



ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

Τμήμα Πληροφορικής & Τηλεματικής



"ΠΙΛΟΤΙΚΗ ΕΦΑΡΜΟΓΗ ΓΙΑ MULTIPOINT ΣΥΝΔΕΣΗ ΜΕ ΤΗΝ ΧΡΗΣΗ ΤΟΥ WEB RTC"

Μακρυωνίτης Χρήστος

A.M: 13207

Επιβλέπων Καθηγητής: **Δημητρακόπουλος Γ.**

Εξεταστική Επιτροπή: **Δημητρακόπουλος Γ.**

Νικολαΐδου Μ.

Μιχαλακέλης Χ.

Αθήνα 2015

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

Περιεχόμενα

Εισαγωγή	5
Κίνητρο	5
Ιστορική Αναδρομή:	5
Σκοπός της εργασίας	6
Δομή της εργασίας	6
Το τεχνολογικό υπόβαθρο του WebRTC.....	7
1 HTML	7
1.1 Εισαγωγή	7
1.2 Σημασιολογία.....	7
1.3 Νέα χαρακτηριστικά.....	7
2 JAVASCRIPT	10
2.1 Εισαγωγή.....	10
2.2 Μοντέλο Εκτέλεσης.....	11
3 Πρωτόκολλα, γενική περιγραφή.....	13
IP	13
TCP	13
UDP.....	14
TLS	15
DTLS	15
ICE	15
STUN	16
TURN	16
SDP	17
SRTP και SRTCP	17
HTTP.....	18
P2P	19
1 Εισαγωγή.....	19
2 Συγκεντρωτικά P2P δίκτυα.....	19
3 Αποκεντρωτικά P2P δίκτυα	19
4 P2P δίκτυα τρίτης γενιάς	20
WebSocket Protocol.....	20
4 WebRTC	22

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

4 1 Τι είναι το WebRTC.....	22
4 2 Η Αρχιτεκτονική του WebRTC	22
4 3 Σενάρια και τρόποι λειτουργίας του Webrtc	25
4 3 1 ΕΦΑΡΜΟΓΗ	25
4 3 2 Αρχικοποίηση	26
4 3 3 Μεταφορά Δεδομένων	27
4 4 WebRTC MULTIPOINT CONNECTION – ΜΕΡΙΚΑ CONFERENCE MODELS ..	28
4 4 1 Ανάπτυξη του WEB RTC BROWSER για μικρά GROUP CONFERENCE.....	28
4 4 2 Μηχανισμός Σηματοδότησης	28
4 4 3 Ανάπτυξη του WEB RTC BROWSER για μεγάλα GROUP CONFERENCE.....	30
4 4 4 Περίπτωση με VoIP Conference	30
4 4 5 Περίπτωση με MMoIP Conference	31
4 5 Η τεχνολογία WebRTC σε κινητές συσκευές	32
5 ASR (automatic speech recognition).....	35
5 1 Speech Recognition.....	35
5 2 Τι είναι η αυτόματη αναγνώριση ομιλίας ASR;	35
5 3 Η Ιστορία της ASR Τεχνολογίας.....	36
5 4 Πως δουλεύει το ASR	38
5 4 1 METHODS, AND ALGORITHMS.....	38
Hidden Markov models(Hidden Markov models).....	38
Δυναμική χρονική στρέβλωση (DTW) αναγνώριση της ομιλίας(Dynamic time warping (DTW)-based speech recognition).....	38
Τα νευρωνικά δίκτυα(Neural networks).....	39
Βαθιά Νευρωνικά Δίκτυα και άλλων Μοντέλων βαθιάς Μάθησης(Deep Neural Networks and Other Deep Learning Models).....	39
5 5 Ποιο είναι το όφελος του ASR.....	41
5 6 Τι έχει συμβεί σε ASR	41
5 6 1 Applications.....	41
5 7 The HTML5 Speech Recognition API	42
Basic usage.....	42
Streaming results.....	43
Accounting for accents.....	43
Security	43

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

6 Διαθέσιμες «Multiconference»εφαρμογές	44
6.1 Skype	44
Χρήσιμα στοιχεία skype.....	44
6.2 Viber	44
6.3 ooVoo	45
6.4 FaceTime	45
Πρωτόκολλα.....	46
6.5 Clear Sea	46
Clear Sea πρωτόκολλα	46
7. Πιλοτική εφαρμογή	47
7.1 Υλοποίηση.....	47
Real time connection	48
Translate and ASR	50
Buttons-Δημιουργίας.....	51
Έναρξη του session room	51
Επιλογή επικοινωνίας της Σύνδεσης	52
Αριθμός Εμφάνισης Χρηστών	53
Session Room.....	55
Εκκίνηση τοπικού video	56
Σύνδεση.....	57
Button	57
Video	58
Frame	58
Buttons.....	58
Textarea-translate «Dialogs»	60
Textarea- «chat-output» και input- «chat-input» « Text Chat»	61
Input «file» « Text Chat»	63
«JOIN»	64
7.2 Console log στην σύνδεση	65
Εύρεση IceServer	66
Δημιουργία Datachannel	67
Δημιουργία SDP σύνδεσης	68
8 Περιγραφή περιβάλλοντος	70

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

8 1 Εικόνα δημιουργίας Room ή Session.	70
8 2 Ερώτηση αποδοχής Video.....	71
8 2 Ερώτηση αποδοχής Audio	71
8 3 Ενεργή σύνδεση	72
8 4 Ενεργή σύνδεση-2.....	74
8 5 Ερώτηση αποδοχής Audio για “audio και data”	75
8 6 Ενεργή σύνδεση “audio και data”	75
8 7 Σύνδεση μέσω FireFox.....	76
8 8 Σύνδεσης.....	77
8 8 1 Multiconference	77
8 8 2 Dialogs	78
8 8 3 Text Chat.....	78
8 8 4 Share File	78
8 9 Σύνδεση 5 ^{ov} συσκευών.....	79
8 10 Σύνδεση 4 ^{ov} συσκευών.....	79
8 11 Σύνδεση 2 συσκευών	80
8 12 Σύνδεση 2 συσκευών	81
8 13 Σύνδεση 2 συσκευών	81
9 Επισκόπηση.....	82
9.1 Συμπεράσματα	82
9.2 Προτεινόμενη μελλοντική μελέτη και πιθανές επεκτάσεις	82
Βιβλιογραφία.....	84

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

Εισαγωγή

Κίνητρο

Ο τομέας της ραγδαίας εξελισσόμενης τεχνολογίας αποτελεί πόλο έλξης ενδιαφέροντος πολλών ερευνητών. Στην περίπτωση μας το ενδιαφέρον μας εστιάστηκε στην πρόσφατα αναπτυχθείσα προηγμένη τεχνολογία του WebRTC η οποία παρέχει την δυνατότητα ανάπτυξης εφαρμογών η χρήση των οποίων διευκολύνει την επικοινωνία μας σε πολλαπλά επίπεδα συνδυάζοντας τόσο την ποιότητα της παρεχόμενης υπηρεσίας όσο και την οικονομία πόρων και χρόνου.

Ιστορική Αναδρομή:

Καθ' όλη την τελευταία δεκαετία γίναμε μάρτυρες μιας μεγάλης εξέλιξης του World Wide Web. Οι ιστοσελίδες έχουν μεταπηδήσει από στατική πληροφόρηση σε ολοκληρωμένες εφαρμογές με μεγάλη διαδραστικότητα και λειτουργικότητα. Ο τομέας των τηλεδιασκέψεων έχει επίσης επηρεαστεί από αυτή την τάση. Οι εφαρμογές των πολυμέσων επικοινωνίας άρχισαν να ξεφυτρώνουν στα Web προγράμματα περιήγησης μας, ενεργοποιούμενες από ιδιόκτητα plug-ins, όπως το Adobe Flash. Τα προϊόντα αυτά σε πολλές περιπτώσεις παρείχαν πλήρεις τηλεδιασκέψεις και συνεργατικές εμπειρίες, τα οποία συχνά βασίζονταν σε ιδιόκτητες λύσεις και τα πρωτόκολλα που επιβάλλονταν από τον προγραμματιστή του browser plug-in, αντί των καθιερωμένων προτύπων[1].

Το Μάιο του 2011, η Google ενσωμάτωσε όλες τις νέες τεχνολογίες και υπηρεσίες της σε ένα open source project για browsers βασισμένο σε επικοινωνία πραγματικού χρόνου, με το όνομα WebRTC. Επιπλέον, συνεργάστηκε με τις κοινότητες προτύπων τεχνολογιών IETF και W3C για να καθορίσει και να εφαρμόσει ένα σύνολο προτύπων για επικοινωνίες πραγματικού χρόνου. Το W3C draft του WebRTC είναι μία δουλειά που βρίσκεται ακόμα σε εξέλιξη με προηγμένες εφαρμογές για τους browsers Chrome και Mozilla. Το συγκεκριμένο API βασίστηκε σε μία προηγούμενη δουλειά, που είχε γίνει στο WHATWG και αναφερόταν ως ConnectionPeer API[2], καθώς επίσης, και σε μία ιδέα που είχε αναπτυχθεί στα εργαστήρια της Ericsson χωρίς όμως να έχει γίνει προτυποποίηση. Το WebRTC από την αρχή είχε την ενεργή υποστήριξη από πολλούς οργανισμούς όπως για παράδειγμα του Mozilla, Cisco και Opera.

Σήμερα, η Google έκανε διαθέσιμο το WebRTC, ένα πακέτο open source code για Real Time Voice and Video στο διαδίκτυο.

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

Σκοπός της εργασίας

Εκμεταλλευόμενοι τις δυνατότητες της WebRTC τεχνολογίας επιλέξαμε να αναπτύξουμε την εφαρμογή που παρουσιάζουμε σ'αυτή την ερευνητική εργασία διευρύνοντας το ανεξάντλητο ερευνητικό πεδίο και ελπίζοντας να προσελκύσουμε το ενδιαφέρον κι άλλων ερευνητών οι οποίοι θα ασχοληθούν να εξελίσουν ακόμη περισσότερο το ερευνητικό έργο αυτού του τομέα.

Συγκεκριμένα η εφαρμογή μας παρέχει την δυνατότητα ταυτόχρονης επικοινωνίας πολλών συμμετεχόντων με ταυτόχρονη εμφάνιση εικόνας και ήχου αλλά και αποτυπώνοντας τον προφορικό λόγο σε γραπτούς υπότιτλους, παρέχοντας την επί πλέον δυνατότητα μετάφρασης του προφορικού λόγου από τα Αγγλικά στα Ελληνικά.

Δομή της εργασίας

Το περιεχόμενο της παρούσας ερευνητικής εργασίας διαρθρώνεται σε εννέα κεφάλαια.

- Στο εισαγωγικό κεφάλαιο το οποίο επικεντρώνεται στα κίνητρα, την δομή της εν λόγω εργασίας καθώς και στην ιστορική αναδρομή της WebRTC τεχνολογίας.
- Στο πρώτο κεφάλαιο όπου γίνεται αναφορά στο τεχνολογικό υπόβαθρο καθώς και τις τεχνολογίες που υποστηρίζουν μία WebRTC εφαρμογή.
- Στο δεύτερο κεφάλαιο περιγράφονται τόσο τα χαρακτηριστικά του WebRTC όσο και ο τρόπος λειτουργίας του.
- Στο τρίτο κεφάλαιο περιγράφονται τα πρωτόκολλα επικοινωνίας που χρησιμοποιούνται στο WebRTC και συνεπώς στην εφαρμογή μας.
- Στο τέταρτο κεφάλαιο περιγράφεται η δομή και η λειτουργία του WebRTC .
- Στο πέμπτο κεφάλαιο περιγράφεται ο τρόπος λειτουργίας του ASR(Automatic Speech Recognition).
- Στο έκτο κεφάλαιο περιγράφονται κάποιες από της ήδη υπάρχουσες εφαρμογές για multiconference επικοινωνία.
- Στο έβδομο κεφάλαιο περιγράφουμε την εφαρμογή μας σε επίπεδο HTML και Javascript.
- Στο όγδοο κεφάλαιο περιγράφουμε το περιβάλλον της εφαρμογής από την πλευρά του χρήστη.

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

Το τεχνολογικό υπόβαθρο του WebRTC

1 HTML

1.1 Εισαγωγή

Η HTML5 είναι μια υπό ανάπτυξη markup language για το Internet η οποία όταν ολοκληρωθεί η ανάπτυξή της θα είναι η επόμενη μεγάλη έκδοση της HTML (HyperText Markup Language). Η ομάδα Web Hypertext Application Technology Working Group (WHATWG) άρχισε δουλειά σε αυτή την έκδοση τον Ιούνιο του 2004 με το όνομα Web Applications 1.0. Το Φεβρουάριο του 2010 το πρότυπο ήταν ακόμη σε κατάσταση "Last Call" στο WHATWG.

Η HTML5 προορίζεται για αντικατάσταση της HTML 4.01, της XHTML 1.0, και της DOM Level 2 HTML. Ο σκοπός είναι η μείωση της ανάγκης για ιδιόκτητα plug-in και πλούσιες διαδικτυακές εφαρμογές (RIA) όπως το Adobe Flash, το Microsoft Silverlight, το Apache Pivot, και η Sun JavaFX[3].

1.2 Σημασιολογία

Πολλές από τις ετικέτες που υπήρχαν από το 1999 είτε καταργήθηκαν είτε διαφοροποιήθηκαν, επιπλέον προστέθηκαν 28 νέες καθώς επίσης και αρκετά νέα γνωρίσματα (attributes) για όλες τις ετικέτες. Παράλληλα το συντακτικό έχει απλουστευτεί με χαρακτηριστικό παράδειγμα την δήλωση του τύπου εγγράφου στην πρώτη σειρά ενός HTML αρχείου. Στην προηγούμενη έκδοση 4.01 η δήλωση ήταν[4]:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

ενώ για την ίδια δήλωση η HTML5 χρειάζεται μόνο 13 χαρακτήρες:

```
<!DOCTYPE html>
```

1.3 Νέα χαρακτηριστικά

Ειδικότερα, η HTML5 προσθέτει πολλά νέα χαρακτηριστικά. Αυτά περιλαμβάνουν το νέο <video> , <audio> και <canvas> elements , καθώς και την ενσωμάτωση των scalable

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

vector γραφικών (SVG) content (replacing generic <object> tags), και MathML για μαθηματικούς τύπους. Ακόμα μερικά από αυτά είναι σημασιολογικές αντικαταστάσεις για κοινές χρήσεις των generic block (<div>) και inline () στοιχείων, για παράδειγμα <nav> (ιστοσελίδα μπλοκ πλοήγησης), <footer> (συνήθως αναφερόμενος στο τέλος της ιστοσελίδας στις τελευταίες γραμμές κώδικα HTML)

Πχ. (Ένα παράδειγμα που χρησιμοποιούμε στην εφαρμογή μας που ενεργοποιεί την camera.)[5]

```
<video autoplay>
```

```
<source src="movie.mp4" type="video/mp4"> Movie </video>
```

Τα χαρακτηριστικά αυτά είναι σχεδιασμένα ώστε να είναι εύκολο να συμπεριληφθούν και να διευκολύνουν τον χειρισμό των πολυμέσων και γραφικών ως περιεχόμενο στο διαδίκτυο, χωρίς να χρειάζεται να καταφύγει κάποιος σε ιδιότητα plugins και APIs . Άλλα νέα στοιχεία είναι, όπως <section> , <Article> , <header> και <nav> , έχουν σχεδιαστεί για να εμπλουτισθεί το σημασιολογικό περιεχόμενο των εγγράφων. Νέα χαρακτηριστικά έχουν εισαχθεί για τον ίδιο σκοπό, ενώ ορισμένα στοιχεία και χαρακτηριστικά έχουν αφαιρεθεί. Ορισμένα στοιχεία, όπως <a> , <cite> και <menu> έχουν αλλάξει, επαναπροσδιορίζεται (χρήζει επαναπροσδιορισμού η τυποποίηση). Τα APIs και Document Object Model (DOM) δεν είναι πλέον δευτερεύουσας σημασίας, αλλά αποτελούν θεμελιώδη στοιχεία της συγγραφής HTML5. Η HTML5 ορίζει επίσης σε κάποιο βαθμό λεπτομέρειας την απαιτούμενη επεξεργασία για τα μη έγκυρα έγγραφα, έτσι ώστε να είναι εφικτό τα συντακτικά λάθη να αντιμετωπίζονται ομοιόμορφα από όλα τα προγράμματα περιήγησης σύμφωνα και με άλλους user agents[3].

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

HTML5 Support

	Chrome	Firefox	Safari	Opera
<code>canvas</code>	✓	✓	✓	✓
<code>video</code>	✓	✓	✓	✓
<code>geolocation</code>	✓	✓	✓ (iPhone)	✓
<code>app cache</code>	✓	✓	✓	✓ (mobile)
<code>database</code>	✓	✓	✓	✓ (mobile)
<code>workers</code>	✓	✓	✓	✓ (mobile)

Εικόνα 1(<http://radar.oreilly.com/2009/05/google-bets-big-on-html-5>)

Επίσης, το Web Storage για την αποθήκευση των δεδομένων χρησιμοποιεί δύο νέες μεθόδους αποθήκευσης, το `sessionStorage` και το `localStorage`[6]. Το πρώτο αποθηκεύει δεδομένα και τα διατηρεί μόνο όσο είναι ανοιχτή η ιστοσελίδα ενώ στην περίπτωση αποσύνδεσης(logout) ή κλεισίματος του συγκεκριμένου παραθύρου του browser, τα δεδομένα διαγράφονται. Σημειώνεται ότι το δεύτερο αποθηκεύει δεδομένα χωρίς χρονικό όριο και τα οποία διατηρούνται ακόμη και αν κλείσει ο browser.

Τέλος, μερικά ακόμη νέα χαρακτηριστικά της HTML5 σε συνδυασμό με την Javascript είναι ότι καθίσταται δυνατόν να χρησιμοποιηθεί κάποια APIs όπως: Timed media playback, Offline Web Applications, Cross-document messaging, Geolocation, Web SQL Database.

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

HTML5

Taxonomy & Status (October 2014)

- Recommendation/Proposed
- Candidate Recommendation
- Last Call
- Working Draft
- Non-W3C Specifications
- Deprecated or inactive



Εικόνα 2(Sergey Mavrody "Sergey's HTML5 & CSS3 Quick Reference. 2nd Edition". Belisso Corp., 2012.)

2 JAVASCRIPT

2.1 Εισαγωγή

Η γλώσσα προγραμματισμού JavaScript δημιουργήθηκε αρχικά από τον Brendan Eich της εταιρείας Netscape με την επωνυμία Mocha. Αργότερα, το Mocha μετονομάστηκε σε LiveScript, και τελικά σε JavaScript, κυρίως επειδή η ανάπτυξή της επηρεάστηκε περισσότερο από τη γλώσσα προγραμματισμού Java. Το LiveScript ήταν το επίσημο όνομα της γλώσσας όταν για πρώτη φορά κυκλοφόρησε στην αγορά σε βήτα (beta) εκδόσεις με το πρόγραμμα περιήγησης στο Web, Netscape Navigator εκδοχή 2.0 τον Σεπτέμβριο του 1995. Το LiveScript μετονομάστηκε σε JavaScript σε μια κοινή ανακοίνωση με την εταιρεία Sun Microsystems στις 4 Δεκεμβρίου, 1995 [9], όταν επεκτάθηκε στην έκδοση του προγράμματος περιήγησης στο Web, Netscape εκδοχή 2.0B3.

Η JavaScript απέκτησε μεγάλη επιτυχία ως γλώσσα από την πλευρά του πελάτη (client-side) για εκτέλεση κώδικα σε ιστοσελίδες, και περιλήφθηκε σε διάφορα προγράμματα περιήγησης στο Web. Κατά συνέπεια, η εταιρεία Microsoft ονόμασε την εφαρμογή της ως JScript για να αποφύγει δύσκολα θέματα εμπορικών σημάτων. Η JScript πρόσθεσε

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

νέες μεθόδους για να διορθώσει τα Y2K-προβλήματα στην JavaScript, τα οποία βασίστηκαν στην `java.util.Date` τάξη της Java. Η JScript περιλήφθηκε στο πρόγραμμα Internet Explorer εκδοχή 3.0, το οποίο κυκλοφόρησε τον Αύγουστο του 1996[7].

Τον Νοέμβριο του 1996, η Netscape ανακοίνωσε ότι είχε υποβάλει τη γλώσσα JavaScript στο Ecma International (μια οργάνωση της τυποποίησης των γλωσσών προγραμματισμού) για εξέταση αυτής ως βιομηχανικό πρότυπο, και στη συνέχεια το project (έργο) είχε ως αποτέλεσμα την τυποποιημένη μορφή, η οποία ονομάζεται ECMAScript.

Η JavaScript έχει γίνει μία από τις πιο δημοφιλείς γλώσσες προγραμματισμού ηλεκτρονικών υπολογιστών στον Παγκόσμιο Ιστό (Web).[8] Αρχικά, όμως, πολλοί επαγγελματίες προγραμματιστές υποτίμησαν τη γλώσσα διότι το κοινό της ήταν ερασιτέχνες συγγραφείς ιστοσελίδων και όχι επαγγελματίες προγραμματιστές. Με τη χρήση της τεχνολογίας Ajax, η JavaScript γλώσσα επέστρεψε στο προσκήνιο και έφερε πιο επαγγελματική προσοχή στον προγραμματισμό. Το αποτέλεσμα ήταν ένα καινοτόμο αντίκτυπο στην εξάπλωση των παισίων και των βιβλιοθηκών, τη βελτίωση προγραμματισμού με JavaScript, καθώς και αυξημένη χρήση της JavaScript έξω από τα προγράμματα περιήγησης στο Web.

Τον Ιανουάριο του 2009, το έργο CommonJS ιδρύθηκε με στόχο τον καθορισμό ενός κοινού προτύπου βιβλιοθήκης κυρίως για την ανάπτυξη της JavaScript έξω από το πρόγραμμα περιήγησης και εντός άλλων τεχνολογιών (π.χ. server-side)[7].

Η αρχική έκδοση της Javascript βασίστηκε στη σύνταξη της γλώσσας προγραμματισμού C, αν και έχει εξελιχθεί, ενσωματώνοντας πια χαρακτηριστικά από νεότερες γλώσσες.

2.2 Μοντέλο Εκτέλεσης

Αρχικά χρησιμοποιήθηκε για προγραμματισμό από την πλευρά του πελάτη (client), που ήταν ο φυλλομετρητής (browser) του χρήστη, και χαρακτηρίστηκε σαν client-side γλώσσα προγραμματισμού. Αυτό σημαίνει ότι η επεξεργασία του κώδικα Javascript και η παραγωγή του τελικού περιεχομένου HTML δεν πραγματοποιείται στο διακομιστή, αλλά στο πρόγραμμα περιήγησης των επισκεπτών, ενώ μπορεί να ενσωματωθεί σε στατικές σελίδες HTML. Αντίθετα, άλλες γλώσσες όπως η PHP εκτελούνται στο διακομιστή (server-side γλώσσες προγραμματισμού).

Π.χ. (Μέσα στην web εφαρμογή, μπορούμε να καλέσουμε την javascript με τον παρακάτω κώδικα)

```
<script src="//cdn.webrtc-experiment.com/RTCMultiConnection.js"></script>
```

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

Και ένα πιο γενικό μοντέλο:

```
<!DOCTYPE HTML>

<html lang="eng">

<head>

  <title> WEB RTC video </title>

  <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>

  <script src="WebRTC.js"></script>

  <link rel="stylesheet" type="text/css" href="WebRTC1.css">

</head>

<body>

  <script>

var connection = new RTCMultiConnection();

connection.channel = 'channel-id';

// easiest way to customize what you need!

var session = {

  audio: true,

  video: true

};

connection.captureUserMedia(function (mediaStream) { }, session);

</script>

</body>

</html>var connection = new RTCMultiConnection();

connection.channel = 'channel-id';

// easiest way to customize what you need!

var session = {

  audio: true,

  video: true

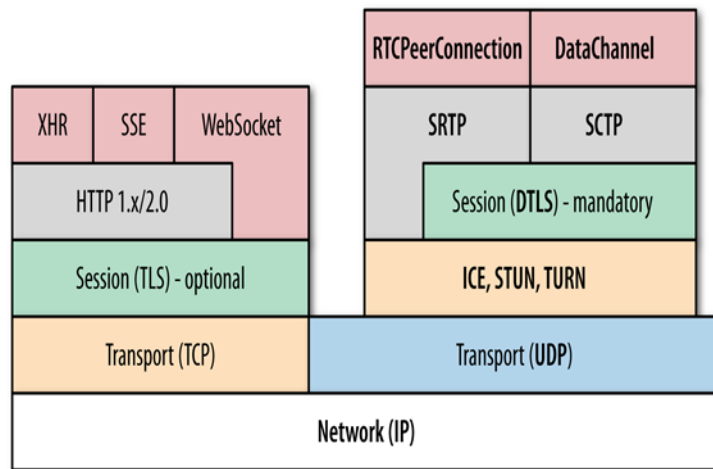
};

connection.captureUserMedia(function (mediaStream) { }, session);

</script>
```

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

3 Πρωτόκολλα, γενική περιγραφή



Στην διπλανή εικόνα βλέπουμε την στοίβα των πρωτοκόλλων που χρησιμοποιεί το webRTC.

Παρακάτω θα αναλύσουμε το κάθε ένα ξεχωριστά.

Εικόνα: WebRTC protocol stack.(http://chimera.labs.oreilly.com/books/1230000000545/ch18.html#WEBRTC_STACK)

IP

Το IP (Internet Protocol) είναι το πιο σημαντικό πρωτόκολλο στο Internet γιατί έχει ως αποστολή να καθοδηγεί τα πακέτα μέσω διαφόρων συνδεδεμένων δικτύων μέχρι το τερματικό που προορίζονται. Ακόμα το IP δεν καθίσταται υπεύθυνο για την σωστή παράδοση του πακέτου ,ούτε για την ενημέρωση του παραλήπτη για τυχόν απώλειες ή σφάλματα. Είναι γενικά ένα μη αξιόπιστο πρωτόκολλο και οι λειτουργίες αυτές επιτυγχάνονται με το TCP πρωτόκολλο, που βρίσκεται στο αμέσως ανώτερο επίπεδο[10].

Ιστορικά, το IP ήταν connectionless datagram service στο αρχικό πρόγραμμα ελέγχου μετάδοσης που εισήγαγαν οι Vint Cerf και Bob Kahn το 1974. Ως εκ τούτου η σουίτα πρωτόκολλο Internet αναφέρεται συχνά και ως το πρωτόκολλο TCP / IP.

Η πρώτη σημαντική έκδοση της IP, internet protocol Έκδοση 4 (IPv4), είναι το κυρίαρχο πρωτόκολλο του Διαδικτύου. Ο διάδοχός του στο Internet είναι το Protocol Version 6 (IPv6)[11].

TCP

Το TCP(Transmission Control Protocol) είναι το Πρωτόκολλο Ελέγχου Μετάδοσης και

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

είναι ένα πρωτόκολλο μεταφοράς, και βρίσκεται στο Transport Layer.

Είναι ένα πρωτόκολλο που παρέχει αξιοπιστία και χρησιμεύει για εφαρμογές για την μετάδοση πακέτων χωρίς διακοπή. Είναι connection-oriented protocol, το οποίο σημαίνει ότι για να γίνει μια επικοινωνία στο TCP, ανοίγει πρώτα μια HTTP σύνδεση (με την χρήση ερώτησης και απάντησης) και στην συνέχεια δημιουργείται ένας δίαυλος επικοινωνίας στέλνοντας συνέχεια πακέτα χωρίς διακοπή. Ο δίαυλος επικοινωνίας κλείνει μόνο όταν διακοπεί η σύνδεση. Επίσης το TCP ευθύνεται για τον έλεγχο δρομολόγησης πακέτων πχ αν ένα πακέτο χαθεί το ξαναστέλνει ή ελέγχει αν υπάρχει συμφόρηση πακέτων μέσα στον δίαυλο ώστε να μην στέλνει συνέχεια πακέτα και χαθούν[10].

Η διαδικασία έχει ως εξής στο TCP: Γίνεται ανταλλαγή δεδομένων μεταξύ του αποστολέα και του παραλήπτη. Ο αποστολέας είναι υπεύθυνος για την αποστολή και την αναμετάδοση των πακέτων που έχουν χαθεί, και ο παραλήπτης για τον έλεγχο των πακέτων που λαμβάνει.

UDP

Το UDP (User Datagram Protocol) είναι ένα από τα βασικά πρωτόκολλα που χρησιμοποιούνται στο Διαδίκτυο. Διάφορα προγράμματα χρησιμοποιούν το πρωτόκολλο UDP για την αποστολή σύντομων μηνυμάτων (γνωστών και ως datagrams) από τον έναν υπολογιστή στον άλλον μέσα σε ένα δίκτυο υπολογιστών. Ένα από τα κύρια χαρακτηριστικά του UDP είναι ότι δεν εγγυάται αξιόπιστη επικοινωνία. Τα πακέτα UDP που αποστέλλονται από έναν υπολογιστή μπορεί να φτάσουν στον παραλήπτη με λάθος σειρά, δηλαδή διπλά ή να μην φτάσουν καθόλου, εάν το δίκτυο έχει μεγάλο φόρτο. Η έλλειψη των μηχανισμών αυτών από το πρωτόκολλο UDP το καθιστά αρκετά πιο γρήγορο και αποτελεσματικό, τουλάχιστον για τις εφαρμογές εκείνες που δεν απαιτούν αξιόπιστη επικοινωνία.

