



ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

**ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ**

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**Σχεδιασμός εφαρμογής κλιμακώσιμης τηλεδιάσκεψης
βασισμένη στο webrtc και stun/turn servers**

Ευάγγελος Χ. Λουκάς

Επιβλέπων : **Παντελής Μπαλαούρας, ΕΡΓΑΣΤΗΡΙΑΚΟ ΔΙΔΑΚΤΙΚΟ
ΠΡΟΣΩΠΙΚΟ (ΕΔΙΠ)**

ΑΘΗΝΑ

ΑΠΡΙΛΙΟΣ 2021

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Σχεδιασμός εφαρμογής κλιμακώσιμης τηλεδιάσκεψης βασισμένη στο webrtc
και stun/turn servers

Ευάγγελος Χ. Λουκάς

A.M.: 1115201400332

ΕΠΙΒΛΕΠΩΝ: **Παντελής Μπαλαούρας**, Εργαστηριακό Διδακτικό
Προσωπικό (ΕΔΙΠ)

ΠΕΡΙΛΗΨΗ

Η συγκεκριμένη εργασία μελετά τις επικοινωνίες σε πραγματικό χρόνο μέσω του διαδικτύου, τα πρωτόκολλα που χρησιμοποιούνται, η υποδομή καθώς επίσης γίνεται αναφορά σε μεθόδους τηλεκπαίδευσης, όπου εκτιμώνται σε πραγματικό πεδίο παράμετροι και μεταβλητές.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Εφαρμογές τηλεδιάσκεψης

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: τηλεπικοινωνίες, διαδίκτυο, τηλεδιάσκεψη, κλιμάκωση, τηλεκπαίδευση

ABSTRACT

This paper studies real-time communications via the Internet, the protocols used, the infrastructure as well as reference to e-learning methods, where parameters and variables are evaluated in real field.

SUBJECT AREA: Video conferencing applications

KEYWORDS: telecommunications, internet, teleconferencing, escalation,
distance learning

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΡΟΛΟΓΟΣ.....	10
1. ΕΙΣΑΓΩΓΗ ΣΤΟ WEBRTC	11
1.1 Ένα πολύ σύντομο ιστορικό του webrtc	11
1.2 Σηματοδότηση: έλεγχος συνεδρίας, πληροφορίες δικτύου και μέσων.....	12
1.3 RTCPeerConnection.....	14
1.4 RTCPeerConnection χωρίς διακομιστές.....	15
1.5 RTCPeerConnection με διακομιστές	16
1.6 Τοπολογίες δικτύου	18
1.7 RTC Data Channel	20
1.8 Ασφάλεια	21
2.STUN, TURN ΚΑΙ ΣΗΜΑΤΟΔΟΤΗΣΗ	22
2.1 Οι λόγοι που η σηματοδότηση δεν ορίζεται από το webrtc	22
2.2 RTCPeerConnection και σηματοδότηση	23
2.3 Δημιουργία υπηρεσίας σηματοδότησης.....	24
2.4 Ασφάλεια σηματοδοσίας	26
2.5 Χρησιμότητα ICE	26
2.6 STUN	28
2.7 TURN.....	29
2.8 Μονάδα Ελέγχου Πολλαπλών Σημείων	30
2.9 VOIP, τηλέφωνα και μηνύματα.....	31
3.ΚΛΙΜΑΚΩΣΗ	32

3.1 Εισαγωγή	32
3.2 Ανάλυση της πολυπλοκότητας.....	32
3.3 Ενεργοί χρήστες σε μια κλήση webrtc.....	36
3.4 Χωρητικότητα ροών σε έναν media server	40
3.5 Cascading σε μια συνεδρία	41
 4.ΜΕΛΕΤΗ ΣΕΝΑΡΙΩΝ ΧΡΗΣΗΣ	46
 5.ΣΥΜΠΕΡΑΣΜΑΤΑ.....	61
 ΠΙΝΑΚΑΣ ΟΡΟΛΟΓΙΑΣ.....	62
 ΣΥΝΤΜΗΣΕΙΣ-ΑΡΚΤΙΚΟΛΕΞΑ-ΑΚΡΩΝΥΜΙΑ	63
 ΑΝΑΦΟΡΕΣ	64

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Εικόνα 1 Αρχιτεκτονική JSEP	14
Εικόνα 2 Αρχιτεκτονική WebRTC.....	15
Εικόνα 3 Εύρεση υποψηφίων σύνδεσης.....	17
Εικόνα 4 Διαδρομές δεδομένων WebRTC	18
Εικόνα 5 Παράδειγμα τοπολογίας μονάδας ελέγχου πολλαπλών σημείων....	19
Εικόνα 6 Teth/Trovo: Επικοινωνίες καταστροφών σε ένα χαρτοφύλακα	20
Εικόνα 7 Αρχιτεκτονική JSEP	23
Εικόνα 8 Η υπηρεσία appr.tc	26
Εικόνα 9 Ένας κόσμος χωρίς NAT και τείχη προστασίας	27
Εικόνα 10 Ο πραγματικός κόσμος	27
Εικόνα 11 Χρήση διακομιστών STUN για τη λήψη δημόσιας IP και θύρας	29
Εικόνα 12 Η πλήρης διαδικασία με STUN, TURN και σηματοδότηση	30
Εικόνα 13 Η πίσω όψη ενός Cisco MCU	30
Εικόνα 14 Ομαδική κλήση.....	33
Εικόνα 15 Στυλ Hangouts	35
Εικόνα 16 9 συμμετέχοντες σε διάταξη παραθύρων	37
Εικόνα 17 Εναλλακτική υλοποίηση	38
Εικόνα 18 Πολυπλοκότητα σε σχέση με αριθμό συμμετεχόντων	39
Εικόνα 19 Cascading	42
Εικόνα 20 Εκδοχή 1	43
Εικόνα 21 Εκδοχή 2	43
Εικόνα 22 Τύπος κατανομής.....	44
Εικόνα 23 Μονόδρομη επικοινωνία από καθηγητή σε 25 μαθητές	47
Εικόνα 24 Αμφίδρομη επικοινωνία μεταξύ καθηγητή και μαθητών	52
Εικόνα 25 Μονόδρομη επικοινωνία σε όλους τους μαθητές εκτός από έναν .	56

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

Πίνακας 1 Σενάριο με 4 χρήστες	33
Πίνακας 2 Σενάριο με μικρότερη ανάλυση	34
Πίνακας 3 Σενάριο με 4 χρήστες σε 720p με simulcast.....	35
Πίνακας 4 Τελική Σύνοψη	36
Πίνακας 5 Απαιτήσεις σε εύρος ζώνης.....	47
Πίνακας 6 Μέση κίνηση ανά συνεδρία για το πρώτο σενάριο υλοποίησης	48
Πίνακας 7 Αριθμός servers ανά συνεδρία για το πρώτο σενάριο υλοποίησης με default ανάλυση.....	49
Πίνακας 8 Αριθμός servers 1Gbps για διαφορετικό αριθμό συνεδριών.....	49
Πίνακας 9 Αριθμός servers 10Gbps για διαφορετικό αριθμό συνεδριών.....	49
Πίνακας 10 Κόστος υλοποίησης για το πρώτο σενάριο	50
Πίνακας 11 Αριθμός servers ανά συνεδρία	50
Πίνακας 12 Αριθμός servers για διαφορετικό πλήθος συνεδριών	50
Πίνακας 13 Αριθμός servers 10Gbps για διαφορετικό πλήθος συνεδριών.....	51
Πίνακας 14 Κόστος υλοποίησης για την μικρή ανάλυση στο σενάριο Α.....	51
Πίνακας 15 Μέση κίνηση ανά συνεδρία για το δεύτερο σενάριο υλοποίησης	52
Πίνακας 16 Αριθμός servers ανά συνεδρία για το δεύτερο σενάριο υλοποίησης.....	53
Πίνακας 17 Αριθμός servers 1Gbps για διαφορετικό πλήθος συνεδριών.....	53
Πίνακας 18 Αριθμός servers 10Gbps για διαφορετικό αριθμό συνεδριών.....	54
Πίνακας 19 Κόστος υλοποίησης για το δεύτερο σενάριο	54
Πίνακας 20 Αριθμός servers ανά συνεδρία	54
Πίνακας 21 Αριθμός servers για διαφορετικό πλήθος συνεδριών	55
Πίνακας 22 Αριθμός servers 10Gbps για διαφορετικό πλήθος συνεδριών.....	55
Πίνακας 23 Κόστος υλοποίησης για τη μικρή ανάλυση στο σενάριο Β	56

Πίνακας 24 Μέση κίνηση ανά συνεδρία για το τρίτο σενάριο υλοποίησης	57
Πίνακας 25 Αριθμός servers ανά συνεδρία για το τρίτο σενάριο υλοποίησης	57
Πίνακας 26 Αριθμός servers 1Gbps για διαφορετικό αριθμό συνεδριών.....	58
Πίνακας 27 Αριθμός servers 10Gbps για διαφορετικό αριθμό συνεδριών.....	58
Πίνακας 28 Κόστος υλοποίησης για το τρίτο σενάριο	59
Πίνακας 29 Αριθμός servers ανά συνεδρία	59
Πίνακας 30 Αριθμός servers 1Gbps για διαφορετικό πλήθος συνεδριών.....	59
Πίνακας 31 Αριθμός servers 10Gbps για διαφορετικό πλήθος συνεδριών.....	60
Πίνακας 32 Κόστος υλοποίησης για τη μικρή ανάλυση στο σενάριο Γ	60

ΠΡΟΛΟΓΟΣ

Η εργασία αυτή υλοποιήθηκε κάτω από τις πρωτόγνωρες συνθήκες που βιώνουμε τον τελευταίο χρόνο, έγινε εξολοκλήρου εξ' αποστάσεως με τηλεδιασκέψεις και θέλω να ευχαριστήσω τον κ. Μπαλαούρα για την βοήθεια και τη στήριξή του.

1. ΕΙΣΑΓΩΓΗ ΣΤΟ WEBRTC

1.1 Ένα πολύ σύντομο ιστορικό του webrtc

Μία από τις τελευταίες μεγάλες προκλήσεις για τον Παγκόσμιο Ιστό είναι να επιτρέψει την ανθρώπινη επικοινωνία μέσω φωνής και βίντεο: Real Time Communication, RTC για συντομία. Το RTC πρέπει να είναι τόσο φυσικό σε μια εφαρμογή ιστού όσο και η εισαγωγή κειμένου σε έναν επεξεργαστή κειμένου.

Ιστορικά, το RTC ήταν εταιρικό και περίπλοκο, απαιτώντας ακριβές τεχνολογίες ήχου και βίντεο που έπρεπε να διαθέτουν άδεια ή να αναπτύσσονται εσωτερικά. Η ενοποίηση της τεχνολογίας RTC με το υπάρχον περιεχόμενο, δεδομένα και υπηρεσίες ήταν δύσκολη και χρονοβόρα, ιδιαίτερα στον Ιστό.

Η συνομιλία μέσω βίντεο του Gmail έγινε δημοφιλής το 2008 και το 2011 η Google παρουσίασε το Hangouts, το οποίο χρησιμοποιεί την υπηρεσία Google Talk (όπως και το Gmail). Η Google αγόρασε το GIPS, μια εταιρεία που είχε αναπτύξει πολλά στοιχεία που απαιτούνται για RTC, όπως κωδικοποιητές και τεχνικές ακύρωσης ηχούς. Το Google open προήλθε από τις τεχνολογίες που ανέπτυξε η GIPS και συνεργάστηκε με σχετικούς φορείς τυποποίησης στο IETF και στο W3C για να διασφαλίσει τη συναίνεση του κλάδου. Τον Μάιο του 2011, η Ericsson δημιούργησε την πρώτη εφαρμογή του RTC μέσω Web, το λεγόμενο webrtc.

Το WebRTC εφάρμοσε ανοιχτά πρότυπα για επικοινωνία βίντεο, ήχου και δεδομένων σε πραγματικό χρόνο, χωρίς πρόσθετα. Η ανάγκη ήταν πραγματική:

- Πολλές υπηρεσίες ιστού χρησιμοποίησαν RTC, αλλά χρειάστηκαν λήψεις, εγγενείς εφαρμογές ή προσθήκες. Σε αυτά περιλαμβάνονται τα Skype, Facebook και Google Hangouts.
- Η λήψη, εγκατάσταση και ενημέρωση προσθηκών είναι περίπλοκη, επιρρεπής σε σφάλματα και ενοχλητική.
- Οι προσθήκες είναι δύσκολο να αναπτυχθούν, να εντοπιστούν τα σφάλματα, να επιλυθούν προβλήματα, να δοκιμαστούν και να συντηρηθούν - και ενδέχεται να απαιτούν άδεια και ενοποίηση με πολύπλοκη, ακριβή τεχνολογία. Είναι συχνά δύσκολο να πείσεις τους χρήστες να εγκαταστήσουν πρώτα προσθήκες.

Οι κατευθυντήριες αρχές του έργου WebRTC είναι ότι τα API του πρέπει να είναι ανοιχτού κώδικα, δωρεάν, τυποποιημένα, ενσωματωμένα σε προγράμματα περιήγησης ιστού και πιο αποτελεσματικά από τις υπάρχουσες τεχνολογίες.

