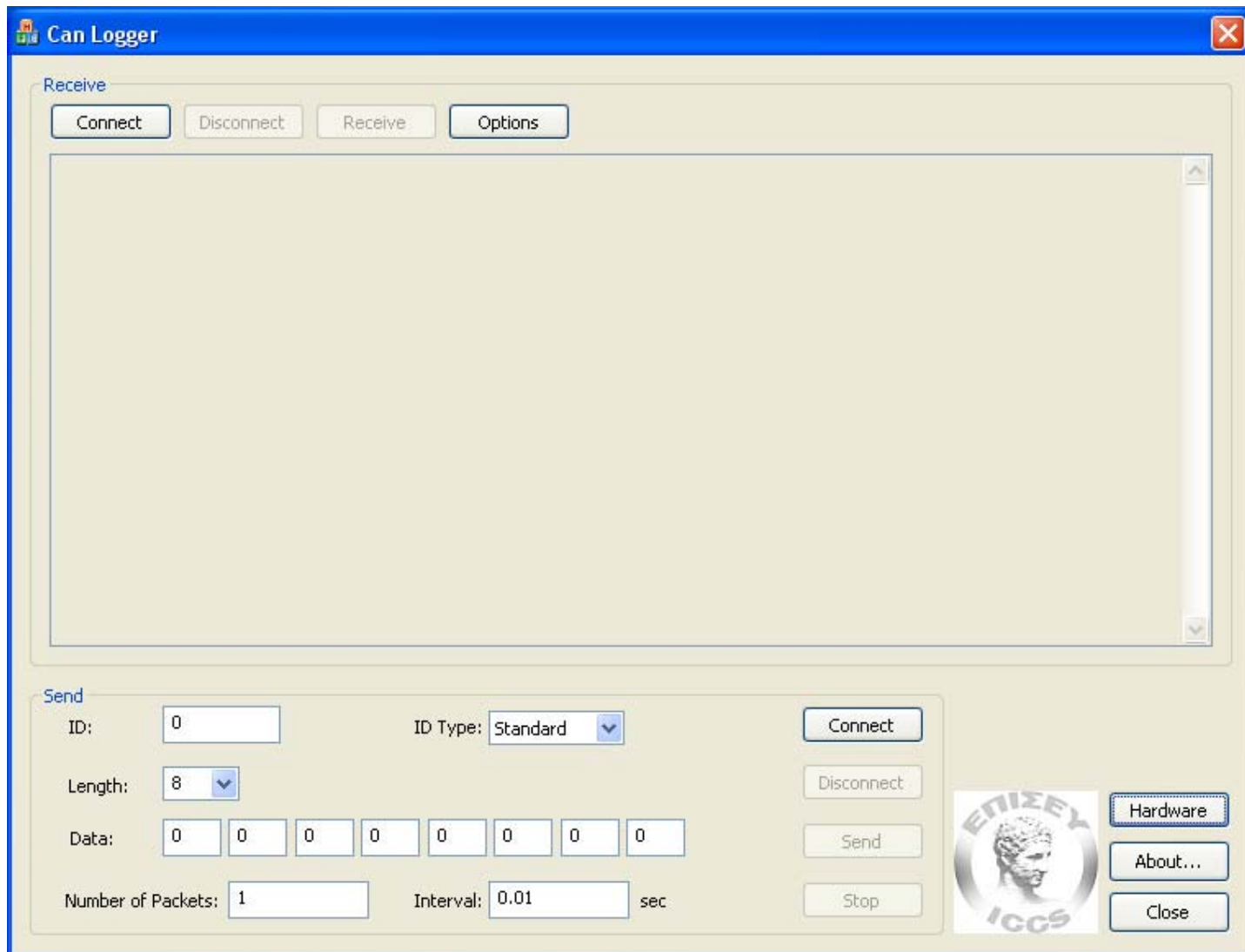
Στην εφαρμογή αυτή ο χρήστης με το που πατήσει το κουμπί “Receive” για να ξεκινήσει η λήψη δεδομένων από τη συσκευή (ή από το ίδιο το πρόγραμμα) επιχειρεί να συνδεθεί με τον DFR (έχει οριστεί με hard coded τρόπο να κάνει σύνδεση στο 127.0.0.1:50000)². Αν καταφέρει να συνδεθεί τότε κάθε πακέτο που θα λαμβάνει θα το προωθεί στον DFR. Συνεπώς πρέπει πρώτα να έχουμε ξεκινήσει τον connections listener του DFR πριν πατήσουμε το κουμπί αυτό.

2.5 Βελτίωση της επίδοσης του συστήματος

Αν στο ίδιο μηχάνημα τρέχουμε πολλούς SCs και ενδεχομένως και τον DFR του συστήματος ενδέχεται το σύστημα να μην μπορεί να επιτύχει την επιθυμητή απόδοση (όπως πχ. στην περίπτωση λήψης μηνυμάτων σαν αυτό του σχήματος 2.7). Στην περίπτωση αυτή μπορούμε να κάνουμε τα εξής:

- Ρύθμιση των SCs ώστε να λαμβάνουν και να προωθούν στον DFR μόνο τα πακέτα που φέρουν χρήσιμες πληροφορίες. Αυτό όπως έχει εξηγηθεί μπορεί να γίνει πατώντας το κουμπί “Set up packets filter”
- Απενεργοποίηση (κλείδωμα) του GUI των SCs.
- Απενεργοποίηση (κλείδωμα) του GUI του DFR ή αύξηση της περιοδου ανανέωσης του.

² Φυσικά θα πρέπει ο Can Logger να τρέχει στον ίδιο υπολογιστή σε σχέση με τον DFR ή αλλιώς να γίνουμε οι απαραίτητες αλλαγές στον κώδικα.



Σχήμα 2.32 Η εφαρμογή CanLogger.

ΚΕΦΑΛΑΙΟ 3: ΑΡΧΕΙΑ LOG

Στο κεφάλαιο αυτό θα παρουσιαστεί η δομή που έχουν τα αρχεία log των συσκευών GPS και radar καθώς και το πως μπορούν να αξιοποιηθούν ώστε να αναπαραχθούν συγχρονισμένα και να μπορεί να επαναληφθεί το πείραμα που έγινε έξω στο δρόμο μέσα στο εργαστήριο. Επίσης θα παρουσιαστεί και ένας απλός τρόπος να κατασκευαστούν τεχνητά πειράματα.