Οι εφαρμογές audio και video streaming χρησιμοποιούν κατά κόρον πακέτα UDP. Για τις εφαρμογές αυτές είναι πολύ σημαντικό τα πακέτα να παραδοθούν στον παραλήπτη σε σύντομο χρονικό διάστημα ούτως ώστε να μην υπάρχει διακοπή στην ροή του ήχου ή της εικόνας. Κατά συνέπεια προτιμάται το πρωτόκολλο UDP διότι είναι αρκετά γρήγορο, παρόλο που υπάρχει η πιθανότητα μερικά πακέτα UDP να χαθούν. Στην περίπτωση που χαθεί κάποιο πακέτο, οι εφαρμογές αυτές διαθέτουν ειδικούς μηχανισμούς διόρθωσης και παρεμβολής ούτως ώστε ο τελικός χρήστης να μην παρατηρεί καμία αλλοίωση ή διακοπή στην ροή του ήχου και της εικόνας λόγω του χαμένου πακέτου. Σε αντίθεση με το πρωτόκολλο TCP, το UDP υποστηρίζει broadcasting, δηλαδή την αποστολή ενός πακέτου σε όλους τους υπολογιστές ενός δικτύου, και multicasting, δηλαδή την αποστολή ενός πακέτου σε κάποιους συγκεκριμένους υπολογιστές ενός δικτύου. Η τελευταία δυνατότητα χρησιμοποιείται πολύ συχνά στις εφαρμογές audio και video streaming ούτως ώστε μία ροή ήχου ή εικόνας να μεταδίδεται ταυτόχρονα σε πολλούς συνδρομητές[12].

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

TLS

Το Transport Layer Security (TLS) είναι ένα πρωτόκολλο που εξασφαλίζει την προστασία ανταλλαγής πακέτων μεταξύ εφαρμογών επικοινωνίας και των χρηστών τους στο Διαδίκτυο. Όταν ένας διακομιστής και ένας πελάτης επικοινωνούν, το TLS διασφαλίζει ότι κανένας τρίτος δεν μπορεί να παρακολουθήσει ή να παρέμβει σε κάποιο μήνυμα. Το TLS είναι ο διάδοχος του Secure Sockets Layer (SSL).

Το πρωτόκολλο TLS έχει σχεδιαστεί για να παρέχει τρεις βασικές υπηρεσίες για όλες τις εφαρμογές που εκτελούνται πάνω από αυτό: κρυπτογράφηση, αυθεντικοποίηση και την ακεραιότητα των δεδομένων. Το TLS αποτελείται από δύο στρώματα: το πρωτόκολλο TLS και το πρωτόκολλο TLS record. Το TLS protocol αποτελείται από το application, alert, ChangeCipherSpec και το Handshake. Το υποπρωτόκολλο TLS record αποτελείται από fragmentation, Compression(opt), Integrity, Authentication, Encryption[13].

Τέλος, το πρωτόκολλο TLS παρέχει επίσης το δικό του μηχανισμό διαμόρφωσης μηνύματος και υπογράφει κάθε μήνυμα με έναν κωδικό επαλήθευσης ταυτότητας μηνύματος (MAC). Ο αλγόριθμος MAC είναι μια μονόδρομη συνάρτηση hash function, και τα κλειδιά ανταλλάσσονται ανάμεσα στις δύο συνδέσεις.

DTLS

Το DTLS (Datagram TLS) πρωτόκολλο είναι ένα παρακλάδι του TLS το οποίο κινείται πάνω από το UDP. Ακόμα παρέχει τα ίδια χαρακτηριστικά ως προς την ακεραιότητα, εμπιστευτικότητα και αυθεντικοποίηση με το TLS[14]. Όμως μέσω του UDP είναι πιο εύκολο να διασχίσει το NAT και ταιριάζει καλύτερα σε P2P εφαρμογές.[15]

ICE

Το ICE (Interactive Connectivity Establishment) είναι μια τεχνική ή ένα framework που χρησιμοποιείται για την δικτύωση των υπολογιστών σε ένα περιβάλλον network area translators (NATs), για web application που χρησιμοποιούν Voice over Internet Protocol (VoIP), peer-to-peer communications, video, instant messaging και άλλα interactive media. Σε τέτοιες εφαρμογές, το NAT traversal είναι ένα σημαντικό συστατικό (component) για να διευκολύνει τις επικοινωνίες με hosts σε κάποιο ιδιωτικό δίκτυο (private network) που συχνά βρίσκονται πίσω από firewalls.

Για να επικοινωνήσουν δυο peers μεταξύ τους θα πρέπει να δημιουργηθεί μια σύνδεση. Αλλά συνήθως είναι δύσκολο να επιτευχθεί αυτό λόγω των διαφορετικών δικτύων που μπορεί να βρίσκεται ο κάθε ένας εκ των peers. Στην περίπτωση όπου έστω ότι βρίσκονται στο ίδιο εσωτερικό δίκτυο, τότε κάθε ένας εκ των peers αναζητεί την IP του, και την επισυνάπτει στην IP και την θύρα του και δημιουργείται ένα SDP Protocol με αυτές της πληροφορίες. Όταν γίνει η ανταλλαγή του SDP τότε δημιουργείται μια σύνδεση peer-to-peer.

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

Σε αντίθετη περίπτωση, όταν οι δύο peers βρίσκονται σε διαφορετικό δίκτυο θα πρέπει να δημιουργηθεί ένα κοινό μονοπάτι μεταξύ τους[16].

- Κάθε RTCPeerConnection σύνδεση περιέχει έναν ICE agent.
- ICE agent είναι υπεύθυνος ώστε να συλλέγει όλες τις τοπικές IP και θύρες.
- ICE agent είναι υπεύθυνος για την εκτέλεση των ελέγχων συνδεσιμότητας μεταξύ των peers.
- ICE agent είναι υπεύθυνος για την διατήρηση της εν ενεργεία σύνδεσης.

Μόλις τεθεί μια περιγραφή συνόδου (τοπικά ή απομακρυσμένα), ο ICE agent ενεργοποιείται και ξεκινά την αναζήτηση για πιθανές υποψήφιες IP και ports.

1. ICE agent, ζητά από το λειτουργικό σύστημα για τις τοπικές διευθύνσεις IP.
2. Εάν έχει ρυθμιστεί, ο ICE agent ζητά από έναν εξωτερικό διακομιστή STUN να ανακτήσει τη δημόσια IP και θύρα από τους peers.
3. Εάν έχει ρυθμιστεί, ο ICE agent προσθέτει τον διακομιστή TURN ως έσχατη λύση αν αποτύχει η σύνδεση. Τα δεδομένα θα μεταφέρονται μέσω ενός καθορισμένου μεσάζοντα.

STUN

Το STUN (Session Traversal Utilities for NAT) είναι ένα από τα πιο σημαντικά πρωτόκολλα του διαδικτύου για την παράκαμψη του NAT [16]. Είναι ένας μηχανισμός που χρησιμοποιείται για την εύρεση δημοσίων IP και ports για τους ενδιαφερόμενους peers στην περίπτωση που βρίσκεται πίσω από ένα NAT περιβάλλον. Είναι πολύ σημαντικό για να επιτευχθεί σύνδεση μεταξύ δυο τερματικών όταν βρίσκονται πίσω από το ίδιο NAT ή ένα Firewall.

Τέλος, το STUN είναι ένα σχετικά απλό client/server πρωτόκολλο. Υποστηρίζει δύο τύπους συναλλαγών. Ο ένας είναι αίτηση/απάντηση, στον οποίο ο client στέλνει αίτηση στο server και ο server στέλνει πίσω απάντηση, ενώ ο δεύτερος είναι μια συναλλαγή αναφοράς κατά την οποία είτε ο server είτε ο client στέλνει μια αναφορά η οποία δεν χρειάζεται απάντηση.

TURN

Το TURN (Traversal Using Relay NAT) είναι ένα πρωτόκολλο το οποίο προορίζεται να διέπει τη λήψη δεδομένων μέσω ενός πρωτοκόλλου ελέγχου μετάδοσης (TCP) ή του πρωτοκόλλου User Datagram Protocol (UDP) που λειτουργεί πίσω ένα Network Address Translator (NAT) ή τείχος προστασίας.

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

Το TURN είναι μια επέκταση του STUN και παρέχει αναμετάδοση πολυμέσων όταν οι άλλες μέθοδοι του ICE αποτύχουν κατά τη διάρκεια μιας κλήσης. Για την αντιμετώπιση αυτού του ζητήματος, όποτε το STUN αποτύχει, μπορούμε να χρησιμοποιήσουμε το Traversal Using Relays around NAT (TURN) πρωτόκολλο (protocol). Με άλλα λόγια, όταν ένα απευθείας μονοπάτι επικοινωνίας δεν μπορεί να βρεθεί για δύο τερματικά, κρίνεται απαραίτητη η χρήση ενδιάμεσων host, οι οποίοι λειτουργούν ως αναμεταδότες για τα πακέτα. Συνήθως τα πακέτα βρίσκονται ανάμεσα στους δύο hosts οι οποίοι βρίσκονται πίσω από NATs. Σε τέτοιες περιπτώσεις το TURN είναι μια καλή λύση, καθώς επιτρέπει σε έναν host που είναι πίσω από ένα NAT (TURN client) να στείλει αίτηση σ' έναν άλλο host (TURN server) ο οποίος λειτουργεί σαν αναμεταδότης. Για τη μεταφορά των μηνυμάτων από TURN server σε peer, το TURN χρησιμοποιεί πάντα UDP, ενώ από TURN server σε TURN client παρέχεται η επιλογή ανάμεσα σε UDP, TCP ή TLS over TCP[16].

Φυσικά, το προφανές μειονέκτημα σε αυτή την ανταλλαγή είναι ότι δεν είναι πλέον peer-to-peer! Το TURN είναι ο πιο αξιόπιστος τρόπος για να παρέχεται συνδεσιμότητα μεταξύ δύο peers σε οποιαδήποτε δίκτυα, αλλά φέρει ένα πολύ υψηλό κόστος λειτουργίας καθώς χρειάζεται υψηλού εύρους ζώνη σύνδεσης στο internet. Ως αποτέλεσμα, η σειρά του είναι καλύτερα να χρησιμοποιείται ως τελευταία εναλλακτική λύση για τις περιπτώσεις όπου η άμεση συνδεσιμότητα αποτυγχάνει[17].

SDP

Το SDP (Session Description Protocol) ή Πρωτόκολλο Περιγραφής Συνόδου είναι ένα σύνολο κανόνων που περιγράφουν τις παραμέτρους εκκίνησης των μέσων ροής. Ο κύριος σκοπός του είναι να περιγράψει τις συνόδους επικοινωνίας των πολυμέσων για θέματα όπως η ανακοίνωση ή η πρόσκληση ή και άλλες μορφές έναρξης μιας συνόδου επικοινωνίας πολυμέσων. Καλό θα ήταν να διευκρινιστεί ότι το συγκεκριμένο πρωτόκολλο είναι πρωτόκολλο περιγραφής και σε καμία περίπτωση πρωτόκολλο μεταφοράς. Το SDP σχεδιάστηκε έτσι ώστε να είναι επεκτάσιμο για να μπορεί να υποστηρίζει νέους τύπους και μορφές πολυμέσων. Δημοσιεύτηκε το 2006 από την IETF [18].

Τέλος, μια περιγραφή συνόδου επικοινωνίας SDP περιλαμβάνει χρήσιμες πληροφορίες όπως για παράδειγμα: το όνομα, την ημερομηνία και την ώρα έναρξης της συνόδου, το σκοπό της συνόδου, καθώς επίσης και πληροφορίες σχετικές με το εύρος ζώνης που θα χρησιμοποιηθεί, διευθύνσεις και θύρες από όλα τα τερματικά που λαμβάνουν μέρος στη σύνοδο.

SRTP και SRTCP

Το SRTP (Secure Real-time Transport Protocol) ορίζει ένα προφίλ της RTP (Real-time Transport Protocol) που προορίζεται για την κρυπτογράφηση, αυθεντικοποίηση και την ακεραιότητα των μηνυμάτων που ανταλλάσσουν οι peers μεταξύ τους, για unicast και multicast εφαρμογές. Αναπτύχθηκε από μια

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

μικρή ομάδα εμπειρογνομόνων της IP protocol και cryptographic experts από την Cisco and Ericsson συμπεριλαμβανομένων των David Oran, David McGrew, Mark Baugher, Mats Naslund, Elisabetta Carrara, Karl Norman, και Rolf Blom. Ήταν η πρώτη που δημοσιεύθηκε από την IETF Μάρτιο του 2004 ως RFC 3711.

Δεδομένου ότι η RTP είναι στενά συνδεδεμένη με RTCP (Real Time Control Protocol) το οποίο μπορεί να χρησιμοποιηθεί για τον έλεγχο μιας RTP συνεδρίας (session) , επίσης το SRTP έχει ένα αδελφό πρωτόκολλο το οποίο ονομάζεται Secure RTCP (SRTCP) και το οποίο παρέχει τα ίδια χαρακτηριστικά ασφάλειας που σχετίζονται με RTCP, όπως αυτές που παρέχονται από SRTP στην RTP.

Για την κρυπτογράφηση και αποκρυπτογράφηση της ροής δεδομένων , SRTP (μαζί με SRTCP) χρησιμοποιείται ο AES αλγόριθμος ως προεπιλεγμένη κρυπτογράφηση. Υπάρχουν δύο τρόποι κρυπτογράφησης ο ένας είναι Segmented Integer Counter Mode και ο άλλος τρόπος είναι f8-mode[19].

HTTP

Το HTTP (HyperText Transfer Protocol, HTTP) είναι ένα πρωτόκολλο επικοινωνίας. Αποτελεί το κύριο πρωτόκολλο που χρησιμοποιείται στους browsers του Παγκοσμίου Ιστού για την μεταφορά δεδομένων ανάμεσα σε έναν διακομιστή (server) και έναν πελάτη (client). Δημιουργήθηκε από τις εταιρείες IETF και W3C και δημοσιεύτηκε τον Ιούνιο του 1999[6].

Το HTTP λειτουργεί ως πρωτόκολλο «αίτημα-απάντηση» στο client-server. Για παράδειγμα, ένα πρόγραμμα περιήγησης στο Web μπορεί να είναι ο πελάτης, ενώ μια εφαρμογή που τρέχει σε έναν υπολογιστή που φιλοξενεί την ιστοσελίδα μπορεί να είναι ο διακομιστής. Ο πελάτης στέλνει ένα HTTP μήνυμα στο διακομιστή. Ο server ο οποίος παρέχει πόρους, όπως HTML αρχεία και άλλο περιεχόμενο, είτε εκτελεί άλλο περιεχόμενο είτε επιστρέφει μια απάντηση στον πελάτη. Η απάντηση περιέχει πληροφορίες για την κατάσταση ολοκλήρωσης σχετικά με την αίτηση και μπορεί επίσης να περιέχει περιεχόμενο που ζητήθηκε στο σώμα του μηνύματος του.

Το HTTP είναι ένα επίπεδο εφαρμογών πρωτοκόλλου το οποίο σχεδιάστηκε στο πλαίσιο του πρωτοκόλλου Internet Suite . Ο ορισμός του προϋποθέτει ένα υποκείμενο και αξιόπιστο στρώμα μεταφοράς πρωτοκόλλου, και χρησιμοποιείται συνήθως το πρωτόκολλο ελέγχου μετάδοσης (TCP). Ωστόσο τα HTTP μπορούν να χρησιμοποιούν αναξιόπιστα πρωτόκολλα όπως το πρωτόκολλο User Datagram Protocol (UDP), για παράδειγμα σε απλό πρωτόκολλο εντοπισμού υπηρεσίας (SSDP).

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

Επιπλέον, το HTTP ανήκει στη κατηγορία stateless protocol δηλ. πρωτόκολλα χωρίς μνήμη.[18] Αυτό σημαίνει ότι το HTTP δεν κρατάει δεδομένα κατάστασης της σύνδεσης και όσες φορές ο χρήστης του στείλει μήνυμα αίτησης τόσες φορές θα πάρει μήνυμα απόκρισης, ακόμη και για το ίδιο αρχείο. Βέβαια σε περίπτωση που είναι επιθυμητή η αναγνώριση των χρηστών από το server, υπάρχουν δύο μηχανισμοί: η πιστοποίηση (authentication) και τα cookies[10].

P2P

1 Εισαγωγή

Ένα δίκτυο υπολογιστών peer-to-peer (ή P2P) είναι ένα δίκτυο που επιτρέπει σε δύο ή περισσότερους υπολογιστές να μοιράζονται τους πόρους τους ισοδύναμα. Το δίκτυο αυτό χρησιμοποιεί την επεξεργαστική ισχύ, τον αποθηκευτικό χώρο και το εύρος ζώνης (bandwidth) των κόμβων. Όλοι οι κόμβοι του δικτύου έχουν ίσα δικαιώματα. Πληροφορίες που βρίσκονται στον ένα κόμβο, ανάλογα με τα δικαιώματα που καθορίζονται, μπορούν να διαβαστούν από όλους τους άλλους και αντίστροφα[20].

Τα Peer-to-Peer δίκτυα χωρίζονται σε τρεις κατηγορίες:

2 Συγκεντρωτικά P2P δίκτυα

Πολλοί, όταν αναφέρονται σε αυτά, χρησιμοποιούν τη φράση «πρώτης γενιάς P2P δίκτυα». Εδώ, υπάρχει ένας κεντρικός Index Server στον οποίο αποθηκεύονται οι πληροφορίες για τα περιεχόμενα των καταλόγων που οι συμμετέχοντες επιθυμούν να μοιράζονται. Οι χρήστες μπορούν να αναζητήσουν στους Index Servers τα αρχεία που ψάχνουν, χρησιμοποιώντας ένα κατάλληλο πρόγραμμα-πελάτη. Όταν το αρχείο βρεθεί, ανοίγει μια σύνδεση μεταξύ των δύο χρηστών για τη μεταφορά του. Σε αυτή τη κατηγορία ανήκουν το Napster το DC++ και το WinMX.

3 Αποκεντρωτικά P2P δίκτυα

Η φιλοσοφία εδώ είναι εντελώς διαφορετική. Κάθε σύστημα που συμμετέχει αποτελεί ταυτόχρονα client και server (ή αλλιώς servent). Μόλις κάποιος συνδεθεί μέσω ενός ανάλογου προγράμματος-πελάτη P2P, κάνει γνωστή την παρουσία του σε ένα μικρό αριθμό υπολογιστών ήδη συνδεδεμένων, οι οποίοι με τη σειρά τους προωθούν τη δήλωση παρουσίας του σε ένα μεγαλύτερο δίκτυο υπολογιστών κ.λ.π. Πλέον ο χρήστης έχει τη δυνατότητα να αναζητήσει οποιαδήποτε πληροφορία μεταξύ των διαμοιραζόμενων αρχείων. Τα δίκτυα αυτά λέγονται και δεύτερης γενιάς. Η μεταφορά των αρχείων είναι όμοια με αυτή των συγκεντρωτικών P2P δικτύων. Σε αυτή τη κατηγορία ανήκουν το Kazaa, το Gnutella και το BearShare.

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

4 P2P δίκτυα τρίτης γενιάς

Είναι αυτά τα οποία διαθέτουν χαρακτηριστικά ανωνυμίας όπως το Freenet, το I2P και το Entropy. Είναι αποκεντρωτικού τύπου και η φιλοσοφία τους βασίζεται εκτός από την ανωνυμία, στην υψηλή βιωσιμότητα τους, στο συνεχή διαμοιρασμό των αρχείων και στην κωδικοποίησή τους έτσι ώστε κανείς να μην μπορέσει ποτέ να αποκτήσει κανένα είδος ελέγχου πάνω σε αυτά. Τα δίκτυα αυτού του τύπου είναι υπό ανάπτυξη και έχουν χαρακτηριστεί ως μικρά παγκόσμια δίκτυα[21].

WebSocket Protocol

Το Πρωτόκολλο WebSocket προτυποποιήθηκε το 2011 από την IETF ως RFC6455[22]. Σχεδιάστηκε για να αντικαταστήσει τις υπάρχουσες τεχνολογίες αμφίδρομης επικοινωνίας, οι οποίες έκαναν συμβιβασμούς μεταξύ της αποτελεσματικότητας και της αξιοπιστίας για να μπορούν να εκμεταλλεύονται την υπάρχουσα υποδομή του HTTP, καθώς το HTTP δεν ήταν αρχικά φτιαγμένο για αμφίδρομη επικοινωνία. Το WebSocket προσπάθησε να συμπεριλάβει όλους τους στόχους από τις υπάρχουσες αμφίδρομες τεχνολογίες στο πλαίσιο της HTTP υποδομής χωρίς όμως συμβιβασμούς και χωρίς αυτή η υλοποίηση να περιορίσει το WebSocket αυστηρά στο HTTP.

Η WebSocket αμφίδρομη επικοινωνία επιτυγχάνεται στέλνοντας μια HTTP αίτηση ρωτώντας τον server για αναβάθμιση της σύνδεσης με το WebSocket πρωτόκολλο.

Ο client στέλνει τον παρακάτω κώδικα:

```
GET /chat HTTP/1.1
Host: server.example.com
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Key: dGhlIHNhbXBsZSBub25jZQ==
Origin: http://example.com
Sec-WebSocket-Protocol: chat, superchat
Sec-WebSocket-Version: 13
```

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

Και στη συνέχεια, αν η αίτηση είναι επιτυχής, ο server στέλνει απάντηση που μοιάζει κάπως έτσι:

```
HTTP/1.1 101 Switching Protocols
```

```
Upgrade: websocket
```

```
Connection: Upgrade
```

```
Sec-WebSocket-Accept: s3pPLMBiTxaQ9kYGzzhZRbK+xOo=
```

```
Sec-WebSocket-Protocol: chat
```

Αυτή η ανταλλαγή ονομάζεται handshake και απαιτείται για να γίνει η εγκαθίδρυση μιας WebSocket σύνδεσης. Στην περίπτωση που το handshake είναι επιτυχές, ο client με τον server μπορούν να ανταλλάζουν δεδομένα ανεξάρτητα ο ένας από τον άλλο καθώς μια αμφίδρομη επικοινωνία ιδρύθηκε.

Γενικά, το WebSocket δημιουργήθηκε για να παρέχει μεγαλύτερη αλληλεπίδραση μεταξύ μιας ιστοσελίδας και ενός browser, διευκολύνοντας τη δημιουργία παιχνιδιών πραγματικού χρόνου και το live content (ζωντανό περιεχόμενο)[15].

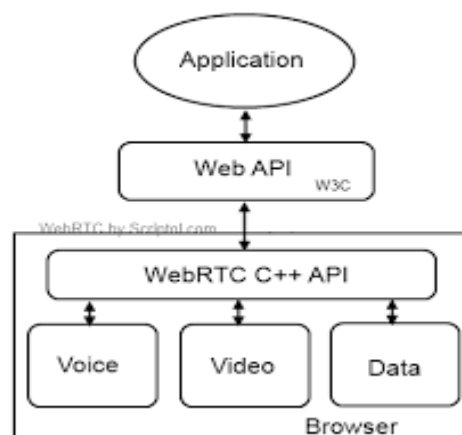
"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

4 WebRTC

4.1 Τι είναι το WebRTC

Τα τελευταία χρόνια έχουν ανακαλυφθεί νέες πλατφόρμες που παρέχουν ζωντανή επικοινωνία. Τα οποία λέγονται browser-embedded application ή Web application. Μεταξύ αυτών των Web applications το Web RTC έλαβε μεγάλο ενδιαφέρον γιατί υποστηρίζεται από I.E.-Google Chrome και Mozilla fireFox που βασίζεται στην HTML5, κατέχει PeerConnection, Media Stream και DataChannels εξαρτήματα API που μπορούν να συνδυαστούν άμεσα με την καθιέρωση P2P επικοινωνίας μεταξύ των browser.

Το Web RTC(Web Real Time Communication) είναι ένα API που κατασκεύασε η Word Wide Web Consortium(W3C). Μέσω ενός Browser μπορεί κάποιος να κάνει video chat , voice chat και P2P κοινή χρήση αρχείων χωρίς Plugins. Επιτρέπει την Real Time Communication με την χρήση απλών JavaScript APIs και HTML5.

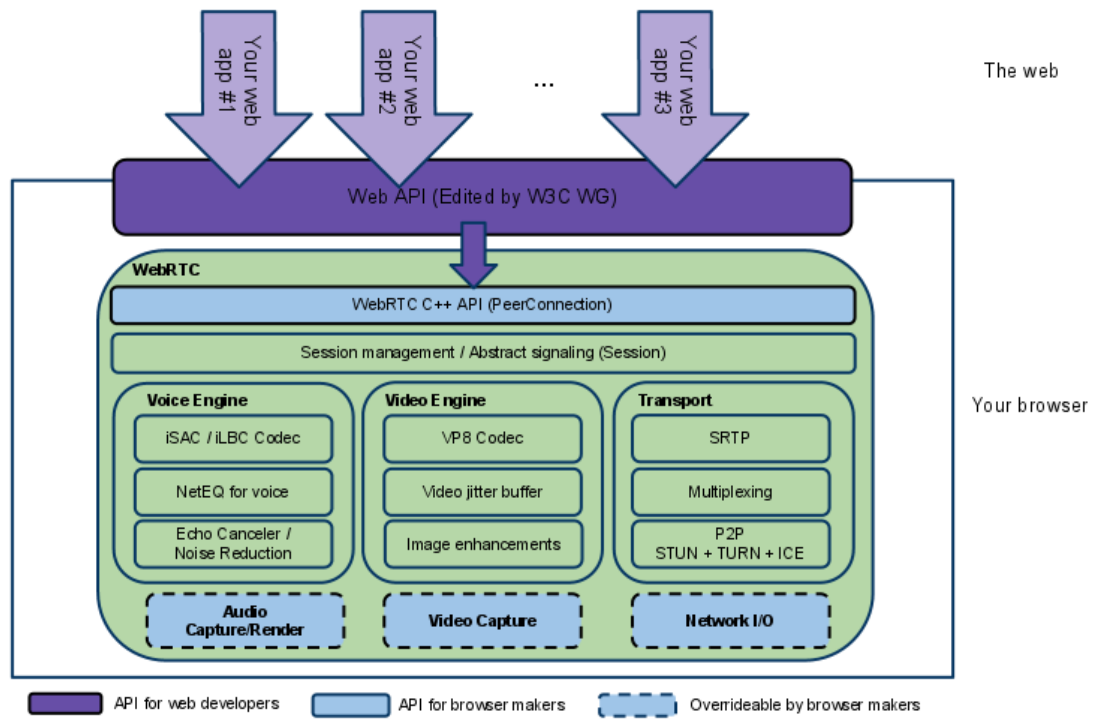


Το WebRTC παρέχει στους προγραμματιστές τη δυνατότητα να κατασκευάσουν εφαρμογές πολυμέσων (συνομιλία μέσω βίντεο) στο διαδίκτυο, χωρίς να απαιτούνται plugins ή άλλες εγκαταστάσεις.

4.2 Η Αρχιτεκτονική του WebRTC

Το WebRTC παρέχει στους προγραμματιστές των web εφαρμογών την ικανότητα να γράφουν εμπλουτισμένα, με εφαρμογές πολυμέσων (σκεφτείτε το βίντεο chat) σε πραγματικό χρόνο στο διαδίκτυο, χωρίς να απαιτούνται plugins, λήψεις ή εγκαταστάσεις. Σκοπός του είναι να βοηθήσει στην κατασκευή μιας ισχυρής πλατφόρμας RTC η οποία λειτουργεί σε πολλαπλά προγράμματα περιήγησης στο Web, σε πολλαπλές πλατφόρμες.

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"



Η αρχιτεκτονική του WebRTC χωρίζεται σε δύο ξεχωριστά κομμάτια: στο Web API, το οποίο αφορά τους προγραμματιστές Web εφαρμογών και στο WebRTC, που περιλαμβάνει το C++ και το PeerConnection API, το οποίο απευθύνεται στους προγραμματιστές που ασχολούνται με browser εφαρμογές.

Η δόμηση του ανωτέρω σχήματος αναφέρεται σε τρεις διαστρωματώσεις οι οποίες αφορούν το **Web App** μας, το **Web API** και το **Web native C++ API**.

Το **Web App** αποτελεί μια τρίτης ομάδας ανάπτυξης web εφαρμογής με δυνατότητες video και ήχου συνομιλίας που τροφοδοτείται από το web API για επικοινωνίες σε πραγματικό χρόνο. Το **Web API** μέλει να χρησιμοποιηθεί από προγραμματιστές τρίτης ομάδας για την ανάπτυξη videochat-like web based εφαρμογών. Τέλος το **Web native C++ API** είναι ένα στρώμα API το οποίο επιτρέπει στους κατασκευαστές browser να εφαρμόσουν εύκολα την Web API πρόταση.

Σχετικά με το Web native C++ API αναφέρονται οι τρεις συνεδρίες διαχείρισης που αντιστοιχούν στην **μεταφορά**, την **μηχανή φωνής** και την **μηχανή video**.