Οι εφαρμογές WebRTC πρέπει να κάνουν πολλά πράγματα:

- Λήψη ροής ήχου, βίντεο ή άλλων δεδομένων.
- Λήψη πληροφοριών δικτύου, όπως διευθύνσεις IP, θύρες και ανταλλαγή τους με άλλους πελάτες WebRTC (γνωστοί ως ομότιμοι-peers) για να ενεργοποιηθεί η σύνδεση, ακόμη και μέσω NAT και τείχους προστασίας.
- Συντονισμό της επικοινωνίας σηματοδότησης για την αναφορά σφαλμάτων και την εκκίνηση ή κλείσιμο συνεδριών.

- Ανταλλαγή πληροφοριών σχετικά με την ικανότητα πολυμέσων και πελατών, όπως ανάλυση και κωδικοποιητές.
- Επικοινωνία ήχου, βίντεο ή δεδομένων ροής.

Για την ανταλλαγή δεδομένων ροής, το WebRTC εφαρμόζει τα ακόλουθα API:

- `MediaStream`: αποκτά πρόσβαση σε ροές δεδομένων, όπως από την κάμερα και το μικρόφωνο του χρήστη.
- `RTCPeerConnection`: κλήσεις ήχου ή βίντεο, με εγκαταστάσεις για κρυπτογράφηση και διαχείριση εύρους ζώνης.
- `RTCDataChannel`: ομότιμη επικοινωνία γενικών δεδομένων.

1.2 Σηματοδότηση: έλεγχος συνεδρίας, πληροφορίες δικτύου και μέσων

Το WebRTC χρησιμοποιεί το `RTCPeerConnection` για την επικοινωνία ροής δεδομένων μεταξύ προγραμμάτων περιήγησης (γνωστοί και ως ομότιμοι), αλλά χρειάζεται επίσης έναν μηχανισμό για τον συντονισμό της επικοινωνίας και την αποστολή μηνυμάτων ελέγχου, μια διαδικασία γνωστή ως σηματοδότηση. Οι μέθοδοι και τα πρωτόκολλα σηματοδότησης δεν καθορίζονται από το WebRTC: η σηματοδότηση δεν αποτελεί μέρος του `RTCPeerConnection` API. [1]

Αντ' αυτού, οι προγραμματιστές εφαρμογών WebRTC μπορούν να επιλέξουν οποιοδήποτε πρωτόκολλο ανταλλαγής μηνυμάτων προτιμούν, όπως SIP ή XMPP και οποιοδήποτε κατάλληλο κανάλι αμφίδρομης επικοινωνίας.

Η σηματοδότηση χρησιμοποιείται για την ανταλλαγή τριών τύπων πληροφοριών:

- **Μηνύματα ελέγχου συνεδρίας:** για να αρχικοποιήσουμε ή να κλείσουμε την επικοινωνία και να αναφέρουμε σφάλματα.
- **Διαμόρφωση δικτύου:** στον εξωτερικό κόσμο, ποια είναι η διεύθυνση IP και η θύρα του υπολογιστή μας;
- **Δυνατότητες πολυμέσων:** με ποιους κωδικοποιητές και αναλύσεις μπορούμε να χειριστούμε το πρόγραμμα περιήγησής μας και το πρόγραμμα περιήγησης με το οποίο θέλουμε να επικοινωνήσουμε;

Η ανταλλαγή πληροφοριών μέσω σηματοδότησης πρέπει να έχει ολοκληρωθεί με επιτυχία πριν ξεκινήσει η ροή μεταξύ των ομοτίμων, η λεγόμενη ροή peer-to-peer ή p2p.

Για παράδειγμα, έστω ότι η Alice θέλει να επικοινωνήσει με τον Bob. Ακολουθεί ένα δείγμα κώδικα από το W3C WebRTC spec, το οποίο δείχνει τη διαδικασία σηματοδότησης σε δράση. Ο κώδικας προϋποθέτει την ύπαρξη κάποιου μηχανισμού σηματοδότησης, που δημιουργήθηκε στη `createSignalingChannel()` μέθοδο.

Αρχικά, η Alice και ο Bob ανταλλάσσουν πληροφορίες δικτύου. (Η έκφραση «εύρεση υποψηφίων» αναφέρεται στη διαδικασία εύρεσης διεπαφών δικτύου και θυρών χρησιμοποιώντας το πλαίσιο ICE. Το ICE σημαίνει Interactive Connectivity Establishment και καθορίζει έναν συστηματικό τρόπο εύρεσης πιθανών επιλογών επικοινωνίας μεταξύ δύο τελικών σημείων [2])

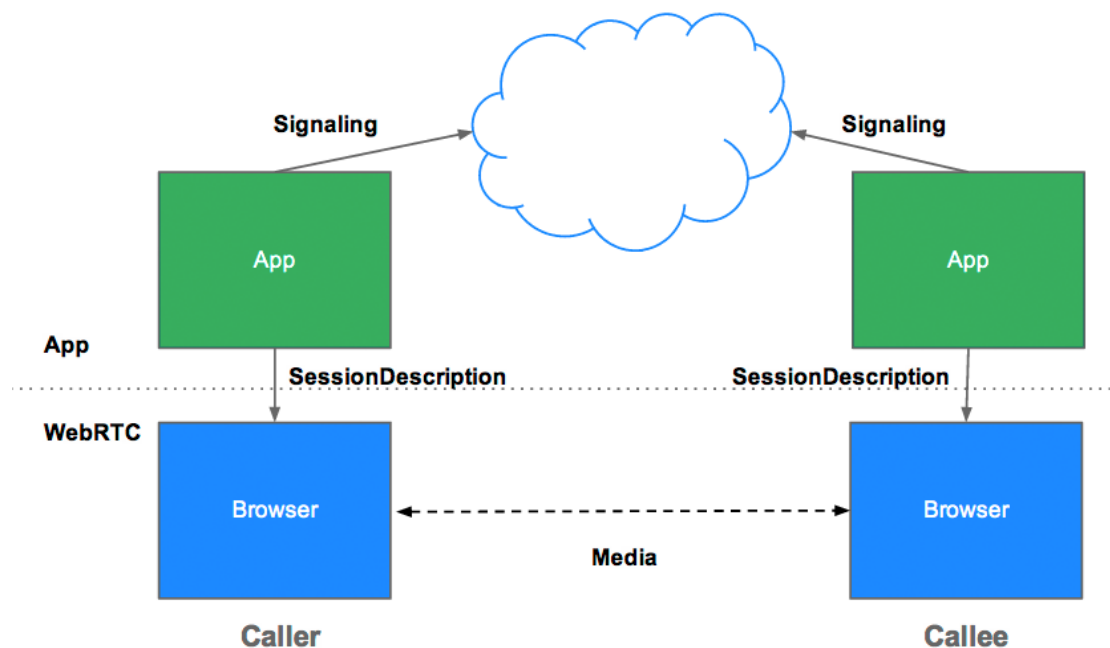
1. Η Alice δημιουργεί ένα αντικείμενο `RTCPeerConnection` με έναν `onicecandidate handler` ή αλλιώς χειριστή που αναλαμβάνει να κάνει τη δουλειά της εύρεσης των πιθανών επιλογών επικοινωνίας.
2. Ο χειριστής εκτελείται όταν οι υποψήφιοι του δικτύου γίνουν διαθέσιμοι.
3. Η Alice στέλνει σειριακά υποψήφια δεδομένα στον Bob, μέσω οποιουδήποτε καναλιού σηματοδότησης που χρησιμοποιεί: `WebSocket` ή κάποιον άλλο μηχανισμό.
4. Όταν ο Bob λαμβάνει ένα υποψήφιο μήνυμα από την Alice, καλεί `addIceCandidate`, για να προσθέσει τον υποψήφιο στην απομακρυσμένη περιγραφή (έχουμε κατά νου ότι υπάρχουν δύο επιλογές, η τοπική και απομακρυσμένη περιγραφή).

Οι πελάτες WebRTC (γνωστοί ως ομότιμοι , Alice και Bob) πρέπει επίσης να εξακριβώσουν και να ανταλλάξουν τοπικές και απομακρυσμένες πληροφορίες πολυμέσων ήχου και βίντεο, όπως δυνατότητες ανάλυσης και κωδικοποιητή. Η σηματοδότηση για την ανταλλαγή πληροφοριών διαμόρφωσης μέσων προχωρά ανταλλάσσοντας μια προσφορά και μια απάντηση χρησιμοποιώντας το πρωτόκολλο περιγραφής περιόδου λειτουργίας (SDP) [1]:

1. Η Alice εκτελεί τη `createOffer()` μέθοδο `RTCPeerConnection` . Η επιστροφή από αυτό περνά ένα `RTCSessionDescription`: Η περιγραφή της τοπικής περιόδου σύνδεσης της Alice.
2. Στην επιστροφή κλήσης, η Alice ορίζει την τοπική περιγραφή χρησιμοποιώντας `setLocalDescription()` και στη συνέχεια στέλνει αυτήν την περιγραφή περιόδου λειτουργίας στον Bob μέσω του καναλιού σηματοδότησης. Σημειώνουμε ότι το `RTCPeerConnection` δεν θα αρχίσει να συγκεντρώνει υποψηφίους έως ότου η εντολή `setLocalDescription()` καλείται: αυτό κωδικοποιείται στο πρόχειρο JSEP IETF .
3. Ο Bob ορίζει την περιγραφή που του έστειλε η Alice ως απομακρυσμένη περιγραφή `setRemoteDescription()`.
4. Ο Bob εκτελεί τη `createAnswer()` μέθοδο `RTCPeerConnection` , περνώντας την απομακρυσμένη περιγραφή που έλαβε από την Alice, ώστε να δημιουργηθεί μια τοπική συνεδρία που να είναι συμβατή με τη δική της. Η `createAnswer()` επιστροφή κλήσης μεταβιβάζεται σε `RTCSessionDescription`: Ο Bob το ορίζει ως τοπική περιγραφή και το στέλνει στην Alice.
5. Όταν η Alice παίρνει την περιγραφή περιόδου λειτουργίας του Bob, το ορίζει ως την απομακρυσμένη περιγραφή με `setRemoteDescription`.
6. Ping!

Η απόκτηση και ανταλλαγή πληροφοριών δικτύου και πολυμέσων μπορεί να γίνει ταυτόχρονα, αλλά και οι δύο διαδικασίες πρέπει να έχουν ολοκληρωθεί προτού ξεκινήσει η ροή ήχου και βίντεο μεταξύ ομότιμων.

Η αρχιτεκτονική προσφοράς / απάντησης που περιγράφεται παραπάνω ονομάζεται JSEP , JavaScript Session Establishment Protocol.



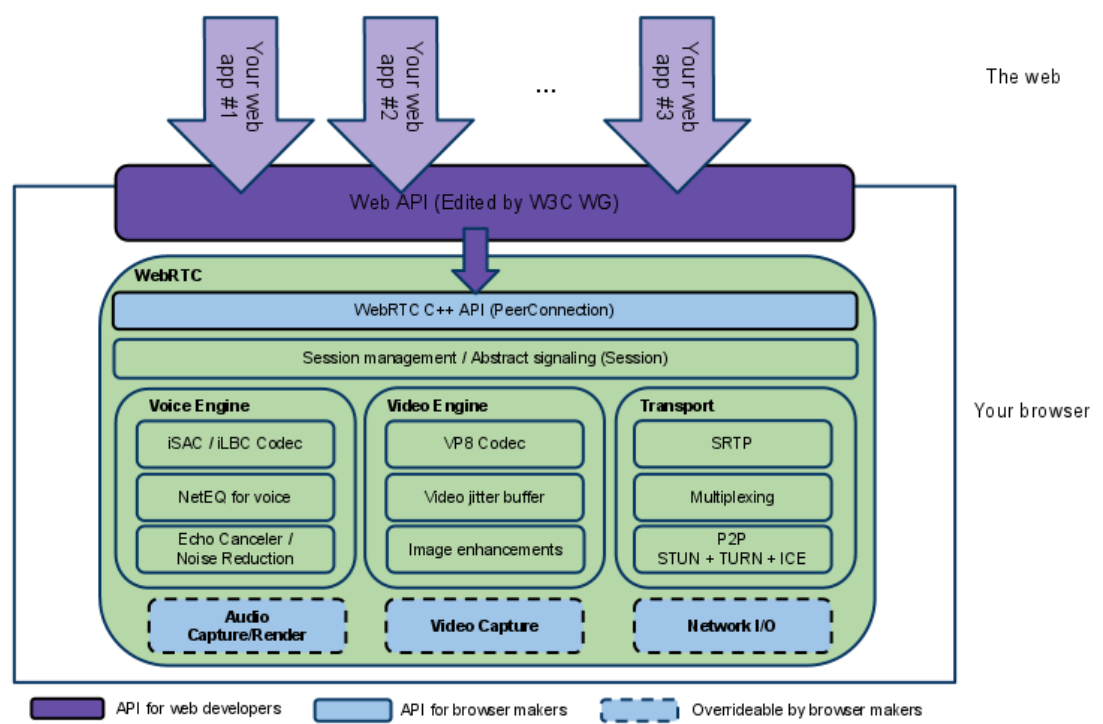
Εικόνα 1 Αρχιτεκτονική JSEP

Μόλις ολοκληρωθεί με επιτυχία η διαδικασία σηματοδότησης, τα δεδομένα μπορούν να μεταδοθούν απευθείας από ομότιμο σε ομότιμο, μεταξύ του καλούντος και του καλούμενου - ή εάν αυτό αποτύχει, μέσω ενός ενδιάμεσου διακομιστή αναμετάδοσης (περισσότερα σχετικά με αυτόν παρακάτω). Η ροή δεδομένων είναι δουλειά του `RTCPeerConnection`.