Δομή των αρχείων log για συσκευές GPS

Γενικά όλα τα αρχεία log, τόσο για το GPS όσο και για το radar, έχουν στην αρχή (δύο πρώτες γραμμές) ένα header το οποίο δηλώνει τι τύπος αρχείου log είναι και επίσης περιέχει ένα timestamp για το πότε ξεκίνησε το logging (πόσα δευτερόλεπτα έχουν περάσει από τα μεσάνυχτα 00:00:00 της 1^{ης} Ιανουαρίου 1970) με ακρίβεια millisecond. Αυτό το timestamp είναι απαραίτητο για τη συγχρονισμένη αναπαραγωγή των αρχείων log.

```

Logging started at: 1157026428.812 seconds since midnight (00:00:00), January 1,
1970, coordinated universal time (UTC).
GPS data from Sensor Client:
 0.1720 $GPRMC,121258.00,V,,,,,,310806,,,N*7C
 0.1870 $GPGGA,121258.00,,,,,0,0,99.99,,,,,*5B
 0.4060 $GPRMC,121258.25,V,,,,,,310806,,,N*7B
 0.4220 $GPGGA,121258.25,,,,,0,0,99.99,,,,,*5C
 0.6720 $GPRMC,121258.50,V,,,,,,310806,,,N*79
 0.6720 $GPGGA,121258.50,,,,,0,0,99.99,,,,,*5E
 0.9060 $GPRMC,121258.75,V,,,,,,310806,,,N*7E
 0.9220 $GPGGA,121258.75,,,,,0,0,99.99,,,,,*59
 1.1720 $GPRMC,121259.00,V,,,,,,310806,,,N*7D
 1.1870 $GPGGA,121259.00,,,,,0,0,99.99,,,,,*5A
 1.4060 $GPRMC,121259.25,V,,,,,,310806,,,N*7A
 1.4220 $GPGGA,121259.25,,,,,0,0,99.99,,,,,*5D
.....
.....
56.9530 $GPGGA,121354.75,3758.86545,N,02347.01251,E,1,4,1.71,190.3,M,35.4,M,,*62
57.1720 $GPRMC,121355.00,A,3758.86474,N,02347.01266,E,15.296,0.79,310806,,,A*55
57.2030 $GPGGA,121355.00,3758.86474,N,02347.01266,E,1,4,1.71,190.3,M,35.4,M,,*66
57.4370 $GPRMC,121355.25,A,3758.86451,N,02347.01276,E,15.269,0.35,310806,,,A*5C
57.4530 $GPGGA,121355.25,3758.86451,N,02347.01276,E,1,4,1.71,190.3,M,35.4,M,,*67
57.6720 $GPRMC,121355.50,A,3758.86399,N,02347.01289,E,15.249,0.80,310806,,,A*51
57.7030 $GPGGA,121355.50,3758.86399,N,02347.01289,E,1,4,1.71,190.3,M,35.4,M,,*66
57.9370 $GPRMC,121355.75,A,3758.86374,N,02347.01301,E,15.238,0.58,310806,,,A*57
57.9530 $GPGGA,121355.75,3758.86374,N,02347.01301,E,1,4,1.71,190.3,M,35.4,M,,*63
58.1720 $GPRMC,121356.00,A,3758.86312,N,02347.01316,E,14.981,0.79,310806,,,A*5B

```

Σχήμα 3.1: Δομή των αρχείων log για συσκευές GPS. Στο σχήμα αυτό φαίνεται ότι το header καταλαμβάνει τρεις γραμμές αλλά στην πραγματικότητα καταλαμβάνει δυο, καθώς οι δύο πρώτες γραμμές είναι στη ουσία μια.

Μετά το header ακολουθεί καταγραφή των πακέτων με τη σειρά που λαμβάνονται (Σχήμα 3.1), προσθέτοντας απλώς μπροστά ένα timestamp (σε second) για τη χρονική στιγμή που ελήφθη το αντίστοιχο πακέτο σε σχέση με την έναρξη λήψης του αρχείου log (timestamp του header).

Για να είναι πιο εύκολη και πιο γρήγορη η ανάγνωση των αρχείων log (ώστε να μην απαιτείται ιδιαίτερο συντακτική ανάλυση (parsing) κάθε φορά που διαβάζεται μια γραμμή του) τόσο τα timestamps των πακέτων όσο και τα ίδια τα πακέτα βρίσκονται σε δεδομένη θέση (offset) σε σχέση με την αρχή της γραμμής. Έτσι στην παρούσα υλοποίηση τα timestamps των πακέτων έχουν 4 δεκαδικά ψηφία και μέχρι 7 ψηφία στο ακέραιο τους μέρος. Αυτό σημαίνει ότι η βιβλιοθήκη για ανάγνωση των αρχείων log μπορεί να χρησιμοποιηθεί ως έχει μόνο για την ανάγνωση αρχείων log των οποίων η διάρκεια λήψης διήρκεσε το πολύ 9.999.999 sec ή αλλιώς περίπου 2778 ώρες. Βέβαια σε περίπτωση που απαιτηθεί (!) να ληφθούν μεγαλύτερα πακέτα οι τροποποιήσεις που απαιτούνται στον κώδικα είναι ελάχιστες. Από την άλλη η στοίχιση αυτή οδηγεί σε ελαφρώς μεγαλύτερα αρχεία log, αλλά όμως έτσι αυτά γίνονται πιο «ευανάγνωστα» τόσο από τον άνθρωπο όσο και από το πρόγραμμα.

Η δομή των πακέτων είναι η κλασσική δομή των πακέτων του πρωτοκόλλου NMEA.

Δομή των αρχείων log για συσκευές radar

Όπως αναφέρθηκε και πιο πριν, τα αρχεία log για συσκευές radar όπως και αυτά για συσκευές GPS, έχουν στην αρχή ένα header που δηλώνει τον τύπο του αρχείου log καθώς και τη χρονική στιγμή που ξεκίνησε η λήψη του αρχείου log.

Στην περίπτωση των συσκευών radar υποστηρίζονται δύο διαφορετικοί τύποι από αρχεία log. Τα αρχεία log τύπου I είναι πανομοιότυπα με τα αρχεία log για συσκευές GPS και η μορφή τους φαίνεται στο σχήμα 3.2. Το ότι πρόκειται για αρχείο log συσκευής radar, και μάλιστα τύπου I, φαίνεται από το “RADAR_1” που υπάρχει στη δεύτερη γραμμή του header.

Τα πακέτα αυτά είναι ASCII πακέτα όπως και αυτά του πρωτοκόλλου NMEA, και έγινε προσπάθεια να έχουν όσο το δυνατό πιο κοινή δομή με αυτά, ώστε τα πακέτα όλων των συσκευών να χειρίζονται τόσο από τον SC όσο και από τον DFR με ομοιόμορφο τρόπο. Αυτό καλό είναι να ληφθεί υπ’οψη στην περίπτωση επέκτασης της πλατφόρμας και για άλλες συσκευές.

Πιο συγκεκριμένα η δομή των πακέτων είναι η εξής:

```
$RADAR,6B5,8,422EC40005000000*12
```

- Το πρώτο πεδίο μετά το αναγνωριστικό “\$RADAR” που έχει την τιμή “6B5” είναι το “standard ID” του πακέτου που δηλώνει τον τύπο του. Για περισσότερες πληροφορίες βλέπε το εγχειρίδιο της Bosch για το LLR radar.

```
Logging started at: 1157025843.6870 seconds since midnight (00:00:00), January
1, 1970, coordinated universal time (UTC).
RADAR_1 data from Canlogger:
 0.0940 $RADAR,6D0,8,0000000000000000*12
 0.1880 $RADAR,6D1,8,0000000000000000*13
 0.1880 $RADAR,6D2,8,0000000000000000*10
 0.1880 $RADAR,6D3,8,0000000000000000*11
 0.1880 $RADAR,6D4,8,0000000000000000*16
 0.1880 $RADAR,6D5,8,0000000000000000*17
 0.1880 $RADAR,6D6,8,000000D200330014*67
 0.1880 $RADAR,6B5,8,8326C40000000000*69
 0.2810 $RADAR,6B6,8,0000000000000000*12
 0.2810 $RADAR,6B7,8,0000000000000000*13
.....
210.8750 $RADAR,6D0,8,0000000000000000*12
210.8750 $RADAR,6B5,8,422EC40005000000*12
210.8750 $RADAR,6B9,8,4E6271FAF3FF04FF*18
210.8750 $RADAR,6C1,8,0000000000000000*14
210.9690 $RADAR,6D1,8,0000000000000000*13
210.9690 $RADAR,6B5,8,432EC40004000000*12
210.9690 $RADAR,6B6,8,BA6AE1F9BB0041FF*68
210.9690 $RADAR,6BE,8,0000000000000000*61
211.0470 $RADAR,6CE,8,0000000000000000*60
211.0630 $RADAR,6D6,8,000000EB002E0014*61
211.0630 $RADAR,6B5,8,442EBE0005000000*64
211.0630 $RADAR,6BB,8,0000000000000000*66
```

Σχήμα 3.2: Δομή των αρχείων log τύπου I για συσκευές radar. Στο σχήμα αυτό φαίνεται ότι το header καταλαμβάνει τρεις γραμμές αλλά στην πραγματικότητα καταλαμβάνει δύο, καθώς οι δύο πρώτες γραμμές είναι στη ουσία μια.