Όσον αφορά την **μεταφορά**, τα συστατικά αυτής της συνεδρίας δομούνται από την επαναχρησιμοποίηση συστατικών από το libjingle, χωρίς να χρησιμοποιείται

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

ή να απαιτείται το πρωτόκολλο XMPP / κουδούνισμα. Βασικά στοιχεία αυτής της συνεδρίας είναι α) το RTP Stack το οποίο είναι μια στοίβα δικτύου για την RTP, το Real Time πρωτόκολλο β) το STUN/ICE το οποίο είναι ένα συστατικό που επιτρέπει στις κλήσεις να χρησιμοποιήσουν τους STUN και ICE μηχανισμούς για την αποκατάσταση συνδέσεων σε διάφορους τύπους δικτύων και γ) την Συνεδρία Διαχείρισης (session management) η οποία είναι ένα στρώμα αφηρημένης συνεδρίας, επιτρέποντας την εγκατάσταση κλήσης και το στρώμα διαχείρισης. Αυτό αφήνει την απόφαση εκτέλεσης πρωτοκόλλου στον προγραμματιστή εφαρμογών.

Όσον αφορά την μηχανή **φωνής**, αποτελεί ένα πλαίσιο για το μέσο ηχητικής αλυσίδας, από την κάρτα ήχου στο δίκτυο. Βασικά στοιχεία τα i) *Isac / iLBC / Opus* ii) *NetEQ για Φωνή* και iii) *Acoustic Echo Canceler και Noise Reduction*.

- Εστιάζοντας στο (i) επισημαίνεται ότι το *Isac* είναι ένας ευρείας ζώνης και υπέρ ευρείας ζώνης κωδικοποιητή ήχου για VoIP και audio ροής, επίσης το *Isac* χρησιμοποιεί 16 kHz ή 32 kHz συχνότητα δειγματοληψίας με ένα προσαρμοστικό και μεταβλητό ρυθμό bit από 12 σε 52 kbps. Το *iLBC* είναι ένας κωδικοποιητής ομιλίας στενής ζώνης για VoIP και audio ροής, χρησιμοποιεί συχνότητα δειγματοληψίας 8 kHz με δόνηση 15,2 kbps για 20ms καρέ και 13.33 kbps για 30ms καρέ και ορίζεται από το IETF RFC οι 3951 και 3952. Το *Opus* υποστηρίζει σταθερό και μεταβλητό δόνηση κωδικοποίησης από 6 kbit /s έως 510 megέθη kbit /s, megέθη καρέ από 2,5 ms έως 60 ms, και διάφορα ποσοστά δειγματοληψίας από 8 kHz (με εύρος ζώνης 4 kHz) έως 48 kHz (με 20 kHz εύρος, όπου ολόκληρο το φάσμα ακρόασης του ανθρώπινου ακουστικού συστήματος μπορεί να αναπαραχθεί) και ορίζεται από το IETF RFC 6176.
- Όσον αφορά το *NetEQ για Φωνή* (ii) είναι ένα δυναμικό ρυθμιστικό jitter και αλγόριθμος απόκρυψης σφάλματος που χρησιμοποιείται για την απόκρυψη των αρνητικών επιπτώσεων του jitter δίκτυο και την απώλεια πακέτων. Διατηρεί την λανθάνουσα κατάσταση όσο το δυνατόν χαμηλότερα, διατηρώντας παράλληλα την υψηλότερη ποιότητα φωνής.
- Το (iii) *Acoustic Echo Canceler* είναι ένα λογισμικό που βασίζεται σε συστατικό επεξεργασίας σήματος που αφαιρείται, σε πραγματικό χρόνο, η ακουστική ηχώ που προκύπτει από τη φωνή που παίζεται έξω έρχεται στο ενεργό μικρόφωνο. Το *Noise Reduction* συστατικό είναι ένα λογισμικό που βασίζεται στο συστατικό επεξεργασίας σήματος που αφαιρεί ορισμένα είδη θορύβου βάθους (background) που συνήθως συνδέονται με το VoIP. (Hiss, ο θόρυβος του ανεμιστήρα, κλπ ...)

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

Αναφορικά με την μηχανή **video** αυτή είναι ένα πλαίσιο μέσου αλυσίδας για βίντεο από την κάμερα στο δίκτυο, και από το δίκτυο στην οθόνη. Εμπλεκόμενα στοιχεία είναι 1) το *VP8* το οποίο είναι βίντεο κωδικοποιητής από το WebM του Έργου, καλά προσαρμοσμένο για το RTC καθώς έχει σχεδιαστεί για χαμηλή καθυστέρηση, 2) το *Video Jitter Buffer* για βίντεο το οποίο βοηθά να κρύψει τις συνέπειες της jitter και την απώλεια πακέτων για τη συνολική ποιότητα του video και 3) το *Image Enhancements*- βελτιώσεις εικόνας π.χ. αφαιρεί το θόρυβο του βίντεο από τη σύλληψη της εικόνας από την κάμερα[22].

4.3 Σενάρια και τρόποι λειτουργίας του WebRTC

Επικεντρώνοντας το ενδιαφέρον στην λειτουργία του WebRTC υπογραμμίζεται ότι όπως έχουμε προαναφέρει επί του παρόντος το WebRTC είναι υπό ανάπτυξη, συνδυάζοντας στοιχεία από το W3C [23] και το IETF [24], αλλά και από τμήματα του σχεδίου προδιαγραφών που έχουν ήδη υλοποιηθεί εν μέρει στο Google Chrome και Mozilla Firefox.

Ενώ τα προηγούμενα πρότυπα Ιστού του W3C επέτρεπαν επικοινωνία μόνο μεταξύ ενός προγράμματος περιήγησης και ενός διακομιστή, το WebRTC επιτρέπει άμεση επικοινωνία μεταξύ των browsers, χωρίς κανένα server στη μέση. Αυτό επιτρέπει θεωρητικά την εφαρμογή των peer-to-peer αλγορίθμων όπως το Coolstreaming ή GridMedia.

Το WebRTC API που τρέχει από το Google Chrome και το Mozilla Firefox αποτελείται από τα ακόλουθα στοιχεία επικοινωνίας λειτουργιών [4]:

- **RTCPeer Connection:** Η **RTCPeerConnection** διεπαφή παρέχει μια σύνδεση μεταξύ peers για την ανταλλαγή δεδομένων.
- **MediaStream:** Η διεπαφή **MediaStream** αντιπροσωπεύει ένα ρεύμα δεδομένων ήχου και βίντεο.
- **RTCDataChannel:** Η διεπαφή **RTCDataChannel** περιγράφει μια πλήρως αμφίδρομη σύνδεση δεδομένων μεταξύ δύο κόμβων.

Ενώ κάποιοι ερευνητές είχαν προτείνει να χρησιμοποιήσουν WebRTC στα BitTorrent συστήματα, εμείς εφαρμόζουμε ένα WebRTC μέσο για ζωντανό βίντεο ροής χρησιμοποιώντας ένα πρωτόκολλο pull-based. Ως αποτέλεσμα, αντί του περιορισμού της ανάγκης για servers αποθηκεύοντας δεδομένα στο δίκτυο, στοχεύουμε στο να μειώσουμε το φόρτο εργασίας του server.

4.3.1 ΕΦΑΡΜΟΓΗ

Στη συνέχεια, θα περιγράψουμε μια γενική λειτουργία του WebRTC για live streaming βίντεο. Το WebRTC είναι χτισμένο σε προγράμματα περιήγησης,

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

όπως ένα API και χρειάζεται να χρησιμοποιηθούν κάποιοι κώδικες JavaScript για να κληθούν οι λειτουργίες του API. Η πιο κατάλληλη λειτουργία για την εφαρμογή μας, ήταν η RTCDatChannel διασύνδεση, καθ'ότι μας επιτρέπει να στείλουμε αυθαίρετα δεδομένα. Επιπλέον στην εφαρμογή μας, χρησιμοποιείται μια τροποποιημένη έκδοση της πειραματικής βιβλιοθήκης PeerJS του JavaScript [25]. Αυτή η βιβλιοθήκη περιλαμβάνει πολλές από τις εγγενείς λειτουργίες του WebRTC σε προσαρμοσμένες λειτουργίες JavaScript και παρέχει λειτουργίες για τη δημιουργία μια σύνδεσης WebRTC μεταξύ των peers.

4.3.2 Αρχικοποίηση

Ο σκοπός του WebRTC δικτύου μας είναι κυρίως να διανέμει πακέτα βίντεο του ίδιου περιεχομένου. Αυτό το βίντεο-περιεχόμενο διανέμεται από έναν κεντρικό εξυπηρετητή, στον οποίο αναφερόμαστε ως πηγή έκδοσης ως προς τα ακόλουθα. Χρησιμοποιούμε ένα συντονιστή WebRTC ο οποίος χρησιμοποιείται για να διατηρήσει το WebRTC δίκτυο στο οποίο ο πελάτης υποστηρίζει την πηγή εκδότη στη διανομή περιεχομένου. Στην εφαρμογή μας ο συντονιστής WebRTC και η πηγή εκδότη είναι ο ίδιος ξενιστής.

Εκτός από το διακομιστή που αναφέραμε χρησιμοποιούμε ένα διακομιστή PeerJS για να υποστηρίζει τον συντονιστή WebRTC για τη δημιουργία μιας σύνδεσης ανάμεσα στους peers. Το WebRTC δεν παρέχει native API χειριστές για να συντονίζουν την επικοινωνία, όπως τη θέσπιση και τον τερματισμό της σύνδεσης. Ωστόσο, υπάρχουν JavaScript βιβλιοθήκες διαθέσιμες, οι οποίες αυτοματοποιούν τη διαχείριση της σύνδεσης. [25].

Σημειώνεται ότι η ίδια η διαχείριση της σύνδεσης είναι αρκετά απλή, δηλαδή κάθε πελάτης που επιθυμεί να ενταχθεί στο peer δίκτυο πρέπει να εγγραφεί στο διακομιστή PeerJS και πρέπει να παρέχει ένα μοναδικό ID. Για να δημιουργηθεί στη συνέχεια μια σύνδεση, ο διακομιστής PeerJS χρησιμοποιεί XmlHttpRequests and WebSockets οι οποίοι και αυτοί επικοινωνούν απ'ευθείας χρησιμοποιώντας το RTCDatChannel του WebRTC.

Για την προσθήκη ενός νέου peer στο δίκτυο δυο προϋπαρχόντων peers απαιτείται η σχετική τεχνική προσαρμογή. Αυτός ο νέος peer δημιουργεί έναν ID χρήστη και εγγράφεται στον PeerJS server και στον WebRTC coordinator. Ο WebRTC coordinator επιλέγει ψευδο τυχαία δύο peers που είναι ήδη μέρος του WebRTC δικτύου και στέλνει αυτές τις πληροφορίες στον peer που επιτυγχάνει την σύνδεση. Αυτός ο peer, στη συνέχεια, ανοίγει ένα RTCDatChannel σε αυτούς τους δύο peers και ως εκ τούτου καταλήγει να αποτελεί μέρος του δικτύου WebRTC. Ως αποτέλεσμα, ο πρώτος peer συμφωνεί μόνο με τον source publisher (εκδότη πηγή), ο δεύτερος peer συμφωνεί τόσο με τον source

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

publisher όσο και τον πρώτο peer, ενώ ο τρίτος peer επιλέγει τυχαία τους δύο peers είτε από τον πρώτο, είτε από τον δεύτερο ή από τον WebRTC coordinator (πηγή εκδότη). Έτσι, ένα τυχαίο πλέγμα δημιουργείται με κάθε νέο peer να ενώνεται με το δίκτυο.

Στην ουσία, ενώ οποιοσδήποτε peer μπορεί να δημοσιεύσει σε πολλούς συνδρομητές, κάθε peer (εκτός από τον πρώτο peer) προσυπογράφει σε ακριβώς τους δύο άλλους peers. Κατά συνέπεια, οι peers οι οποίοι έχουν πρώτοι συνδεθεί με το δίκτυο WebRTC είναι πιο κοντά στον WebRTC coordinator (πηγή εκδότη) και έχουν φυσικά περισσότερους subscribers. Ομοίως, για τους peers που παραμένουν στο δίκτυο για εκτεταμένα χρονικά διαστήματα, κερδίζουν επίσης περισσότερους subscribers.

Η συμπεριφορά αυτή εξασφαλίζει ότι οι peers με περισσότερους subscribers είναι επίσης πιο πιθανό να είναι πιο σταθεροί και πλησιέστερα στον source publisher (πηγή εκδότη).

4.3.3 Μεταφορά Δεδομένων

Για να συλληφθεί η ουσία των pull-based πρωτοκόλλων [26], αναφέρουμε ότι τα pull-based πρωτόκολλα χρησιμοποιούν δύο τύπους πακέτων: τα BufferMap and PacketRequest. Και οι δύο τύποι των pf πακέτων ανταλλάσσονται σε τακτά χρονικά διαστήματα μεταξύ των publishers (εκδοτών) και των subscribers (συνδρομητών). Ένας BufferMap περιέχει την ακολουθία των IDs όλων των δεδομένων των τμημάτων τα οποία ο peer έχει στο ρυθμιστικό του. Οι Peers οι οποίοι είναι επίσης εκδότες στέλνουν περιοδικά BufferMap πακέτα σε όλους τους συνδρομητές τους. Όταν ένας συνδρομητής λαμβάνει ένα BufferMap πακέτο, συγκρίνει το περιεχόμενο του BufferMap με το δικό του ρυθμιστικό. Τα request packets (πακέτα ζήτησης) αποστέλλονται αυτόματα από τους συνδρομητές στον κατάλληλο εκδότη για όλα τα missing streaming chunks που ένας peer ανακαλύπτει στον δικό του buffer map.

Οι Peers επικοινωνούν μεταξύ τους χρησιμοποιώντας το RTCDataChannel, που επιτρέπει την αυθαίρετη αποστολή δεδομένων. Αποφασίσαμε να χρησιμοποιήσουμε τη μορφή JSON για την κωδικοποίηση των μηνυμάτων μας, επειδή εύκολα αναλύεται από την JavaScript και επειδή η υψηλή απώλεια πακέτων μπορεί σοβαρά να επηρεάσει την εμπειρία θέασης, χρησιμοποιώντας το πρωτόκολλο TCP.

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

4.4 WebRTC MULTIPOINT CONNECTION – ΜΕΡΙΚΑ CONFERENCE MODELS

4.4.1 Ανάπτυξη του WEB RTC BROWSER για μικρά GROUP CONFERENCE

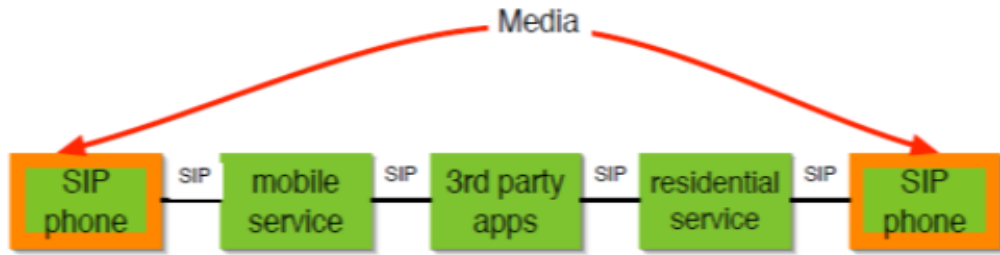
Στη περίπτωση Multipoint connection για μικρά Conference Groups αναφερόμαστε στο μοντέλο κατά το οποίο γίνεται η ανάμιξη και η ανταλλαγή των δεδομένων από έναν κεντρικό web server, και για το οποίο θα πρέπει να συμβιβαστούμε σε έναν μικρό N αριθμό συμμετεχόντων σε ένα conference. Ωστόσο αυτή η διαχείριση των δεδομένων μπορεί να πραγματοποιηθεί, στην ίδια HTTP σύνδεση η οποία είναι εγκατεστημένη με τον Web server και χρησιμοποιώντας τους κατάλληλους javascript κώδικες. Οι browsers του Web RTC θα πρέπει να χρησιμοποιούν χαμηλή κωδικοποίηση iLBC (internet Low BitRate Codec) που καταναλώνει εύρος ζώνης 13-15 Kbps [27]. Ο αριθμός των συμμετεχόντων που μπορούν να συνδεθούν $N < 10$ ώστε να μπορούν να επικοινωνούν μεταξύ τους ως ενεργοί χρήστες.

4.4.2 Μηχανισμός Σηματοδότησης

Η σηματοδότηση γίνεται με την χρήση του Web Socket. Δημιουργείται μια Web Socket σύνδεση με σκοπό να δημιουργηθεί μια Peer 2 Peer σύνδεση μεταξύ του Web Server και των συμμετεχόντων. Για να συμμετέχει κάποιος participant στο conference θα πρέπει να στείλει ένα αίτημα HTTP χρησιμοποιώντας το κατάλληλο URI. Αυτό το URI μπορεί να σταλεί με (email) ή με κάποια δημοσίευση σε κάποιο forum. Συγκεκριμένα το URI είναι η HTTP σύνδεση για να μπορέσει κανείς να συνδεθεί μέσω του Web Server στο Conference Room. Σε γενικές γραμμές, ο web server που διαχειρίζεται τέτοια διάσκεψη θα πρέπει να υποστηρίζει ορισμένες λειτουργίες ως εξής:

- Initialization (Αρχικοποίηση): Διαχειρίζεται την πρώτη πρόσβαση του συμμετέχοντα δημιουργώντας αίθουσα συνεδριάσεων στο διακομιστή, δηλαδή τη δημιουργία ενός τυχαίου και μοναδικού αριθμού για τον προσδιορισμό του συνεδρίου.
- Ένταξη Συμμετεχόντων: Για των χειρισμό των media, δηλαδή να στέλνουν και να δέχονται δεδομένα οι συμμετέχοντες. Ο web server χρησιμοποιεί την φιλοσοφία Third Party Call Control (η οποία χρησιμοποιεί το SIP protocol [28]).

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"



{Third-Party Call Control}

- Dynamic Web content: Για να στέλνονται σε κάθε συμμετέχοντα οι κατάλληλοι javascript κώδικες θα πρέπει να είναι προσαρμοσμένοι στο κατάλληλο conference, δηλαδή ο τρίτος συμμετέχοντας θα πρέπει να στέλνει και να εκτελεί ένα Javascript κώδικα ο οποίος δυναμικά θα στείλει 2 media offers στον server και στη συνέχεια κάθε media offer θα σταλεί στον κατάλληλο participant.
- Support Participant αναχωρήσεις/αποτυχίας: Σε ενδεχόμενη αποτυχία ή κάποιου προβλήματος θα πρέπει να ανιχνεύεται (χρησιμοποιώντας ένα HTML link) και σε αυτή την περίπτωση θα ενημερώνει τους υπόλοιπους συμμετέχοντες.
- Εφαρμογή πολιτικής στο Conference: Αυτή η πολιτική μπορεί να οριστεί από τον δημιουργό του room για την διαχείριση των κανόνων πρόσβασης (π.χ. Black list, κάποια προκαθορισμένα μέλη του conference , ακόμα για το πόσο θα είναι το room ενεργό).

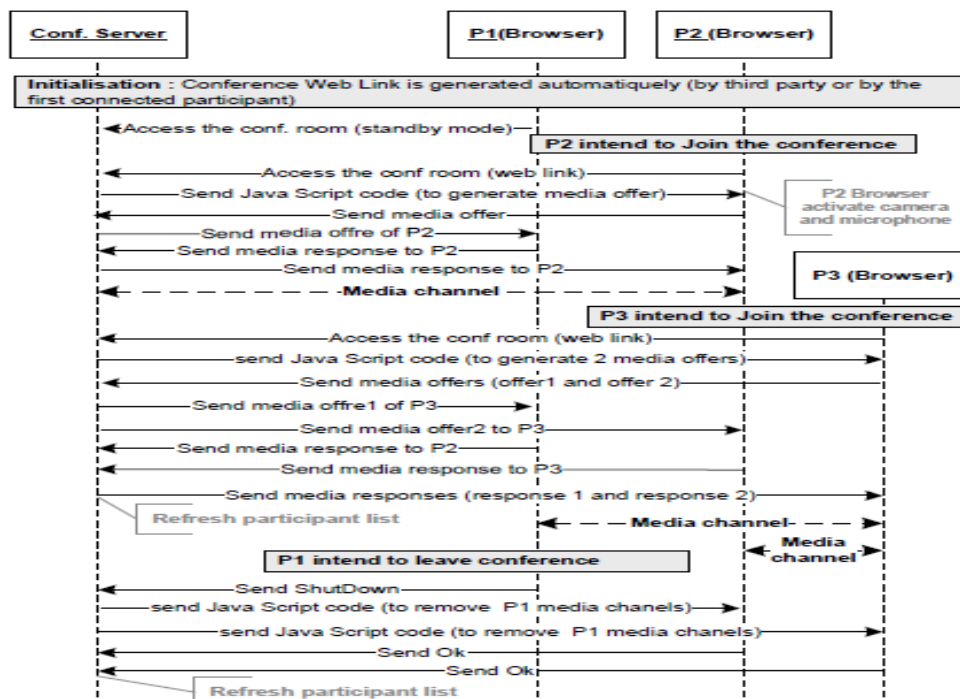


Fig. 2. Example of some signaling operations to manage fully-coupled conference

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

Το παραπάνω σχήμα δείχνει ένα παράδειγμα κάποιων εργασιών που πρέπει να εφαρμοσθούν από τον μηχανισμό σηματοδότησης.

4.4.3 Ανάπτυξη του WEB RTC BROWSER για μεγάλα GROUP CONFERENCE

Στην περίπτωση της Multipoint connection για μεγάλα Group Conference θα αναλύσουμε το μοντέλο ALM (Application Layer Multicast) και σημεία (and points), το οποίο μπορεί να υποστηρίξει μεγάλο αριθμό συμμετεχόντων[29-31].

Η ALM λύση τείνει να παράγει αυτό-οργάνωση, αποτελεσματικότητα και αυτό-βελτίωση στην επικοινωνία μεταξύ των συμμετεχόντων, που μπορεί να προσαρμόζεται δυναμικά σε διαφορετικές παραλλαγές του δικτύου. Σε αυτά τα συστήματα, ένας κόμβος συμμετεχόντων μπορεί να χρησιμοποιήσει τα media flows που παρέχονται από τους "parents"(ανωτέρους) ώστε αυτοί να μεταναστεύσουν με το constructed tree για να ελαχιστοποιηθεί η περιττή διαβίβαση media στις φυσικές συνδέσεις. Ενώ η επεξεργασία των media χρησιμοποιεί δυναμικά σχήμα δέντρου decentralized, ο έλεγχος δικτύου βασίζεται στην τοπολογία centralized γύρω από τον Web Server. Κάθε χρήστης που συνδέεται στο σύστημα μπορεί να προσφέρει υπηρεσίες επεξεργασίας των media στον administrator που ελέγχει την διανομή των media, χρησιμοποιώντας τον κατάλληλο Web Socket και την HTML5 τεχνολογία εξ' αποστάσεως για την καθιέρωση, για την ενημέρωση και για την αφαίρεση των media session μεταξύ των συμμετεχόντων. Το συγκεκριμένο μοντέλο μπορεί να κατασκευαστεί δυναμικά ανάλογα με της δραστηριότητες του χρήστη (πχ μόνο audio) και τις προτιμήσεις του χρήστη, να προσφέρει ανάμικτες υπηρεσίες (mix services) στους άλλους συμμετέχοντες και αντίστοιχα οι άλλοι συμμετέχοντες να προσφέρουν ανάμικτες υπηρεσίες σε αυτόν. Αυτό το μοντέλο μπορεί να χρησιμοποιηθεί από το Web RTC browsers για να δημιουργήσει και VoIP και MMoiP conference.

4.4.4 Περίπτωση με VoIP Conference

Στην περίπτωση του VoIP για μεγάλο αριθμό συμμετεχόντων το προτιμώμενο μοντέλο έχει δύο διαφορετικά mashed δίκτυα τα οποία επιτρέπουν την κατανομή της φωνής μεταξύ του Browser και του γενικού conference control . Το δίκτυο των media για την κατανομή της φωνής χρησιμοποιεί το σχήμα δέντρου (tree-shaped). Να σημειωθεί ότι παρακάτω θα υπάρχουν οι εξής συμβολισμοί ώστε να γίνει πιο εύκολη η περιγραφή του συστήματος P2P. Θα χρησιμοποιήσουμε 3 διαφορετικές συνιστώσες: MDN (Mixer/Distributor Node), DN(Distributor Node), και LN (Leaf Node).

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

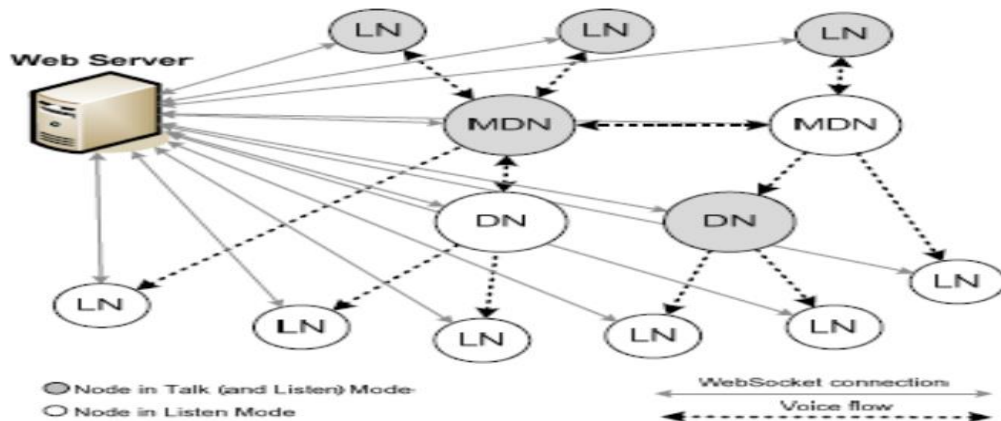


Fig. 3. WebRTC browser interconnection for large scale VoIP conference

Ο ρόλος του MDN είναι ότι μπορεί να υποστηρίξει πάρα πάνω από μια audio σύνδεση στο συνέδριο. Επίσης μπορεί να διαχειριστεί και να διανείμει media audio προς άλλα μέλη του conference. Ο κόμβος DN(Distributor Node) παίρνει τα media από τους ανωτέρους του, μπορεί να τα <<παίξει>> τοπικά και μπορεί να τα στείλει στα <<παιδιά του>> (στα κατώτερα μέλη). Τέλος ο ρόλος του LN(Leaf Node) μπορεί να λάβει μόνο μια συνεδρία ήχου. Η αναχώρηση των (LN) από το συνέδριο δεν θα πρέπει να επηρεάζει κάποιους από τους συμμετέχοντες. Αυτός ο ρόλος έχει δημιουργηθεί για απλές φορητές συσκευές με IP ή για χρήστες που δεν θα προσφέρουν μέρος υπολογιστικής ισχύος στο συνέδριο. Επίσης η αρχιτεκτονική αυτή λαμβάνει υπόψη και τις λειτουργίες που θα θέλει ο κάθε χρήστης (ακροατής ή ομιλητής).

4.4.5 Περίπτωση με MMoIP Conference

Σε αυτή την περίπτωση χρησιμοποιεί την ίδια αρχιτεκτονική με το VoIP μόνο που στο Video conference δεν χρειάζεται κάποιος κόμβος να κάνει την ανάμιξη. Οι κόμβοι που χρησιμοποιούνται είναι οι DN(Distributor Node) και οι LN (Leaf Node) Διαφορετικά και παράλληλα κανάλια θα δημιουργηθούν μεταξύ συμμετεχόντων κόμβων για video streaming επικοινωνία .Με την χρήση του RTCDatChannel interfaces εντός του Data API Peer-To-Peer, ο DN κόμβος λειτουργεί σαν μεσολαβητής(proxy) και είναι σε θέση να λάβει και να στείλει media. Ανά πάσα στιγμή, οι DN μπορούν να λειτουργήσουν σαν "Presenter" μέσα στο conference. Σε αυτή την περίπτωση, το video streaming μπορεί να μεταδοθεί στους γειτονικούς κόμβους,δημιουργώντας μια τοπολογία στην οποία επιτρέπεται η μετάδοση και video και audio μέσα στην ίδια conference. Επίσης θα πρέπει να σχεδιαστεί μια topologia media ώστε DN να λειτουργούν και σαν VoIP MDN.