1.3 RTCPeerConnection

Το `RTCPeerConnection` είναι το στοιχείο WebRTC που χειρίζεται σταθερή και αποτελεσματική επικοινωνία ροής δεδομένων μεταξύ ομότιμων.

Ακολουθεί ένα διάγραμμα αρχιτεκτονικής WebRTC που δείχνει το ρόλο του `RTCPeerConnection`.



Εικόνα 2 Αρχιτεκτονική WebRTC

Από την άποψη της JavaScript, το κύριο πράγμα που πρέπει να κατανοήσουμε από αυτό το διάγραμμα είναι ότι το `RTCPeerConnection` προστατεύει τους προγραμματιστές ιστού από τις μυριάδες πολυπλοκότητες που κρύβονται από κάτω. Οι κωδικοποιητές και τα πρωτόκολλα που χρησιμοποιούνται από το WebRTC κάνουν τεράστια δουλειά για να καταστήσουν δυνατή την επικοινωνία σε πραγματικό χρόνο, ακόμη και σε αναξιόπιστα δίκτυα [1]:

- απόκρυψη πακέτων
- ακύρωση ηχούς
- προσαρμοστικότητα εύρους ζώνης
- δυναμικό jitter buffering
- automatic gain control (AGC)
- μείωση θορύβου και καταστολή

1.4 RTCPeerConnection χωρίς διακομιστές

Αυτό δεν αποτελεί τίποτα πολύ χρήσιμο - ο καλών και ο καλούμενος βρίσκονται στην ίδια σελίδα - αλλά καθιστά τη λειτουργία του `RTCPeerConnection` API λίγο πιο ξεκάθαρη, καθώς τα αντικείμενα `RTCPeerConnection` στη σελίδα μπορούν να ανταλλάσσουν δεδομένα και μηνύματα απευθείας χωρίς να χρειάζεται να χρησιμοποιούν ενδιάμεσο μηχανισμό σηματοδότησης.

1.5 RTCPeerConnection με διακομιστές

Στον πραγματικό κόσμο, το WebRTC χρειάζεται διακομιστές, όσο απλοί και να είναι, ώστε να συμβούν τα εξής:

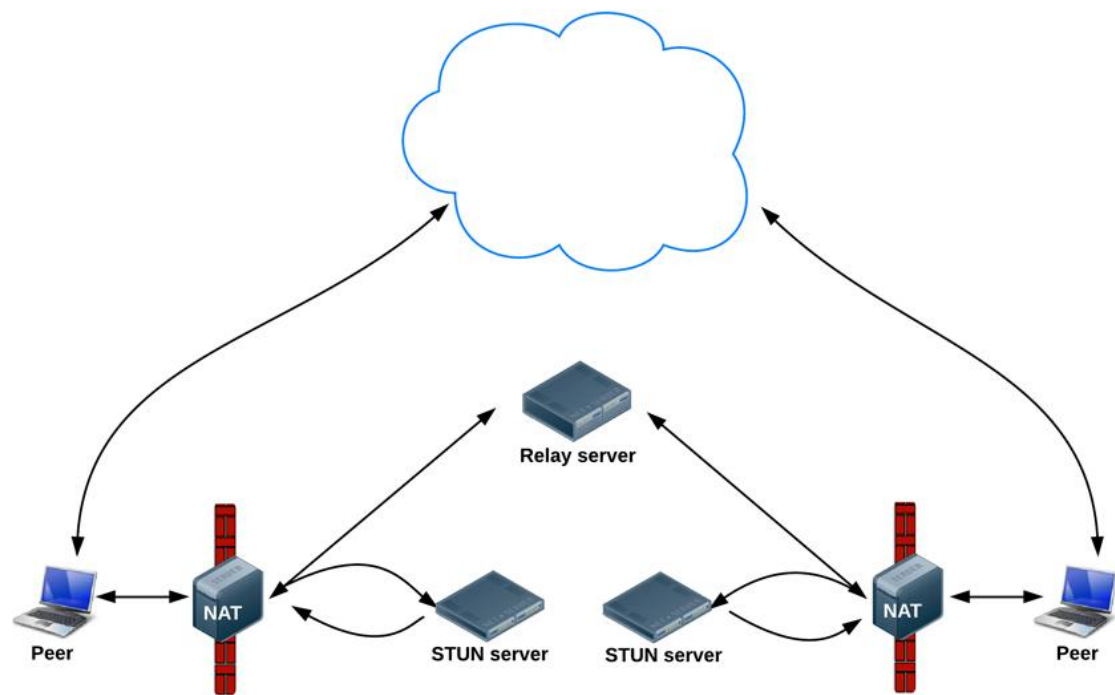
- Οι χρήστες ανακαλύπτουν ο ένας τον άλλον και ανταλλάσσουν λεπτομέρειες «πραγματικού κόσμου» όπως ονόματα.
- Οι εφαρμογές πελατών WebRTC (peers) ανταλλάσσουν πληροφορίες δικτύου.
- Ομότιμοι χρήστες ανταλλάσσουν δεδομένα σχετικά με μέσα όπως μορφή βίντεο και ανάλυση.
- Οι εφαρμογές πελατών WebRTC διασχίζουν πύλες NAT και τείχη προστασίας.

Με άλλα λόγια, το WebRTC χρειάζεται τέσσερις τύπους λειτουργιών από την πλευρά του διακομιστή:

- Ανακάλυψη και επικοινωνία χρήστη.
- Σηματοδότηση.
- Διαδρομή NAT / τείχους προστασίας.
- Διακομιστές αναμετάδοσης σε περίπτωση αποτυχίας επικοινωνίας peer-to-peer.

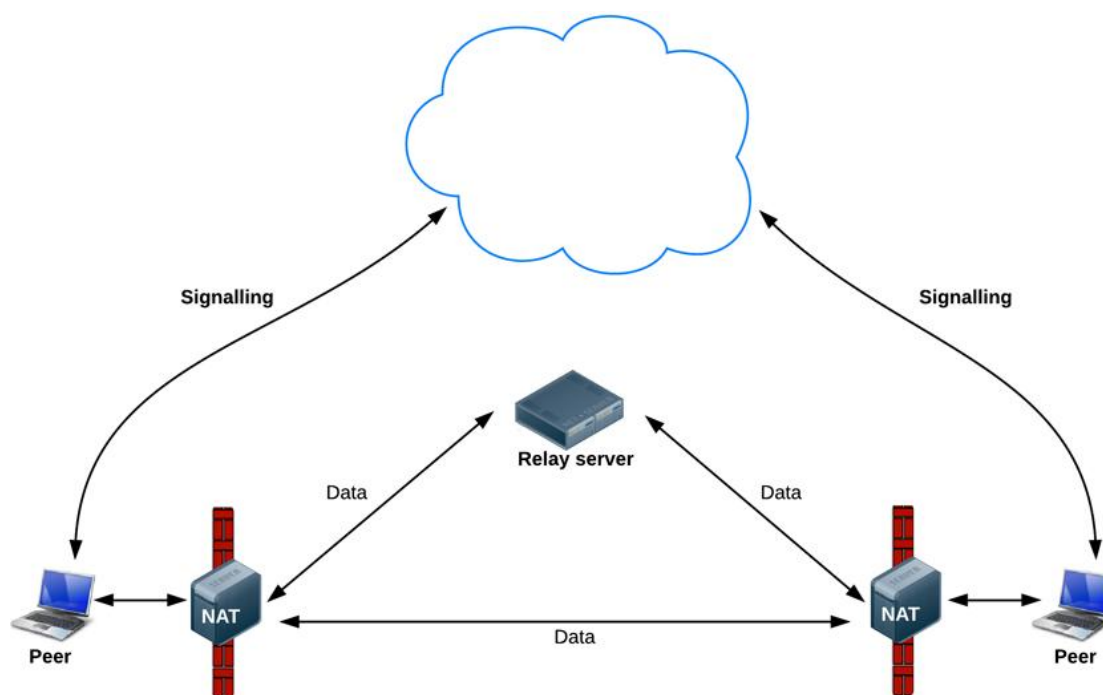
Αρκεί να πούμε ότι το πρωτόκολλο STUN (STUN σημαίνει Session Traversal Utilities για το NAT και παρέχει στο αιτούν τελικό σημείο τη δημόσια διεύθυνση IP του [2]) και η επέκτασή του TURN (TURN σημαίνει Traversal Using Relay NAT [2]), χρησιμοποιούνται από το πλαίσιο ICE για να επιτρέψουν στο RTCPeerConnection να αντιμετωπίσει το NAT traversal και άλλα δικτυακά προβλήματα που εμποδίζουν την αποκατάσταση μιας σύνδεσης μεταξύ ομοτίμων με την πρώτη προσπάθεια.

Το ICE είναι ένα πλαίσιο για τη σύνδεση δύο μερών, όπως δύο πελάτες συνομιλίας μέσω βίντεο. Αρχικά, το ICE προσπαθεί να συνδέσει τα δύο μέρη απευθείας, με τη χαμηλότερη δυνατή καθυστέρηση, μέσω UDP. Σε αυτήν τη διαδικασία, οι διακομιστές STUN έχουν μία μόνο εργασία: να επιτρέψουν σε έναν ομότιμο πίσω από ένα NAT να βρει τη δημόσια διεύθυνση και τη θύρα του.



Εικόνα 3 Εύρεση υποψηφίων σύνδεσης

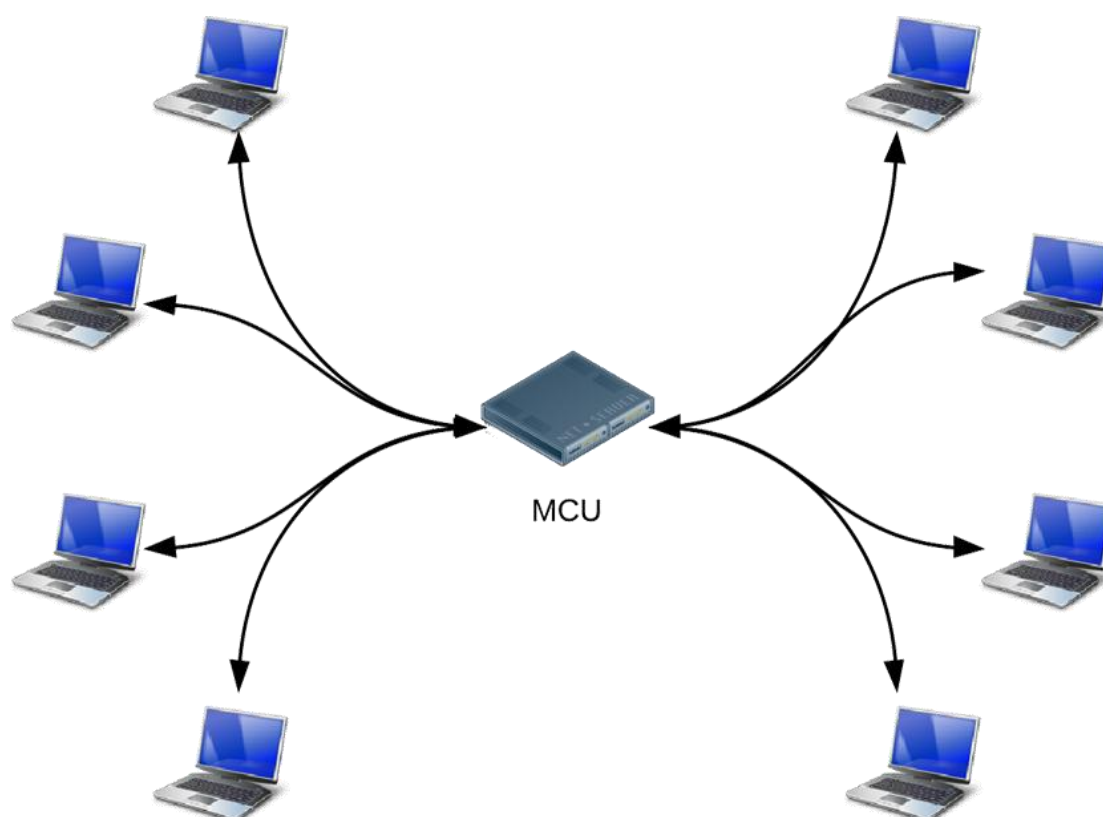
Εάν το UDP αποτύχει, το ICE δοκιμάζει το TCP. Εάν η άμεση σύνδεση αποτύχει - συγκεκριμένα, λόγω ενός εταιρικού NAT traversal και firewalls - το ICE χρησιμοποιεί έναν ενδιάμεσο (relay) διακομιστή TURN. Με άλλα λόγια, το ICE θα χρησιμοποιήσει πρώτα το STUN με UDP για άμεση σύνδεση ομότιμων και, εάν αποτύχει, θα επιστρέψει σε έναν διακομιστή αναμετάδοσης TURN. Η έκφραση «εύρεση υποψηφίων» αναφέρεται στη διαδικασία εύρεσης διεπαφών δικτύου και θυρών.