- Το δεύτερο πεδίο που έχει την τιμή “8” δηλώνει το μέγεθος του πακέτου σε bytes.
- Το τρίτο πεδίο που έχει την τιμή “422EC40005000000” είναι τα δεδομένα που φέρει το πακέτο σε δεκαεξαδική μορφή. Στην περίπτωση αυτή όπως είπαμε το μέγεθος του πακέτου είναι 8 (bytes) οπότε χρειάζονται 16 (8 ζεύγη) δεκαεξαδικοί αριθμοί για την αναπαράστασή τους. Ο πρώτος δεκαεξαδικός κάθε ζεύγους είναι τα 4 MSBs (Most Significant Bits) του αντίστοιχου byte ενώ ο δεύτερος τα 4 LSBs (Least Significant Bits).

Η δομή αυτή δικαιολογεί τον ισχυρισμό ότι το πρωτόκολλο αυτό που χρησιμοποιήθηκε για τα πακέτα radar είναι ASCII πρωτόκολλο και μάλιστα ότι μοιάζει αρκετά με το NMEA. Ο λόγος που προτιμήθηκε ένα τέτοιο ASCII

πρωτόκολλο για τα αρχεία log, αλλά και για την επικοινωνία των SCs με τον DFR είναι ότι είναι platform independent και ότι δεν έχει προβλήματα με το endianness. Αυτό σημαίνει ότι μπορεί σε άλλη αρχιτεκτονική να πήραμε τα αρχεία log και σε άλλη να τα αναπαράγουμε. Επίσης, ότι σε άλλη αρχιτεκτονική μπορεί να τρέχει(ουν) ο SC(s) και σε άλλη ο DFR. Το γεγονός αυτό κρίθηκε ως ιδιαίτερα σημαντικό και έτσι αποφασίστηκε να θυσιάσει κάποια βελτίωση στην απόδοση του συστήματος που θα επέφερε ένα δυαδικό (binary) πρωτόκολλο.

Τα αρχεία log τύπου II διαθέτουν και αυτά ένα πανομοιότυπο header στις πρώτες δύο γραμμές αλλά έχουν αρκετά διαφορετική δομή (επιλεγμένη έτσι ώστε να διαβάζεται εύκολα μέσα από εφαρμογές MATLAB) για την αποθήκευση των πακέτων (Σχήμα 3.3). Τα αρχεία log αυτά έχουν κατάληξη “.asc” και για περισσότερες πληροφορίες μπορείτε να απευθυνθείτε στο I-SENSE. Η μόνη διαφοροποίηση με τα αρχεία log για radar του I-SENSE είναι η ύπαρξη του header. Έτσι προσθέτοντας χειροκίνητα ένα header στα αρχεία log αυτά μπορούν άμεσα να χρησιμοποιηθούν από την εφαρμογή Sensor Client.

```

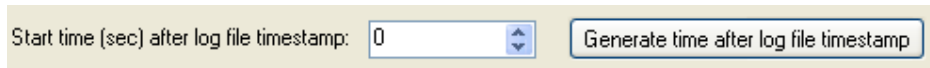
Logging started at: 1153670832.4059999 seconds since midnight (00:00:00),
January 1, 1970, coordinated universal time (UTC).
RADAR_2 data from UNKNOWN:
date Wed Apr 19 05:10:28 pm 2006
base hex timestamps absolute
internal events logged
Begin Triggerblock Wed Apr 19 05:10:28 pm 2006
 0.0000 Start of measurement
 0.0274 1 6B5 Rx d 8 F2 42 C4 00 01 00 00 00
 0.0277 1 6B6 Rx d 8 B2 1E D3 00 9F 01 C4 F9
 0.0279 1 6B7 Rx d 8 00 00 00 00 00 00 00 00
 0.0282 1 6B8 Rx d 8 00 00 00 00 00 00 00 00
 0.0284 1 6B9 Rx d 8 00 00 00 00 00 00 00 00
 0.0287 1 6BA Rx d 8 00 00 00 00 00 00 00 00
.....
16.5222 1 6D4 Rx d 8 00 00 00 00 00 00 00 00
16.5225 1 6D5 Rx d 8 00 00 00 00 00 00 00 00
16.5227 1 6D6 Rx d 8 00 00 00 EF 00 27 00 14
16.6048 1 6B5 Rx d 8 9F 43 C4 00 04 00 00 00
16.6050 1 6B6 Rx d 8 A6 2F 31 FB CA 02 41 00
16.6052 1 6B7 Rx d 8 A3 1D 93 FB C8 FD 42 FF
16.6055 1 6B8 Rx d 8 A4 13 22 01 50 00 43 F9
16.6057 1 6B9 Rx d 8 41 09 E3 00 CF FF 84 FC
16.6060 1 6BA Rx d 8 00 00 00 00 00 00 00 00
16.6062 1 6BB Rx d 8 00 00 00 00 00 00 00 00
16.6065 1 6BC Rx d 8 00 00 00 00 00 00 00 00

```

Σχήμα 3.3 Δομή των αρχείων log τύπου II για συσκευές radar.

Συγχρονισμένη αναπαραγωγή των αρχείων log

Η πλατφόρμα αυτή, όπως έχει αναφερθεί, εκτός από την real time λειτουργία προσφέρει και δυνατότητα για μετέπειτα «off-line» αναπαραγωγή του πειράματος είτε ολόκληρου είτε ενός μέρους του, αντλώντας τα δεδομένα από τα αντίστοιχα αρχεία log που δημιουργήθηκαν κατά τη διάρκεια του πραγματικού πειράματος. Αυτό επιτυγχάνεται αναπαράγοντας τα αρχεία log από έναν υπολογιστή (ή και πολλούς διαφορετικούς) με τρόπο συγχρονισμένο, σύμφωνα με τον τρόπο με τον οποίο αποκτήθηκαν. Αυτό σημαίνει ότι αν κάποιο συγκεκριμένο δεδομένο πχ από το GPS του οχήματος X το πήραμε τη χρονική στιγμή t , ενώ ένα άλλο συγκεκριμένο δεδομένο από το radar του οχήματος Y τη χρονική στιγμή $t+t_0$, τότε πάλι όταν μετέπειτα αναπαράγουμε τα αντίστοιχα αρχεία log τα δεδομένα θα τα πάρουμε με το ένα να έχει την ίδια χρονική καθυστέρηση t_0 ως προς το άλλο. Έτσι μπορούμε να αναπαράγουμε όλο το πείραμα με μια μετάθεση στο χρόνο.



Σχήμα 3.4. Τα components αυτά είναι απαραίτητα για τη συγχρονισμένη αναπαραγωγή των αρχείων log.