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

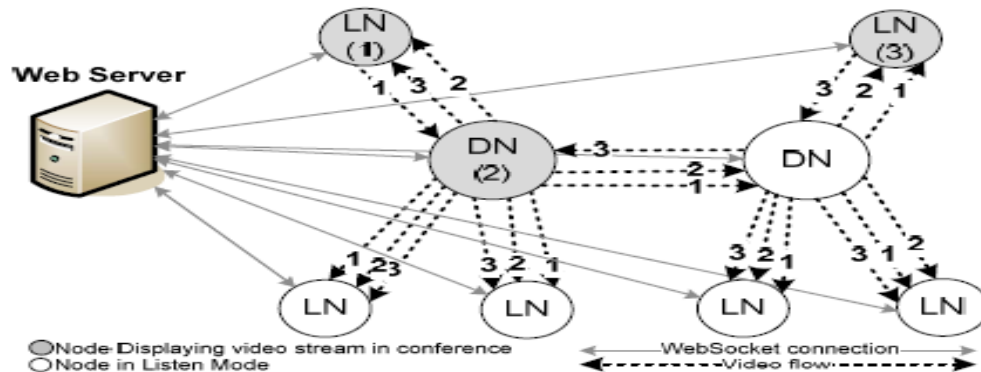


Fig. 4. Media flow distribution

4.5 Η τεχνολογία WebRTC σε κινητές συσκευές

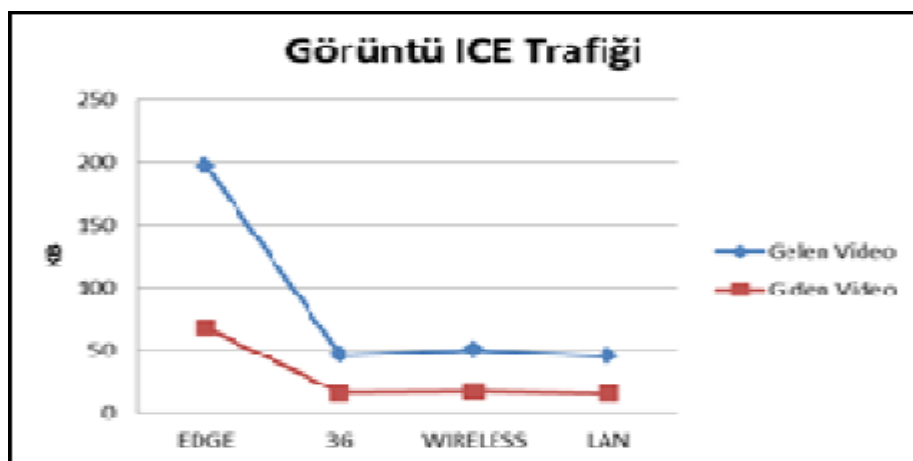
Το WebRTC μαζί με την εξασφάλιση της σύνδεσης μεταξύ προσωπικών υπολογιστών έχει ήδη λάβει θέση στον κόσμο των κινητών από το πρώτο τρίμηνο του 2014.

Ωστόσο, υπάρχει ανάγκη για ανάπτυξη της ισχύος του επεξεργαστή που απαιτεί η υπάρχουσα βιβλιοθήκη και σχετικά με τις ανάγκες της σύνδεσης του δικτύου, ειδικά για κλίσεις εικόνες σε κωδικοποιητές (σειρά VP) και στο θέμα των μηχανισμών ελέγχου σφαλμάτων στην μετάδοση των media. Η ταχύτητα και η χωρητικότητα που χρησιμοποιείται στους TURN διακομιστές για τον προσανατολισμό των media, μπορεί να προκαλέσει την καθυστέρηση των πακέτων UDP και την αξιολόγηση αυτών ως χαμένα. Τα συστήματα ελέγχου των σφαλμάτων που προτείνονται ως λύση για τα χαμένα πακέτα και απασχολούν το δίκτυο και τον επεξεργαστή, ο θόρυβος στον ήχο/εικόνα και η μπαταρία της συσκευής δημιουργούν αρνητικές αντανακλάσεις.

Όταν είναι απασχολημένο το δίκτυο και ο επεξεργαστής, η αμφίδρομη σύνδεση (έναρξη κλήσης και κοινοποίηση εισερχόμενων κλήσεων) όπως απαιτείται για την δημιουργία σύνδεσης και η κατανάλωση ενέργειας δημιουργούν αρνητικές επιπτώσεις. Υπάρχει ανάγκη στην παραλαβή της κοινοποίησης της κλήσης στην περίπτωση χρήσης της μεθόδου Longpolling, να προκαλέσει την κίνηση των πρόσθετων δεδομένων μαζί με την ρύθμιση της σύνδεσης σε συγκεκριμένα διαστήματα. Μπορεί να χρησιμοποιηθεί αποτελεσματικά ο PUSH μηχανισμός (Apple Push Notification (APNS) και Android Push Notification (C2DM)) ειδικά στα λειτουργικά συστήματα για φορητές συσκευές. Αναμένεται στην χρήση της τεχνολογίας PUSH η αξιοπιστία του διακομιστή και τα ενημερωτικά μηνύματα να μεταφέρονται στη συσκευή σε μερικά δευτερόλεπτα. Απορρίπτονται τα ICE

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

Connectivity Check (STUN) μηνύματα με στόχο την διασφάλιση της συνέχειας των ανοιχτών θυρών για NAT συσκευές. Στην έκδοση 32.0.1700.76 του Chrome για την επικοινωνία ήχου και εικόνας (ήχος και εικόνα σε ξεχωριστές θύρες) προκαλείται κίνηση περίπου 2.240 byte το δευτερόλεπτο. Σε ένα περιβάλλον για μια συνομιλία 60 λεπτών την ημέρα μόνο για τα STUN μηνύματα προκύπτει κυκλοφορία περίπου 230 MB ανά μήνα. Είναι σαφές ότι θα οδηγήσει σε υψηλότερο κόστος η κατανάλωση ενέργειας που χρειάζεται για την κυκλοφορία καθώς και για τους χρήστες με ιδιόκτητες γραμμές δεδομένων. Υπάρχει ανάγκη για εργασίες για έρευνα και ανάπτυξη με θέμα τις εφαρμογές πελατών που είναι ιδιοκτήτες σε σχέδιο δεδομένων 3G/4G. Μεταξύ των λύσεων που μπορούν να εφαρμοστούν, για την διατήρηση της σύνδεσης σε ένα περιβάλλον υπάρχουν μελέτες με στόχο την εξάλειψη των ICE Connectivity Check μηνυμάτων. Μπορεί να εμποδιστεί η αποστολή των STUN πακέτων με τις ρυθμίσεις που θα γίνουν για τις WebRTC εφαρμογές που θα δουλέψουν στα δίκτυα των φορέων που παρέχουν 3G/4G υπηρεσίες για τις NAT συσκευές και με τις ρυθμίσεις που θα γίνουν στις WebRTC βιβλιοθήκες των πελατών.[32]

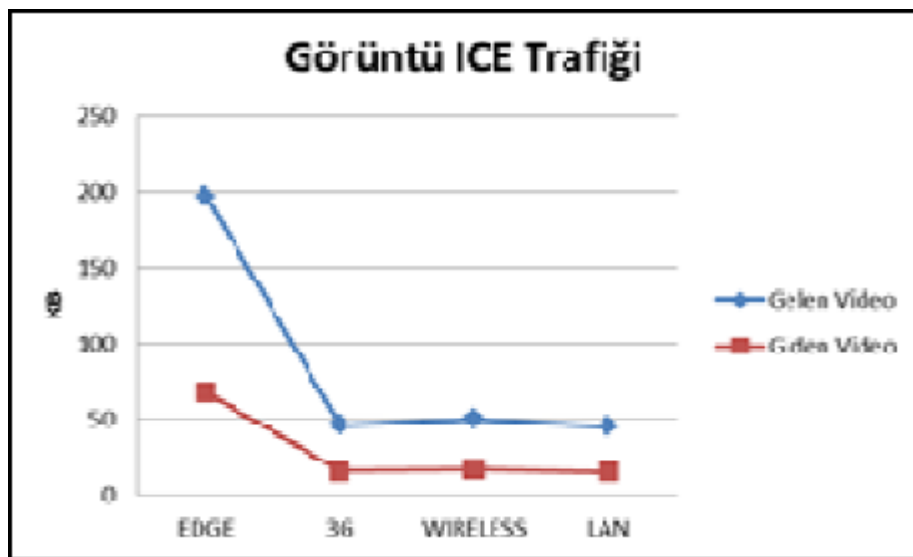


Σχήμα 3. Κυκλοφορία ICE Ήχου

Ses&Görüntü=Ήχος&Εικόνα **Ses=Ήχος**

Gelen Ses=Εισερχόμενος Ήχος **Giden Ses=Εξερχόμενος Ήχος**

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"



Σχήμα 4. Κυκλοφορία ICE Εικόνας

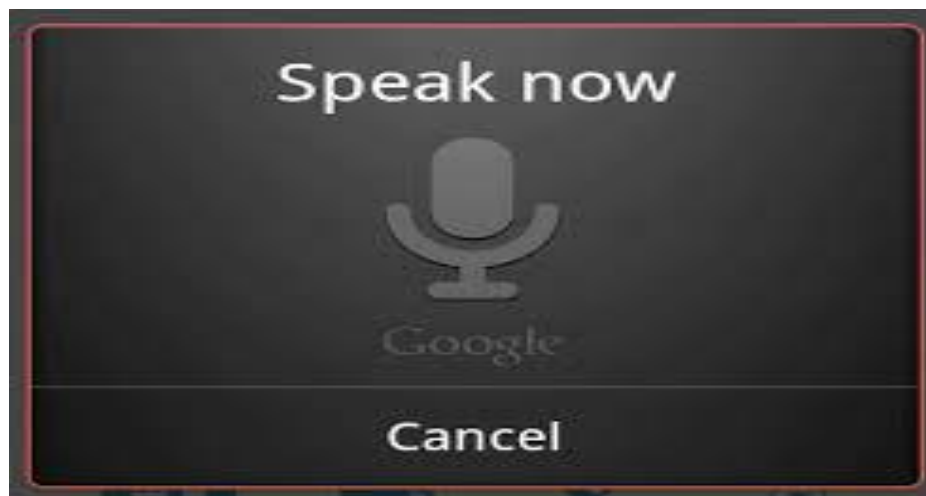
Gelen Video=Εισερχόμενο Βίντεο Giden Video=Εξερχόμενο Βίντεο

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

5 ASR (automatic speech recognition)

5.1 Speech Recognition

Το SR (Speech Recognition) στην επιστήμη των υπολογιστών και των μηχανικών το SR σημαίνει αναγνώριση ομιλίας, δηλαδή είναι η μετάφραση των προφορικών λέξεων σε κείμενο. Είναι, επίσης, γνωστό και ως αυτόματη αναγνώριση ομιλίας "automatic speech recognition" (ASR) ή απλά σε κείμενο την ομιλία "speech to text" (STT).



Κάποια συστήματα αναγνώρισης ομιλίας χρησιμοποιούν "αναγνώριση ομιλίας ανεξάρτητη από τον "ομιλητή" ενώ άλλα χρησιμοποιούν "εξάσκηση", όπου ένα άτομο διαβάζει κομμάτια κειμένου σε ένα σύστημα αναγνώρισης ομιλίας. Τότε, αυτού του είδους τα συστήματα αναλύουν τη φωνή ενός ομιλητή και την χρησιμοποιούν για να προσαρμόσουν την αναγνώριση της ομιλίας του συγκεκριμένου ατόμου από τον υπολογιστή, με αποτέλεσμα την πιο ακριβή καταγραφή της. Συστήματα που δεν χρησιμοποιούν εξάσκηση ονομάζονται συστήματα "ανεξάρτητα από τον ομιλητή".

5.2 Τι είναι η αυτόματη αναγνώριση ομιλίας ASR;

Η αυτόματη αναγνώριση ομιλίας (ASR) μπορεί να οριστεί ως η ανεξάρτητη καθοδηγούμενη από τον υπολογιστή μεταγραφή της προφορικής γλώσσας σε αναγνώσιμο κείμενο σε πραγματικό χρόνο (Stuckless, 1994). Με λίγα λόγια, ASR είναι η τεχνολογία που επιτρέπει σε έναν υπολογιστή να εντοπίζει τις λέξεις που ένα άτομο εκφέρει προφορικά σε ένα μικρόφωνο ή τηλέφωνο και να τις μετατρέπει σε γραπτό κείμενο.

Με το δεδομένο ότι έχουμε μια μηχανή που καταλαβαίνει τον ευχερή προφορικό λόγο, έχει οδηγηθεί η έρευνα ομιλίας για περισσότερα από 50 χρόνια. Αν και η

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

ASR τεχνολογία δεν είναι ακόμη στο σημείο όπου οι μηχανές δύνανται να κατανοήσουν όλη την ομιλία, σε οποιοδήποτε ακουστικό περιβάλλον, ή από οποιοδήποτε πρόσωπο, παρ'όλα αυτά χρησιμοποιείται σε καθημερινή βάση σε μια σειρά εφαρμογών και υπηρεσιών.

Ο απώτερος στόχος της ASR έρευνας είναι να επιτρέπει σε έναν υπολογιστή να αναγνωρίζει σε πραγματικό χρόνο, με ακρίβεια 100%, όλες τις λέξεις που είναι κατανοητά ομιλούμενες από οποιοδήποτε πρόσωπο, ανεξαρτήτως του μεγέθους του λεξιλογίου, του θορύβου, των χαρακτηριστικών των ηχείων ή της προφοράς. Σήμερα, αν το σύστημα εκπαιδευτεί να μάθει τη φωνή ενός ομιλητή, τότε πολύ πιο πλούσια λεξιλόγια καθίστανται δυνατά και η ακρίβεια μπορεί να είναι μεγαλύτερη από 90%.

Τα εμπορικά διαθέσιμα συστήματα ASR συνήθως απαιτούν μόνο μια σύντομη περίοδο εκπαίδευσης ομιλητή και μπορεί με επιτυχία να συλλάβουν συνεχή ομιλία με πλούσιο λεξιλόγιο σε κανονικό ρυθμό με πολύ μεγάλη ακρίβεια. Οι περισσότερες εμπορικές εταιρείες ισχυρίζονται ότι το λογισμικό αναγνώρισης μπορεί να επιτύχει ακρίβεια μεταξύ 98% και 99% εάν λειτουργεί υπό βέλτιστες συνθήκες. «Οι βέλτιστες συνθήκες» συνήθως υποθέτουν ότι οι χρήστες: έχουν χαρακτηριστικά ομιλίας που ταιριάζουν στα εκπαιδευτικά δεδομένα, μπορούν να επιτύχουν τη κατάλληλη προσαρμογή του ομιλητή, και να εργαστούν σε ένα καθαρό από θόρυβο περιβάλλον (π.χ. ήσυχο χώρο)

Αυτό εξηγεί γιατί ορισμένοι χρήστες, ιδίως εκείνοι των οποίων η ομιλία είναι με βαριά προφορά, μπορεί να επιτύχουν ποσοστά αναγνώρισης πολύ χαμηλότερα από τα αναμενόμενα.

5.3 Η Ιστορία της ASR Τεχνολογίας

Οι πρώτες προσπάθειες για να αναπτύξουν συστήματα για την αυτόματη αναγνώριση ομιλίας μέσω μηχανής έγιναν στη δεκαετία του 1950. Μεγάλο μέρος της αρχικής έρευνας που οδήγησε στην ανάπτυξη της ενεργοποίησης του λόγου και την τεχνολογία αναγνώρισης χρηματοδοτήθηκε από το Εθνικό Ίδρυμα Επιστημών (NSF) και του Υπουργείου Άμυνας Defense Advanced Research Οργανισμός Έργων (DARPA). Μεγάλο μέρος της αρχικής έρευνας, που εκτελέστηκε με χρηματοδότηση της NSA και της NSF, διεξήχθη τη δεκαετία του 1980. (Πηγή: Παγκόσμια Security.Org)

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

Η τεχνολογία αναγνώρισης ομιλίας σχεδιάστηκε αρχικά για τα άτομα με αναπηρία. Για παράδειγμα, η αναγνώριση φωνής μπορεί να βοηθήσει τα άτομα με μυοσκελετικές αναπηρίες, που προκλήθηκαν από σκλήρυνση κατά πλάκας, εγκεφαλική παράλυση, ή αρθρίτιδα, να επιτύχουν μέγιστη παραγωγικότητα μέσω υπολογιστών.

Στις αρχές της δεκαετίας του 1990, εμφανίστηκαν τεράστιες ευκαιρίες στην αγορά για την αναγνώριση ομιλίας της τεχνολογίας των υπολογιστών. Οι πρώτες εκδόσεις των προϊόντων αυτών ήταν ογκώδεις και δύσκολες στη χρήση. Τα αρχικά συστήματα αναγνώρισης γλωσσών έπρεπε να κάνουν συμβιβασμούς: ήταν «συντονισμένοι» να είναι εξαρτημένα από ένα συγκεκριμένο ομιλητή, ή είχαν μικρό λεξιλόγιο, ή χρησιμοποιούσαν μία πολύ στυλιζαρισμένη και άκαμπτη σύνταξη. Ωστόσο, στην βιομηχανία των υπολογιστών, τίποτα δεν μένει το ίδιο για πολύ καιρό και μέχρι το τέλος της δεκαετίας του 1990 υπήρχε μια ολόκληρη νέα γενιά εμπορικών πακέτων λογισμικού αναγνώρισης ομιλίας που ήταν πιο εύκολα στη χρήση και πιο αποτελεσματικά από τους προκατόχους τους.

Κατά τα τελευταία χρόνια, η τεχνολογία αναγνώρισης ομιλίας έχει προχωρήσει στο σημείο όπου χρησιμοποιείται από εκατομμύρια άτομα για την αυτόματη δημιουργία εγγράφων με υπαγόρευση. Ιατρικοί δακτυλογράφοι ακούνε υπαγορευμένες καταγραφές που πραγματοποιούνται από γιατρούς και άλλους επαγγελματίες υγείας και τα μετατρέπουν σε ιατρικές εκθέσεις, αλληλογραφία και άλλο διοικητικό υλικό. Μια ολοένα και πιο δημοφιλής μέθοδος χρησιμοποιεί την τεχνολογία αναγνώρισης ομιλίας, με την οποία μεταφράζεται σε ηλεκτρονική μορφή ο ήχος σε κείμενο και δημιουργεί αντίγραφα καθώς και σχέδια των εκθέσεων.

Οι μεταγραφές και εκθέσεις στη συνέχεια διαμορφώνονται, επεξεργάζονται για λάθη στη μετάφραση, τη στίξη, ή τη γραμματική και ελέγχονται για τη συνοχή και τυχόν λάθη. Δακτυλογράφοι που εργάζονται σε τομείς με τυποποιημένη ορολογία, όπως ακτινολογία ή παθολογία, είναι πιο πιθανό να αντιμετωπίσουν τεχνολογία αναγνώρισης ομιλίας. Η χρήση της τεχνολογίας αναγνώρισης φωνής θα γίνει πιο διαδεδομένη, καθώς η τεχνολογία εξελίσσεται όλο και περισσότερο.

Μερικοί συγγραφείς φωνής παράγουν ένα αντίγραφο σε πραγματικό χρόνο, χρησιμοποιώντας την τεχνολογία υπολογιστών αναγνώρισης ομιλίας. Οι συγγραφείς ομιλίας που παρέχουν δυνατότητα αναγνώρισης φωνής επιδιώκουν όχι μόνο σταδιοδρομία σε δικαστικές εκθέσεις, αλλά επίσης και σταδιοδρομία όπως αυτών με κλειστές λεζάντες (σύντομες εξηγήσεις) και τους παρόχους κειμένων συνεχούς ροής στο Διαδίκτυο ή τους παρόχους λεζάντας.

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

5.4 Πως δουλεύει το ASR

Ο στόχος ενός συστήματος ASR είναι με ακρίβεια και αποτελεσματικότητα να μετατρέπει ένα σήμα ομιλίας σε ένα αντίγραφο μηνύματος κειμένου προφορικών λέξεων ανεξάρτητα από τον ομιλητή, το περιβάλλον ή την συσκευή που χρησιμοποιείται για την καταγραφή της ομιλίας (δηλαδή το μικρόφωνο).

Αυτή η διαδικασία ξεκινά όταν ένας ομιλητής αποφασίζει τι να πει και πραγματικά εκφέρει μια φράση. (Αυτό είναι μια ακολουθία λέξεων, ενδεχομένως με παύσεις, εεε, και εμμ...) Στη συνέχεια το λογισμικό παράγει μια κυματομορφή ομιλίας, η οποία ενσωματώνει τα λόγια της πρότασης, καθώς και τους εξωτερικούς ήχους και παύσεις στην ομιλούμενη είσοδο. Ακολουθώς, το λογισμικό προσπαθεί να αποκωδικοποιήσει την ομιλία στην καλύτερη εκτίμηση της πρότασης. Πρώτα μετατρέπει το σήμα ομιλίας σε μία ακολουθία διανυσμάτων τα οποία μετρώνται καθ'όλη τη διάρκεια του σήματος ομιλίας. Κατόπιν, χρησιμοποιώντας ένα συντακτικό αποκωδικοποιητή, δημιουργεί μια έγκυρη ακολουθία των παραστάσεων. (Rabiner & Juang, 2004)[33]

5.4.1 METHODS, AND ALGORITHMS

Τόσο η ακουστική μοντελοποίηση όσο και η γλώσσα μοντελοποίησης είναι σημαντικά μέρη των σύγχρονων στατιστικά βασισμένων αλγορίθμων αναγνώρισης ομιλίας. Τα Hidden Markov models(HMMs) χρησιμοποιούνται ευρέως σε πολλά συστήματα.

Hidden Markov models(Hidden Markov models)

Τα σύγχρονα συστήματα αναγνώρισης ομιλίας βασίζονται στα Hidden Markov Models. Αυτά είναι στατιστικά μοντέλα τα οποία εξάγουν μια ακολουθία από σύμβολα ή ποσότητες. Τα HMM χρησιμοποιούνται στην αναγνώριση ομιλίας, διότι ένα σήμα λόγου μπορεί να θεωρηθεί ως ένα τμηματικά στάσιμο σήμα ή ένα στατικό σήμα σύντομου χρόνου. Σε κλίμακα σύντομου χρονικού διαστήματος (π.χ., 10 χιλιοστά του δευτερολέπτου), η ομιλία μπορεί να προσεγγιστεί ως μία στάσιμη διαδικασία. Η ομιλία μπορεί να θεωρηθεί ως ένα μοντέλο Markov για πολλούς στοχαστικούς σκοπούς[34].

Δυναμική χρονική στρέβλωση (DTW) αναγνώριση της ομιλίας(Dynamic time warping (DTW)-based speech recognition)

Δυναμική χρονική στρέβλωση είναι μια προσέγγιση που ιστορικά χρησιμοποιήθηκε για την αναγνώριση ομιλίας, αλλά έχει πλέον σε μεγάλο βαθμό αντικατασταθεί από την πιο επιτυχημένη προσέγγιση βασισμένη στο HMM.

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

Η δυναμική χρονική στρέβλωση είναι ένας αλγόριθμος για τη μέτρηση της ομοιότητας μεταξύ δύο ακολουθιών που μπορούν να ποικίλουν στο χρόνο ή την ταχύτητα. Για παράδειγμα, οι ομοιότητες στα πρότυπα με τα πόδια θα ανιχνεύονταν, ακόμη και αν σε ένα βίντεο το πρόσωπο περπατούσε αργά και εάν σε άλλο αυτός ή αυτή περπατούσαν πιο γρήγορα, ή ακόμα και αν υπήρχαν επιταχύνσεις και επιβραδύνσεις κατά τη διάρκεια μιας παρατήρησης. Η DTW έχει εφαρμοστεί σε βίντεο, ήχου και γραφικών - πράγματι, όλα τα δεδομένα που μπορεί να μετατραπούν σε μια γραμμική αναπαράσταση μπορεί να αναλυθούν με DTW[35].

Τα νευρωνικά δίκτυα(Neural networks)

Τα νευρωνικά δίκτυα αναδείχτηκαν ως μια ελκυστική προσέγγιση ακουστικής μοντελοποίησης ASR στα τέλη της δεκαετίας του 1980. Από τότε, τα νευρωνικά δίκτυα έχουν χρησιμοποιηθεί σε πολλές πτυχές της αναγνώρισης ομιλίας, όπως η ταξινόμηση φωνήματος, [36] απομονωμένη αναγνώριση λέξεων, [37] και η προσαρμογή των ηχείων.

Σε αντίθεση με τα HMMs, τα νευρωνικά δίκτυα δεν κάνουν υποθέσεις σχετικά με τις ιδιότητες των στατιστικών χαρακτηριστικών και έχουν πολλές ιδιότητες που τα καθιστούν ελκυστικά μοντέλα αναγνώρισης για την αναγνώριση ομιλίας. Όταν χρησιμοποιούνται για να εκτιμήσουν τις πιθανότητες ενός χαρακτηριστικού τμήματος της ομιλίας, τα νευρωνικά δίκτυα επιτρέπουν διακριτική εκπαίδευση με ένα φυσικό και αποτελεσματικό τρόπο. Κάποιες παραδοχές σχετικά με τις στατιστικές των χαρακτηριστικών εισόδου γίνονται με νευρωνικά δίκτυα. Ωστόσο, παρά την αποτελεσματικότητά τους, στον χαρακτηρισμό των μονάδων σύντομου χρόνου, όπως τα ατομικά τηλέφωνα και οι μεμονωμένες λέξεις, [38] τα νευρωνικά δίκτυα είναι σπανίως επιτυχή για τα έργα συνεχούς αναγνώρισης, κυρίως λόγω της έλλειψης της ικανότητας να διαμορφώσουν χρονικές εξαρτήσεις.

Βαθιά Νευρωνικά Δίκτυα και άλλων Μοντέλων βαθιάς Μάθησης(Deep Neural Networks and Other Deep Learning Models)

Ένα βαθύ νευρωνικό δίκτυο (DNN) είναι ένα τεχνητό νευρωνικό δίκτυο με πολλαπλά κρυμμένα στρώματα μονάδων μεταξύ των στρωμάτων εισόδου και εξόδου. [39] Παρόμοια με ρηχά νευρωνικά δίκτυα, οι DNNs μπορούν να μοντελοποιήσουν πολύπλοκες μη γραμμικές σχέσεις. Οι DNN αρχιτεκτονικές παράγουν συνθετικά μοντέλα, όπου επιπλέον στρώματα καθιστούν δυνατή τη σύνθεση των χαρακτηριστικών από τα χαμηλότερα στρώματα, δίνοντας

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

τεράστιες δυνατότητες μάθησης και έτσι το δυναμικό της μοντελοποίησης πολύπλοκων μορφών δεδομένων ομιλίας. [40] Οι DNN είναι το πιο δημοφιλές είδος των αρχιτεκτονικών βαθιάς μάθησης που χρησιμοποιείται επιτυχώς ως ένα ακουστικό μοντέλο για την αναγνώριση ομιλίας από το 2010.

Η επιτυχία των DNNs σε μεγάλο λεξιλόγιο αναγνώρισης ομιλίας παρουσιάστηκε το 2010 από βιομηχανικούς ερευνητές, σε συνεργασία με ακαδημαϊκούς ερευνητές, όπου μεγάλα στρώματα εξόδου των DNNs βασισμένα σε πλαίσιο εξαρτώμενο από τις HMM καταστάσεις κατασκευασμένα από δένδροειδείς αποφάσεις, υιοθετήθηκαν.

Από το αρχικό επιτυχημένο ντεμπούτο των DNNs για την αναγνώριση ομιλίας περίπου το 2009-2011, υπήρξαν τεράστιες νέες πρόοδοι που έχουν επιτευχθεί. Αυτή η πρόοδος (καθώς και οι μελλοντικές κατευθύνσεις) έχουν συνοψιστεί στους ακόλουθους οκτώ κύριους τομείς: [41] [40] [42]

1. Κλιμάκωση / αποκλιμάκωση και επιτάχυνση DNN κατάρτισης και αποκωδικοποίησης
2. Ακολουθία διακριτικής εκπαίδευσης των DNNs
3. Επεξεργασία χαρακτηριστικών από βαθιά μοντέλα με στερεά κατανόηση των υποκείμενων μηχανισμών
4. Προσαρμογή των DNNs και των συναφών βαθέων μοντέλων
5. Multi-εργασία και μεταφορά μάθησης από DNNs και συναφών βαθέων μοντέλων
6. Συνέλιξη νευρικών δικτύων και πώς να τα σχεδιάσουν ώστε καλύτερα να αξιοποιήσουν τις βέλτιστες γνώσεις του πεδίου του λόγου
7. Επαναλαμβανόμενο νευρωνικό δίκτυο και την πλούσια LSTM παραλλαγή του
8. Άλλοι τύποι βαθέων μοντέλων, συμπεριλαμβανομένων μοντέλων βασισμένων σε ένταση και ολοκληρωμένα βαθιά παραγωγικά / διακριτικά μοντέλα.

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

5.5 Ποιο είναι το όφελος του ASR

Υπάρχουν ουσιαστικά τρεις βασικοί λόγοι για τους οποίους τόσο πολλή έρευνα και προσπάθεια έχει καταβληθεί σχετικά με το πρόβλημα να διδάξει τις μηχανές πως να αναγνωρίσουν και να κατανοήσουν την ομιλία:

- Προσβασιμότητα για κωφούς και βαρήκοους
- Μείωση του κόστους μέσω αυτοματισμού
- Δυνατότητα αναζήτησης κειμένου

5.6 Τι έχει συμβεί σε ASR

Εκτός από τους επιστήμονες και τεχνικούς που ασχολούνται με την έρευνα και την ανάπτυξη ASR, οι περισσότεροι άνθρωποι που σκέφτονται για ASR υποτιμούν την πολυπλοκότητα του. Είναι κάτι περισσότερο από αυτόματο «κείμενο προς λόγο» (text-to-speech), το ASR απαιτεί γρήγορους υπολογιστές με μεγάλη χωρητικότητα δεδομένων και μνήμης – μία απαραίτητη προϋπόθεση για πολύπλοκους σκοπούς αναγνώρισης, καθώς και τη συμμετοχή των επιστημόνων ομιλίας, των γλωσσολόγων, των επιστημόνων πληροφορικής, μαθηματικών και μηχανικών.

Η έρευνα εστιάζεται για τα ASR συστήματα που ενσωματώνουν τα τρία χαρακτηριστικά: πλούσια λεξιλόγια, δυνατότητες συνεχούς ομιλίας, και την ανεξαρτησία των ομιλητών. Σήμερα, υπάρχουν πολυάριθμα συστήματα που ενσωματώνουν αυτούς τους συνδυασμούς[33].