Εικόνα 4 Διαδρομές δεδομένων WebRTC

1.6 Τοπολογίες δικτύου

Το WebRTC, όπως αυτή τη στιγμή εφαρμόζεται, υποστηρίζει μόνο επικοινωνία one-to-one, αλλά θα μπορούσε να χρησιμοποιηθεί σε πιο περίπλοκα σενάρια δικτύου: για παράδειγμα, με πολλαπλούς ομότιμους ομιλητές να επικοινωνούν μεταξύ τους απευθείας, peer-to-peer ή μέσω μονάδας ελέγχου πολλαπλών σημείων (MultipointControlUnit-MCU) , ένας διακομιστής που μπορεί να χειριστεί μεγάλο αριθμό συμμετεχόντων, να κάνει επιλεκτική προώθηση ροής καθώς και μίξη ή εγγραφή ήχου και βίντεο:



Εικόνα 5 Παράδειγμα τοπολογίας μονάδας ελέγχου πολλαπλών σημείων

Πολλές υπάρχουσες εφαρμογές WebRTC επιδεικνύουν μόνο επικοινωνία μεταξύ προγραμμάτων περιήγησης ιστού, αλλά οι διακομιστές πύλης μπορούν να επιτρέψουν σε μια εφαρμογή WebRTC που εκτελείται σε ένα πρόγραμμα περιήγησης να αλληλεπιδρά με συσκευές όπως τηλέφωνα (γνωστός και ως PSTN) και με συστήματα VOIP. Τον Μάιο του 2012, η Doubango Telecom έκανε ανοιχτό λογισμικό τον πελάτη sipml5 SIP , που δημιουργήθηκε με WebRTC και WebSocket, το οποίο (μεταξύ άλλων πιθανών χρήσεων) επιτρέπει κλήσεις βίντεο μεταξύ προγραμμάτων περιήγησης και εφαρμογών που εκτελούνται σε iOS ή Android. Στο Google I / O, ο Tethr και ο Trope επέδειξαν ένα πλαίσιο για επικοινωνίες καταστροφών «σε ένα χαρτοφύλακα», χρησιμοποιώντας ένα κελί OpenBTS για την ενεργοποίηση των επικοινωνιών μεταξύ κινητών τηλεφώνων και υπολογιστών μέσω WebRTC. Με άλλα λόγια αυτό σημαίνει τηλεφωνική επικοινωνία χωρίς μεταφορέα.



Εικόνα 6 Tethr/Tropo: Επικοινωνίες καταστροφών σε ένα χαρτοφύλακα

1.7 RTC Data Channel

Εκτός από τον ήχο και το βίντεο, το WebRTC υποστηρίζει επικοινωνία σε πραγματικό χρόνο και για άλλους τύπους δεδομένων.

Το RTCDataChannel API επιτρέπει την ομότιμη ανταλλαγή αυθαίρετων δεδομένων, με χαμηλό λανθάνοντα χρόνο και υψηλή απόδοση.

Υπάρχουν πολλές πιθανές περιπτώσεις χρήσης για το API, όπως:

- Παιχνίδια
- Εφαρμογές απομακρυσμένης επιφάνειας εργασίας
- Συνομιλία κειμένου σε πραγματικό χρόνο
- Μεταφορά αρχείων
- Αποκεντρωμένα δίκτυα
- Το API διαθέτει πολλές δυνατότητες για να αξιοποιήσει στο έπακρο το RTCPeerConnection και να επιτρέψει ισχυρή και ευέλικτη επικοινωνία peer-to-peer:
- Αξιοποίηση της ρύθμισης περιόδου λειτουργίας RTCPeerConnection.
- Πολλαπλά ταυτόχρονα κανάλια, με προτεραιότητα.
- Αξιόπιστη και αναξιόπιστη σημασιολογία παράδοσης.
- Ενσωματωμένη ασφάλεια (DTLS) και έλεγχος συμφόρησης.
- Δυνατότητα χρήσης με ή χωρίς ήχο ή βίντεο.

Η επικοινωνία πραγματοποιείται απευθείας μεταξύ των προγραμμάτων περιήγησης, οπότε το RTCDataChannel μπορεί να είναι πολύ πιο γρήγορο από το WebSocket, ακόμα και αν απαιτείται ένας διακομιστής αναμετάδοσης (TURN), όταν το «άνοιγμα τρύπας» για την αντιμετώπιση τείχους προστασίας και NAT αποτύχει.

Το RTCDataChannel είναι διαθέσιμο σε Chrome, Safari, Firefox, Opera και Samsung Internet. Το παιχνίδι Cube Slam χρησιμοποιεί το API για να επικοινωνήσει την κατάσταση του παιχνιδιού. Η καινοτόμος πλατφόρμα Sharefest επέτρεψε την κοινή χρήση αρχείων μέσω RTCDataChannel και το peerCDN προσέφερε μια ματιά για το πώς το WebRTC θα μπορούσε να επιτρέψει την ομαδική κατανομή περιεχομένου.

1.8 Ασφάλεια

Υπάρχουν διάφοροι τρόποι με τους οποίους μια εφαρμογή επικοινωνίας σε πραγματικό χρόνο μπορεί να θέσει σε κίνδυνο την ασφάλεια. Για παράδειγμα:

- Μη κρυπτογραφημένα μέσα ή δεδομένα ενδέχεται να υποκλέπτονται κατά τη διαδρομή μεταξύ προγραμμάτων περιήγησης ή μεταξύ προγράμματος περιήγησης και διακομιστή.
- Μια εφαρμογή ενδέχεται να εγγράψει και να διανείμει βίντεο ή ήχο χωρίς να γνωρίζει ο χρήστης.
- Το κακόβουλο λογισμικό ή οι ιοί ενδέχεται να εγκατασταθούν παράλληλα με ένα φαινομενικά αβλαβές πρόσθετο ή εφαρμογή.

Το WebRTC έχει πολλές δυνατότητες για την αποφυγή αυτών των προβλημάτων:

- Οι υλοποιήσεις WebRTC χρησιμοποιούν ασφαλή πρωτόκολλα όπως DTLS και SRTP.
- Η κρυπτογράφηση είναι υποχρεωτική για όλα τα στοιχεία WebRTC, συμπεριλαμβανομένων των μηχανισμών σηματοδότησης.
- Το WebRTC δεν είναι πρόσθετο: τα στοιχεία του εκτελούνται στο περιβάλλον δοκιμών του προγράμματος περιήγησης και όχι σε ξεχωριστή διαδικασία, τα στοιχεία δεν απαιτούν ξεχωριστή εγκατάσταση και ενημερώνονται κάθε φορά που ενημερώνεται το πρόγραμμα περιήγησης.
- Η πρόσβαση στην κάμερα και το μικρόφωνο πρέπει να παρέχεται ρητά και, όταν λειτουργεί η κάμερα ή το μικρόφωνο, αυτό φαίνεται σαφώς από τη διεπαφή χρήστη.

2.STUN, TURN ΚΑΙ ΣΗΜΑΤΟΔΟΤΗΣΗ

Η σηματοδότηση είναι η διαδικασία συντονισμού της επικοινωνίας. Προκειμένου μια εφαρμογή WebRTC να δημιουργήσει μια «κλήση», οι πελάτες της πρέπει να ανταλλάσσουν πληροφορίες[3]:

- Μηνύματα ελέγχου συνεδρίας που χρησιμοποιούνται για το άνοιγμα ή το κλείσιμο της επικοινωνίας.
- Μηνύματα σφάλματος.
- Μεταδεδομένα μέσω των οποίων κωδικοποιητές και ρυθμίσεις κωδικοποιητή, εύρος ζώνης και τύποι μέσων.
- Βασικά δεδομένα, που χρησιμοποιούνται για τη δημιουργία ασφαλών συνδέσεων.
- Δεδομένα δικτύου, όπως η διεύθυνση IP του κεντρικού υπολογιστή και η θύρα, όπως φαίνεται από τον έξω κόσμο.

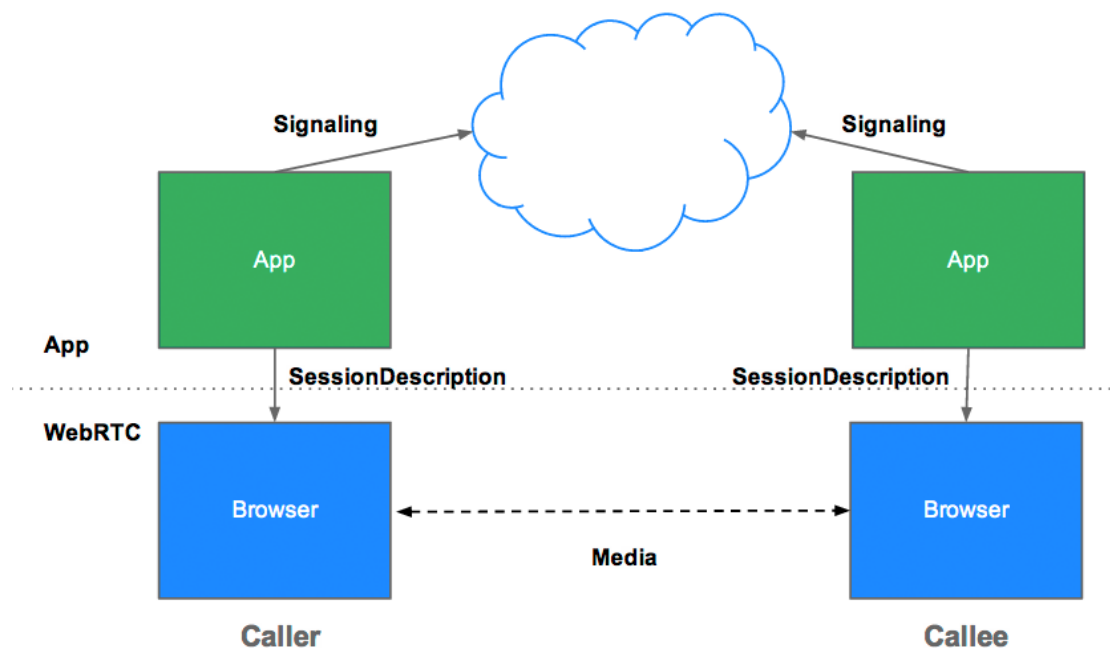
Αυτή η διαδικασία σηματοδότησης χρειάζεται έναν τρόπο για τους πελάτες να μεταφέρουν μηνύματα μπρος-πίσω. Αυτός ο μηχανισμός δεν εφαρμόζεται από τα API WebRTC, πρέπει να το δημιουργήσουμε μόνοι μας. Παρακάτω περιγράφουμε μερικούς τρόπους δημιουργίας μιας υπηρεσίας σηματοδότησης. Πρώτα όμως ας αναφέρουμε κάποια εισαγωγικά.

2.1 Οι λόγοι που η σηματοδότηση δεν ορίζεται από το webrtc

Για να αποφευχθεί ο πλεονασμός και να μεγιστοποιηθεί η συμβατότητα με καθιερωμένες τεχνολογίες, οι μέθοδοι σηματοδότησης και τα πρωτόκολλα δεν καθορίζονται από τα πρότυπα WebRTC. Αυτή η προσέγγιση περιγράφεται από το JSEP, το JavaScript Session Establishment Protocol.

Η σκέψη πίσω από τη ρύθμιση κλήσεων WebRTC ήταν ο πλήρης προσδιορισμός και ο έλεγχος του επιπέδου μέσων, αλλά η ανάθεση του επιπέδου σηματοδότησης όσο το δυνατόν περισσότερο στην εφαρμογή. Η λογική είναι ότι διαφορετικές εφαρμογές μπορεί να προτιμούν να χρησιμοποιούν διαφορετικά πρωτόκολλα, όπως τα υπάρχοντα πρωτόκολλα σηματοδότησης κλήσεων SIP(Session Initiation Protocol)[7] ή JingleProtocol[7], ή κάτι προσαρμοσμένο στη συγκεκριμένη εφαρμογή, ίσως για μια νέα περίπτωση χρήσης. Σε αυτήν την προσέγγιση, οι βασικές πληροφορίες που πρέπει να ανταλλάσσονται είναι η περιγραφή συνεδρίας πολυμέσων, η οποία καθορίζει τις απαραίτητες πληροφορίες μεταφοράς και διαμόρφωσης μέσων που είναι απαραίτητες για τη δημιουργία του επιπέδου.

Η αρχιτεκτονική του JSEP αποτρέπει επίσης ένα πρόγραμμα περιήγησης να χρειάζεται να αποθηκεύει κατάσταση, δηλαδή να λειτουργεί ως μηχανή κατάστασης σηματοδότησης. Αυτό θα ήταν προβληματικό εάν, για παράδειγμα, τα δεδομένα σηματοδότησης χάνονταν κάθε φορά που μια σελίδα φορτωνόταν ξανά. Αντ' αυτού, η κατάσταση σηματοδότησης μπορεί να αποθηκευτεί σε έναν διακομιστή.



Εικόνα 7 Αρχιτεκτονική JSEP

Το JSEP απαιτεί την ανταλλαγή μεταξύ ομοτίμων(peers), προσφοράς και απάντησης, τα μεταδεδομένα μέσω των οποίων αναφέρονται παραπάνω. Οι προσφορές και οι απαντήσεις κοινοποιούνται σε μορφή πρωτοκόλλου περιγραφής περιόδου σύνδεσης (SDP, SessionDescriptionProtocol[7]).

2.2 RTCPeerConnection και σηματοδότηση

Το RTCPeerConnection είναι το API που χρησιμοποιείται από εφαρμογές WebRTC για τη δημιουργία σύνδεσης μεταξύ ομοτίμων και την επικοινωνία ήχου και βίντεο.