Για να το επιτύχουμε αυτό στον Sensor Client (Input File Settings groupbox) υπάρχουν τα components που φαίνονται στο σχήμα 3.4. Αφού έχουμε επιλέξει πατώντας το κουμπί “Browse” ένα κατάλληλο αρχείο log, αν πατήσουμε το κουμπί “Generate time after log file timestamp”, στο numeric up-down component που υπάρχει ακριβώς δίπλα εισάγεται ο χρόνος που έχει περάσει σε second από τη στιγμή που ελήφθη το αρχείο log μέχρι που ο χρήστης πάτησε το κουμπί αυτό.

Για να αναπαραχθούν τώρα τα αρχεία log συγχρονισμένα πρέπει να γίνουν τα εξής:

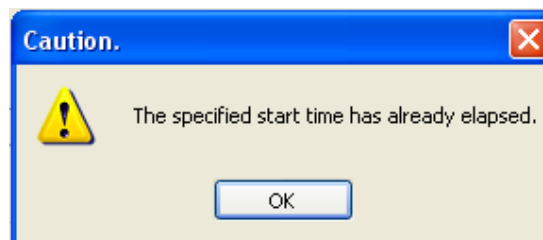
- 1) Επιλέγουμε τα αρχεία log που θέλουμε να τρέξουμε σε όλους τους SCs.
- 2) Πατάμε το κουμπί “Generate time after log file timestamp” σε όλους τους SCs και καθορίζουμε ποια είναι η μεγαλύτερη τιμή που έχει εισαχθεί στο εν λόγω πεδίο.
- 3) Επιλέγουμε τελικά μια τιμή που είναι αρκετά μεγαλύτερη από τη μέγιστη αυτή τιμή κατά ένα χρονικό διάστημα «x» και την εισάγουμε σε

όλους τους SCs. Το χρονικό αυτό διάστημα «x» πρέπει να είναι αρκετό καθώς μέσα σε αυτό πρέπει να έχουμε ολοκληρώσει τα βήματα 2 έως 4.

- 4) Πατάμε το πράσινο κουμπί (play) για να ξεκινήσουν όλοι οι SCs. Έτσι αν τελικά εισάγαμε την τιμή «y», τότε όλοι οι SCs αφού πατήσουμε το play θα ξεκινήσουν να διαβάζουν τα πακέτα που υπάρχουν στο αρχείο log τους όταν θα έχει παρέλθει το χρονικό αυτό διάστημα «y» σε σχέση με το timestamp που υπάρχει στο header τους. Έτσι όλοι οι SCs καθυστερούν αυτόματα κατά διαφορετικό χρονικό διάστημα, ώστε να αρχίσουμε να λαμβάνουμε τα πακέτα τους με την ίδια χρονική διάταξη που τα λαμβάναμε και στο παραγματικό πείραμα.

Αν η τιμή αυτή «y» που επιλέξαμε να εισάγουμε δεν είναι αρκετά μεγάλη και για κάποιο Sensor Client όταν πατήσουμε το play έχει ήδη παρέλθει τότε θα πάρουμε το παρακάτω μήνυμα του σχήματος 3.5. Στην περίπτωση αυτή θα πρέπει να επαναλάβουμε τα βήματα (2) 3 και 4.

Σαν γρήγορη εποπτική επιβεβαίωση ότι τα αρχεία log (για GPS) αναπαράγονται συγχρονισμένα μπορούμε να ελέγξουμε αν τα ρολόγια τους λένε την ακριβώς ίδια ώρα (με εξαίρεση βέβαια την περίπτωση των τεχνητών πειραμάτων).



Σχήμα 3.5: Προειδοποιητικό μήνυμα που εμφανίζεται αν δεν επιλέξουμε αρκετά μεγάλη τιμή για το «y».

Δημιουργία τεχνητών πειραμάτων

Λόγω του τρόπου αυτού με τον οποίο αναπαράγονται συγχρονισμένα τα αρχεία log είναι δυνατή η δημιουργία μη πραγματικών, τεχνητών πειραμάτων και η ακριβής επανάληψη τους. Πχ. μπορούμε να δοκιμάσουμε ένα αλγόριθμο για αποφυγή (μετωπικών) συγκρούσεων διαθέτοντας ένα μόνο όχημα με το οποίο διαγράφουμε

ένα κύκλο της πολυτεχνειούπολης προς τη μια κατεύθυνση, και επείτα (ή και μετά από αυθαίρετα μεγάλο χρονικό διάστημα) προς την άλλη κατεύθυνση. Μετά το τέλος λήψης των αρχείων log αλλάζουμε το timestamp ενός από τα δύο κατάλληλα, ώστε το σύστημα να νομίζει ότι τα είχαμε πάρει ταυτόχρονα και να αναπαράξει το επιθυμητό πλασματικό πείραμα.

Λήψη συγχρονισμένων αρχείων log

Όπως έχει αναφερθεί στο header των αρχείων log φυλάσσεται ένα timestamp για τη χρονική στιγμή που ξεκίνησε το logging. Αν και η απόλυτη τιμή αυτή δεν μας ενδιαφέρει να έχει μεγάλη ακρίβεια (στην πραγματικότητα ίσως να μην μας ενδιαφέρει και καθόλου), μας ενδιαφέρει αν ξεκινήσουμε να παίρνουμε δύο αρχεία log με χρονική διαφορά t τότε τα timestamps τους να έχουν την ακριβώς ίδια χρονική διαφορά t . Μόνο στην περίπτωση αυτή μπορούμε να αναπαράγουμε αργότερα πιστά το πείραμα.

Έτσι, όπως μόλις αναφέρθηκε δε μας ενδιαφέρει η απόλυτη τιμή των timestamps παρά μόνο οι σχετικές τιμές τους. Συνεπώς, αν όλα τα αρχεία log τα παίρνουμε από τον ίδιο υπολογιστή δε μας πειράζει αν ο ρολόι του δεν είναι ρυθμισμένο στη σωστή ώρα καθώς μπορεί οι απόλυτες τιμές των timestamps να έχουν μια απόκλιση από τις πραγματικές αλλά θα έχουν όλες την ίδια αποκλιση και συνεπώς οι σχετικές τιμές (διαφορές) τους θα είναι σωστές.

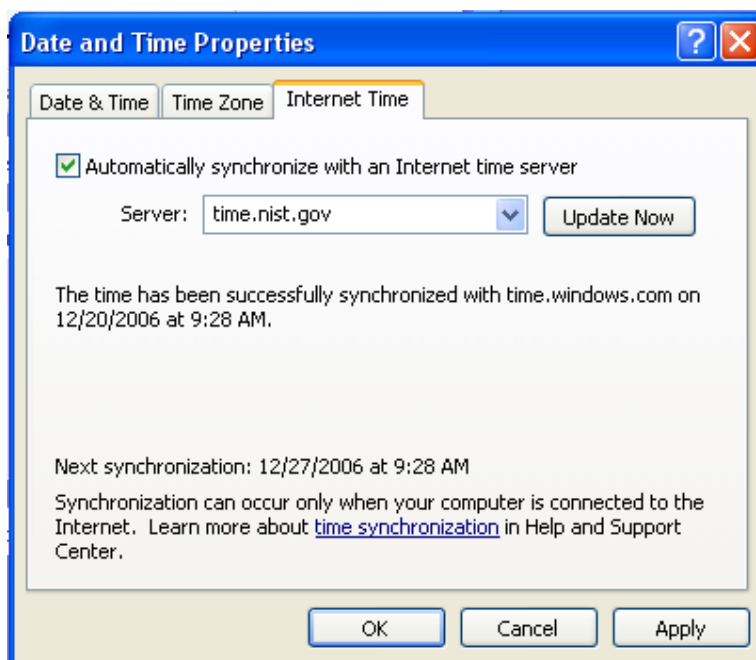
Πρόβλημα προκύπτει όταν τα αρχεία log τα παίρνουμε σε διαφορετικούς υπολογιστές, που είναι και η συνήθης περίπτωση, καθώς μπορεί ο καθένας (και συνεπώς τα αρχεία log που κρατάνε) να έχουν διαφορετική απόκλιση από την πραγματική ώρα. Για το λόγο αυτό, καλό είναι ακριβώς πριν την έναρξη του πειράματος να συγχρονίζουμε τα ρολόγια των υπολογιστών με τον ίδιο³ time server (Σχήμα 3.6).