Επίσης κρίνεται σκόπιμο να αναφέρουμε για ποιους σκοπούς ήδη χρησιμοποιείται:

5.6.1 Applications

Το SR API μπορεί να χρησιμοποιηθεί στα παρακάτω[43]:

1. Εσωτερικά συστήματα αυτοκινήτου (In-car systems).
2. Φροντίδα Υγείας (Health care)
3. Στρατός (Military).
4. Τηλεφωνία και άλλα συναφή (Telephony and other domains)
5. Χρήση στην Εκπαίδευση και στην καθημερινή ζωή (Usage in education and daily life)
6. Άνθρωπους με αναπηρίες (People with disabilities)

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

5.7 The HTML5 Speech Recognition API

Η HTML5 Speech Recognition API επιτρέπει στην javascript να έχει πρόσβαση σε έναν Browser στο audio stream και να το μετατρέπουν σε κείμενο.

Basic usage

```
var recognition = new webkitSpeechRecognition();

recognition.onresult = function(event) {

    console.log(event)

}
```

Αυτό θα καλέσει το χρήστη να επιτρέψει την πρόσβαση σε μια σελίδα μικροφώνου του. Αν επιτρέψουμε την πρόσβαση, μπορούμε να αρχίσουμε να μιλάμε και όταν σταματήσουμε η onResult event θα ενεργοποιηθεί, καθιστώντας τα αποτελέσματα της καταγραφής ως object JavaScript.

Μετά παίρνουμε το javascript object με τα αποτελέσματα μας.

```
{
  ..
  results: {
    o: {
      o: {
        confidence: 0.695017397403717,
        transcript: "look Mum I'm talking into a web page!"
      },
      isFinal:true,
      length:1
    },
    length:1
  }
```

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

Streaming results

Η εμφάνιση των αποτελεσμάτων θα μπορεί να γίνει είτε όταν σταματήσουμε να μιλάμε είτε όταν θα έχει γίνει η επεξεργασία των δεδομένων τους.

```
var recognition = new webkitSpeechRecognition();

recognition.continuous = true;

recognition.interimResults = true;

recognition.onresult = function(event) {

    console.log(event)

}

recognition.start();
```

Αυτό σημαίνει ότι μπορεί να ξεκινήσει την μετατροπή των object σε κείμενο πρώτου αρχίσει να μιλάει. (πχ: <http://blogs.sitepointstatic.com/examples/tech/speech-input/index.html>)

Accounting for accents

Μπορούμε επίσης να ρυθμίσουμε την γλώσσα αναλόγως με τον ομιλητή. Αν είμαστε από την Ελλάδα μπορούμε να βάλουμε «el-GR».

```
var recognition = new webkitSpeechRecognition();

recognition.lang = "el-GR";
```

Security

Οι Page hosts στην HTML χρειάζονται να ζητήσουν άδεια (permission) κάθε φορά που θέλουν να κάνουν καταγραφή της φωνής. Είναι ο ίδιος τρόπος που ακολουθείται όταν μας κάνει ερώτηση ο Browser μας. Οι σελίδες της Https δεν ξανά κάνουν ερώτηση για πρόσβαση. Αυτό είναι ένα καλό στοιχείο για την ασφάλεια, διότι αν κάποιος παραβιάσει το room θα πρέπει είτε να έχει τα δικαιώματα είτε να ξανά κάνει την ερώτηση.

Στο πλαίσιο ολόκληρης της σελίδας η javascript ,έχει πρόσβαση στην έξοδο του audio capture.Τότε αν τα δεδομένα είναι σε κίνδυνο δεν θα μπορέσουν να διαβαστούν[44].

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

6 Διαθέσιμες

«Multiconference»εφαρμογές

6.1 Skype

Το Skype είναι μια εφαρμογή VoIP (Voice over Internet Protocol) . Το Skype μπορεί να χρησιμοποιηθεί από μια ποικιλία συσκευών όπως PC, Laptop, Tablets, smart phone με πλατφόρμες όπως Android, IOS και με λειτουργικά όπως MAC, Windows , LINUX.

Επίσης, μπορεί να το χρησιμοποιηθεί και για την αποστολή μηνυμάτων SMS(IM Instant message), για συνδιασκέψεις και αποστολή αρχείων(FTP File Transfer Protocol).

Το Skype παρέχει σύνδεση <<peer-to-peer>> χωρίς κάποιον ενδιάμεσο διακομιστή . Η εφαρμογή του Skype από το 2014 και έπειτα χρησιμοποιεί το Skype protocol της Microsoft.

Ένα από τα ελαττώματα του Skype είναι ότι χρησιμοποιεί wall garden το οποίο είναι ένα κλειστό σύστημα λογισμικού για το οποίο ο πάροχος ή ο φορέας έχει τον έλεγχο των αιτήσεων , των περιεχομένων και περιορίζει την επικοινωνία μεταξύ των άλλων μέσων.

Ένα άλλο μειονέκτημα είναι ότι οι συνδιασκέψεις με πολλά άτομα χρειάζονται διαφορετικό λογαριασμό και αυτό γίνεται μέσω κάποιας πληρωμής[45].

Χρήσιμα στοιχεία skype

- Το σήμα κρυπτογραφείται με τον αλγόριθμο RCA και τα πακέτα φωνής με AES. Επίσης για την κρυπτογράφηση των πακέτων χρησιμοποιείται και ο αλγόριθμος RC4.
- Για την σύνδεση στο δίκτυο του skype χρησιμοποιείται μια TCP σύνδεση.
- Για την ανταλλαγή πακέτων χρησιμοποιείται το UDP πρωτόκολλο.

6.2 Viber

Το Viber είναι μια εφαρμογή VoIP(Voise over IP) και χρησιμοποιεί για text message μηνύματα την instant message (IM). Το viber χρειάζεται εγκατάσταση από τον χρήστη .

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

Το Viber είναι διαθέσιμο για πλατφόρμες και λειτουργικά MAC OS, Android, Black Berry OS, iOS , Symbian 40, Bada , Windows Phone και Microsoft Windows.

Ακόμα το Viber χρησιμοποιεί την τεχνολογία SPIRIT DSP η οποία παρέχει καλύτερη μετάδοση ήχου και εικόνας.

Ένα μειονέκτημα του Viber είναι ότι ενώ παρέχει δυνατότητα επικοινωνίας δύο χρηστών μεταξύ τους δεν παρέχει την δυνατότητα για conference.

Ένα πλεονέκτημα είναι ότι η εφαρμογή είναι δωρεάν. Ακόμα το Viber μπορεί να στείλει την τοποθεσία που βρίσκεται κάποιος, μέσω GPS.

Τέλος, το Viber συγχρονίζεται με τις επαφές του κινητού τηλεφώνου του χρήστη (αριθμό) και άμα έχει η επαφή του χρήστη εγγραφεί στο Viber εμφανίζεται και το account του στο Viber[46].

6.3 ooVoo

Το ooVoo είναι και αυτό μια εφαρμογή που δουλεύει over IP . Το ooVoo μοιάζει πολύ με το Microsoft skype.

Το ooVoo επιτρέπει στους εγγεγραμμένους χρήστες να επικοινωνούν μέσω δωρεάν άμεσων μηνυμάτων, φωνή και video chat.

Το ooVoo conferencing επιτρέπει υψηλής ποιότητας βίντεο και ήχου κλήσεις με έως δώδεκα θέσεις ταυτόχρονα.

Επίσης μπορεί κάποιος να κάνει δωρεάν share την εικόνα του desktop, ακόμα να κάνει call recording.

Το ooVoo με την απελευθέρωση του έγινε SDK (Software Development Kit) στις 22 Ιουλίου του 2013 και είναι διαθέσιμο δωρεάν.

Τέλος, το ooVoo δεν διαθέτει πρωτόκολλο κρυπτογράφησης και είναι ευάλωτο σε επιθέσεις, επίσης δεν χρησιμοποιείται τόσο πολύ όπως οι άλλες εφαρμογές[47].

6.4 FaceTime

Το FaceTime είναι μια εφαρμογή που υποστηρίζει Video και Audio επικοινωνία (videotelephony), δουλεύει και το face time over Ip[48].

Το facetime λειτουργεί μόνο σε κινητές συσκευές που έχουν λειτουργικό iOS και σε υπολογιστές με λειτουργικό Mac OS X 10.6.6 και μετά.

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

Η έκδοση του βίντεο FaceTime υποστηρίζεται σε οποιαδήποτε συσκευή iOS με μια φωτογραφική μηχανή που κοιτάει μπροστά και σε κάθε Macintosh υπολογιστή εξοπλισμένο με μια κάμερα FaceTime , παλαιότερα γνωστή ως iSight κάμερα.

Η έκδοση του FaceTime ήχου είναι διαθέσιμη σε οποιαδήποτε συσκευή iOS, οι οποίες υποστηρίζουν το iOS 7 και τις μεταγενέστερες εκδόσεις και κάθε Macintosh με μέτωπο προς τα εμπρός κάμερα τρέχει Mac OS X 10.9.2 ή νεότερη έκδοση.

Το facetime είναι δωρεάν, χρειάζεται εγκατάσταση (i phone και MAC PC). Μπορεί κάποιος να επικοινωνήσει με ένα χρήστη Point-to-Point σύνδεση.

Πρωτόκολλα

- H.264 και AAC -ELD - video και audio codecs , αντίστοιχα.
- SIP - IETF signaling protocol for VoIP .
- STUN , TURN και ICE - IETF τεχνολογίες για την διάσχιση firewalls και NAT .
- RTP και SRTP - IETF πρότυπα για την παροχή σε πραγματικό χρόνο και κρυπτογραφημένες ροές πολυμέσων για VoIP.

6.5 Clear Sea

Το ClearSea δεν χρησιμοποιείται ευρέως γιατί δεν είναι δωρεάν εφαρμογή. Για να χρησιμοποιήσει κάποιος αυτή την εφαρμογή πρέπει να αγοράσει license από την εταιρία Life Size. Το ClearSea είναι εφαρμογή VoIP . Έχει την δυνατότητα για text message και επικοινωνία με Video και Audio. Ακόμα επιτρέπει video conference με άλλους χρήστες και μπορεί να καταγράψει την εικόνα και τον ήχο.

Το ClearSea είναι κλειστού κώδικα(wall garden) και είναι client-to-server .

Είναι συμβατό για λειτουργικά Microsoft, MAC OS και για πλατφόρμες iOS και Android.

Τέλος το ClearSea όπως και οι προηγούμενες εφαρμογές , χρειάζονται εγκατάσταση.

Clear Sea πρωτόκολλα

Κωδικοποίηση: Audio: G.711A,G.711μ,G.722.1,G.722.1/C

Video:H.263,H.264

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

7. Πιλοτική εφαρμογή



Ref.foto(<http://blog.mgechev.com/2014/12/26/multi-user-video-conference-webrtc-angularjs-yeoman/>)

7.1 Υλοποίηση

Αρχικά αξίζει να σημειωθεί ότι ολόκληρη η εφαρμογή λαμβάνει χώρα σ' ένα domain της alien dragon innovative creations. Το url της εφαρμογής είναι aliendragoninnovativecreations.com/cmak/grayscale/grayscale.php

Η εφαρμογή είναι προσβάσιμη σε όλους μέσω web.

Για την εφαρμογή δεν χρειάζεται να γίνει καμία εγκατάσταση webserver διότι η διαχείριση γίνεται αυτόματα από τον server που είναι η εφαρμογή, επίσης δεν χρειάζεται η εγκατάσταση κάποιου άλλου plug-in.

Η εφαρμογή υλοποιεί τρεις κύριες λειτουργίες Real time connection, Text messaging, File sharing και δύο λειτουργίες για διαφορετικό τρόπο σύνδεσης many-to-many και one-to many. Επίσης κατά την διάρκεια της σύνδεσης καθίσταται δυνατόν να εκτελούνται και οι ακόλουθες λειτουργίες recording, mute audio, mute video, full-screen and change volume. Τέλος η εφαρμογή είναι μέσα σε ένα template theme του Bootstrap και για την εν λόγω εφαρμογή χρησιμοποιούνται τα APIs του WebRTC.

Κύριες λειτουργίες:

1. Real time connection, καθίσταται δυνατή η επικοινωνία με έναν ή περισσότερους ενδιαφερόμενους σε πραγματικό χρόνο.
2. Text messaging, καθίσταται δυνατή η ανταλλαγή μηνυμάτων σε πραγματικό χρόνο με έναν ή περισσότερους ενδιαφερομένους.
3. File sharing, ανταλλαγή αρχείων μεταξύ των ενδιαφερόμενων μέσα σε μια σύνδεση.

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

Τρόπος σύνδεσης:

1. Many-to-many: Δημιουργείται μια σύνδεση και καθίσταται δυνατή η επικοινωνία πολλών με πολλούς.
2. One-to-many: Δημιουργείται μια σύνδεση και καθίσταται δυνατή η επικοινωνία ενός με πολλούς.

Real time connection

Το real time connection της εφαρμογής είναι το πιο κύριο τμήμα της πιλοτικής εφαρμογής της παρούσας ερευνητικής εργασίας και του WebRTC. Για την εφαρμογή μας πρέπει να δημιουργηθεί μια σύνδεση για να καταστεί εφικτή η επικοινωνία.

```
<script src="//cdn.webrtc-experiment.com/RTCMultiConnection.js">
```

Για αρχή χρησιμοποιούμε το javascript codes του WebRTC <<RTCMultiConnection.js>> το οποίο είναι μια βιβλιοθήκη η οποία τρέχει πάνω από RTCPeerConnection APIs και είναι απαραίτητο για την παροχή multi-session σεναρίων. Επίσης είναι απαραίτητο και για την εργασία μας για να δημιουργηθεί η σύνδεση μεταξύ των browser και την κωδικοποίηση των πακέτων που ανταλλάσσονται μεταξύ των ενδιαφερόμενων. Είναι μια βιβλιοθήκη η οποία καθίσταται δυνατόν να χρησιμοποιηθεί σε οποιαδήποτε RTCWeb APIs και ταιριάζει σε οποιοδήποτε περιβάλλον. Ακόμα καθίσταται δυνατόν να χρησιμοποιηθεί σε οποιοδήποτε PHP project, σε οποιοδήποτε ASP.net project και σε οποιοδήποτε Rubu. Τέλος στη συγκεκριμένη library δημιουργείται μια μέθοδος η "openSignalingChannel" η οποία είναι μια απλή μέθοδος που είναι δυνατόν να μπει σε οποιοδήποτε HTML σελίδα. Η μέθοδος αυτή χρησιμοποιείται για να τοποθετήσει όλα τα πράγματα σηματοδότησης σε ενιαίο χώρο. Όποιος και αν είναι ο μηχανισμός σηματοδότησης, είτε αν είναι XMPP ή SIP ή ακόμα και για υπηρεσίες 3rd party protocol Pusher / firebase / PubNub κλπ[27].

Επίσης η Web εφαρμογή μας χρησιμοποιεί και άλλη βιβλιοθήκη του WebRTC την <<getMediaElement.min.js>> η οποία ενεργοποιείται όταν έχει δημιουργηθεί η σύνδεση μέσω του «openSignalingChannel». Το «getMediaElement.min.js» ενεργοποιεί στην HTML και CSS το video και το audio για την επικοινωνία μέσω streaming. Έχει προσαρμοσμένο Flash και Silverlight player που χρησιμοποιούνται στην HTML5 MediaElement API για παλαιότερα προγράμματα περιήγησης.

```
<script src="//cdn.webrtc-experiment.com/getMediaElement.min.js">
```

Η λειτουργία του είναι αντί να προσφέρει ένα πρόγραμμα αναπαραγωγής HTML5 σε σύγχρονα προγράμματα περιήγησης και ένα εντελώς ξεχωριστό Flash player, όπως

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

ίσχυε για τα παλαιότερα προγράμματα περιήγησης, η MediaElement.js αναβαθμίζεται με το custom Flash και το Silverlight plugins που χρησιμοποιούνται στην HTML5 MediaElement API.

Οι Browsers χρησιμοποιούν για την κωδικοποίηση τα εξής:Firefox, Safari και Opera τα οποία υποστηρίζουν H264 για <video> και <audio>,και ο IE9 υποστηρίζει WebM. Στο μέλλον το adobe θα δημιουργήσει την υποστήριξη του VP8.

Browser και Device που το υποστηρίζουν:

	 IE9+	 IE6-8	 Firefox	 Opera	 Safari	 Chrome	 iOS	 Android	 WP7
h.264 only									Native
WebM only	 1	Flash11 ²			Flash11 ²		-	-	-
WebM + h.264									Native
WMV							-	-	Native
FLV							-		-
MP3								Native	Native
YouTube									

Ref(<http://mediaelementjs.com/>)

Τέλος στην εφαρμογή μας γίνεται χρήση μιας ακόμα βιβλιοθήκης του WebRTC «firebase.js». Το Firebase είναι μια σχετικά καινούργια τεχνολογία δεδομένου ότι ξεκίνησε να λειτουργεί τον Απρίλιο του 2012, Το firebase προσφέρει μια βάση δεδομένων σε πραγματικό χρόνο που παρέχει ένα API το οποίο επιτρέπει στους προγραμματιστές να αποθηκεύουν τα δεδομένα συγχρονισμού σε πολλούς πελάτες.

```
<script src="//cdn.webrtc-experiment.com/firebase.js">
```

Η υπηρεσία παρέχει στους application developers ένα API που επιτρέπει στα δεδομένα της εφαρμογής να είναι συγχρονισμένα μεταξύ των πελατών και να αποθηκεύονται στο cloud firebase cloud. [49] [50] Η εταιρεία παρέχει library clients που επιτρέπουν την ενσωμάτωση με το Android, iOS, το JavaScript, Java, Objective-C και του Node.js applications.. Η βάση δεδομένων είναι επίσης προσβάσιμη μέσω ενός REST API και συνδέσεων για αρκετά JavaScript frameworks, όπως AngularJS, React, Ember.js και Backbone.js. Το REST API χρησιμοποιεί το Server-Sent Events protocol, το οποίο είναι ένα API για τη δημιουργία μιας HTTP σύνδεσης για να λαμβάνει push notifications από ένα server.

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

Στην εφαρμογή μας το firebase το χρησιμοποιούμε στο text message για την ανταλλαγή μηνυμάτων μεταξύ των συμμετεχόντων και για το file sharing.

Translate and ASR

Η καινοτομία της εφαρμογής μας βρίσκεται σε αυτό το τμήμα με την βοήθεια του Google Translate και του WebRTC. Στην εφαρμογή μας ένας χρήστης έχει την δυνατότητα ενώ μιλάει να καταγράφεται σε ένα «inbox» η σχετική ομιλία του, ακόμα παρέχεται η δυνατότητα μετάφρασης από αγγλικά σε ελληνικά. Τεχνικά αυτό επιτυγχάνεται, καλώντας στον κώδικα μας ένα API από το Google library τον Translator. Φυσικά όπως για όλα τα API έτσι και σ' αυτό για να καθίσταται δυνατή η χρήση όλων των λειτουργιών του χρειάζεται ένα κλειδί[51] ApiKey=

```
api_key: 'AIzaSyAKLM48_aFj-ktsgdXMQTmTHaeqtKog5bo', // use your own key
```

το οποίο είναι μοναδικό για κάθε εφαρμογή.

Ακόμα, για να μπορέσουμε να χρησιμοποιήσουμε τον «Translator»στην εφαρμογή μας θα πρέπει να χρησιμοποιήσουμε ένα Javascripte,

```
<script src="//cdn.webrtc-experiment.com/Translator.js"> </script>
```

στο συγκεκριμένο script καλεί το API της Google Translate. Ουσιαστικά παίρνει μια μεταβλητή και μας δίνει μια άλλη μεταβλητή στον κώδικα μας. Όπως και τα προηγούμενα script έτσι και αυτό τρέχει στην αρχή της εφαρμογής

Για την συγκεκριμένη σύνθεση του WebRTC και του Google Translate παρατηρούνται ορισμένα αρνητικά :

- 1^ο η εφαρμογή μας με την συγκεκριμένη λειτουργία ενεργή, δουλεύει μόνο σε συγκεκριμένο Browser τον Chrome(λόγω Google, γιατί και οι δύο τεχνολογίες είναι δικές της),
- 2^ο παρουσιάζει ορισμένα προβλήματα στην μετάφραση,
- 3^ο ακόμα παρουσιάζει και κάποια προβλήματα στις τοπικές συσκευές του μικροφώνου, επειδή γίνεται ερώτηση και στον Translator για την αποδοχή του μικροφώνου, και τέλος
- 4^ο πρόβλημα είναι ότι γίνεται μετάφραση μόνο από Αγγλικά σε Ελληνικά.

Το συγκεκριμένο project έστω και πιλοτικά, δεν υπάρχει πουθενά κάπου επίσημα. Δηλαδή δεν έχει εντοπισθεί καταγεγραμμένη εφαρμογή στην οποία να είναι εφικτή η

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

σύνδεση σε πραγματικό χρόνο και να υπάρχει γραπτή καταγραφή της ομιλίας η οποία ταυτόχρονα να εμφανίζεται και στον χρήστη, μέσω Web application χωρίς κάποιο Plug-in. Αναφορικά στο εμπόριο θα αρχίσει να κυκλοφορεί κάτι παρόμοιο από το Skype, το οποίο έχει ως προϋπόθεση για να λειτουργεί σωστά η νέα λειτουργία του Skype, ο χρήστης να έχει εγκατεστημένο λειτουργικό σύστημα Windows 8.1 ή Windows 10 Technical Preview. Ενώ η δική μας εφαρμογή χρειάζεται κυρίως Google Chrome.

Buttons-Δημιουργίας

Στην εφαρμογή για να δημιουργηθεί από κάποιον ένα room προτείνεται να εισαχθεί ένα user name και ένα name room τα οποία βοηθάνε τους ενδιαφερόμενους για λόγους πληροφόρησης μέσα στο room.

```
<input type="text" id="session-name" color: #ooffoo;>
```

```
<input type="text" id="moderator-name" color: #ooffoo;>
```

Η ανωτέρω εικόνα μας δείχνει δυο input της HTML5 στα οποία ο χρήστης μπορεί να εισάγει δυο τιμές τις οποίες θα χρησιμοποιήσουμε για το room name και για το όνομα του user-moderator(δημιουργός του room).

Έναρξη του session room

Στην εφαρμογή για να δημιουργήσεις ένα room, πρέπει να πατήσεις ένα <button>

```
<button id="setup-new-session" class="setup">ΣΥΝΔΕΣΗ </button>
```

Η ανωτέρω εικόνα μας δείχνει ένα tag <button> με id= setup-new-session το οποίο ενεργοποιεί την function για να ξεκινήσει μια καινούργια σύνδεση.

```
var setupNewSession = document.getElementById('setup-new-session');  
setupNewSession.onclick = function() {
```

Η συγκεκριμένη εντολή σε συνδυασμό με το library RTCMultiConnection.js του WebRTC , ενεργοποιούν ένα νέο room.

Τέλος στην εφαρμογή μας αυτός που δημιουργεί το conference room λέγεται moderator, και στην εφαρμογή μας απεικονίζετε ως εξής:

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

```
connection.session = session;

connection.maxParticipantsAllowed = maxParticipantsAllowed;

connection.sessionid = sessionName || 'Anonymous'; //χωρίς ονομά

var ownerName = document.getElementById('moderator-name').value;

if(ownerName==""){

    connection.userid="Moderator";

}
```

Επιλογή επικοινωνίας της Σύνδεσης

Στην εφαρμογή μας δίνουμε την δυνατότητα στον δημιουργό του room να επιλέξει πόσοι ενδιαφερόμενοι θα συνδεθούν και με ποιο τρόπο θα επικοινωνούν οι συμμετέχοντες.

Τρόπος Σύνδεσης

Οι δυνατότητες επιλογών σύνδεσης του κώδικά μας απεικονίζονται κάτωθι .

```
<select id="session" title="Type">

    <option>audio+video+data</option>

    <option>audio+data</option>

</select>
```

Στην εφαρμογή μας default είναι όλες οι επιλογές ενεργοποιημένες video, audio και data. Στην περίπτωση που ο δημιουργός επιθυμεί να πραγματοποιήσει σύνδεση χωρίς video μπορεί να απενεργοποιήσει μόνο το video.

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

```
var connection = new RTCMultiConnection();  
  
    connection.session = {  
  
        audio: true,  
  
        video: true,  
  
        data:true  
  
    };
```

Έτσι στην συνέχεια απεικονίζεται η επιλογή του δημιουργού σύνδεσης με απενεργοποίηση του video.

```
var connection = new RTCMultiConnection();  
  
    connection.session = {  
  
        audio: true,  
  
        video: false,  
  
        data:true  
  
    };
```

Αριθμός Εμφάνισης Χρηστών

Ο τρόπος σύνδεσης μπορεί να γίνει με δύο διαφορετικούς τρόπους. Αυτό καθορίζεται από τον αριθμό των συμμετεχόντων. Ο πρώτος τρόπος είναι «many-to-many»(πολλοί με πολλούς), και ο δεύτερος «one-to-many» ένας με πολλούς.

Many-to-many

Ο many-to-many είναι ο πιο σημαντικός τρόπος για την εφαρμογή μας γιατί είναι και ο σκοπός της. Αυτός ο τρόπος είναι MULTIPOINT CONNECTION, δημιουργείται ένα conference room και σε αυτό μπορούν να συνδεθούν ένας μεγάλος αριθμός ανθρώπων $N < 10$.

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

```
for (var i = 0; i < splittedSession.length; i++) {  
    session[splittedSession[i]] = true;  
}  
  
var maxParticipantsAllowed = 256; //default  
  
if (direction == 'one-to-many') session.broadcast = true;  
  
var sessionName = document.getElementById('session-name').value;  
  
connection.extra = {  
    'session-name': sessionName || 'Anonymous'  
};
```

Στην ανωτέρω εικόνα βλέπουμε μια μεταβλητή «maxParticipantsAllowed» η οποία έχει ορισθεί από την βιβλιοθήκη του WebRTC μέσα στο RTCMultiConnection.js για multipoint connection. Στην εφαρμογή μας είναι προκαθορισμένο, και μόνο αν επιλέξει κανείς τις άλλες επιλογές θα αλλάξει.

One-to-many

```
for (var i = 0; i < splittedSession.length; i++) {  
    session[splittedSession[i]] = true;  
}  
  
var maxParticipantsAllowed = 256; //default  
  
if (direction == 'one-to-many') session.broadcast = true;  
  
var sessionName = document.getElementById('session-name').value;  
  
connection.extra = {  
    'session-name': sessionName || 'Anonymous'  
};
```

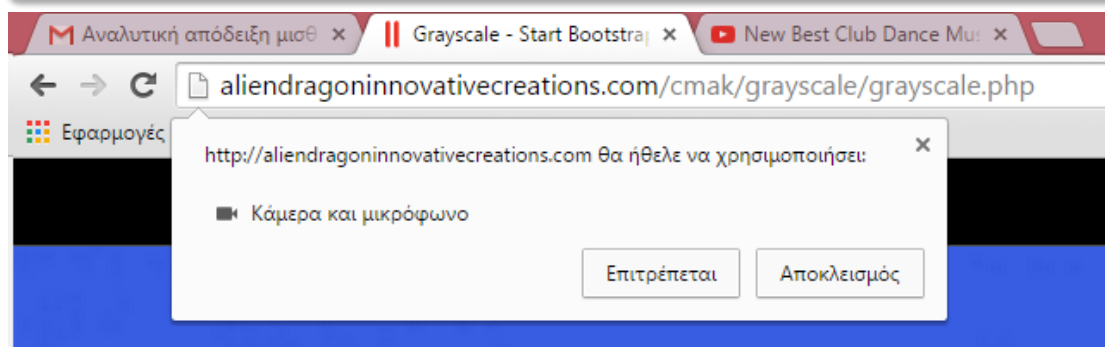
"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

Στην πιο πάνω εικόνα φαίνεται ότι αλλάζει η μεταβλητή και γίνεται «session.broadcast» Αυτό σημαίνει ότι κάνει streaming το session που έχει δημιουργηθεί προς όλους τους ενδιαφερόμενους. Η διαφορά με την πρώτη επιλογή είναι ότι κάνει μετάδοση video και audio μόνο ο δημιουργός του σε όλους τους συμμετέχοντες. Ενώ ο δημιουργός μπορεί να τους δει όλους, οι άλλοι συμμετέχοντες δεν μπορούν να επικοινωνήσουν μεταξύ τους..