Για να ξεκινήσει αυτή η διαδικασία, το RTCPeerConnection έχει δύο εργασίες:

- Εξακρίβωση τοπικών συνθηκών μέσω, όπως δυνατότητες ανάλυσης και κωδικοποιητή. Αυτά είναι τα μεταδεδομένα που χρησιμοποιούνται για το μηχανισμό προσφοράς και απάντησης.
- Λήψη πιθανών διευθύνσεων δικτύου για τον κεντρικό υπολογιστή της εφαρμογής, γνωστές ως υποψήφιοι (candidates) .

Μόλις εξακριβωθούν αυτά τα τοπικά δεδομένα, πρέπει να ανταλλάσσονται μέσω ενός μηχανισμού σηματοδότησης με τον απομακρυσμένο ομότιμο.

Έστω ότι η Alice προσπαθεί να καλέσει την Eve . Παρακάτω είναι ο πλήρης μηχανισμός προσφοράς / απάντησης με όλες τις λεπτομέρειες:

1. Η Alice δημιουργεί ένα αντικείμενο RTCPeerConnection.
2. Η Alice δημιουργεί μια προσφορά (περιγραφή περιόδου λειτουργίας SDP) με τη μέθοδο RTCPeerConnection createOffer () .
3. Η Alice καλεί το setLocalDescription () με την προσφορά.

4. Η Alice μετατρέπει σε συμβολοσειρά την προσφορά και χρησιμοποιεί έναν μηχανισμό σηματοδότησης για να την στείλει στην Eve.
5. Η Eve καλεί το `setRemoteDescription()` με την προσφορά της Alice, έτσι ώστε το `RTCPeerConnection` της να γνωρίζει για τη ρύθμιση της Alice.
6. Η Eve καλεί την `createAnswer()` και η επιτυχημένη επιστροφή κλήσης της, μεταβιβάζεται σε μια περιγραφή τοπικής περιόδου σύνδεσης: Αυτή θα είναι η απάντηση της Eve .
7. Η Eve ορίζει την απάντησή της ως τοπική περιγραφή καλώντας τη μέθοδο `setLocalDescription()` .
8. Στη συνέχεια, η Eve χρησιμοποιεί τον μηχανισμό σηματοδότησης για να στείλει την απάντησή της που την έχει μετατρέψει σε συμβολοσειρά, στην Alice.
9. Η Alice ορίζει την απάντηση της Eve ως την απομακρυσμένη περιγραφή περιόδου λειτουργίας χρησιμοποιώντας τη μέθοδο `setRemoteDescription()` .

Η Alice και η Eve πρέπει επίσης να ανταλλάξουν πληροφορίες δικτύου. Η έκφραση «εύρεση υποψηφίων» αναφέρεται στη διαδικασία εύρεσης διεπαφών δικτύου και θυρών χρησιμοποιώντας το πλαίσιο ICE .

1. Η Alice δημιουργεί ένα αντικείμενο `RTCPeerConnection` με έναν handler (χειριστή) `onicecandidate` .
2. Ο χειριστής καλείται όταν οι υποψήφιοι του δικτύου γίνουν διαθέσιμοι.
3. Στο χειριστή, η Alice στέλνει δεδομένα υποψηφίων (σαν συμβολοσειρές) στην Eve, μέσω του καναλιού σηματοδότησης.
4. Όταν η Eve λαμβάνει ένα υποψήφιο μήνυμα από την Alice, καλεί τη μέθοδο `addIceCandidate()` για να προσθέσει τον υποψήφιο στην απομακρυσμένη περιγραφή.

Το JSEP υποστηρίζει το ICE Candidate Trickling[8] , το οποίο επιτρέπει στον καλούντα να παρέχει σταδιακά υποψηφίους στον καλούμενο μετά την αρχική προσφορά και στον καλούμενο να αρχίσει να ενεργεί στην κλήση και να δημιουργεί μια σύνδεση χωρίς να περιμένει να φτάσουν όλοι οι υποψήφιοι.

2.3 Δημιουργία υπηρεσίας σηματοδότησης

Ας επαναλάβουμε ότι τα πρωτόκολλα και οι μηχανισμοί σηματοδότησης δεν καθορίζονται από τα πρότυπα WebRTC. Ό, τι κι αν επιλέξουμε, θα χρειαστούμε έναν ενδιάμεσο διακομιστή για την ανταλλαγή μηνυμάτων σηματοδότησης και δεδομένων εφαρμογής μεταξύ πελατών. Δυστυχώς, μια εφαρμογή ιστού δεν μπορεί απλά να φωνάζει στο Διαδίκτυο «Συνδέστε με τον φίλο μου!»

Ευτυχώς, τα μηνύματα σηματοδότησης είναι μικρά και ανταλλάσσονται κυρίως στην αρχή μιας κλήσης. Κατά τη δοκιμή με `appr.tc` , βρήκαμε ότι, για μια συνεδρία συνομιλίας μέσω βίντεο, η υπηρεσία σηματοδότησης χειρίστηκε συνολικά περίπου 30-45 μηνύματα , με συνολικό μέγεθος για όλα τα μηνύματα περίπου 10kB.

Εκτός από το ότι είναι σχετικά μη απαιτητικές όσον αφορά το εύρος ζώνης, οι υπηρεσίες σηματοδότησης WebRTC δεν απαιτούν πολύ επεξεργασία ούτε καταναλώνουν πολύ μνήμη, καθώς χρειάζεται μόνο να μεταδίδουν μηνύματα

και να διατηρούν ένα μικρό αριθμό δεδομένων κατάστασης περιόδου σύνδεσης (όπως οι πελάτες που είναι συνδεδεμένοι).

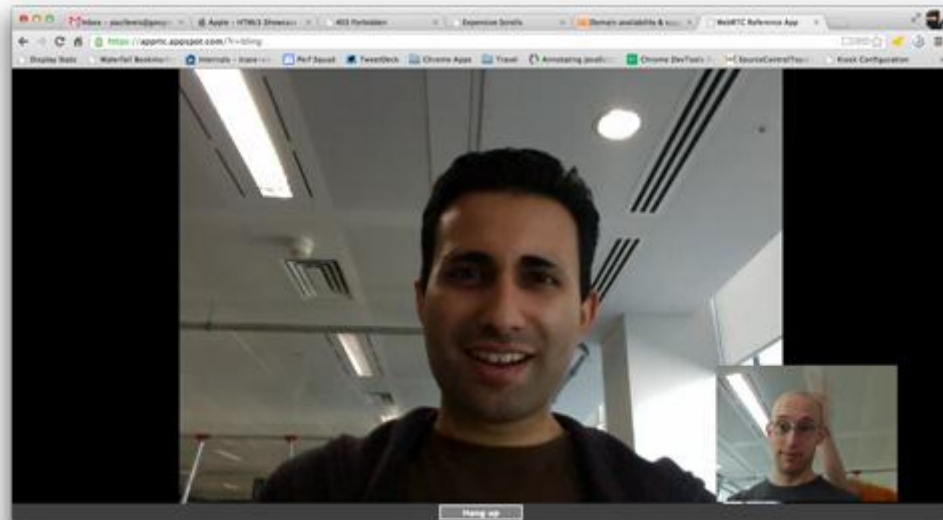
Ο μηχανισμός σηματοδοσίας που χρησιμοποιείται για την ανταλλαγή μεταδεδομένων συνεδρίας μπορεί επίσης να χρησιμοποιηθεί για την επικοινωνία δεδομένων εφαρμογής, αφού είναι απλώς μια υπηρεσία ανταλλαγής μηνυμάτων.

Μια υπηρεσία μηνυμάτων για σηματοδότηση πρέπει να είναι αμφίδρομη, δηλαδή από πελάτη σε διακομιστή και από διακομιστή σε πελάτη. Η αμφίδρομη επικοινωνία έρχεται σε αντίθεση με το μοντέλο αίτησης-απόκρισης πελάτη-διακομιστή HTTP, αλλά έχουν αναπτυχθεί διάφορες τεχνικές εδώ και πολλά χρόνια, προκειμένου να προωθηθούν δεδομένα από μια υπηρεσία που εκτελείται σε έναν διακομιστή ιστού σε μια εφαρμογή ιστού που εκτελείται σε ένα πρόγραμμα περιήγησης.

Το EventSource API έχει εφαρμοστεί ευρέως . Αυτό επιτρέπει «συμβάντα που αποστέλλονται από διακομιστή» ή αλλιώς server-sent, δηλαδή δεδομένα που αποστέλλονται από διακομιστή ιστού σε πρόγραμμα-πελάτη προγράμματος περιήγησης μέσω HTTP. Το EventSource έχει σχεδιαστεί για ανταλλαγή μηνυμάτων μονής κατεύθυνσης, αλλά μπορεί να χρησιμοποιηθεί σε συνδυασμό με το XHR(XMLHttpRequest) [7] για τη δημιουργία υπηρεσίας ανταλλαγής μηνυμάτων σηματοδοσίας, μια υπηρεσία σηματοδοσίας μεταδίδει ένα μήνυμα από έναν καλούντα, που παραδίδεται με αίτημα XHR, ωθώντας το μέσω EventSource στον καλούμενο .

Το WebSocket είναι μια πιο φυσική λύση, σχεδιασμένη για πλήρη αμφίδρομη επικοινωνία πελάτη-διακομιστή (τα μηνύματα μπορούν να ρέουν ταυτόχρονα και στις δύο κατευθύνσεις) [4]. Ένα πλεονέκτημα μιας υπηρεσίας σηματοδοσίας που δημιουργήθηκε με συμβάντα WebSocket ή Server-Sent (EventSource) είναι ότι το back-end για αυτά τα API μπορεί να εφαρμοστεί σε μια ποικιλία από web frameworks, κοινών στα περισσότερα πακέτα φιλοξενίας ιστοσελίδων, για γλώσσες όπως PHP, Python και Ruby.

Περίπου τα τρία τέταρτα των προγραμμάτων περιήγησης υποστηρίζουν το WebSocket και το πιο σημαντικό είναι ότι όλα τα προγράμματα περιήγησης που υποστηρίζουν WebRTC υποστηρίζουν επίσης το WebSocket, τόσο σε επιτραπέζιους υπολογιστές όσο και σε κινητά. Το TLS(Transport Layer Security) [7] πρέπει να χρησιμοποιείται για όλες τις συνδέσεις, για να διασφαλίζεται ότι τα μηνύματα δεν μπορούν να υποκλαπούν, όντας μη κρυπτογραφημένα, καθώς και για τη μείωση των προβλημάτων κατά τη διέλευση μέσω διακομιστή μεσολάβησης .



Εικόνα 8 Η υπηρεσία appr.tc

2.4 Ασφάλεια σηματοδοσίας

Η κρυπτογράφηση είναι υποχρεωτική για όλα τα στοιχεία WebRTC, ωστόσο οι μηχανισμοί σηματοδότησης δεν καθορίζονται από τα πρότυπα WebRTC, οπότε εξαρτάται από εμάς να κάνουμε ασφαλή τη σηματοδοσία. Εάν ένας εισβολέας καταφέρει να εισβάλει στη σηματοδοσία, μπορεί να σταματήσει τις συνεδρίες, να ανακατευθύνει τις συνδέσεις και να ηχογραφήσει, να αλλάξει ή να εισάγει περιεχόμενο.

Ο πιο σημαντικός παράγοντας για την εξασφάλιση της σηματοδοσίας είναι η χρήση ασφαλών πρωτοκόλλων, HTTPS και WSS (δηλ. TLS), τα οποία διασφαλίζουν ότι τα μηνύματα δεν μπορούν να υποστούν κρυπτογράφηση. Προσέχουμε επίσης να μην μεταδίδουμε μηνύματα σηματοδοσίας με τρόπο που να έχουν πρόσβαση άλλοι καλούντες χρησιμοποιώντας τον ίδιο διακομιστή σηματοδοσίας.

Για να εξασφαλίσουμε μια εφαρμογή WebRTC, είναι απολύτως επιτακτική ανάγκη, η σηματοδοσία να χρησιμοποιεί TLS .

2.5 Χρησιμότητα ICE

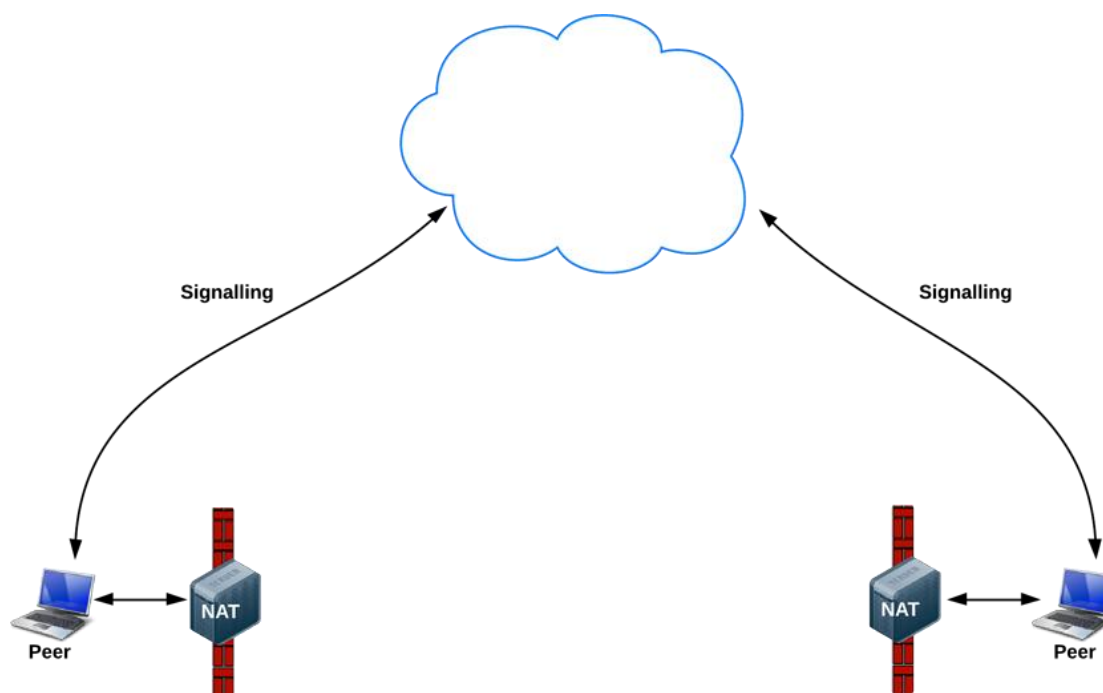
Για τη σηματοδοσία μεταδεδομένων, οι εφαρμογές WebRTC χρησιμοποιούν έναν ενδιάμεσο διακομιστή, αλλά για την πραγματική ροή μέσω και δεδομένων μόλις δημιουργηθεί μια περίοδος σύνδεσης, το `RTCPeerConnection` επιχειρεί να συνδέσει απευθείας τους πελάτες: peer to peer.

