

Επικοινωνία UDP για Εφαρμογή Ανταλλαγής Μηνυμάτων και Ήχου

Δημήτριος - Χρήστος Κυριάκου, ΑΕΜ: 10842

Στόχοι

Η εφαρμογή έχει ως στόχο την υλοποίηση ενός προγράμματος επικοινωνίας σε Java, η οποία επιτρέπει την ανταλλαγή μηνυμάτων κειμένου και κλήσεων ήχου σε πραγματικό χρόνο, μέσω των IP διευθύνσεων και των θυρών τους. Η επικοινωνία πραγματοποιείται μέσω του πρωτοκόλλου UDP, χρησιμοποιώντας DatagramSocket για αποστολή και λήψη δεδομένων, διασφαλίζοντας peer-to-peer επικοινωνία. Ακόμη τα δεδομένα ήχου κωδικοποιούνται και αποκωδικοποιούνται με χρήση Java Sound API.

Ανάλυση κώδικα

Αρχικά όσον αφορά τη γραφική διεπαφή (GUI), η εφαρμογή παρέχει ένα παράθυρο που περιλαμβάνει πεδία κειμένου (TextField), ένα πεδίο κειμένου πολλαπλών γραμμών (TextArea) για την εμφάνιση των μηνυμάτων και δύο κουμπιά ("Send" και "Call"). Το sendButton επιτρέπει την αποστολή μηνυμάτων κειμένου, ενώ το callButton ξεκινά και τερματίζει τη σύνδεση ήχου μεταξύ των δύο χρηστών.

Τώρα οι βιβλιοθήκες που χρησιμοποιούνται είναι:

- **java.net:** Χρησιμοποιείται για τη δημιουργία sockets και την ανταλλαγή δεδομένων μέσω UDP.
- **javax.sound:** Χρησιμοποιείται για την πρόσβαση σε συσκευές ήχου (μικρόφωνο και ηχεία) και την επεξεργασία δεδομένων ήχου.

Στην υλοποίηση του προγράμματος χρησιμοποιείται το πρωτόκολλο UDP για την επικοινωνία, λόγω της χαμηλής καθυστέρησης και της υποστήριξης real-time

εφαρμογών. Ακόμη η θύρα 12345 χρησιμοποιείται για τα μηνύματα κειμένου και η θύρα 12346 για τη φωνητική επικοινωνία.

Αναφορικά με τα λειτουργικά μέρη του προγράμματος :

Αποστολή και λήψη μηνυμάτων

Η αποστολή και λήψη μηνυμάτων πραγματοποιείται μέσω του πρωτοκόλλου UDP, χρησιμοποιώντας την κλάση DatagramSocket. Όταν ο χρήστης εισάγει ένα μήνυμα στο πεδίο κειμένου και πατήσει "Send", το μήνυμα μετατρέπεται σε byte array και αποστέλλεται στον peer χρήστη:

```
150          // Send message via textSocket
151          byte[] buffer = message.getBytes();
152          DatagramPacket packet = new DatagramPacket(buffer, buffer.length, peerAddress, peerPort);
153          textSocket.send(packet);
```

Η λήψη μηνυμάτων υλοποιείται σε ξεχωριστό νήμα. Το DatagramSocket λαμβάνει τα πακέτα και εμφανίζει το περιεχόμενο στο GUI:

```
103          byte[] buffer = new byte[1024];
104          DatagramPacket packet = new DatagramPacket(buffer, buffer.length);
105          // Receive packet on textSocket
106          textSocket.receive(packet);
107          // Convert to String and append to textArea
108          String message = new String(packet.getData(), 0, packet.getLength());
109          textArea.append("Peer: " + message + newline);
```

Αμφίδρομη φωνητική επικοινωνία (VoIP)

Κατά την εκκίνηση μιας φωνητικής κλήσης (πατώντας "Call"), δημιουργούνται δύο νήματα, το ένα για την αποστολή και το άλλο για τη λήψη ήχου από τον χρήστη. Το μικρόφωνο και τα ηχεία ανοίγουν και αρχίζουν να μεταδίδουν δεδομένα χρησιμοποιώντας τον ακόλουθο κώδικα:

```
167          microphone = (TargetDataLine) AudioSystem.getLine(micInfo);
168          speakers = (SourceDataLine) AudioSystem.getLine(speakerInfo);
169
170          microphone.open(format);
171          speakers.open(format);
172
173          microphone.start();
174          speakers.start();
```