Βέβαια ακόμα και να μη έχουμε πρόσβαση στον παγκόσμιο ιστό ώστε να συγχρονίσουμε τα ρολόγια των υπολογιστών που συθέτουν το δίκτυο μας με κάποιο time server, μπορούμε να χρησιμοποιούμε τα αρχεία log που κρατάει το DFR. Στην περίπτωση αυτή το timestamp κάθε πακέτου στο timestamp αντιστοιχεί στο χρόνο

³ Ο time sever πρέπει να είναι ο ίδιος καθώς έχει παρατηρηθεί ο time.windows.com να έχει διαφορετική ώρα από τον πιο ακριβή time.nist.gov. Βέβαια στην περίπτωση μας, δεν μας ενδιαφέρει η ώρα των υπολογιστών να είναι σωστή αλλά να έχουν όλη την ίδια. Φυσικά το ίδιο ισχύει και για την ημερομηνία.

λήψης του πακέτου. Τα πακέτα αυτά είναι πάντα συγχρονισμένα μεταξύ τους καθώς λαμβάνονται από τον ίδιο υπολογιστή και συνεπώς τα timestamps βασίζονται στο ίδιο κοινό ρολόι. Στην περίπτωση που ο συγχρονισμός των ρολογιών των υπολογιστών δεν είναι πολύ ακριβής τα αρχεία log αυτά που λαμβάνουμε από τον DFR μπορεί να αποδειχτούν καλύτερα.

Σημειώνεται βέβαια ότι υπάρχουν και ειδικά πρωτόκολλα για το συγχρονισμό των υπολογιστών ενός δικτύου με καλύτερη ακρίβεια από τον απλό τρόπο που υποδείχτηκε προτύτερα. Τέτοιο πρωτόκολλο που διατίθεται για δωρεάν τόσο για συστήματα UNIX όσο και για συστήματα Linux είναι το Network Time Protocol (NTP). Με το πρωτόκολλο αυτό μπορούμε να επιτύχουμε ακρίβεια μερικών μόνο milliseconds.



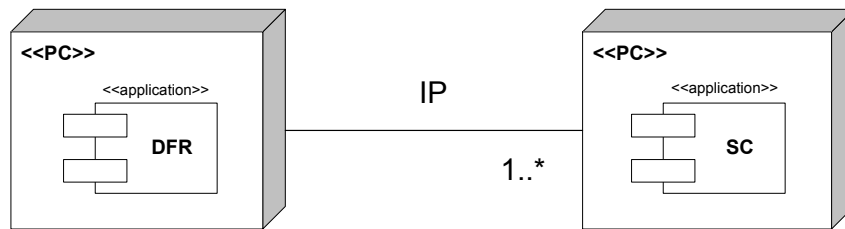
Σχήμα 3.6: Παράθυρο των Windows® για το συγχρονισμό του ρολογιού του υπολογιστή με κάποιο time server.

ΚΕΦΑΛΑΙΟ 4: ΕΓΧΕΙΡΙΔΙΟ ΠΡΟΓΡΑΜΜΑΤΙΣΤΗ

Στο εγχειρίδιο αυτό θα δοθεί μια σύντομη περιγραφή της δομής συστήματος καθώς και άλλες χρήσιμες πληροφορίες που θα βοηθήσουν στην επέκτασή του.

4.1 Πλατφορμική Όψη

Στο Σχήμα 4.1 φαίνονται οι υπολογιστές που αποτελούν την υλισμική πλατφόρμα του συστήματος σύζευξης δεδομένων.



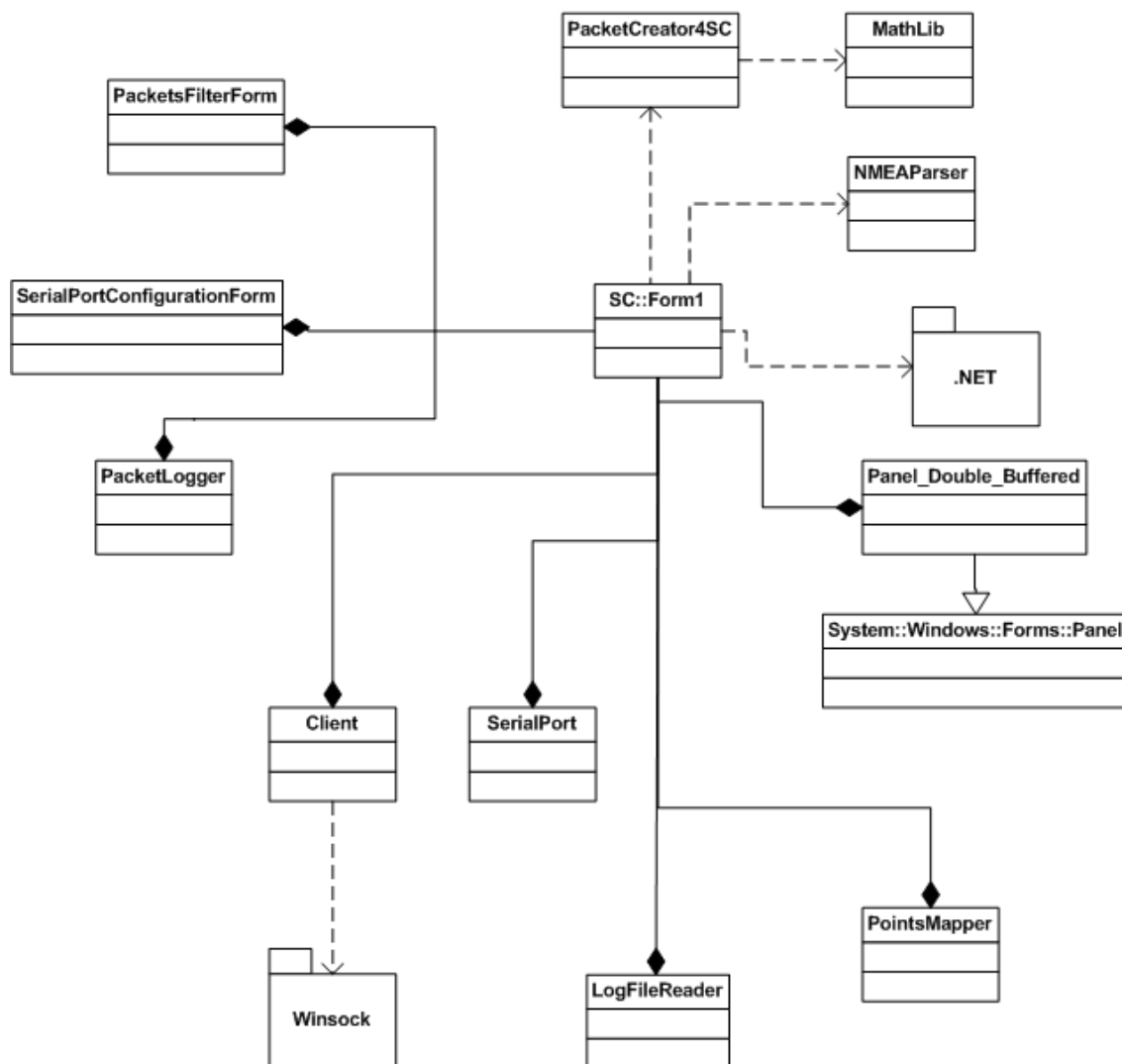
Σχήμα 4.1: Πλατφορμική όψη του συστήματος DF.