Σε έναν απλούστερο κόσμο, κάθε τελικό σημείο WebRTC θα έχει μια μοναδική διεύθυνση που θα μπορούσε να ανταλλάξει με άλλους ομότιμους για να επικοινωνήσει απευθείας.



Εικόνα 9 Ένας κόσμος χωρίς NAT και τείχη προστασίας

Στην πραγματικότητα, οι περισσότερες συσκευές ζουν πίσω από ένα ή περισσότερα επίπεδα NAT, μερικές διαθέτουν λογισμικό προστασίας από ιούς που αποκλείει συγκεκριμένες θύρες και πρωτόκολλα και πολλές βρίσκονται πίσω από διακομιστές μεσολάβησης(proxy servers) και εταιρικά τείχη προστασίας. Ένα τείχος προστασίας (firewall) και NAT μπορούν, στην πραγματικότητα, να εφαρμοστούν από την ίδια συσκευή, όπως ένας οικιακός δρομολογητής WiFi.



Εικόνα 10 Ο πραγματικός κόσμος

Οι εφαρμογές WebRTC μπορούν να χρησιμοποιήσουν το πλαίσιο ICE για να ξεπεράσουν τις πολυπλοκότητες της πραγματικής δικτύωσης. Για να συμβεί αυτό, η εφαρμογή μας πρέπει να μεταφέρει τις διευθύνσεις URL διακομιστή ICE στο `RTCPeerConnection`, όπως περιγράφεται παρακάτω.

Το ICE προσπαθεί να βρει την καλύτερη διαδρομή για να συνδέσει τους ομότιμους. Δοκιμάζει όλες τις δυνατότητες παράλληλα και επιλέγει την πιο

αποτελεσματική περίπτωση που λειτουργεί. Το ICE προσπαθεί πρώτα να πραγματοποιήσει μια σύνδεση χρησιμοποιώντας τη διεύθυνση κεντρικού υπολογιστή που λαμβάνεται από το λειτουργικό σύστημα και την κάρτα δικτύου μιας συσκευής, εάν αυτό αποτύχει (το οποίο θα κάνει για συσκευές πίσω από NAT), το ICE αποκτά μια εξωτερική διεύθυνση χρησιμοποιώντας έναν διακομιστή STUN και εάν και αυτό αποτύχει, η κίνηση δρομολογείται μέσω ενός διακομιστή μεσολάβησης TURN.

Με άλλα λόγια:

- Ένας διακομιστής STUN χρησιμοποιείται για τη λήψη εξωτερικής διεύθυνσης δικτύου
- Οι διακομιστές TURN χρησιμοποιούνται για την αναμετάδοση της κυκλοφορίας εάν αποτύχει η άμεση (peer to peer) σύνδεση.

Κάθε διακομιστής TURN υποστηρίζει STUN, δηλαδή ένας διακομιστής TURN είναι ένας διακομιστής STUN με ενσωματωμένη πρόσθετη λειτουργία αναμετάδοσης. Το ICE αντιμετωπίζει επίσης τις πολυπλοκότητες των ρυθμίσεων NAT, το «άνοιγμα τρύπας» στο NAT μπορεί να απαιτεί κάτι περισσότερο από μια δημόσια διεύθυνση IP και θύρα.

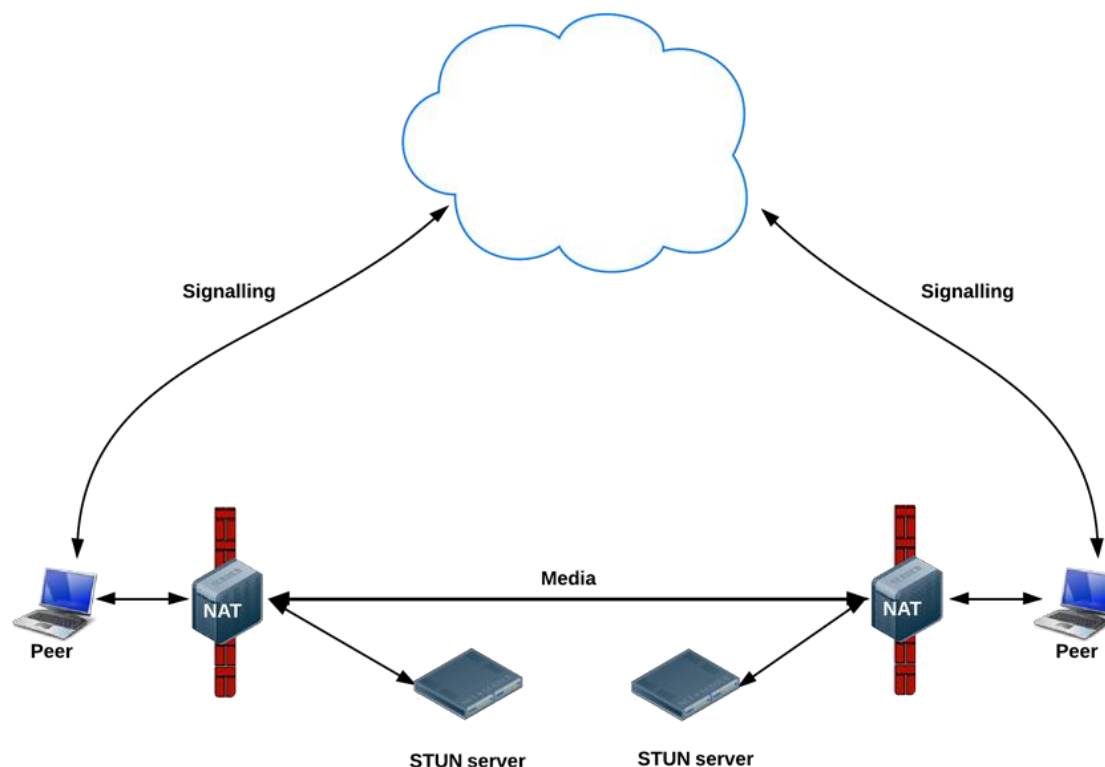
2.6 STUN

Οι NAT παρέχουν σε μια συσκευή μια διεύθυνση IP για χρήση σε ένα ιδιωτικό τοπικό δίκτυο, αλλά αυτή η διεύθυνση δεν μπορεί να χρησιμοποιηθεί εξωτερικά. Χωρίς δημόσια διεύθυνση, δεν υπάρχει τρόπος επικοινωνίας με τους ομότιμους WebRTC [5]. Για να αντιμετωπίσει αυτό το πρόβλημα, το WebRTC χρησιμοποιεί STUN .

Οι διακομιστές STUN ζουν στο δημόσιο Διαδίκτυο και έχουν μια απλή εργασία, να ελέγχουν τη διεύθυνση IP και θύρα ενός εισερχόμενου αιτήματος (από μια εφαρμογή που τρέχει πίσω από ένα NAT) και να στέλνουν τη διεύθυνση πίσω ως απάντηση. Με άλλα λόγια, η εφαρμογή χρησιμοποιεί έναν διακομιστή STUN για να ανακαλύψει την IP και τη θύρα της, από μια δημόσια προοπτική. Αυτή η διαδικασία επιτρέπει σε έναν ομότιμο WebRTC να αποκτήσει μια δημόσια προσβάσιμη διεύθυνση και, στη συνέχεια, να τη μεταβιβάσει σε έναν άλλο ομότιμο μέσω ενός μηχανισμού σηματοδότησης, προκειμένου να δημιουργήσει έναν άμεσο σύνδεσμο. (Στην πράξη, διαφορετικά NAT λειτουργούν με διαφορετικούς τρόπους και μπορεί να υπάρχουν πολλά επίπεδα NAT, αλλά η αρχή παραμένει η ίδια.)

Οι διακομιστές STUN δεν χρειάζεται να κάνουν πολλά ή να θυμούνται πολλά, επομένως οι διακομιστές STUN σχετικά χαμηλών προδιαγραφών μπορούν να χειριστούν μεγάλο αριθμό αιτημάτων.

Οι περισσότερες κλήσεις WebRTC πραγματοποιούν επιτυχώς μια σύνδεση χρησιμοποιώντας το STUN, περίπου το 86%, σύμφωνα με το webrtcstats.com , αν και αυτό μπορεί να είναι μικρότερο για κλήσεις μεταξύ ομοτίμων πίσω από τείχη προστασίας και πιο σύνθετες ρυθμίσεις NAT.



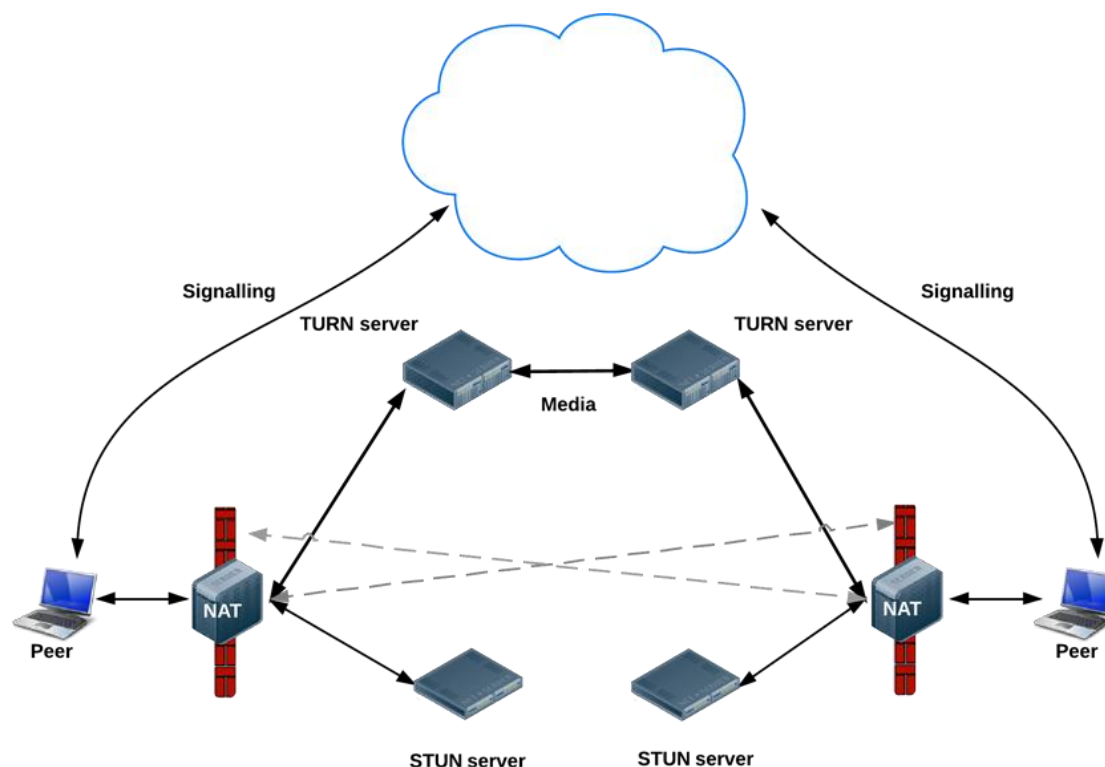
Εικόνα 11 Χρήση διακομιστών STUN για τη λήψη δημόσιας IP και θύρας

2.7 TURN

Το `RTCPeerConnection` προσπαθεί να ρυθμίσει την άμεση επικοινωνία μεταξύ των ομοτίμων μέσω του UDP. Εάν αποτύχει, το `RTCPeerConnection` καταφεύγει στο TCP. Εάν αποτύχει και αυτό οι διακομιστές TURN μπορούν να χρησιμοποιηθούν ως εναλλακτικό μέσο, μεταδίδοντας δεδομένα μεταξύ τελικών σημείων.

Να επισημάνουμε λοιπόν ότι το TURN χρησιμοποιείται για την αναμετάδοση ροής ήχου, βίντεο ή και δεδομένων μεταξύ ομοτίμων, όχι για δεδομένα σηματοδότησης.

Οι διακομιστές TURN έχουν δημόσιες διευθύνσεις, οπότε μπορούν να εντοπιστούν από τους ομότιμους ακόμη και αν οι ομότιμοι βρίσκονται πίσω από τείχη προστασίας ή διακομιστές μεσολάβησης. Οι διακομιστές TURN έχουν μια εννοιολογικά απλή εργασία, την αναμετάδοση μιας ροής, αλλά σε αντίθεση με τους διακομιστές STUN, καταναλώνουν πιο μεγάλο εύρος ζώνης. Με άλλα λόγια, οι διακομιστές TURN πρέπει να είναι καλύτεροι.