Τα δεδομένα ήχου διαβάζονται από το μικρόφωνο και αποστέλλονται ως πακέτα UDP:

```
186          int bytesRead = microphone.read(buffer, 0, buffer.length);
187          DatagramPacket packet = new DatagramPacket(buffer, bytesRead, peerAddress, audioPort);
188          audioSocket.send(packet);
```

Παρόμοια, τα πακέτα UDP που λαμβάνονται μέσω του `audioSocket` αποκωδικοποιούνται και αναπαράγονται από τα ηχεία:

```
206 DatagramPacket packet = new DatagramPacket(buffer, buffer.length);
207 audioSocket.receive(packet);
208 speakers.write(packet.getData(), 0, packet.getLength());
209 }
```

Τέλος, η χρήση των `Threads` εξασφαλίζει ότι η αποστολή/λήψη δεδομένων και η διαχείριση του GUI γίνονται ανεξάρτητα ώστε να αποφευχθεί η επικάλυψη της πληροφορίας.

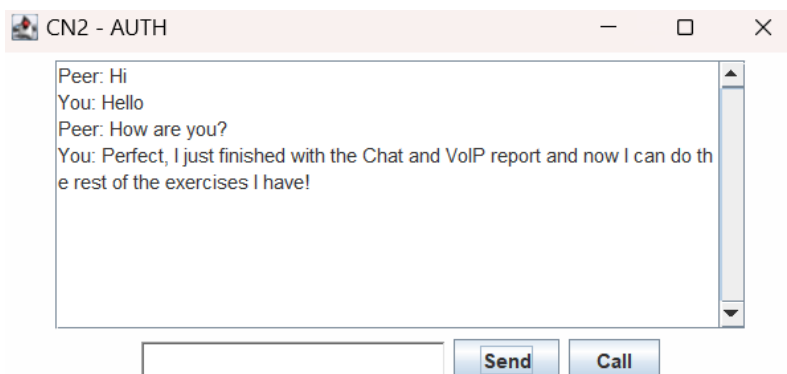
Ανάλυση με Wireshark

Η ανάλυση περιλαμβάνει την καταγραφή και την επιθεώρηση της κίνησης δικτύου με τη χρήση του Wireshark, με έμφαση στην ανταλλαγή μηνυμάτων κειμένου και τη ροή φωνητικών κλήσεων.

Ανταλλαγή Μηνυμάτων Κειμένου

Τα μηνύματα κειμένου αποστέλλονται μέσω πακέτων UDP στη θύρα 12345. Κάθε πακέτο περιέχει το φορτίο (payload) του μηνύματος, το οποίο μπορεί να εξαχθεί και να επιθεωρηθεί για τα δικτυακά headers και το περιεχόμενό του σε μορφή κειμένου.

Αρχικά χρησιμοποιείται το φίλτρο `udp.port == 12345` για την απομόνωση της θύρας UDP.



Λεπτομέρειες Κεφαλίδας Δικτύου:

- Πρωτόκολλο: UDP
- Πηγή IP και Θύρα του αποστολέα/ παραλήπτη

Current filter: udp.port == 12345

No.	Time	Source	Destination	Protocol	Length	Info
1145...	-468.226313	192.168.1.116	192.168.1.107	UDP	44	12345 → 12345 Len=2
1155...	-309.684018	192.168.1.107	192.168.1.116	UDP	47	12345 → 12345 Len=5
1203...	-295.723817	192.168.1.116	192.168.1.107	UDP	54	12345 → 12345 Len=12
1209...	-167.470246	192.168.1.107	192.168.1.116	UDP	147	12345 → 12345 Len=105