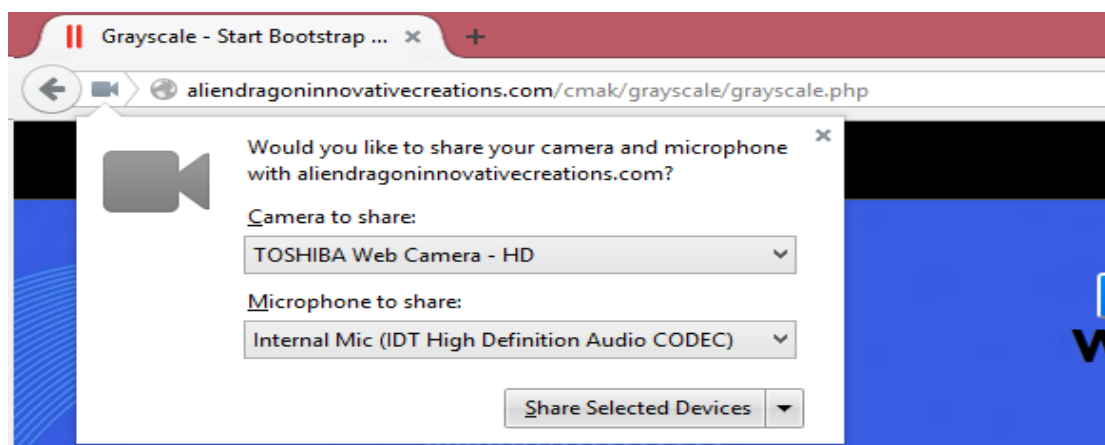
Session Room

Σε αυτό το σημείο έχουμε ήδη δημιουργήσει το room. Όταν θα ξεκινήσει η σύνδεση, ο Web Server στέλνει ένα μήνυμα στο JavaScript που τρέχει στον browser του πρώτου συμμετέχοντα ώστε να δημιουργήσει μια peer σύνδεση με δύο κανάλια πολυμέσων. Ένα για τον ήχο και ένα για το βίντεο. Στη συνέχεια, ο browser του πρώτου συμμετέχοντα και δημιουργού ζητά την άδεια του χρήστη για να έχει η εφαρμογή πρόσβαση στην κάμερα και στο μικρόφωνό του. Αυτό συμβαίνει με ένα αναδυόμενο παράθυρο που εμφανίζεται κάτω από το URL. Η εικόνα 1 απεικονίζει το αναδυόμενο παράθυρο του Chrome ενώ η 2 του Firefox.

```
navigator.getMedia({audio: true, video: true}, successCallback, errorCallback);
```



:chrome



:firefox

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

Ζητώντας την άδεια του χρήστη, επιτυγχάνονται δύο πράγματα. Πρώτον προσδιορίζει ποια ιστοσελίδα ζητάει την άδεια για χρήση των τοπικών συσκευών και δεύτερον δίνεται η επιλογή στο χρήστη να επιτρέψει ή όχι την πρόσβαση στην κάμερα και στο μικρόφωνό του. Άρα ο χρήστης έχει τον πλήρη έλεγχο.

Εάν ο χρήστης αποδεχτεί το αίτημα, στην οθόνη του εμφανίζεται το βίντεο από την τοπική κάμερα, μέσα στο html video element.

```
sourcevid.src = window.URL.createObjectURL(stream);
```

Εκκίνηση τοπικού video

Μετά την αποδοχή, η οποία θα μας δώσει πρόσβαση στην camera και στο μικρόφωνο αναλόγως με τον τρόπο που θα συνδεθούμε (με video ή χωρίς) ανοίγουν οι τοπικές συσκευές. Για την υλοποίησή αυτής της πρόσβασης γίνεται χρήση του MediaStream API και του PeerConnection API του WebRTC.

```
var connection = new RTCMultiConnection();  
  
connection.session = {  
  
    audio: true,  
  
    video: true  
  
};
```

Στη συνέχεια είναι απαραίτητο να οριστεί το navigator.getMedia που χρησιμοποιεί το MediaStream API για να αποκτήσει πρόσβαση στην κάμερα ή στο μικρόφωνο. Ορίζεται, για να μπορέσει να λειτουργήσει η εφαρμογή με όλους τους browser που υποστηρίζουν το WebRTC.

```
var navigator = window.navigator;  
  
navigator.getMedia = ( navigator.getUserMedia || navigator.webkitGetUserMedia  
||  
navigator.mozGetUserMedia || navigator.msGetUserMedia);
```

Έπειτα, για την εγκαθίδρυση της σύνδεσης μεταξύ των δύο peers μέσω Websockets απαιτείται ένας WebSocket server. Στην περίπτωση μας, ο WebSocket server είναι ο

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

Server που είναι επάνω η εφαρμογή μας (foodtorch.com-185.4.135.96) και δεν χρησιμοποιήσαμε κάποιο τοπικό π.χ. node.js.

Ας σημειωθεί ακόμη ότι χρησιμοποιείται και ο STUN server της Google, όπου έχει διατεθεί ελεύθερα για χρήση από τους προγραμματιστές.

```
var stunServer = "stun.l.google.com:19302";
```

Σύνδεση

Button

Όταν ξεκινήσει η σύνδεση εμφανίζονται δύο κουμπιά τα οποία το καθ' ένα εκτελεί και την δική του συνάρτηση. Τα κουμπιά που εμφανίζονται είναι τα εξής «leave» και «reload page».

Leave

```
<input type="button" value="Close" id="close">
```

Η ανωτέρω εντολή είναι ένα button στην HTML 5 για να εμφανίζεται στον χρήστη μέσω της οποίας καλείται η πιο κάτω function.

```
document.querySelector('#close').onclick = function() {  
    connection.close();  
};
```

Το οποίο δίνει την δυνατότητα στο χρήστη να διακόψει την σύνδεση και να εξέλθει από το room.

Reload Page

```
<input type="button" value="Reload Page" onClick="history.go(0)">
```

Με αυτήν την επιλογή ο χρήστης μπορεί να κάνει ανανέωση της σελίδας και να ξανά τρέξει από την αρχή. Αυτή η εντολή εισήχθη για ευκολία των χρηστών στην περίπτωση που χρειάζεται νέα σύνδεση ή στην περίπτωση που υπάρχει κάποιο πρόβλημα στην ήδη υπάρχουσα.

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

Video

Frame

Στην εφαρμογή μας δεν ορίζουμε κάπου κάποιο tag «input type=video» για να ανοίξει η τοπική camera. Στην εφαρμογή μας ορίζουμε ένα «div»

```
<div id="videos-container"></div>
```

Το οποίο έχει ένα id= videos-container που το καλούμε στο JavaScript για να δημιουργήσουμε το frame για το video και της διαστάσης.

```
var mediaElement = getMediaElement(e.mediaElement, {  
    width: (videosContainer.clientWidth / 2) - 50,
```

Buttons

Μέσα στο frame έχουμε βάλει να εμφανίζονται ορισμένα buttons. Τα οποία εκτελούν κάποιες λειτουργίες .

1^ο Mute audio και Mute video

```
onMuted: function(type) {  
    connection.streams[e.streamid].mute({  
        audio: type == 'audio',  
        video: type == 'video'  
    });
```

2^ο Unmute audio και Unmute Video

Αντίστοιχα τα κουμπιά που περιγράψαμε πιο πάνω έχουν και τις επιλογές να επαναφέρουν πάλι τον ήχο και την εικόνα.

```
onUnMuted: function(type) {  
    connection.streams[e.streamid].unmute({  
        audio: type == 'audio',  
        video: type == 'video'  
    });
```

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

3^ο RecordingStarted

Όπως και για τον ήχο και την εικόνα είχαμε δύο διαφορετικά κουμπιά έτσι έχουμε και για την καταγραφή. Η εφαρμογή μας δίνει την επιλογή στον χρήστη να κάνει καταγραφή της εικόνας ή του ήχου. Με τα συγκεκριμένα κουμπιά ξεκινάει η καταγραφή.

```
onRecordingStarted: function(type) {  
  
    // www.RTCMultiConnection.org/docs/startRecording/  
  
    connection.streams[e.streamid].startRecording({  
  
        audio: type == 'audio',  
  
        video: type == 'video'
```

4^ο RecordingStopped

Αντίστοιχα τα ίδια κουμπιά σταματάνε την καταγραφή. Εδώ όμως όταν πατήσουμε το κουμπί αυτόματα αποθηκεύεται ένα αρχείο με κωδικοποίηση WAV το οποίο περιέχει το αρχείο που καταγράψαμε. Εδώ να προσθέσουμε ότι γίνεται η χρήση του firebase που αναφέραμε πιο πάνω, αν σταματήσει η σύνδεση το αρχείο χάνεται.

```
onRecordingStopped: function(type) {  
  
    // www.RTCMultiConnection.org/docs/stopRecording/  
  
    connection.streams[e.streamid].stopRecording(function(blob) {  
  
        if (blob.audio) connection.saveToDisk(blob.audio);  
  
        else if (blob.video) connection.saveToDisk(blob.video);  
  
        else connection.saveToDisk(blob);  
  
    }, type);
```

5^ο Full-Screen και Volume-Slider

Με τα συγκεκριμένα κουμπιά δίνουμε την επιλογή στον χρήστη να μπορεί να διευρύνει την εικόνα σε full-screen(σε ολόκληρη την οθόνη) και να μπορεί να ρυθμίσει τον ήχο μέσα από την εφαρμογή. Όταν πατήσουμε τα συγκεκριμένα κουμπιά ενεργοποιείται το `getMediaElement.min.js` και τα `css` της εφαρμογής. Εδώ δεν μπορούμε να βάλουμε

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

κάποια εικόνα του κώδικα διότι το συγκεκριμένο js που εκτελείται έχει μια λειτουργία (function) η οποία είναι αρκετά μεγάλη και άσκοπο για να αναλυθεί.

Textarea-translate «Dialogs»

Το επόμενο τμήμα είναι η καινοτομία της εφαρμογής μας η οποία ωστόσο είναι ακόμα σε πειραματικό στάδιο. Στον HTML κώδικα μας εμφανίζεται μια «textarea».

```
<textarea id="text-translator" rows="5" style="font-size: 1.2em; width:100%;"
placeholder="dialogs" disabled ></textarea>
```

Η οποία παίρνει τιμές από τον javascript κωδικα με το ID= text-translator

```
translator.translateLanguage(text, {
    from: 'en',
    to: 'el',
    api_key: 'AIzaSyAKLM48_aFj-ktsgdXMQTmTHaeqtKog5bo', // use your own
    key
    callback: function (translatedText) {

        var div=document.getElementById('text-translator');
        div.value+='me:'+translatedText+"\n";
        connection.send("$"+ connection.userid+": "+translatedText);

        console.log('translated text', translatedText);
    }
});
```

Ο παραπάνω κώδικας είναι για τον δημιουργό (moderator) και αντίστοιχα έχουμε και για τον συμμετέχοντα. Η διαφορά μέσα στον κώδικα μας είναι ότι το ένα task, του δημιουργού μπαίνει στην function setupNewSession ενώ του συμμετέχοντα μπαίνει στην function join την οποία θα αναλύσουμε πιο κάτω.

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

Textarea- «chat-output» και input- «chat-input» « Text Chat»

Το επόμενο box που βλέπουμε είναι και αυτό μια «text area» η οποία εμφανίζει τιμές του «text message» , δηλαδή μπορούμε να επικοινωνούμε μέσω γραπτών μηνυμάτων με τους συμμετέχοντες.

```
<textarea id="chat-output" rows="5" style="font-size: 1.2em; width:100%;"
placeholder="chat" disabled ></textarea>
```

Το πιο πάνω tag της HTML παίρνει τιμές από τον javascript κωδικά μας με το ID=="chat-output".

```
var chatOutput = document.getElementById('chat-output')
```

Η πιο πάνω εντολή είναι σε συνδυασμό με το «chat-input» το οποίο είναι ένα tag της HTML που παίρνει τιμές από τον χρήστη και τις εμφανίζει στο «chat-output»

```
<input type="text" id="chat-input" style="font-size: 1.2em;" placeholder="chat
message" disabled>
```

Εδώ ο χρήστης δίνει κάποιες τιμές, οι οποίες στον κώδικα μας αποθηκεύονται ως ένα «string». Οι τιμές αυτές είναι τα μηνύματα που θέλουμε να στείλουμε στον άλλο συμμετέχοντα και τα οποία και θα εμφανιστούν στο «chat-output». Στον javascript κώδικα είναι λίγο πιο πολύπλοκο διότι δέχεται τιμές και από το «chat-translator»

```
connection.onmessage = function(e) {
    appendDIV(e.data);
```

Η συγκεκριμένη εντολή ενεργοποιεί την «onmessage» από το javascript «RTCMultiConnection.js», η οποία παίρνει τις τιμές και τις βάζει σε ένα DIV στον κώδικα μας.

```
chatInput.onkeypress = function(e) {
    if (e.keyCode !== 13 || !this.value) return;
    appendDIV("@me: " +this.value);
    // sending text message
    connection.send("@"+ connection.userid+": "+this.value);
    this.value = "";
```

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

Στην συνέχεια η ανωτέρω λειτουργία (function) ενεργοποιείται όταν «πατήσουμε το «enter= onkeypress» τότε παίρνει την τιμή και είναι έτοιμη να την στείλει. Πριν την στείλει γίνεται ένας έλεγχος.

```
function appendDIV(div, parent) {  
    if (typeof div === 'string') {  
        var pos = div.charAt(0);  
        if (pos == '@'){  
            div = div.substr(1);  
            var content = div;  
            div = document.getElementById('chat-output');  
            div.innerHTML += content+"\n";  
        }  
        else if ( pos == '$'){  
            div = div.substr(1);  
            var content = div;  
  
            div=document.getElementById('text-translator');  
            div.value+=content+"\n";  
        }  
    }  
}
```

Ο συγκεκριμένος έλεγχος καταλαβαίνει τον πρώτο χαρακτήρα από το «string» που θα πάρουμε από τον χρήστη ή από το «text-input» ή από «text-translate» και το εμφανίζει στο κατάλληλο ID.

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

Input «file» « Text Chat»

Το επόμενο και τελευταίο κομμάτι μέσα στην σύνδεση που θα αναλύσουμε είναι το «file-sharing». Εδώ δίνουμε την δυνατότητα στον χρήστη να μπορεί να στείλει κάποια αρχεία στους άλλους συμμετέχοντες.

```
<input type="file" id="file" disabled>
```

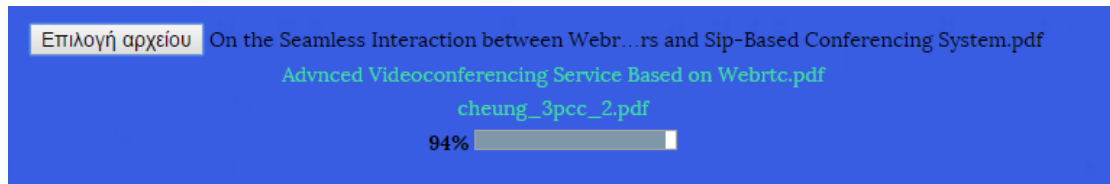
Το παραπάνω «tag» είναι ένα input το οποίο έχει type=file. Αυτό σημαίνει ότι δημιουργείται ένα κουμπί με το οποίο δύναται κανείς να κάνει αναζήτηση σε τοπικά αρχεία στον υπολογιστή του.

```
connection.onFileStart = function(file) {  
  
    var div = document.createElement('div');  
  
    div.title = file.name;  
  
    div.innerHTML = '<label>ο%</label> <progress></progress>';  
  
    appendDIV(div, fileProgress);  
  
    progressHelper[file.uuid] = {  
  
        div: div,  
  
        progress: div.querySelector('progress'),  
  
        label: div.querySelector('label')  
  
    };  
  
    progressHelper[file.uuid].progress.max = file.maxChunks;
```

Στην παραπάνω εικόνα φαίνεται η διαδικασία που γίνεται για να ανέβει το αρχείο, που έχει επιλέξει ο χρήστης. Εδώ γίνεται η χρήση του «firebase» το οποίο καλείται μέσω του «RTCMultiConnection.js», δημιουργείται ένας εικονικός χώρος στον οποίο

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

μπορούμε να ανεβάσουμε τα αρχεία μας.



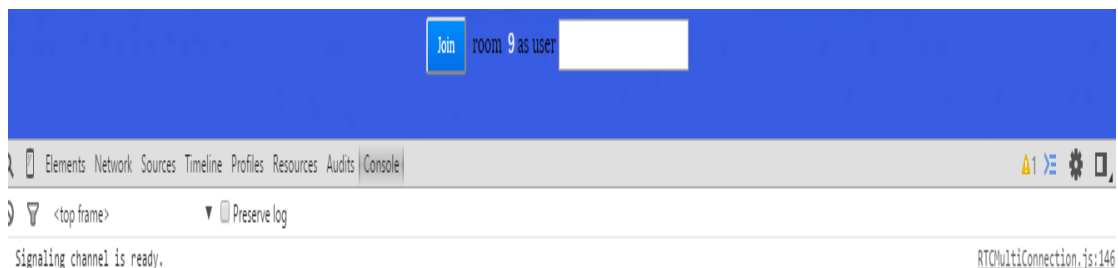
Στην ανωτέρω εικόνα βλέπουμε την πρόοδο του αρχείου που ανεβαίνει μέσω της εντολής της HTML 5 `<label></label> <progress></progress>`.

```
connection.onFileEnd = function(file) {  
  
    progressHelper[file.uuid].div.innerHTML = '<a href="' + file.url + "'  
    target="_blank" download="' + file.name + "'>' + file.name + '</a>';  
  
};
```

«JOIN»

Ένα από τα κυριότερα τμήματα της εφαρμογής μας είναι το «join». Είναι σημαντικό διότι με αυτήν την επιλογή ο συμμετέχοντας θα μπορέσει να συνδεθεί στο session.

Όταν δημιουργηθεί ένα session από τον moderator, και ο συμμετέχοντας θα θελήσει να συνδεθεί τότε θα του εμφανισθεί στην οθόνη σύνδεσης ένα κουμπί.



Στην ανωτέρω εικόνα βλέπουμε το κουμπί σύνδεσης και πιο κάτω στον debugger του Google Chrome στην επιλογή Console, ότι έχει δημιουργηθεί και RTCDATAChannel για να εκκινήσει η σύνδεση με τον άλλο συμμετέχοντα μέσω P2P.

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

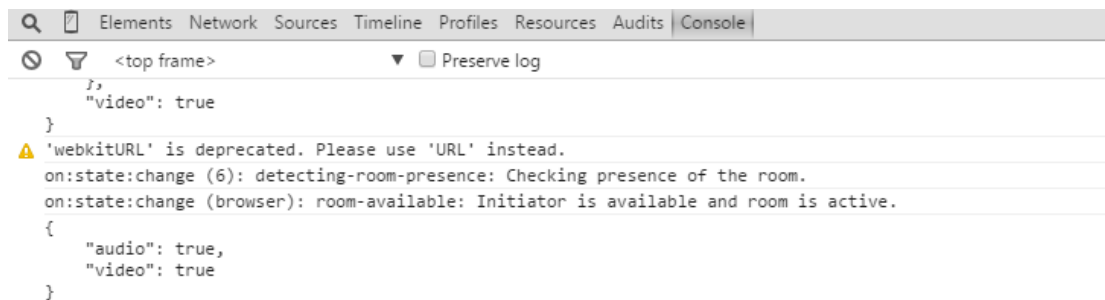
```
var roomsList = document.getElementById('rooms-list'),  
  
    sessions = {};  
  
connection.onNewSession = function(session) {  
  
    if (sessions[session.sessionid]) return;  
  
    sessions[session.sessionid] = session;  
  
    var tr = document.createElement('tr');  
  
    tr.innerHTML = '<td><button class="join">Join</button>&nbsp;  room
```

Μετά από έλεγχο που γίνεται στην αρχή της εφαρμογής και εφ'όσον έχει δημιουργηθεί κάποιο room, τότε με την δημιουργία του room και του αντίστοιχου channel εμφανίζεται ένα «button-Join» για την σύνδεση και ένα «input» για να βάλει ο συμμετέχοντας ένα όνομα.

7.2 Console log στην σύνδεση

Αφού γίνει η διαδικασία της δημιουργίας και της έναρξης σύνδεσης με τους συμμετέχοντες. Αρχίζουν κάποιοι έλεγχοι

Η πρώτη ερώτηση που γίνεται από τον Browser είναι αν είναι ενεργές οι τοπικές συσκευές video και μικρόφωνο για να ενεργοποιηθεί το room .



Έπειτα, ο συμμετέχοντας λαμβάνει την προσφορά για την αποδοχή του audio και του video, η οποία έχει δημιουργηθεί από τον δημιουργό .

```
target user has both audio/video streams.  
target user's SDP has? {  
  "OfferToReceiveAudio": true,  
  "OfferToReceiveVideo": true  
}
```

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

Πιο κάτω βλέπουμε την αποδοχή του συμμετέχοντα:

```
accepting request from 8dvicedbbihpvi
sdp-mandatory-constraints {
  "OfferToReceiveAudio": true,
  "OfferToReceiveVideo": true
}
```

Εύρεση IceServer

1. Το επόμενο βήμα είναι να δημιουργηθεί από το δημιουργό του room μια `RTCPeerConnection` μαζί με κάποιον `onicecandidate handler` η οποία επιτρέπει να γίνονται συνδέσεις μεταξύ διαφορετικών peers.
2. Ο `Onicecandidate handler` καλείται όταν οι ενδιαφερόμενοι είναι διαθέσιμοι στο δίκτυο.
3. Στέλνει ο δημιουργός τα στοιχεία τον διαθέσιμο `IceServer` στον δημιουργό μέσω ενός stringified (JavaScript Object Notation (JSON) string).
4. Τότε ο ενδιαφερόμενος παίρνει ένα μήνυμα που λέγεται `addIceCandidate()`.

```
rtc-configuration {
  "iceServers": [
    {
      "url": "stun:stun.l.google.com:19302"
    },
    {
      "url": "stun:stun.anyfirewall.com:3478"
    },
    {
      "url": "turn:turn.bistri.com:80",
      "credential": "homeo",
      "username": "homeo"
    },
    {
      "url": "turn:turn.anyfirewall.com:443?transport=tcp",
      "credential": "webrtc",
      "username": "webrtc"
    },
    {
      "url": "stun:turn1.xirsys.com"
    },
    {
      "username": "4260cd34-464e-4605-973c-27ac0d098450",
      "url": "turn:turn1.xirsys.com:443?transport=udp",
      "credential": "87df66b5-89d7-4644-b891-811cdd589cd"
    },
    {
      "username": "4260cd34-464e-4605-973c-27ac0d098450",
      "url": "turn:turn1.xirsys.com:443?transport=tcp",
      "credential": "87df66b5-89d7-4644-b891-811cdd589cd"
    }
  ],
  "iceTransports": "all"
}
```

Όπως φαίνεται και παραπάνω, οι `iceServers` είναι ένας πίνακας τύπου `RTCIceServer`, ο οποίος περιέχει όλους τους διαθέσιμους STUN και TURN servers που χρησιμοποιούνται από το ICE framework. Μπορεί να είναι πολλοί servers από κάθε τύπο, καθώς επίσης και κάθε TURN server μπορεί να παίζει το ρόλο ενός STUN server.

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

Το ICE προσπαθεί να συνδέει δύο peers απευθείας με τη μικρότερη δυνατή καθυστέρηση μεταφοράς μέσω UDP. Σε αυτή την προσπάθεια παίρνουν μέρος και δύο άλλοι servers, οι STUN και TURN.

Σε περίπτωση που η UDP σύνδεση αποτύχει, το ICE προσπαθεί αρχικά HTTP και στη συνέχεια HTTPS. Εάν η απευθείας σύνδεση αποτύχει, κυρίως λόγω των NAT και firewalls, το ICE χρησιμοποιεί έναν ενδιάμεσο (relay) TURN server. Με άλλα λόγια, το ICE πρώτα χρησιμοποιεί STUN με UDP για απευθείας σύνδεση και αν αυτή αποτύχει θα χρησιμοποιήσει τον TURN relay server. Αν και σύμφωνα με το «<http://webrtcstats.com/>» Εικόνα 9 , το 86% των κλήσεων χρησιμοποιούν STUN server.



Εικόνα 9.

Δημιουργία Datachannel

Μετά δημιουργείται ένα «createDataChannel();», για την απευθείας σύνδεση μεταξύ των peers.

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

```
invoked: createDataChannel
dataChannelDict {}
attaching stream: 7hbqv1fcjd78pvi {
  "onremovetrack": null,
  "onaddtrack": null,
  "oninactive": null,
  "onactive": null,
  "active": true,
  "ended": false,
  "id": "VI12tJ228tevM6KP1Rv1DYTyradksXqb8LoT",
  "label": "VI12tJ228tevM6KP1Rv1DYTyradksXqb8LoT",
  "streamid": "7hbqv1fcjd78pvi",
  "isScreen": false,
  "isVideo": true,
  "isAudio": false,
  "preMuted": {
    "audio": false,
    "video": false
  }
}
```

Δημιουργία SDP σύνδεσης

Ο δημιουργός του room καλεί την startIce() μέθοδο της peer σύνδεσης. Στη συνέχεια στέλνει στον B έναν υποψήφιο ICE candidate, ο οποίος περιλαμβάνει πληροφορίες όπως την IP διεύθυνση και την πόρτα της επικοινωνίας.

```
(getLocalDescription) peer createType is offer
offer sdp v=0
o=- 496511221239838042 2 IN IP4 127.0.0.1
s=-
t=0 0
a=group:BUNDLE audio video data
a=msid-semantic: WMS VI12tJ228tevM6KP1Rv1DYTyradksXqb8LoT
m=audio 9 RTP/SAVPF 111 103 104 9 0 8 126
c=IN IP4 0.0.0.0
a=rtcp:9 IN IP4 0.0.0.0
a=ice-ufrag:jymUHOEjkL2Aj9xt
a=ice-pwd:3o+wCGTyMrcQYU0XD7RiVodK
a=fingerprint:sha-256 94:D9:7E:D0:B5:CE:A8:57:3A:2D:C4:6F:42:51:73:CD:E4:DD:3A:2C:39:11:09:4C:EE:8D:2A:95:9F:39:C9:53
a=setup:actpass
a=mid:audio
a=extmap:1 urn:iETF:params:rtp-hdrext:ssrc-audio-level
a=extmap:3 http://www.webrtc.org/experiments/rtp-hdrext/abs-send-time
a=sendrecv
a=rtcp-mux
a=rtpmap:111 opus/48000/2
a=fmtp:111 minptime=10; useinbandfec=1
a=rtpmap:103 ISAC/16000
a=rtpmap:104 ISAC/32000
a=rtpmap:9 G722/8000
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:126 telephone-event/8000
a=maxptime:60
a=ssrc:3240711972 cname:599oerwS2XH1GJHk
a=ssrc:3240711972 msid:VI12tJ228tevM6KP1Rv1DYTyradksXqb8LoT 48090190-0f3d-4e54-87c3-c94764b2688a
```

Ο συμμετέχοντας λαμβάνει, επεξεργάζεται τον ICE candidate, δημιουργεί έναν νέο και

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

τον στέλνει στον A.

```
got remote streamid 5kjpj6717g9olor from 8dvicedbbihpvi
setting remote description answer v=0
o=- 3915697511219896421 2 IN IP4 127.0.0.1
s=-
t=0 0
a=group:BUNDLE audio video data
a=msid-semantic: WMS eQRMwHU1xIqLYutMH4qBTKFcXmqFb6z8tTBW
m=audio 1 RTP/SAVPF 111 103 0 8 126
c=IN IP4 0.0.0.0
a=rtcp:1 IN IP4 0.0.0.0
a=ice-ufrag:14ZkDyKgFk3L9Iw4
a=ice-pwd:8fz5+C/yQrXgAERmpTdMo+s
a=fingerprint:sha-256 28:37:F2:5F:C4:C3:DD:EF:D8:07:F5:53:7B:31:65:8A:7C:46:EF:35:AC:C9:C9:F1:16:79:CB:C9:7F:4E:CE:24
a=setup:active
a=mid:audio
a=extmap:1 urn:ietf:params:rtp-hdrext:ssrc-audio-level
a=extmap:3 http://www.webrtc.org/experiments/rtp-hdrext/abs-send-time
a=sendrecv
a=rtcp-mux
a=rtpmap:111 opus/48000/2
a=fmtp:111 minptime=10
a=rtpmap:103 ISAC/16000
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:126 telephone-event/8000
a=maxptime:60
```

Τέλος, ο δημιουργός το λαμβάνει και για να ολοκληρωθεί η εγκαθίδρυση της RTCPeer σύνδεσης ο χρήστης A (ο δημιουργός) καλεί την `setRemoteDescription` και έτσι η σύνδεση μεταξύ των δύο peers είναι πλέον εφικτή.

```
sdp success
```

Παράλληλα, από το `onaddstream` event handler καλείται μια συνάρτηση, η οποία περιμένει να ληφθεί το απομακρυσμένο stream για να το προβάλει στην οθόνη του χρήστη, μέσα στο `html video element` για το συγκεκριμένο stream.

```
sdp success
onaddstream {
  "onremovetrack": null,
  "onaddtrack": null,
  "onended": null,
  "oninactive": null,
  "onactive": null,
  "active": true,
  "ended": false,
  "id": "eQRMwHU1xIqLYutMH4qBTKFcXmqFb6z8tTBW",
  "label": "eQRMwHU1xIqLYutMH4qBTKFcXmqFb6z8tTBW"
}
on:add:stream ► Object
```

Τώρα πια οι δύο χρήστες μπορούν να επικοινωνήσουν σε πραγματικό χρόνο με βίντεο και ήχο μέσω του browser τους.

```
added: audio candidate:704553097 1 udp 2122260223 192.168.1.3 56904 typ host generation 0
added: data candidate:704553097 1 udp 2122260223 192.168.1.3 56904 typ host generation 0
added: video candidate:704553097 1 udp 2122260223 192.168.1.3 56904 typ host generation 0
```

Αυτή η αρχιτεκτονική ονομάζεται JSEP, Java Session Establishment Protocol.

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

8 Περιγραφή περιβάλλοντος

Το περιβάλλον της πιλοτικής εφαρμογής που αναπτύχθηκε, αποτελείται από μια html σελίδα, η οποία θα παρουσιαστεί αναλυτικά στη συνέχεια.