Εικόνα 12 Η πλήρης διαδικασία με STUN, TURN και σηματοδότηση

Αυτό το διάγραμμα δείχνει το TURN σε δράση: το καθαρό STUN δεν πέτυχε, έτσι κάθε ομότιμος καταφεύγει στη χρήση ενός διακομιστή TURN.

2.8 Μονάδα Ελέγχου Πολλαπλών Σημείων

Μια καλύτερη επιλογή για μεγάλο αριθμό τελικών σημείων είναι να χρησιμοποιήσουμε μια μονάδα ελέγχου πολλαπλών σημείων (MCU). Αυτός είναι ένας διακομιστής που λειτουργεί ως γέφυρα για τη διανομή πολυμέσων μεταξύ μεγάλου αριθμού συμμετεχόντων. Οι MCU μπορούν να αντιμετωπίσουν διαφορετικές αναλύσεις, κωδικοποιητές και ρυθμούς καρέ σε μια διάσκεψη βίντεο, να χειριστούν τη διακωδικοποίηση, να κάνουν εκλεκτική προώθηση ροής και να αναμίζουν ή να εγγράψουν ήχο και βίντεο. Για κλήσεις πολλών μερών, υπάρχουν ορισμένα ζητήματα που πρέπει να ληφθούν υπόψη, συγκεκριμένα πώς να προβάλλουμε πολλές εισόδους βίντεο και να συνδυάζουμε ήχο από πολλές πηγές. Οι πλατφόρμες cloud προσπαθούν επίσης να βελτιστοποιήσουν τη δρομολόγηση της κίνησης.



Εικόνα 13 Η πίσω όψη ενός Cisco MCU

Είναι δυνατό να αγοράσουμε ένα πλήρες πακέτο υλικού MCU ή να δημιουργήσουμε το δικό μας. Διατίθενται πολλές επιλογές λογισμικού ανοιχτού κώδικα MCU. Για παράδειγμα, το OpenTok έχει Mantis[6] .

2.9 VOIP, τηλέφωνα και μηνύματα

Η τυποποιημένη φύση του WebRTC καθιστά δυνατή τη δημιουργία επικοινωνίας μεταξύ μιας εφαρμογής WebRTC που εκτελείται σε ένα πρόγραμμα περιήγησης και μιας συσκευής ή πλατφόρμας που εκτελείται σε μια άλλη πλατφόρμα επικοινωνίας, όπως ένα τηλέφωνο ή συστήματα τηλεδιάσκεψης.

Το SIP (Session Initiation Protocol) [7] είναι ένα πρωτόκολλο σηματοδότησης που χρησιμοποιείται από VoIP (Voice Over IP) [7] και συστήματα τηλεδιάσκεψης. Για να ενεργοποιηθεί η επικοινωνία μεταξύ μιας εφαρμογής WebRTC και ενός προγράμματος-πελάτη SIP, όπως ένα σύστημα τηλεδιάσκεψης, το WebRTC χρειάζεται διακομιστή μεσολάβησης για τη διαμεσολάβηση της σηματοδότησης. Η σηματοδότηση πρέπει να ρέει μέσω της πύλης, αλλά, μόλις πραγματοποιηθεί η επικοινωνία, η κυκλοφορία SRTP (Secure Realtime Transport Protocol) [7] (βίντεο και ήχος) μπορεί να ρέει απευθείας από ομότιμο σε ομότιμο.

Το PSTN (Public Switched Telephone Network) , το Δημόσιο Τηλεφωνικό Δίκτυο Μεταγωγής, είναι το δίκτυο εναλλαγής κυκλωμάτων όλων των «απλών παλιών» αναλογικών τηλεφώνων. Για κλήσεις μεταξύ εφαρμογών ιστού WebRTC και τηλεφώνων, η κίνηση πρέπει να διέρχεται από πύλη PSTN. Ομοίως, οι εφαρμογές ιστού WebRTC χρειάζονται έναν ενδιάμεσο διακομιστή XMPP (Extensible Messaging and Presence Protocol) για να επικοινωνούν με τελικά σημεία Jingle , όπως πελάτες IM (Instant Messaging) [7]. Το πρωτόκολλο Jingle αναπτύχθηκε από την Google ως επέκταση στο XMPP για την ενεργοποίηση φωνητικών και βίντεο για υπηρεσίες ανταλλαγής μηνυμάτων. Οι τρέχουσες υλοποιήσεις WebRTC βασίζονται στη βιβλιοθήκη C++ libjingle , μια εφαρμογή του Jingle που αναπτύχθηκε αρχικά για το Google Talk.

3.ΚΛΙΜΑΚΩΣΗ

3.1 Εισαγωγή

Μια αυθόρμητη απάντηση στο πόσοι χρήστες χωρούν σε μια συνεδρία, θα ήταν ότι σε μια κλήση WebRTC χωρούν όσοι χρήστες επιθυμούμε. Μπορούμε να εξυπηρετήσουμε από ένα χρήστη έως ένα εκατομμύριο χρήστες σε μια κλήση WebRTC.

Εάν μας ζητηθεί να δημιουργήσουμε μια ομαδική βιντεοκλήση, η τεχνολογία που θα επιλεγεί για το έργο δε μπορεί να είναι άλλη από το WebRTC. Είναι σχεδόν η μόνη εναλλακτική λύση εκεί έξω και σίγουρα αυτή με την καλύτερη σχέση τιμής-απόδοσης. Εδώ είναι η μεγάλη ερώτηση: Πόσους χρήστες μπορούμε να χωρέσουμε σε αυτήν την κλήση WebRTC μιας ομάδας;

Πρέπει να σκεφτούμε το WebRTC ως ένα σύνολο τεχνολογικών δομικών μονάδων που τα συνδυάζουμε και τα ταιριάζουμε όπως θέλουμε. Η εφαρμογή του WebRTC του προγράμματος περιήγησης είναι μόνο ένα δομικό στοιχείο.

Το πιο κοινό δομικό στοιχείο σήμερα στο WebRTC για υποστήριξη ομαδικών βιντεοκλήσεων είναι το SFU (SelectiveForwardingUnit), ένας δρομολογητής πολυμέσων που λαμβάνει ροές πολυμέσων από όλους τους συμμετέχοντες σε μια συνεδρία και αποφασίζει σε ποιον θα δρομολογήσει το εκάστοτε περιεχόμενο.

Αυτό που θέλουμε να πετύχουμε σε αυτή την εργασία, είναι να εξετάσουμε μερικές από τις πτυχές και τις αποφάσεις που θα πρέπει να λάβουμε όταν προσπαθούμε να δημιουργήσουμε εφαρμογές που υποστηρίζουν μεγάλες ομαδικές συνεδρίες βίντεο χρησιμοποιώντας το WebRTC.

3.2 Ανάλυση της πολυπλοκότητας

Το πρώτο βήμα στην εργασία μας θα είναι να αναλύσουμε την πολυπλοκότητα της περίπτωσης που μελετούμε. Με το WebRTC και τις επικοινωνίες βίντεο σε πραγματικό χρόνο γενικά, θα μας απασχολήσουν δύο πράγματα:

1. Η ανάλυση και ο ρυθμός bit που περιμένουμε στην υπηρεσία μας
2. Το πλήθος των ροών της μεμονωμένης περιόδου σύνδεσης

Ας ξεκινήσουμε με ένα παράδειγμα.



Εικόνα 14 Ομαδική κλήση

Ας υποθέσουμε ότι θέλουμε να εκτελέσουμε μια υπηρεσία ομαδικής κλήσης για την επιχείρηση που δραστηριοποιείται σε παγκόσμιο επίπεδο. Οι άνθρωποι θα συμμετάσχουν μαζί σε συνεδρίες εργασίας και σκοπεύουμε να περιορίσουμε τις ομαδικές συνεδρίες σε 4 άτομα. Σίγουρα θα θέλαμε περισσότερα, αλλά ας κρατήσουμε τα πράγματα απλά προς το παρόν. Η παραπάνω εικόνα μας δείχνει πώς θα μοιάζει ένα συνέδριο 4 συμμετεχόντων.

Εάν η ποιότητα που θέλουμε για αυτό το συνέδριο είναι υψηλής ευκρίνειας (HD, 720p), τότε σκοπεύουν να στείλουν όλοι οι συμμετέχοντες ανάλυση βίντεο 720p, με στόχο οθόνες WQHD (δηλαδή 2560×1440). Ας πούμε ότι κάθε ροή καταναλώνει 1,5Mbps (μπορεί να χρειαστεί και περισσότερο), οπότε:

- Κάθε συμμετέχων στη συνεδρία αποστέλλει 1,5Mbps και λαμβάνει 3 ροές 1,5Mbps
- Σε 4 συμμετέχοντες, ο διακομιστής πολυμέσων πρέπει να λάβει 6Mbps και να στείλει 18Mbps

Συνοψίζοντας σε έναν απλό πίνακα, παίρνουμε[9]:

Πίνακας 1 Σενάριο με 4 χρήστες

Ανάλυση	720p
Ρυθμός	1,5Mbps
Εξερχόμενος χρήστης	1,5Mbps (1 ροή)
Εισερχόμενος χρήστης	4,5Mbps (3 ροές)

Εξερχόμενη SFU	18Mbps (12 ροές)
Εισερχόμενη SFU	6Mbps (4 ροές)

Εάν δεν μας ενδιαφέρει τόσο πολύ η ανάλυση, μπορούμε να στοχεύσουμε στην ανάλυση VGA και ακόμη και να περιορίσουμε τα bitrate στα 600Kbps:

Πίνακας 2 Σενάριο με μικρότερη ανάλυση

Ανάλυση	VGA
Ρυθμός	600Kbps
Εξερχόμενος χρήστης	0,6Mbps (1 ροή)
Εισερχόμενος χρήστης	1,8Mbps (3 ροές)
Εξερχόμενη SFU	7,2Mbps (12 ροές)
Εισερχόμενη SFU	2,4Mbps (4 ροές)

Αυτό που ίσως θα θέλαμε να αποφύγουμε όταν πηγαίνουμε σε VGA είναι η ανάγκη αναβάθμισης της ανάλυσης στην οθόνη. Μπορεί να φαίνεται άσχημα, ειδικά στις μεγαλύτερες οθόνες 4K.

Με μια πρώτη ματιά φαίνεται ότι αν επιλέγαμε ανάλυση VGA θα μπορούσαμε να χωρέσουμε τρεις ταυτόχρονες κλήσεις σε μια συνεδρία, ενώ με ανάλυση HD θα υποστηριζόταν μόνο μία.

Τι γίνεται όμως αν η διάταξή μας είναι λίγο διαφορετική; Αν είχαμε έναν ομιλητή και μικρότερα σημεία προβολής για τους άλλους συμμετέχοντες:



Εικόνα 15 Στυλ Hangouts

Το ονομάζουμε στυλ Hangouts, επειδή το Hangouts είναι πολύ γνωστό για αυτήν τη διάταξη και ήταν από τα πρώτα που το χρησιμοποίησε αποκλειστικά χωρίς να προσφέρει ένα μεγαλύτερο σύνολο πρόσθετων διατάξεων.

Αυτή τη φορά, θα χρησιμοποιήσουμε ταυτόχρονη εκπομπή, όλοι θα στέλνουν βίντεο υψηλής ποιότητας και το SFU θα αποφασίζει ποια εισερχόμενη ροή θα χρησιμοποιήσει ως τον κυρίαρχο ομιλητή, επιλέγοντας την υψηλότερη ανάλυση για αυτόν και ποια ροή θα επιλέξει για την χαμηλότερη ανάλυση.

Θα στοχεύσουμε σε 720p, επειδή μετά από μερικά πειράματα, αποφασίσαμε ότι οι χαμηλότερες αναλύσεις, όταν κλιμακώνονται στις μεγαλύτερες οθόνες δεν φαίνονται τόσο καλά. Καταλήγουμε στα εξής:

- Κάθε συμμετέχων στη συνεδρία αποστέλλει 2,2Mbps (δηλαδή 1,5Mbps για τη ροή 720p και τα επιπλέον 700Kbps για τις άλλες αναλύσεις που θα κάνουμε ταυτόχρονη μετάδοση)
- Κάθε συμμετέχων στη συνεδρία λαμβάνει 1,5Mbps από τον κυρίαρχο ομιλητή και 2 επιπλέον εισερχόμενες ροές ~ 300Kbps για τα μικρότερα παράθυρα βίντεο
- Σε 4 συμμετέχοντες, ο διακομιστής πολυμέσων πρέπει να λάβει 8,8Mbps και να στείλει 8,4Mbps

Πίνακας 3 Σενάριο με 4 χρήστες σε 720p με simulcast

Ανάλυση	720p ως υψηλότερο (σε ταυτόχρονη εκπομπή)
Ρυθμός	150Kbps - 1,5Mbps
Εξερχόμενος χρήστης	2.2Mbps (1 ροή)
Εισερχόμενος χρήστης	1,5Mbps (1 ροή)

	0,3Mbps (2 ροές)
Εξερχόμενη SFU	8,4Mbps (12 ροές)
Εισερχόμενη SFU	8,8Mbps (4 ροές)

Το συμπέρασμα είναι το εξής:

Διαφορετικές περιπτώσεις χρήσης του ομαδικού βίντεο με τον ίδιο αριθμό χρηστών, μεταφράζονται σε διαφορετικό φόρτο εργασίας στο διακομιστή πολυμέσων.