Frame 120990: 147 bytes on wire (1176 bits), 147 bytes captured (1176 bits) on interface \Device\NPF...
 Ethernet II, Src: Intel_ab:8a:9c (a8:7e:ea:ab:8a:9c), Dst: Intel_51:30:4c (e4:fd:45:51:30:4c)
 Internet Protocol Version 4, Src: 192.168.1.107, Dst: 192.168.1.116
 User Datagram Protocol, Src Port: 12345, Dst Port: 12345
 Data (105 bytes)

wireshark - Wi-Fi0UYPY2.ncannq Packets: 132167 · Displayed: 4 (0.0%) Profile: Default

Περιεχόμενο:

- Δεκαεξαδική αναπαράσταση του μηνύματος κειμένου.
- Αναπαράσταση σε ASCII για ευανάγνωστη μορφή.

Wireshark · Follow UDP Stream (udp.stream eq 806) · Wi-Fi

00000000	48 69	Hi
00000000	48 65 6c 6c 6f	Hello
00000002	48 6f 77 20 61 72 65 20 79 6f 75 3f	How are you?
00000005	50 65 72 66 65 63 74 2c 20 49 20 6a 75 73 74 20	Perfect, I just
00000015	66 69 6e 69 73 68 65 64 20 77 69 74 68 20 74 68	finished with th
00000025	65 20 43 68 61 74 20 61 6e 64 20 56 6f 49 50 20	e Chat and VoIP
00000035	72 65 70 6f 72 74 20 61 6e 64 20 6e 6f 77 20 49	report and now I
00000045	20 63 61 6e 20 64 6f 20 74 68 65 20 72 65 73 74	can do the rest
00000055	20 6f 66 20 74 68 65 20 65 78 65 72 63 69 73 65	of the exercise
00000065	73 20 49 20 68 61 76 65 21	s I have !

Ροή κλήσεων Φωνής

Οι κλήσεις φωνής υλοποιούνται ως ροές πραγματικού χρόνου πακέτων UDP που ανταλλάσσονται μεταξύ χρηστών μέσω της θύρας 12346. Τα δεδομένα ήχου κωδικοποιούνται και μεταδίδονται σε μικρά πακέτα για συνεχή αναπαραγωγή.

Αρχικά χρησιμοποιείται το φίλτρο `udp.port == 12346` για την απομόνωση της θύρας UDP.

Λεπτομέρειες Κεφαλίδας Δικτύου:

- Πρωτόκολλο: UDP
- Πηγή IP και Θύρα του αποστολέα/ παραλήπτη
- Μέγεθος πακέτου και χρονικές σημάνσεις.

udp.port == 12346

No.	Time	Source	Destination	Protocol	Length	Info
945	28.484145	192.168.1.107	192.168.1.116	QUIC	554	Protected Payload (KP0)
949	28.606809	192.168.1.116	192.168.1.107	QUIC	554	Protected Payload (KP0)
950	28.607419	192.168.1.116	192.168.1.107	QUIC	554	Protected Payload (KP0)
951	28.609079	192.168.1.107	192.168.1.116	QUIC	554	Protected Payload (KP0)
952	28.609295	192.168.1.107	192.168.1.116	QUIC	554	Protected Payload (KP0)
967	28.734028	192.168.1.116	192.168.1.107	QUIC	554	Protected Payload (KP0)
968	28.734561	192.168.1.107	192.168.1.116	QUIC	554	Protected Payload (KP0)
969	28.734886	192.168.1.107	192.168.1.116	QUIC	554	Protected Payload (KP0)

▶ Frame 999: 554 bytes on wire (4432 bits), 554 bytes captured (4432 bits) on interface \Device\NPF_{C732E5B9-191B-4B0A-9E31-02322B786BB0}, id 0
 ▶ Ethernet II, Src: Intel_51:30:4c (e4:fd:45:51:30:4c), Dst: Intel_ab:8a:9c (a8:7e:ea:ab:8a:9c)
 ▶ Internet Protocol Version 4, Src: 192.168.1.116, Dst: 192.168.1.107
 ▶ User Datagram Protocol, Src Port: 12346, Dst Port: 12346
 ▶ QUIC IETF

Λεπτομέρειες Πακέτου:

- Πληροφορίες κεφαλίδας δικτύου παρόμοιες με τα μηνύματα κειμένου.
- Περιεχόμενο πακέτου σε δυαδική ή δεκαεξαδική μορφή.