4.2 Διαγράμματα κλάσεων

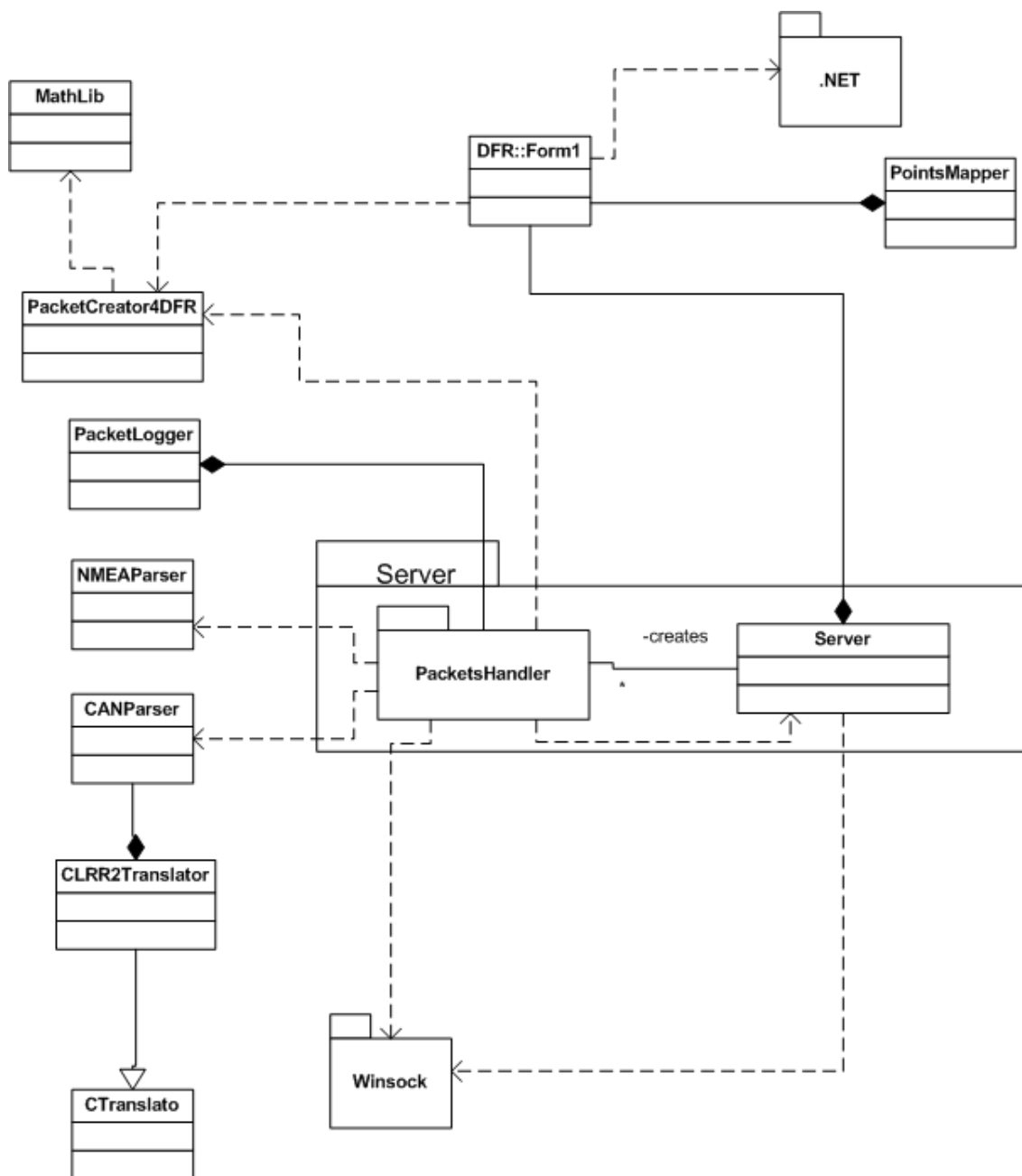
Όπως έχει ήδη αναφερθεί το σύστημα σύζευξης δεδομένων έχει υλοποιηθεί σε Visual C++ .NET και οι δύο εφαρμογές που το συνθέτουν, DFR και SC, είναι τελείως ανεξάρτητες μεταξύ τους. Η δομή της κάθε ανεξάρτητη εφαρμογής συνοψίζεται στο αντίστοιχο διάγραμμα κλάσεων (Σχήματα 4.2, 4.3).

Όπως φαίνεται από το διάγραμμα κλάσεων της εφαρμογής SC (Σχήμα 3.3) η κεντρική κλάση είναι η Form1. Η κλάση αυτή είναι η κλάση του GUI της εφαρμογής και συντονίζει την ολη λειτουργία της χρησιμοποιώντας τις υπόλοιπες κλάσεις που στην πλειοψηφία τους της είναι διαθέσιμες ως στατικές βιβλιοθήκες (.lib αρχεία).

Παρόμοια κατάσταση είναι και στη περίπτωση της εφαρμογής DFR. Στη περίπτωση αυτή, πάλι κεντρικό ρόλο έχει η κλάση του GUI Form1. Όπως έχει αναφερθεί όταν ο χρήστης πατήσει το κουμπί “START Accepting Connections” τότε δημιουργείται ένας connections listener που ανήκει στην κλάση Server. Κάθε φορά που ο listener αυτός δεχτεί μια σύνδεση από έναν Sensor Client δημιουργεί ένα νέο thread (packetsHandler) που χειρίζεται τα εισερχόμενα πακέτα της σύνδεσης αυτής.



Σχήμα 4.2: Διάγραμμα κλάσεων της εφαρμογής SC.



Σχήμα 4.3: Διάγραμμα κλάσεων της εφαρμογής DFR.

Έτσι κάθε τέτοιος handler φροντίζει να αναλύει (parse) τα πακέτα που λαμβάνει, να τα φυλάει σε αρχείο log και να ενημερώνει την κοινή global δομή του DFR, όπου συγκεντρώνονται όλα τα δεδομένα ώστε να είναι διαθέσιμα στο thread που θα κάνει το fusion.

Λόγω του γεγονότος ότι η εν λόγω δομή προσπελαύνεται από πολλά διαφορετικά threads, η πρόσβαση τους σε αυτήν συγχρονίζεται με critical sections για να αποφύγουμε να καταλήξουμε σε ασυνεπείς καταστάσεις. Επίσης, για να επιτύχουμε όσο το δυνατόν μεγαλύτερο παραλληλισμό των threads δεν χρησιμοποιούμε ένα critical section για όλη τη δομή αλλά πολλά. Έτσι, για να κλειδώνουμε την πρόσβαση στα δεδομένα μόνο όταν χρειάζεται, και για όσον το δυνατόν μικρότερο χρονικό διάστημα, χρησιμοποιείται διαφορετικό critical section ανά συσκευή, και φροντίζουμε να βγαίνουμε από αυτό όσο το δυνατό πιο γρήγορα.