Επίσης η παρακάτω ανάλυση θα γίνει σε Google Chrome.

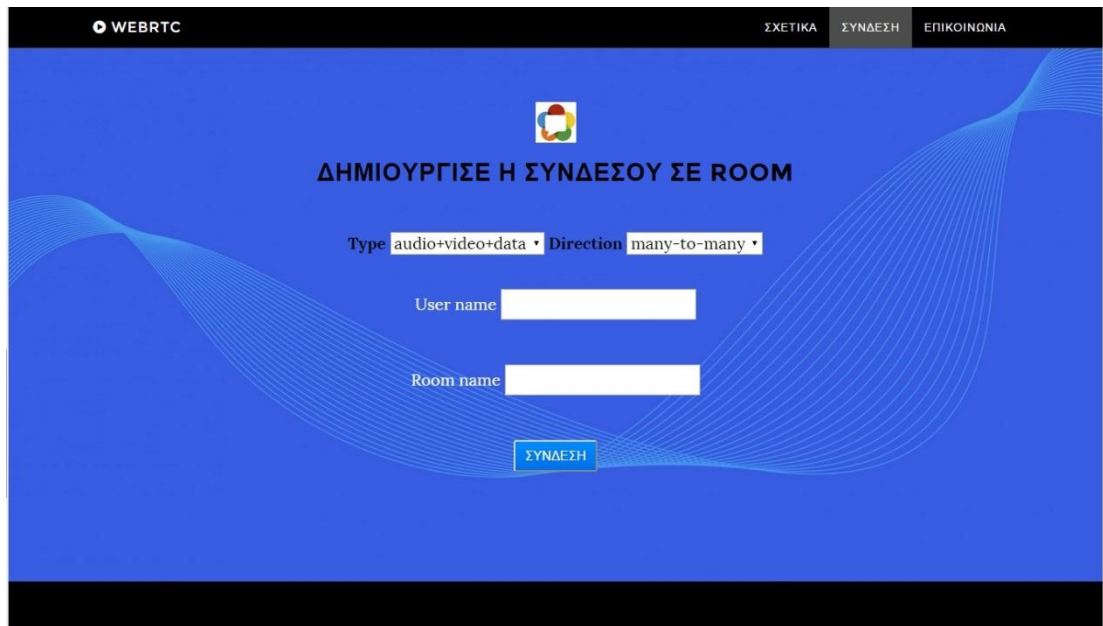
Η εφαρμογή μας βρίσκεται:

<http://aliendragoninnovativecreations.com/cmak/cmakwrtc/grayscale.php>

Η εφαρμογή μας είναι μέσα σε ένα template

«<http://ironsummitmedia.github.io/startbootstrap-grayscale/>» «one page» του Bootstrap.

Εμείς θα ξεκινήσουμε από το σημείο που βρίσκεται ήδη η εφαρμογή μας.



8.1 Εικόνα δημιουργίας Room ή Session.

Εδώ ο χρήστης έχει 5 επιλογές:

1. Το «type» που θα συνδεθούν, δηλαδή τι είδος σύνδεση θα έχουν (audio+video+data ή audio+data).
2. Και σε αυτό το κομμάτι πάλι έχει να κάνει με το είδος της σύνδεσης αλλά και με πόσους χρηστές (δηλαδή ένας με πολλούς ή όλοι με όλους) θα μπορούν να επικοινωνούν μεταξύ τους.
3. Το επόμενο κομμάτι είναι το User Name στο οποίο ο χρήστης θα μπορεί να βάλει ένα όνομα χρήστη.

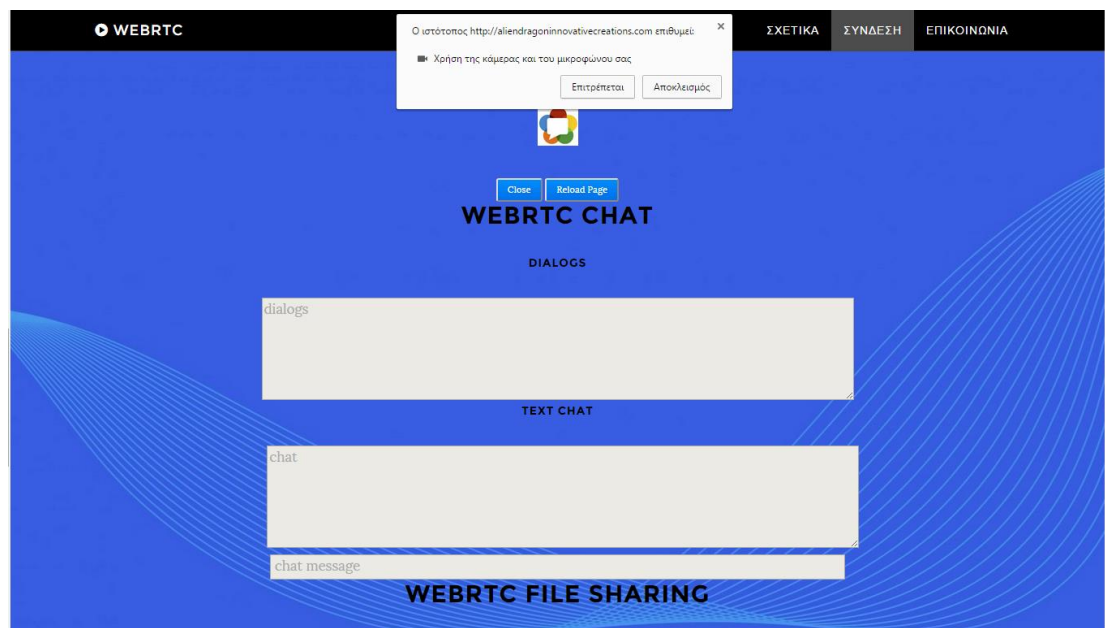
"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

4. Μετά έχουμε το Room Name στο οποίο ο χρήστης θα μπορεί να δώσει ένα όνομα στο Room που θα δημιουργήσει.
5. Το τελευταίο κομμάτι είναι ένα κουμπί, που παίρνει όλα τα πιο πάνω στοιχεία και δημιουργεί το Room.

Σε περίπτωση που είναι Default όλες οι επιλογές (many-to-many και audio+video+data) τότε ακολουθούν τα παρακάτω.

8.2 Ερώτηση αποδοχής Video

Η επόμενη εικόνα είναι όταν ο browser μέσω του «server» μας κάνει ερώτηση για να ενεργοποιήσουμε το video (την camera του υπολογιστή μας).

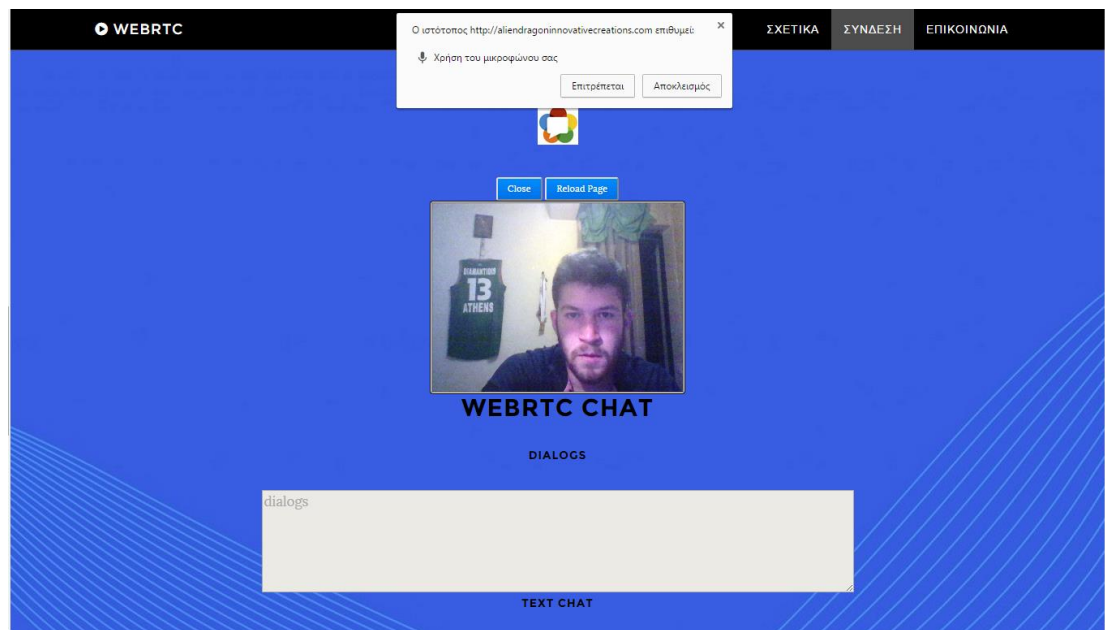


Εικόνα ερώτησης Video

8.2 Ερώτηση αποδοχής Audio

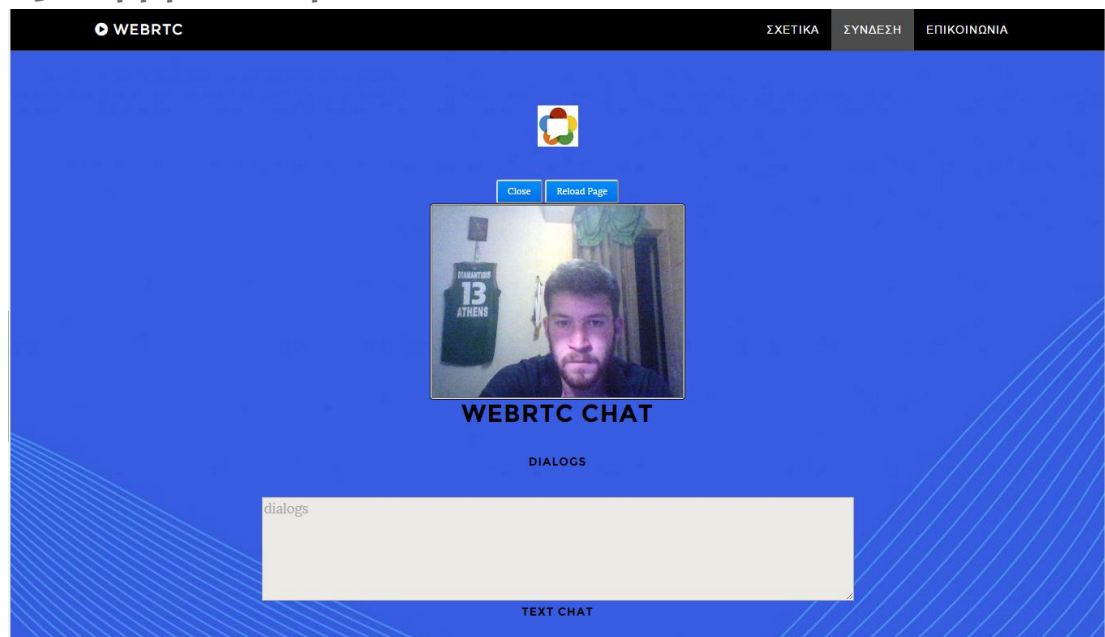
Η επόμενη εικόνα είναι αντίστοιχα για το audio.

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"



Εικόνα ερώτησης Audio.

8.3 Ενεργή σύνδεση



"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

Στην πιο πάνω εικόνα βλέπουμε πως είναι το interface όταν έχουμε δημιουργήσει το Room.(θα έπρεπε να ήταν μια εικόνα αλλά δυστυχώς είναι δύο)

Εδώ ο χρήστης βλέπει επτά διαφορετικές κατηγορίες:

1. Η πρώτη κατηγορία είναι ένα κουμπί «close» το οποίο του δίνει την δυνατότητα να κλείσει την σύνδεση του.
2. Η δεύτερη κατηγορία είναι πάλι ένα κουμπί με το οποίο ο χρήστης μπορεί να ανανεώσει την σελίδα για τυχόν προβλήματα η για δημιουργία καινούργιου Room.
3. Η Τρίτη κατηγορία είναι το canvas του video, που στην συγκεκριμένη εφαρμογή είναι ένα Div(στον κώδικα). Το οποίο έχει μέσα στο canvas ορισμένα κουμπιά τα οποία είναι:
 - a. Το πρώτο κουμπί είναι για απενεργοποίηση του ήχου.
 - b. Το δεύτερο κουμπί είναι για απενεργοποίηση του video.
 - c. Το τρίτο κουμπί είναι για καταγραφή του ήχου
 - d. Το τέταρτο κουμπί είναι για καταγραφή του video.
 - e. Το επόμενο δεν είναι κουμπί (αλλά δεν πειράζει),είναι μια μπάρα για την ένταση του volume του αντιστοίχου συμμετέχοντα.
 - f. Το επόμενο κουμπί δίνει την δυνατότητα στον χρήστη να κάνει full-screen την εικόνα του.



Εικόνα εργαλείων οθόνης

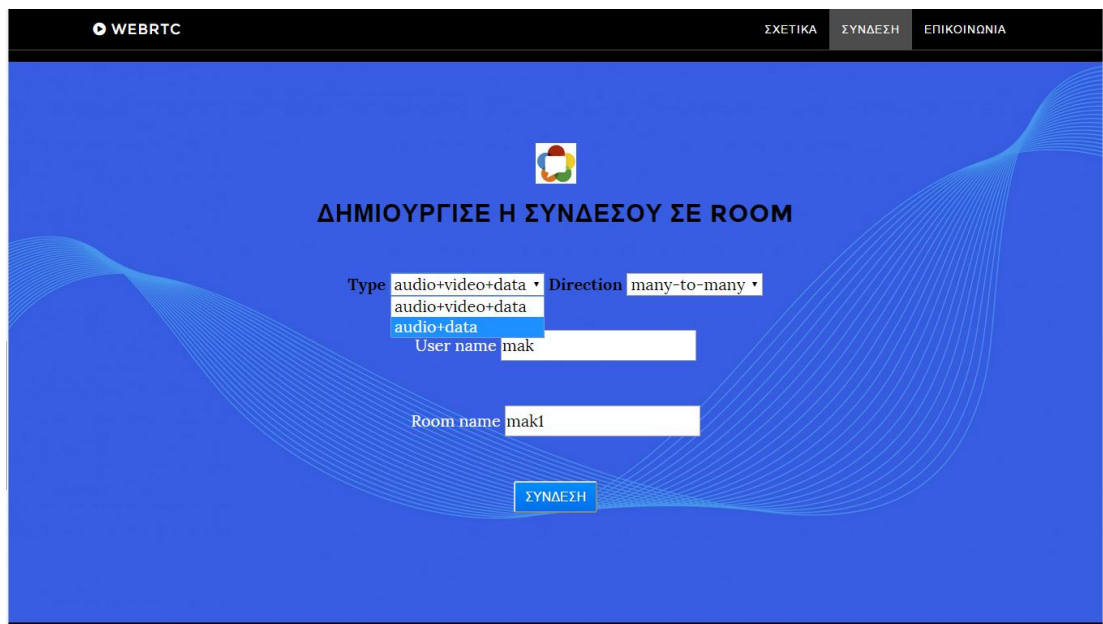
4. Η τέταρτη κατηγορία είναι μια text area στην οποία γίνεται η χρήση του ASR και Google Translation. Μέσα στην text area εμφανίζονται η καταγραφή της αναγνώρισης της ομιλίας σε κείμενο (δεν πειράζουμε κάτι μόνο διαβάζουμε).

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

5. Η πέμπτη κατηγορία είναι πάλι μια text area η οποία παίρνει τιμές από το input chat και τις εμφανίζει σαν διάλογο (πάλι δεν πειράζουμε κάτι μόνο διαβάζουμε).
6. Η έκτη κατηγορία είναι ένα input text στο οποίο ο χρήστης μέσα μπορεί να γράψει ένα μήνυμα που θα το διαβάσουν όλοι οι συμμετέχοντες.
7. Η έβδομη και τελευταία κατηγορία είναι ένα input file στο οποίο ο χρήστης μπορεί να ανεβάσει ένα αρχείο προς τους συμμετέχοντες

8.4 Ενεργή σύνδεση-2

Το επόμενο κομμάτι που θα αναλύσουμε είναι η σύνδεση μόνο με audio και video.

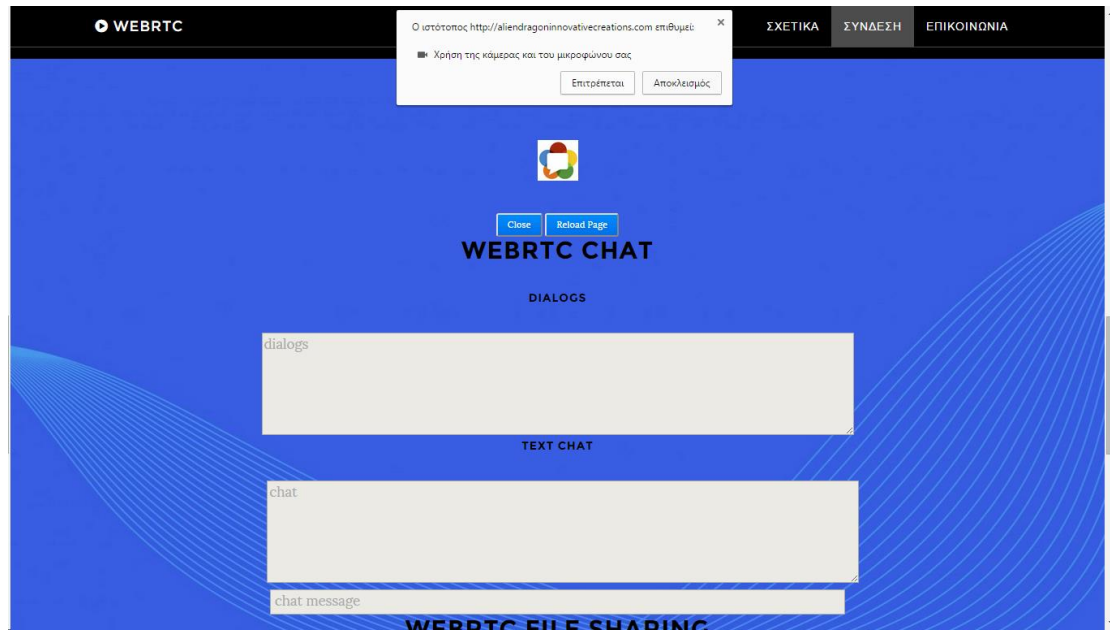


Εικόνα δημιουργία Room ή Session "audio+data"

Εδώ δημιουργούμε ένα session στο οποίο δεν θα υπάρχει εικόνα στην σύνδεση, θα μπορεί ο χρήστης να κάνει μόνο ακουστική σύνδεση ή να ανταλλάξει μηνύματα.

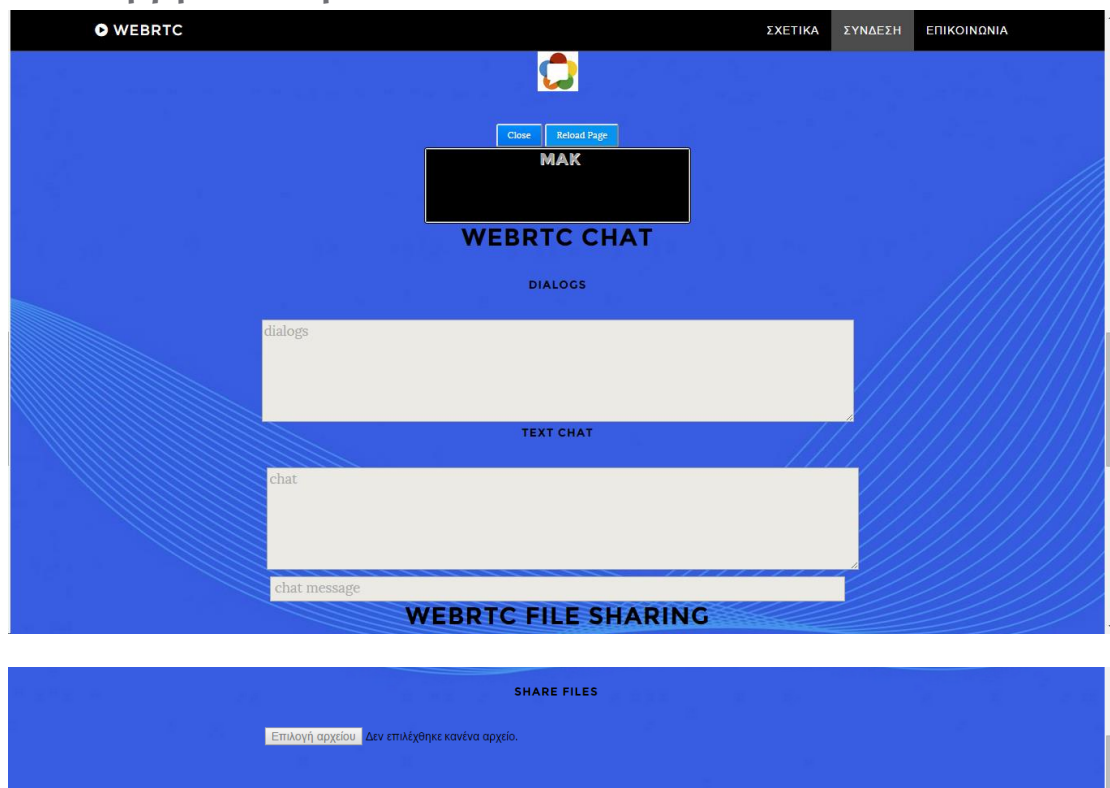
"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

8 5 Ερώτηση αποδοχής Audio για “audio και data”



Και σε αυτήν την σύνδεση γίνεται η ερώτηση αποδοχής για το μικρόφωνο.

8 6 Ενεργή σύνδεση “audio και data”



Εικόνα ενεργής σύνδεσης με “audio και data”

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

Στην πιο πάνω εικόνα βλέπουμε πως είναι το interface όταν έχουμε δημιουργήσει το Room για σύνδεση με ήχο και δεδομένα. (θα έπρεπε πάλι να ήταν μια εικόνα αλλά δυστυχώς είναι δύο).

Σε αυτήν την περίπτωση δεν έχουμε πολλές διαφορές με την προηγούμενη σύνδεση (video+audio+data). Γι αυτό θα επικεντρωθούμε στις διαφορές.

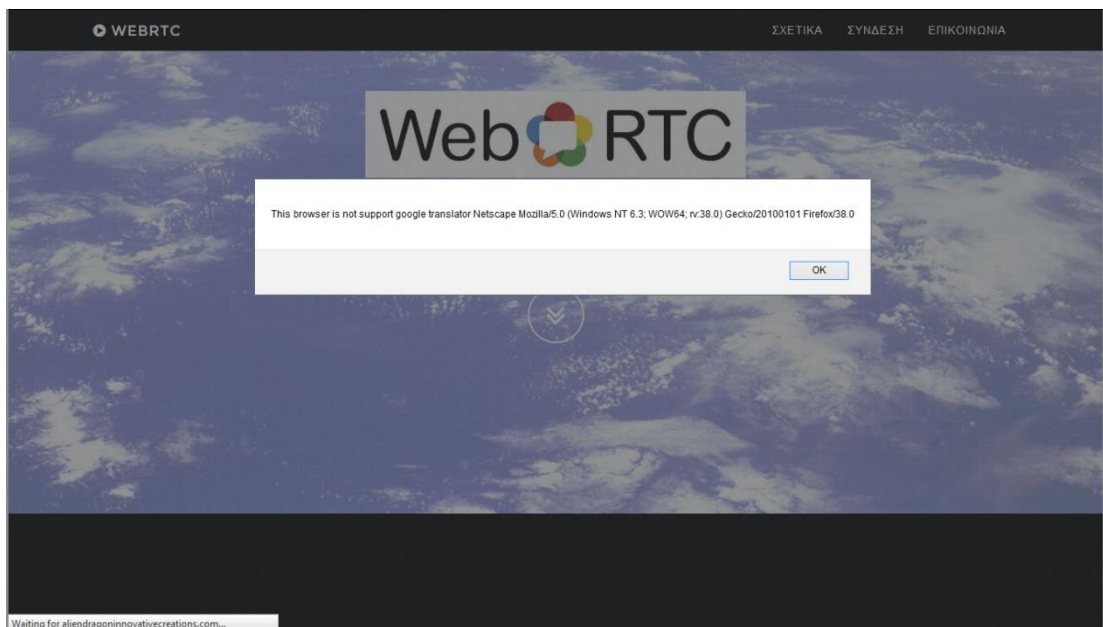
1. Πρώτη διαφορά είναι όπως επιλέξαμε δεν έχουμε εικόνα μόνο ένα πλαίσιο στο οποίο φαίνεται το όνομα του χρήστη.
2. Δεύτερη διαφορά είναι τα εργαλεία που αναλύσαμε πιο πάνω, μειώθηκαν και έγιναν ένα μόνο η μπάρα του Volume.



Τέλος, όλες οι άλλες λειτουργίες παραμένουν ίδιες.

8.7 Σύνδεση μέσω FireFox

Σε αυτό το κομμάτι θα αναλύσουμε την σύνδεση μέσω του Mozilla Firefox.



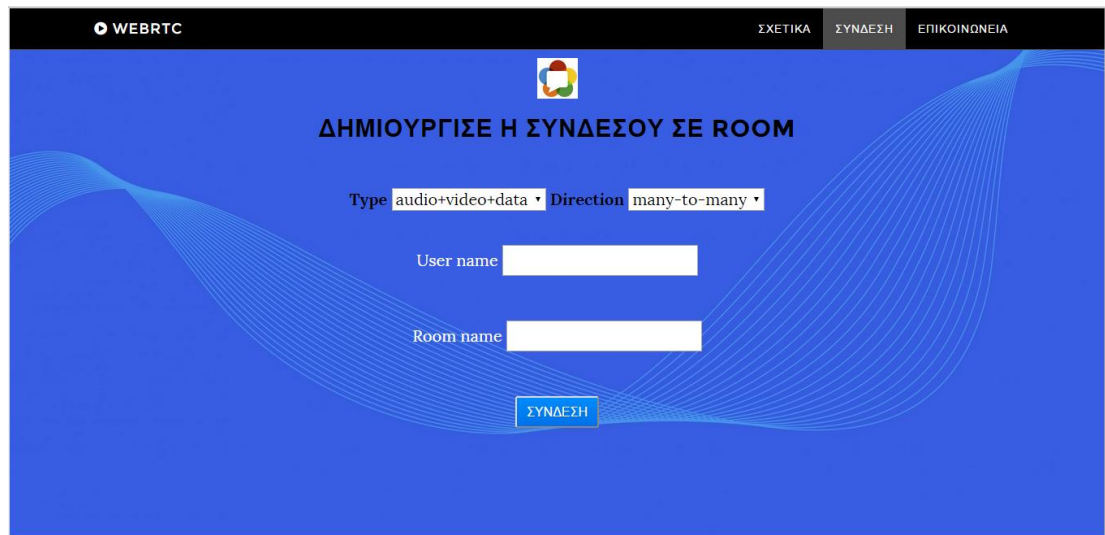
Εικόνα σύνδεση από Firefox.

Όταν θα πάμε να συνδεθούμε από τον Firefox θα δούμε ένα μήνυμα το οποίο μας λέει ότι η αναγνώριση της ομιλίας δεν υποστηρίζεται.

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

Επίσης μας παρέχει και κάποιες πληροφορίες για τον browser που έχουμε συνδεθεί όπως πχ ποιος browser είναι και την version του.

Αφού πατήσουμε ok (στο alert μήνυμα), η συνέχεια είναι ίδια δεν αλλάζει κάτι.



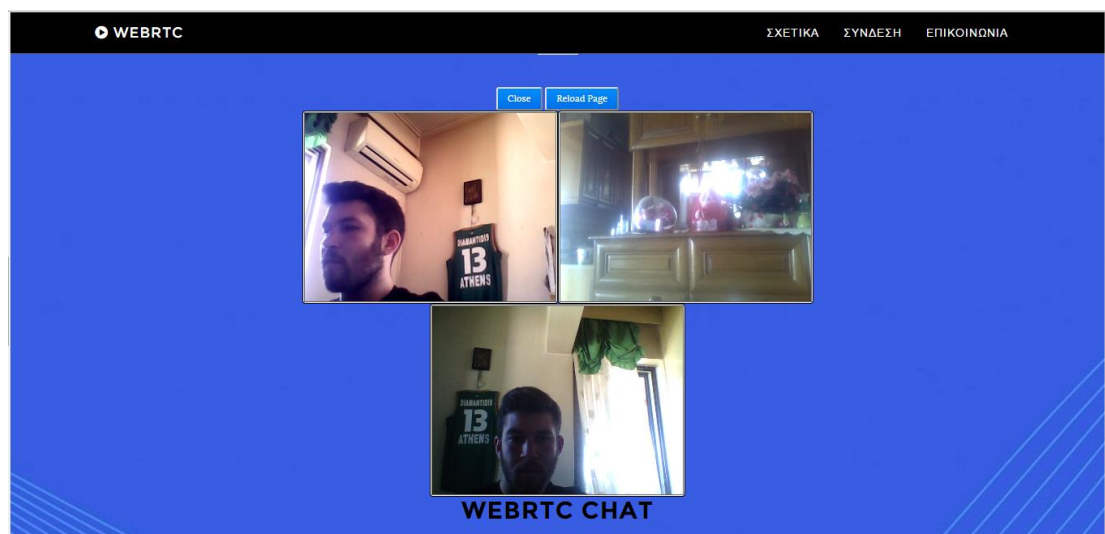
Εικόνα δημιουργίας Room ή Session.