Wireshark · Follow UDP Stream (udp.stream eq 8) · Wi-Fi

000020C0	01 00 00 ff 00 00 00 00	00 00 00 00 00 01 00 01
000020D0	00 00 01 ff 01 00 00 00	ff ff 00 00 00 ff 00 ff
000020E0	00 00 00 ff 00 00 00 00	01 00 00 00 00 00 00 01
000020F0	00 01 00 00 00 00 00 00	01 00 00 00 01 01 00 00
00002100	00 00 00 00 00 00 ff 01	00 00 00 00 00 00 00 01
00002110	00 00 ff 00 00 00 ff ff	00 00 01 00 ff 00 00 00
00002120	00 00 00 ff ff 00 00 ff	01 00 01 00 00 00 00 00
00002130	00 01 00 00 00 ff 00 00	00 00 00 01 ff 00 00 00
00002140	00 00 00 00 00 00 01 00	00 00 00 00 00 ff 00 00
00002150	ff 00 00 ff 00 00 00 00	00 00 00 00 01 00 00 ff
00002160	01 00 00 ff ff 00 00 00	01 ff 00 00 00 ff 01 00
00002170	00 00 01 ff 00 00 00 00	00 00 00 00 00 00 00 ff
00002180	00 00 00 00 00 00 00 00	00 00 01 ff 00 00 ff 01
00002190	00 00 ff 00 00 00 00 00	00 00 00 00 00 00 ff 00
000021A0	00 00 00 00 00 00 ff 00	00 00 ff 00 00 ff 00 00
000021B0	00 00 00 00 01 01 ff 00	00 00 00 00 00 00 00 00
000021C0	00 01 00 00 00 00 ff 00	00 00 01 00 00 00 00 00
000021D0	00 01 00 00 00 00 00 ff	00 ff 01 00 01 00 00 01
000021E0	00 00 01 00 00 00 00 ff	ff 01 00 00 00 ff 00 00
000021F0	01 00 00 00 00 00 00 ff	01 00 00 ff 00 00 01 01
00007E00	00 00 ff 00 00 00 00 ff	00 00 00 00 00 00 00 00
00007E10	00 00 00 00 00 00 ff 00	00 00 00 00 00 00 00 00
00007E20	01 00 ff 00 00 00 00 00	00 00 00 01 00 00 00 00
00007E30	00 00 00 00 00 00 ff 00	00 00 00 00 00 00 00 00
00007E40	00 00 00 00 01 00 00 00	00 00 00 00 01 00 00 00
00007E50	01 01 00 00 00 ff 01 01	00 00 00 00 00 00 00 01
00007E60	00 00 00 01 00 00 00 00	00 ff 00 00 00 00 01 00
00007E70	00 00 00 00 ff ff ff 00	00 00 00 ff 00 00 00 00
00007E80	00 00 00 00 00 01 00 01	00 00 00 00 00 00 01 ff
00007E90	00 01 ff 00 00 00 00 00	00 00 ff 00 00 00 ff 00
00007EA0	ff 00 00 00 00 00 ff 00	00 00 00 00 00 ff 00 00
00007EB0	00 00 00 00 00 00 ff ff	00 00 00 00 00 01 00 00
00007EC0	01 00 00 ff 00 00 00 00	00 00 00 00 00 01 00 01
00007ED0	00 00 01 ff 01 00 00 00	ff ff 00 00 00 ff 00 ff

Συμπέρασμα

Η ανάλυση επιβεβαιώνει την επιτυχή μετάδοση δεδομένων κειμένου και ήχου μέσω UDP. Τα χαρακτηριστικά χαμηλής καθυστέρησης του UDP το καθιστούν κατάλληλο για αυτόν τον τύπο εφαρμογής. Τα καταγεγραμμένα πακέτα επιβεβαιώνουν ότι τα δεδομένα μεταδίδονται όπως αναμένεται, με σαφείς κεφαλίδες και περιεχόμενα που είναι ορατά στο Wireshark.