Συνεπώς αν ο αλγόριθμος επεξεργασίας είναι βαρύς, μια καλή ιδέα θα ήταν, ανά περίοδο επεξεργασίας να κλειδώνουμε τα δεδομένα της αντίστοιχης συσκευής, να τα αντιγράφουμε κάπου προσωρινά (στην ανάγκη να κρατάμε πλήρες αντίγραφο ολόκληρης αυτής της δομής), μετά κατευθείαν να τα ξεκλειδώνουμε, να επαλαμβάνουμε ενδεχομένως τη διαδικασία αυτή και για άλλες συσκευές και μετά να κάνουμε την επεξεργασία με όλα τα δεδομένα ξεκλειδωμένα ώστε να μπορούν παράλληλα να ενημερώνονται με τις νέες τιμές.

4.3 Επέκταση της πλατφόρμας για επιπλέον συσκευές

Δεδομένου ότι η πλατφόρμα στην παρούσα της έκδοση (Version 1.0) υποστηρίζει δύο συσκευές και όχι μόνο μία, η δομή της είναι τέτοια ώστε να μπορεί να υποστηρίξει εύκολα και άλλες συσκευές. Μέσα στον κώδικα υπάρχουν εντολές ελέγχου (switch statements) που διαφοροποιούν την ροή εκτέλεσης ανάλογα με τον τύπο της συσκευής. Για την υποστήριξη επιπλέον συσκευών πρέπει να επεκταθούν οι εντολές αυτές, να οριστούν επιπλέον αγνωριστικά συσκευών (device IDs) και φυσικά να αναπτυχθούν οι αντίστοιχοι drivers και να οριστεί η μορφή που θα έχουν τα αρχεία log τους καθώς και πακέτα τους όταν προωθούνται στον DFR. Βέβαια καλό είναι να επιλεχτεί πάλι μια μορφή όσο το δυνατό πιο όμοια με τα πακέτα του πρωτοκόλλου NMEA ώστε να υπάρχει ομοιόμορφη αντιμετώπιση τους.

4.4 Υλοποίηση αλγορίθμων σύζευξης δεδομένων στον DFR.

Η πλατφόρμα στην παρούσα της έκδοση είναι πλήρως έτοιμη για την υλοποίηση αλγορίθμων σύζευξης δεδομένων. Μάλιστα έχει οριστεί και το thread που θα κάνει το fusion, και δημιουργείται με την εκκίνηση του DFR.

Μέσα στην κύρια κλάση Form1 του DFR υπάρχει η μέθοδος processReceivedData(). Η μέθοδος αυτή είναι η μέθοδος που θα εκτελεί το thread για το fusion. Η μέθοδος αυτή προτείνει και ένα βασικό σκελετό για την ανάπτυξη αλγορίθμων και κυρίως όσον αφορά στη συγχρονισμένη πρόσβαση στη δομή που συγκεντρώνονται όλα τα δεδομένα.

```
private: void processReceivedData()
{
    while(1) {
        // -----start of data fusion code-----
        updateProcessingProgressBar(0);
        for(int i=0; i<MAX_VEHICLES; i++)
            for(int j=0; j<MAX_DEVICES_PER_VEHICLE; j++){
                EnterCriticalSection(&vehicleDataList[i].devices[j].cs);
                Thread::Sleep(1); //get thread critical data
                LeaveCriticalSection(&vehicleDataList[i].devices[j].cs);
                Thread::Sleep(10); //some processing maybe
                updateProcessingProgressBar(
                    (100*(i*MAX_DEVICES_PER_VEHICLE+j+1))/
                    (MAX_VEHICLES*MAX_DEVICES_PER_VEHICLE));
            }
        //processing -> enter here data processing algorithm
        Thread::Sleep(processPeriod);
        // -----end of data fusion code-----
    }
}
```

Σχήμα 3.5: Η μέθοδος processReceivedData της κλάσης Form1 του DFR που έχει οριστεί να εκτελεί το thread που θα κάνει την επεξεργασία. Η μέθοδος αυτή παρέχεται ως σκελετός για την ανάπτυξη αλγορίθμων DF. Στην παρούσα υλοποίηση της απλώς ανανεώνει ένα progress bar.

ΚΕΦΑΛΑΙΟ 5: ΠΡΟΤΑΣΕΙΣ ΓΙΑ ΒΕΛΤΙΩΣΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ

- Μια εύκολη αλλά σημαντική βελτίωση που μπορεί να γίνει, έχει να κάνει με τη διαχείριση μνήμης του DFR. Πιο συγκεκριμένα στην τρέχουσα υλοποίηση δεσμεύεται μνήμη στατικά για όλες τις πιθανές συσκευές ανά όχημα. Αυτό έχει σαν αποτέλεσμα να δεσμεύεται μνήμη η οποία τελικά μένει πιθανότατα αχρησιμοποίητη. Αυτό γενικά είναι αρκετά κακή πρακτική και ειδικά όσο αυξάνεται ο αριθμός των οχημάτων αλλά και των συσκευών ανά όχημα ενώ παράλληλα δεν αξιοποιούμε παρά μόνο λίγα από αυτά. Η στατική δέσμευση μνήμης προτιμήθηκε για τη φάση του debugging αλλά όπως μόλις ειπώθηκε δεν αποτελεί καλή προγραμματιστική πρακτική. Συνεπώς, προτείνεται να δεσμεύεται μνήμη δυναμικά με το που συνδέεται μια συσκευή και να αποδεσμεύεται με το που αποσυνδέεται.
- Προσθήκη του driver για CAN στον SC ώστε να μπορεί να διαβάζει απευθείας από συσκευές radar και όχι μόνο τα αρχεία log τους.
- Υλοποίηση του πρωτοκόλλου UDP στις εφαρμογές DFR και SC. Η πλατφόρμα αυτή στηρίζομενη στο στοίβα πρωτοκόλλων TCP/IP (ή UDP/IP μελλοντικά) λειτουργεί για κάθε υπόστρωμα προσπέλασης στο μέσο μετάδοσης (802.3, 802.11 κ.τ.λ.). Παρόλα αυτά επειδή αναμένεται να χρησιμοποιηθεί κυρίως πάνω από ασύρματο δίκτυο η χρήση TCP δεν συνίσταται. Στα ασύρματα δίκτυα υπάρχει πολλή πιο υψηλή πιθανότητα απώλειας πακέτων (υψηλό bit error rate) εξαιτίας μείωσης ισχύος του σήματος λόγω απωλειών διάδοσης, παρεμβολές με άλλες πηγές ηλεκτρομαγνητικής ακτινοβολίας και φαινομένου πολλαπλής διόδευσης. Το πρωτόκολλο TCP επειδή δεν είχε σχεδιαστεί λαμβάνοντας υπ' όψη ασύρματα δίκτυα παρερμηνεύει την απώλεια πακέτων ως ένδειξη συμφόρησης στο δίκτυο και μειώνει το ρυθμό μετάδοσης (το παράθυρο μετάδοσης). Επίσης λόγω κινητικότητας κάποιου host παρουσιάζονται μεταβατικές διακοπές λήψης και συνεπώς αλλαγές στο round trip time ενός πακέτου. Το πρωτόκολλο TCP όμως χρησιμοποιεί το RTT⁴ για να καθορίσει την επαναμετάδοση πακέτων. Τέλος το πρωτόκολλο TCP δεν συμπεριφέρεται καλά στη λήψη πακέτων με λανθασμένη σειρά. Για όλους αυτούς του λόγους η χρήση του πρωτοκόλλου UDP ενδείκνυται και κυρίως εφόσον έχουμε να κάνουμε με σύστημα

⁴ RTT: Round Trip Time

πραγματικού χρόνου, στο οποίο πιθανότατα ένα πακέτο που έχει αργήσει πολύ να έρθει μπορεί να μας είναι πλέον άχρηστο και να μην αξίζει να το επαναμεταδώσουμε.