8 8 Σύνδεσης

8 8 1 Multiconference

Σύνδεση 3^{ον} συσκευών

Θα ξεκινήσουμε μια σύνδεση με τρεις συσκευές με browser Google Chrome.



Εικόνα σύνδεσης 3^{ον} συσκευών

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

Επίσης να προσθέσω ότι η σύνδεση έγινε από το ίδιο access point. Εκ των οποίων η μία συσκευή ήταν ένα tablet «Turbo-X».

8 8 2 Dialogs

Εδώ βλέπουμε τους δύο υπολογιστές να έχουν ανταλλάξει κάποια μηνύματα.



Εδώ δεν έχουμε τόσο μεγάλη επιτυχία όπως αναμένονταν λόγω του ενός υπολογιστή και του tablet(χαμηλών προδιαγραφών για το ASR).

8 8 3 Text Chat

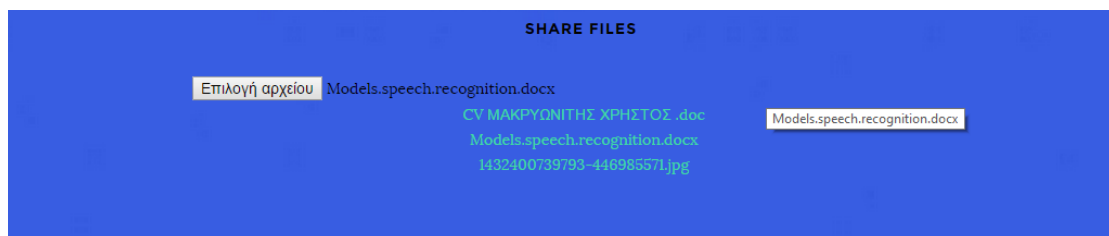


Εικόνα Text Chat

Στην πιο πάνω εικόνα φαίνεται η συνομιλία που ανταλλάξαν οι τρεις χρήστες μεταξύ τους.

8 8 4 Share File

Θα ανεβάσει ο κάθε χρήστης ξεχωριστά ένα αρχείο, μέσα στην σύνδεση.



Εικόνα share file

Ο χρήστης «mike» ανέβασε το CV μου (ένα αρχείο doc),ο χρήστης makι «εγώ» δηλαδή ανέβασα το SR.docx και τέλος ο χρήστης mak2 ανέβασε μια φωτογραφία με κωδικοποίηση .jpg .

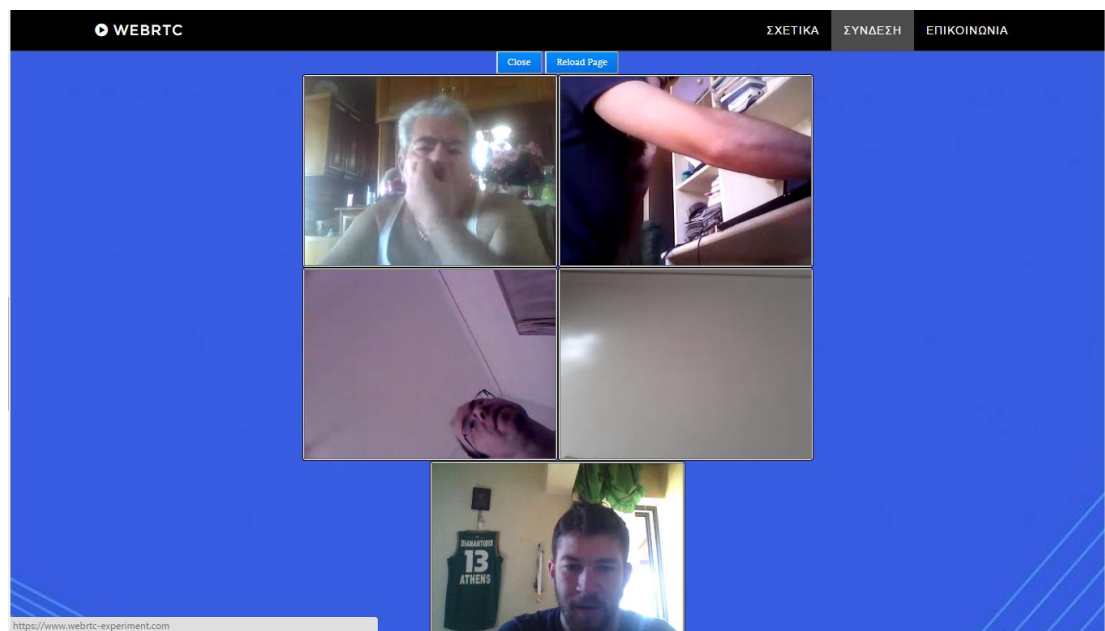
"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

8 9 Σύνδεση 5^ο συσκευών

Στην παρακάτω εικόνα θα απεικονίσουμε την σύνδεση, 5 διαφορετικών συσκευών.

Συσκευές

1. 2 tablet-Android.
2. 3 pc-windows.



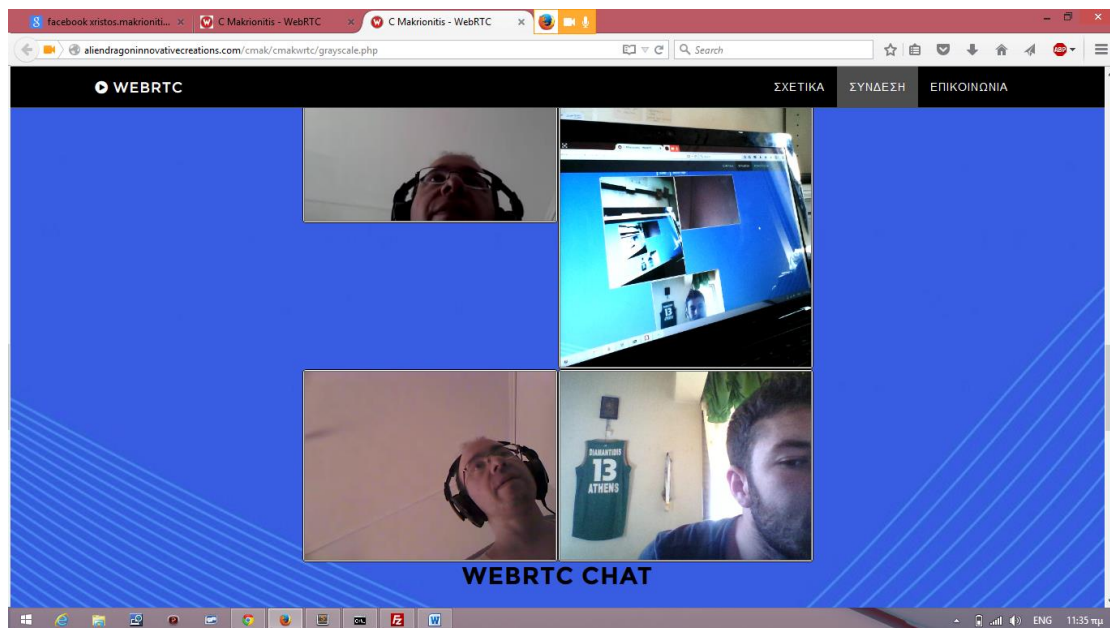
8 10 Σύνδεση 4^ο συσκευών

Στην παρακάτω εικόνα θα απεικονίσουμε την σύνδεση, 5 διαφορετικών συσκευών.

Συσκευές

1. 2 pc-Windows
2. 2 Tablet-Android

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

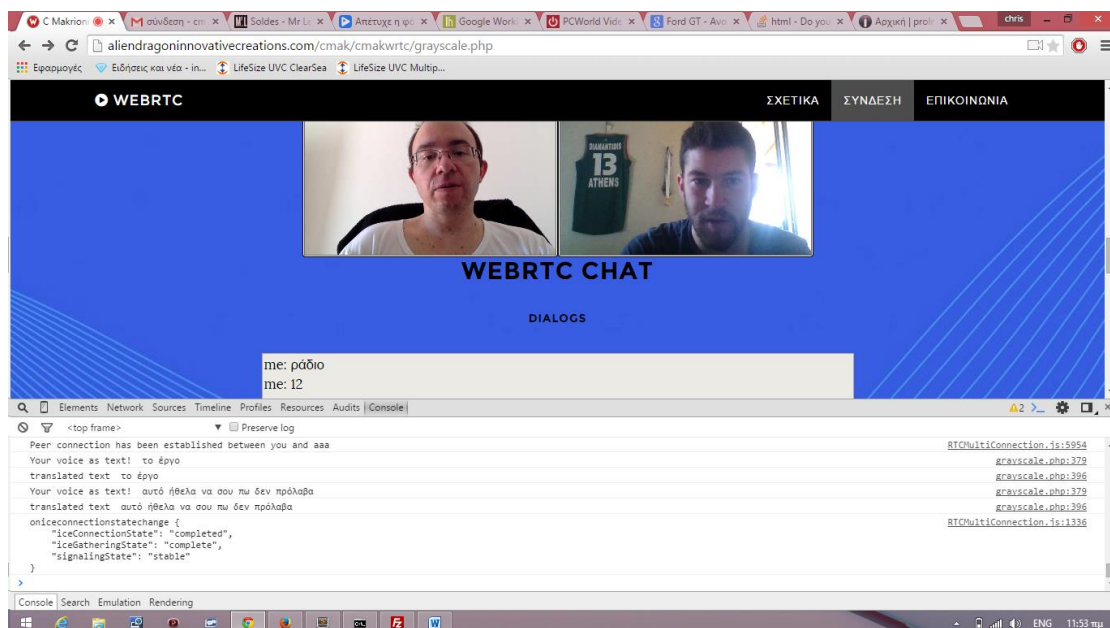


8.11 Σύνδεση 2 συσκευών

Στην παρακάτω εικόνα θα απεικονίσουμε την σύνδεση, 2 διαφορετικών συσκευών.

Συσκευές

1. Pc-Windows
2. Mac Book Pro-Mac



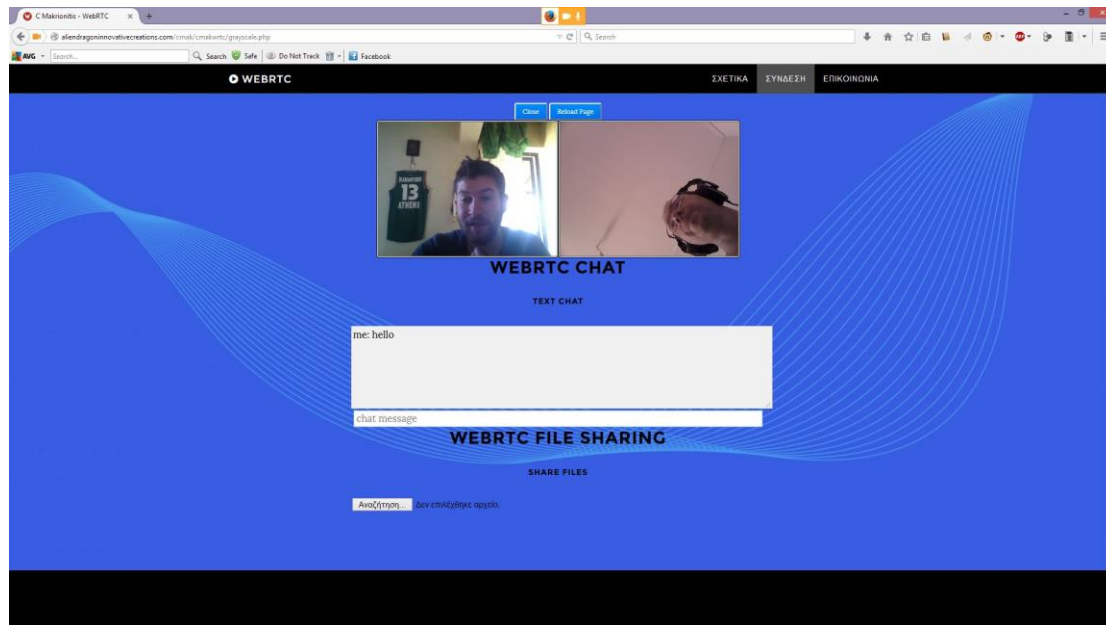
"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

8.12 Σύνδεση 2 συσκευών

Στην παρακάτω εικόνα θα απεικονίσουμε την σύνδεση, 2 διαφορετικών συσκευών με διαφορετικές συσκευές.

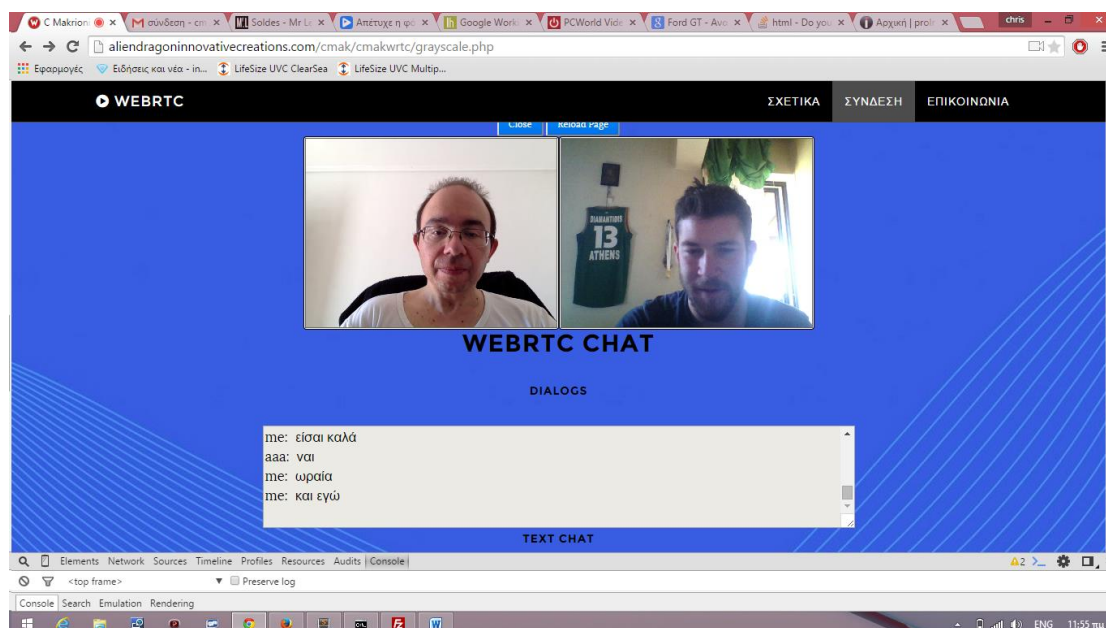
Συσκευές

1. 2 pc-Windows(Chrome-Firefox).



8.13 Σύνδεση 2 συσκευών

Στην παρακάτω εικόνα θα απεικονίσουμε την σύνδεση, 2 διαφορετικών συσκευών με την χρήση του Google Translate.



"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

9 Επισκόπηση

9.1 Συμπεράσματα

Από την επεξεργασία των ερευνητικών μας δεδομένων συμπεραίνουμε ότι το WebRTC είναι η τεχνολογία που έχει προάγει τον τομέα των επικοινωνιών σε σημαντικό επίπεδο και όπως τεκμαίρεται παραμένει να θεωρείται ότι είναι η πολλά υποσχόμενη τεχνολογία στον χώρο των επικοινωνιών πραγματικού χρόνου στο μέλλον.

Προβλέπεται ότι με την WebRTC τεχνολογία οι χρήστες θα φθάσουν στο επίπεδο να επικοινωνούν μεταξύ τους χωρίς να χρειάζεται να κάνουν λήψη κάποιας εφαρμογής στον υπολογιστή, στο κινητό ή στο tablet τους δεδομένου ότι θα υπάρχει ένας σύγχρονος browser εγκατεστημένος στο τερματικό τους.

Επί πλέον σημειώνεται ότι η τεχνολογία WebRTC προτιμάται από τους προγραμματιστές διότι διευκολύνει το έργο τους. Δηλαδή απαιτείται λιγότερος επίπονος χρόνος για να αναπτυχθούν οι σχετικές εφαρμογές, απ'ότι απαιτούνταν στο παρελθόν.

Επίσης υπογραμμίζεται ότι το WebRTC είναι ένα ανοικτό πρότυπο το οποίο βοηθάει στην υλοποίηση των ιδεών των προγραμματιστών που εστιάζουν π.χ. στις τηλεδιασκέψεις, στην μεταφορά αρχείων, στις κλήσεις κ.λ.π.

Τέλος συμπερασματικά αναφέρουμε ότι στην τρέχουσα περίοδο το WebRTC είναι ακόμη σε πειραματικό στάδιο και ως εκ τούτου τίποτα δεν έχει οριστικοποιηθεί με αποτέλεσμα πολλά απ'αυτά τα καθιερωμένα δεδομένα να κινδυνεύουν να ανατραπούν.

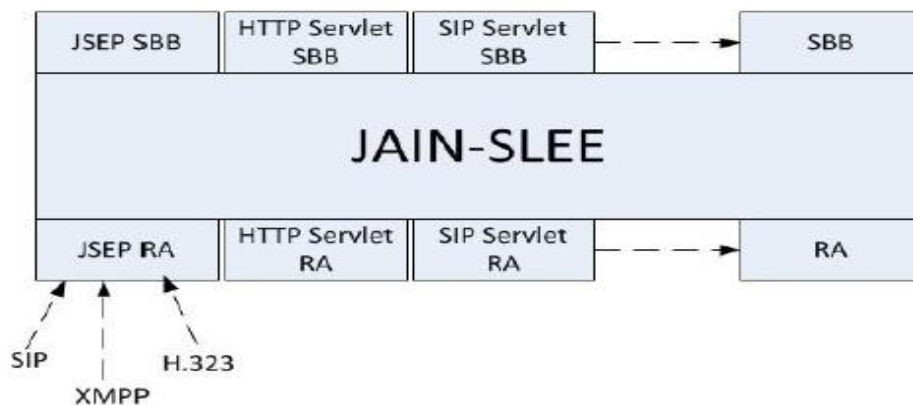
9.2 Προτεινόμενη μελλοντική μελέτη και πιθανές επεκτάσεις

Σε μελλοντική εργασία, έχουμε την πρόθεση να εφαρμόσουμε πρωτόκολλα βίντεο συνεχούς ροής παρόμοια με GridMedia και Coolstreaming για τη σημαντική αύξηση των επιδόσεων. Η εργασία μας έχει δείξει ότι τα pullbased πρωτόκολλα μπορούν να εφαρμοστούν σε WebRTC. Έτσι, φαίνεται πολύ πιθανό ότι τα τρέχοντα υβριδικά push-pull πρωτόκολλα μπορούν να εφαρμοστούν, λόγω των πολλών ομοιοτήτων με τα απλά pull-based πρωτόκολλα. Το μόνο που απομένει είναι η πρόκληση, δημιουργίας συστημάτων μεγάλης κλίμακας. Ωστόσο, αν και, κανένας περιηγητής δεν έχει εφαρμόσει πλήρως το τρέχον σχέδιο WebRTC, είναι πιθανό ότι θα εφαρμοστούν σε μελλοντικές εκδόσεις. Με τις πολλές peer-to-peer εφαρμογές και λύσεις που υπάρχουν μέχρι σήμερα, το WebRTC θα μπορούσε να ενδυναμώσει σημαντικά τις διαδικτυακές του εφαρμογές στο μέλλον.

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

Ακόμα, οι μελλοντικές προσπάθειες θα στοχεύσουν στην ανάπτυξη ενός πρωτοκόλλου-αγνωστικιστής WebRTC B2BUA. Ο στόχος είναι να επιτρέψει σε διάφορους πράκτορες χρήστες (user agents) (που μπορεί να είναι προγράμματα περιήγησης στο Web ή κανονικές κινητές ή desktop εφαρμογές επικοινωνίας) να λειτουργούν σε συνεργασία παρά το γεγονός ότι μπορεί ο καθένας να χρησιμοποιεί ένα διαφορετικό υπο-πρωτόκολλο (π.χ. SIP, XMPP ή H.323). Η εργασία θα χρησιμοποιήσει το Mobicents Java API για ολοκληρωμένο δίκτυο υπηρεσιών, Logic Execution Environment (JAIN SLEE) σαν πλατφόρμα επικοινωνίας της.

Το Σχ. 3 απεικονίζει την αναμενόμενη αρχιτεκτονική λύση που βασίζεται στο Mobicents JAIN SLEE, και χρησιμοποιώντας την Javascript Session Establishment Protocol (JSEP) ως ένα ενοποιητικό πρωτόκολλο για όλα τα υποκείμενα του χρήστη-πράκτορα (user agent) συγκεκριμένα υπο-πρωτόκολλα όπως το SIP, XMPP ή H.323. Όπως προκύπτει από την παρούσα ερευνητική εργασία, το μόνο WebRTC έργο αφορώντας τη χρήση του Mobicents server εφαρμογής χρησιμοποιεί το SIP Servlet και ως εκ τούτου, αναλαμβάνει μια all-SIP επικοινωνία[52].



Σχήμα 3. Το σχέδιο υλοποίησης JAIN-SLEE WebRTC

Το προτεινόμενο πρωτόκολλο-αγνωστικιστής B2BUA βασίζεται σε JSEP και οποιοδήποτε πρωτόκολλο σηματοδότησης της επιλογής (π.χ. SIP, H.323 και XMPP) μπορεί να συνδεθεί σε αυτό. Λόγω της ευελιξίας WebRTC σχετικά με τα υπο-πρωτόκολλα που χρησιμοποιούνται από τους πελάτες πάνω σε WebSocket, ο server εφαρμογής θα πρέπει να έχει την υποστήριξη πολλαπλών πρωτοκόλλων. Mobicents JAIN SLEE επιλέγεται επειδή ανταποκρίνεται σε αυτή την απαίτηση μέσω του πλαισίου Resource Adaptor (RA). Οι προσπάθειες βρίσκονται σε εξέλιξη για την ανάπτυξη της JSEP RA που απαιτείται για την κατανόηση του προτεινόμενου πρωτοκόλλου-αγνωστικιστής WebRTC B2BUA.

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

Βιβλιογραφία

- [1]. ADVANCED VIDEOCONFERENCING SERVICES BASED ON WEBRTC(Pedro Rodríguez, Javier Cerviño, Irena Trajkovska and Joaquín Salvachúa)
- [2]. <http://en.wikipedia.org/wiki/WebRTC>
- [3].<http://en.wikipedia.org/wiki/HTML5>
- [4]. Gauchat, J. D. (2012) HTML5 for Masterminds, US: Mini Books]
- [5]. http://www.w3schools.com/html/html5_video.asp
- [6]. Lubbers, P., Salim, F. & Albers, B. (2011) Pro HTML5 Programming, NY: Apress.
- [7]. <http://el.wikipedia.org/wiki/JavaScript>
- [8].<https://dspace.lib.uom.gr/dspace/bitstream/2159/16156/3/TsikopoulosSpyridonMsc2014.pdf>
- [9].<http://www.rtcmulticonnection.org/docs/session/>
- [10]. Βενιέρης, Ι. & Νικολούζου, Ε. (2006) Τεχνολογίες Διαδικτύου, Θεσσαλονίκη: ΤΖΙΟΛΑ
- [11]. http://en.wikipedia.org/wiki/IP_address
- [12]. <http://el.wikipedia.org/wiki/UDP>
- [13]. <http://searchsecurity.techtarget.com/definition/Transport-Layer-Security-TLS>.
- [14]. Johnston, B.A. & Burnett, C.D. (2012) WebRTC, APIs and RTCWEB Protocols of the HTML5 Real-Time Web. MO: C: Digital Codex LLC.
- [15]. Λυμπεροπούλου Στελίνα, (2013) Πτυχιακή Εργασία «Ανάπτυξη πιλοτικής εφαρμογής με χρήση της τεχνολογίας WebRTC»,Αθήνα.
- [16]. Grigorik, L. (2013) High Performance Browser Networking: What every web developer should know about networking and web performance, US: O'Reilly Media.
- [17]. R. Mahy & P. Matthews,(2010) Traversal Using Relays around NAT (TURN):Relay Extensions to Session Traversal Utilities for NAT (STUN).
- [18]. http://en.wikipedia.org/wiki/Session_Description_Protocol
- [19]. http://en.wikipedia.org/wiki/Secure_Real-time_Transport_Protocol
- [20]. <http://el.wikipedia.org/wiki/Peer-to-peer>.

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

- [21]. Stephanos Androutsellis-Theotokis, (2002) A Survey of Peer-to-Peer File Sharing Technologies, Athens
- [22]. <http://www.webrtc.org/architecture#TOC-WebRTC-Native-C-API>
- [23]. A. Bergkvist, D. C. Burnett, C. Jennings, and A. Narayanan. (2013) Webrtc 1.0: Real-time communication between browsers. [Online]. Available: <http://dev.w3.org/2011/webrtc/editor/webrtc.html>
- [24]. IETF. (2013) Rtcweb status pages. [Online]. Available: <http://tools.ietf.org/wg/rtcweb/charters>
- [25]. M. Bu and E. Zhang. (2013) PeerJS - peer-to-peer data in the web browser. [Online]. Available: <http://peerjs.com/>
- [26]. M. Zhang, Q. Zhang, L. Sun, and S. Yang, "Understanding the power of pull-based streaming protocol: Can we do better?" Selected Areas in Communications, IEEE Journal on, vol. 25, no. 9, pp. 1678–1694, 2007.
- [27]. WebRTC, Official Web Site : www.webrtc.org, 2013.
- [28]. J. Rosenberg, J. Peterson, , H. Schulzrinne, , G. Camarillo, "Best Current Practices for Third Party Call Control (3pcc) in the Session Initiation Protocol (SIP)", IETF RFC 3725, 2004.
- [29]. Elleuch, W., Houle, A.C, "Multiparty Voice over IP (MVoIP) Peer-based System for Large-scale Conference Support", 4th IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, 2008.
- [30]. Elleuch, W., Houle, A.C, "SIP-based Protocol for P2P Large-scale Multiparty VoIP (MVoIP) Conference Support", 6th Annual IEEE Consumer Communications & Networking Conference, 2009.
- [31]. J.W. Byers, J. Considine, M. Mitzenmacher, and S. Rost, "Informed Content Delivery Across Adaptive Overlay Networks," IEEE/ACM Trans. Networking, 2004.
- [32]. Dr. Cengiz Toğay NETAŞ, "Mobil Cihazlarda WebRTC Teknolojisi WebRTC Technology for Mobile Devices", İstanbul, Türkiye ctogay@netas.com.tr, 2014.
- [33]. <http://www.docsoft.com/Resources/Studies/Whitepapers/whitepaper-ASR.pdf>
- [34]. Mohri, M. (2002). "Edit-Distance of Weighted Automata: General Definitions and Algorithms" (PDF). International Journal of Foundations of Computer Science 14 (6): 957–982.
- [35]. http://en.wikipedia.org/wiki/Dynamic_time_warping

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

- [36].Waibel, A.; Hanazawa, T.; Hinton, G.; Shikano, K.; Lang, K. J. (1989). "Phoneme recognition using time-delay neural networks". IEEE Transactions on Acoustics, Speech and Signal Processing 37: 328–339.
- [37].Wu, J.; Chan, C. (1993). "Isolated Word Recognition by Neural Network Models with Cross-Correlation Coefficients for Speech Dynamics". IEEE Trans. Pattern Anal. Mach. Intell 15: 1174–1185.
- [38].S. A. Zahorian, A. M. Zimmer, and F. Meng, (2002) "Vowel Classification for Computer based Visual Feedback for Speech Training for the Hearing Impaired," in ICSLP 2002.
- [39].G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, B. Kingsbury (2012). "Deep Neural Networks for Acoustic Modeling in Speech Recognition --- The shared views of four research groups," IEEE Signal Processing Magazine, vol. 29, no. 6, pp. 82-97.
- [40].L. Deng and D. Yu (2014) "Deep Learning: Methods and Applications"
<http://research.microsoft.com/pubs/209355/DeepLearning-NowPublishing-Vol7-SIG-039.pdf>.
- [41].Keynote talk: "Achievements and Challenges of Deep Learning - From Speech Analysis and Recognition To Language and Multimodal Processing," Interspeech, September 2014 (by Li Deng).
- [42].Yu, D.; Deng, L. (2014). "Automatic Speech Recognition: A Deep Learning Approach (Publisher: Springer)".
- [43].http://en.wikipedia.org/wiki/Speech_recognition.
- [44].<http://shapedshed.com/html5-speech-recognition-api/>
- [45].http://en.wikipedia.org/wiki/Skype_protocol
- [46].http://www.academia.edu/5717224/Viber_Communication_Security_unscramble_the_scrambled_Contents
- [47].<http://www.oovoo.com/TermsAndPolicies.aspx>.
- [48].<http://en.wikipedia.org/wiki/FaceTime>
- [49].Farr, Christina (February 13, 2013). "Firebase's scalable backend makes it '10 times easier' to build apps". VentureBeat. Retrieved June 12, 2014.
- [50].Marshall, Matt (August 29, 2013). "Firebase is building a Dropbox for developers". VentureBeat. Retrieved June 12, 2014.
- [51].<https://cloud.google.com/translate/docs>.

"Πιλοτική εφαρμογή για Multipoint σύνδεση με την χρήση του Web RTC"

[52]. Michael Adeyeye¹, Member, IEEE, Ishmeal Makitla², and Thomas Fogwill², Member, IEEE "Determining the Signalling Overhead of two common WebRTC methods:JSON via XMLHttpRequest and SIP over WebSocket",2013.