- Τα πρωτόκολλα TCP και UDP έχουν πολύ πιο ισχυρούς μηχανισμούς για έλεγχο της ορθότητας μετάδοσης των πακέτων σε σχέση με το μηχανισμό (XOR checksum) που έχει το NMEA για μετάδοση των πακέτων (μέσω σειριακής, USB κτλ.). Συνεπώς, είναι άσκοπο να ελέγχονται αυτά τα checksums όταν ο DFR (τα PacketsHandler threads) λαμβάνουν ένα πακέτο. Έτσι θα μπορούσαν να καταργηθούν και να εξοικονομήσουμε resources .
- Όπως έχει ειπωθεί ο τρόπος σχεδίασης της πλατφόρμας ευνοεί την προσθήκη υποστήριξης για επιπλέον συσκευές πράγμα που και συνίσταται.
- Λήψη νέου χάρτη δυναμικά από το Google.
- Προσθήκη δυνατότητας για ρύθμιση-απενεργοποίηση/ενεργοποίηση στοιχείων της διαπροσωπείας χρήστη για τη μείωση της επεξεργαστικής ισχύος που δαπανάται σε αυτό σε περίπτωση που υπάρξει ανάγκη για κάτι τέτοιο. Στην παρούσα υλοποίηση παρέχεται δυνατότητα ρύθμισης της περιόδου ανανέωσης του χάρτη του DFR.
- Απαλοιφή όσο το δυνατόν περισσότερου συγχρονισμού κατά την πρόσβαση των threads στα διάφορα συγχρονισμένα κομμάτια της δομής του DFR όπου συγκεντρώνονται τα δεδομένα που λαμβάνονται από τις συσκευές. Αυτό μπορεί να υλοποιηθεί ως εξής: Όταν ένα thread παραγωγός λαβει νέα δεδομένα από τη συσκευή κάνει malloc ένα στοιχείο DevDataElem⁵ και συμπληρώνει τα διάφορα πεδία του. Όταν τελειώσει αλλάζει με μια μοναδική εντολή ανάθεσης το δείκτη του συγκεκριμένου πεδίου της αναφερθείσας δομής. Τα threads καταναλωτές κρατάνε/αντιγράφουν⁶ απλώς με μια μοναδιαία εντολή αναθέσης την τιμή του δείκτη και χρησιμοποιώντας το νέο αυτό δείκτη μπορούν να προσπελαύνουν ένα ένα τα πεδία του χωρίς να χρησιμοποιούν τους υπάρχοντες μηχανισμούς αμοιβαίου αποκλεισμού. Προσοχή πρέπει να δοθεί στην αποδέσμευση μνήμης για τα στοιχεία DevDataElem.
- Τέλος μια πολύ σημαντική βελτίωση που θα επιταχύνει σημαντικά το χρόνο εκτέλεσης της πλατφόρμας είναι η βελτίωση των μεθόδων ανάγνωσης από τη σειριακή θύρα, τα αρχεία log και τα sockets. Στην παρούσα υλοποίηση

⁵ Device Data Element: Μια δομή που περιλαμβάνει τα δεδομένα που λαμβάνονται από μια συσκευή. Στην παρούσα υλοποίηση ονομάζεται "DB_". Η υπάρχουσα δομή "DeviceData_" αντί για ένα "DB_db" πεδίο πρέπει να αλλάξει και να έχει ένα "DB_*db".

⁶ Απαραίτητο να γίνει καθώς μπορεί ένα thread παραγωγός που τρέχει παράλληλα να το αλλάξει.

πραγματοποιείται μια κλήση συστήματος για κάθε byte/χαρακτήρα. Προτείνεται η υλοποίηση μι άλλης μεθόδου (μια για καθε περίπτωση ή μια κοινή) που θα διαβάζει με μια κλήση συστήματος από το stream εισόδου όσα bytes/χαρακτήρες είναι διαθέσιμα και να τα αποθηκεύει σε ένα buffer από τον οποίο θα τροφοδοτεί τις υπάρχουσες μεθόδους ανάγνωσης όταν την καλούν. Η νέα αυτή μέθοδος θα κάνει μια νέα κλήση συστήματος για λήψη από το stream εισόδου μόνο όταν αδειάσει ο buffer της. Ο buffer αυτός ενδείκνυται να έχει μέγεθος μερικών δεκάδων ή και εκατοντάδων bytes. Μια πολύ καλή τέτοια τιμή για το μέγεθος του buffer είναι όσο και το μέσο πακέτο που συνήθως λαμβάνεται. Περισσότερα σχετικά με αυτό μπορούν να βρεθούν στο [8].

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] Εμμανουήλ Τσόγκας, Νικόλαος Φλούδας, Συστήματα Σύζευξης Δεδομένων (Data Fusion) – Ανάλυση και Προσομοίωση Υποσυστημάτων, Σεπτέμβριος 2002. Διπλωματική εργασία.
- [2] Άρης Πολυχρονόπουλος, Σχεδίαση και ανάπτυξη συστήματος σύντηξης δεδομένων από πολλαπλές πηγές με χρήση εκτιμητών Kalman με εφαρμογή στην ανασύνθεση του οδικού περιβάλλοντος, Ιούνιος 2004,.
- [3] www.safespot-eu.org
- [4] Feng Zhao, Leonidas Guibas, Wireless Sensor Networks, An Information Processing Approach, Morgan Kaufmann Publications.
- [5] Andrew Tanenbaum, Δίκτυα Υπολογιστών, Εκδόσεις Παπασωτηρίου.
- [6] Larry Peterson, Bruce Davie, Computer Networks: A System Approach, Morgan Kaufmann.
- [7] Richard Stevens, TCP/IP Illustrated, Volume 1: The Protocols, Addison Wesley Professional (Computing Series).
- [8] Richard Stevens, Bill Fenner, Andrew Rudoff, Unix Network Programming, Volume 1: The Sockets Networking API, Addison Wesley Professional (Computing Series).
- [9] Matthew Gast, 802.11 Wireless Networks – The Definitive Guide, O'REILLY.
- [10] Templeman, Andy Olsen, Microsoft Visual C++ .NET, Julian, Microsoft Press.
- [11] Tom Vercauteren, Dong Guo, Xiaodong Wang, Joint Multiple Target Tracking and Classification in Collaborative Sensor Networks, IEEE Journal on selected areas in communications, Vol. 23, No. 4, April 2005.
- [12] Ullrich Schneunert, Heiko Cramer, Gerd Wanielik, “Precise Vehicle Localization Using Multiple Sensors and Natural Landmarks”.
- [13] Ren C. Luo, Ogst Chen, L.C. Tu, “Nodes Localization through Data Fusion in Sensor Network”. IEEE Proceedings of the 19th International Conference on Advanced Information Networking and Applications (AINA'05).