Και αν δεν αναφέρθηκε συγκεκριμένα, η ταυτόχρονη εκπομπή λειτουργεί εξαιρετικά και βελτιώνει την αποτελεσματικότητα και την ποιότητα των ομαδικών κλήσεων (το simulcast(ταυτόχρονη εκπομπή) είναι αυτό που χρησιμοποιήσαμε στη συνάντηση του Hangouts Style).

Στα 3 σενάρια που παρουσιάσαμε εδώ για βιντεοκλήση 4 κατευθύνσεων, έχουμε αυτήν την ποικιλία δραστηριότητας στο SFU:

Πίνακας 4 Τελική Σύνοψη

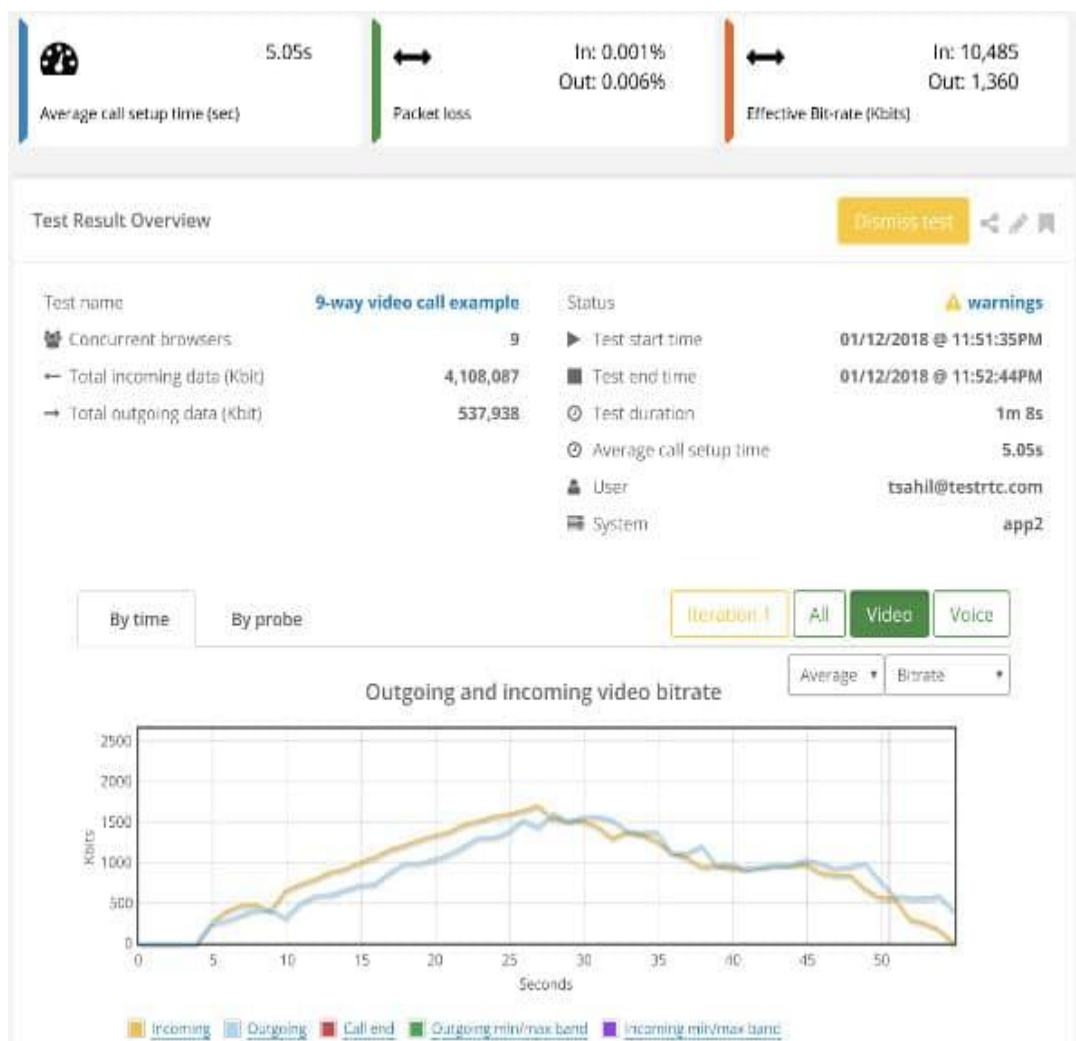
	720p	VGA	Στυλ Hangouts
Εξερχόμενη SFU	18Mbps	7,2Mbps	8,4Mbps
Εισερχόμενη SFU	6Mbps	2,4Mbps	8,8Mbps

3.3 Ενεργοί χρήστες σε μια κλήση webrtc

Καταρχήν είναι δύσκολο να απαντηθεί το ερώτημα πόσοι χρήστες μπορούν να είναι ενεργοί σε μια κλήση. Εάν χρησιμοποιούμε ένα MCU , μπορούμε να λάβουμε όσους περισσότερους χρήστες μπορεί να χειριστεί το MCU. Εάν χρησιμοποιούμε SFU, εξαρτάται από 3 διαφορετικές παραμέτρους:

- 1.Το επίπεδο πολυπλοκότητας του διακομιστή πολυμέσων μας, μαζί με την απόδοση που έχει
2. Την ισχύ που έχουμε στη διάθεσή μας στις συσκευές πελάτη
- 3.Τον τρόπο με τον οποίο δημιουργήσαμε την υποδομή μας

Οποιαδήποτε υλοποίηση με περίπου 8-10 χρήστες σε μία κλήση γίνεται περίπλοκη. Ακολουθεί ένα παράδειγμα μιας δημόσια διαθέσιμης υπηρεσίας.



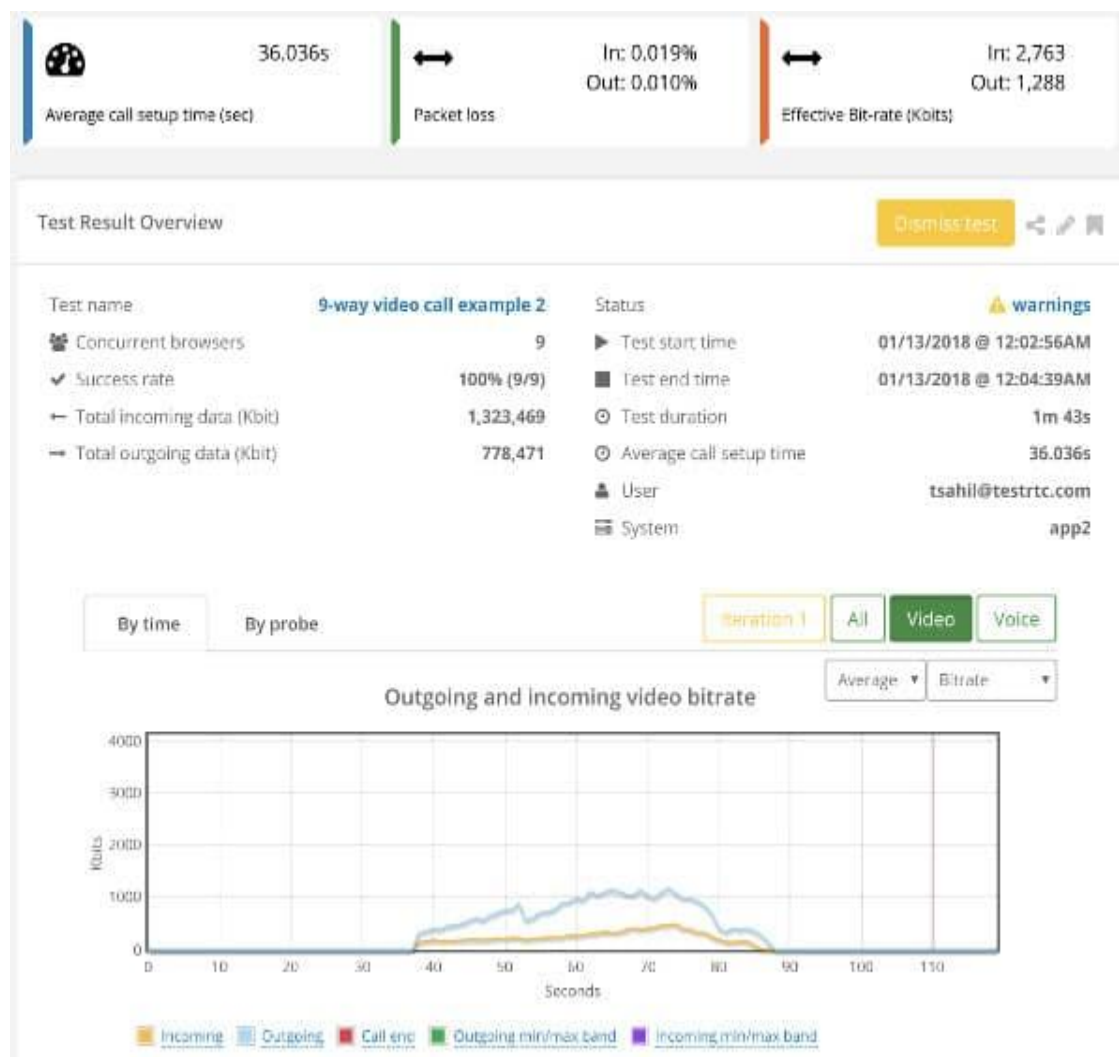
Εικόνα 16 9 συμμετέχοντες σε διάταξη παραθύρων

Το σενάριο είναι το εξής:

- 9 συμμετέχοντες σε μία συνεδρία σε διάταξη παραθύρων στους ομιλητές.
- Χρησιμοποιούμε το testRTC για να βάλουμε τους χρήστες στη συνεδρία, οπότε όλα αυτοματοποιούνται
- Το τρέχουμε για ένα λεπτό. Μετά από αυτό, σταματάει η συνεδρία, καθώς είναι demo
- Λαμβάνει υπόψη ότι συμμετέχουν 9 άτομα στην οθόνη, με μειωμένες αναλύσεις για όλους σε VGA και διαθέτει 1,3Mbps για αυτήν την ανάλυση
- Οι browsers λαμβάνουν 10Mbps δεδομένων για επεξεργασία

Ο διακομιστής πολυμέσων αποφάσισε εδώ πώς να περιορίσει και να μετρήσει την κυκλοφορία.

Και εδώ είναι μια άλλη υπηρεσία με ένα online demo που εκτελεί το ίδιο σενάριο:



Εικόνα 17 Εναλλακτική υλοποίηση

Τώρα ο εισερχόμενος ρυθμός bit κατά μέσο όρο ανά πρόγραμμα περιήγησης ήταν μόνο 2,7Mbps - σχεδόν το ένα τέταρτο της άλλης υπηρεσίας.

Τώρα θα μελετήσουμε τι συμβαίνει με ορισμένες δημοφιλείς υπηρεσίες που κάνουν βιντεοδιάσκεψη σε ένα μοντέλο δρομολόγησης με SFU. Θα δούμε τι είδους περιορισμούς μεγέθους θέτουν στις εφαρμογές τους.

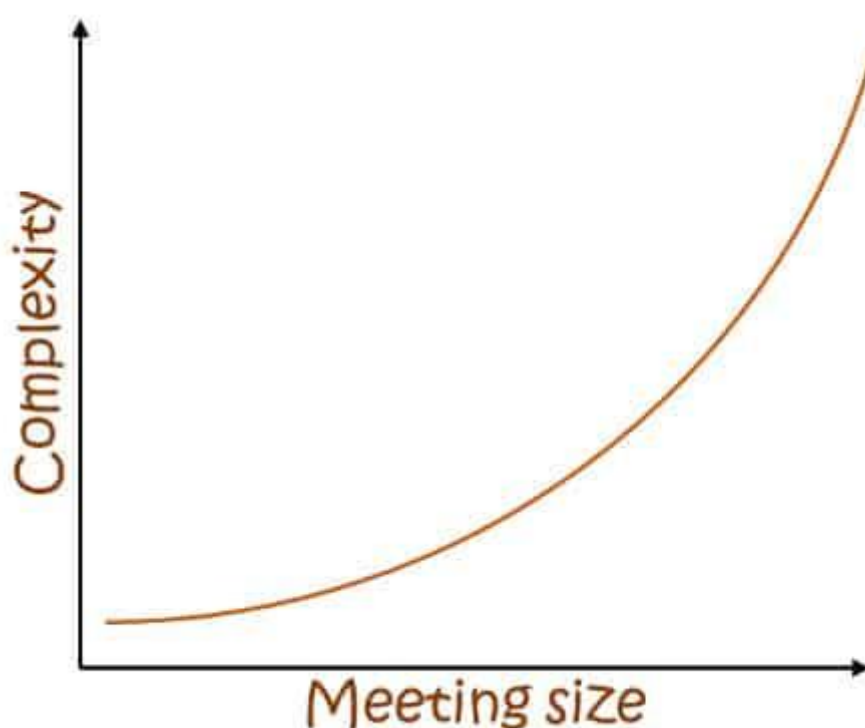
Έχουμε τα εξής:

- Το Google Hangouts επιτρέπει έως και 25 συμμετέχοντες σε μία συνεδρία, ενώ αυτό το όριο ήταν 10 στο παρελθόν.
- Το Hangouts Meet έχει θέσει τον μέγιστο αριθμό του σε 50 συμμετέχοντες σε μία συνεδρία
- Το Houseparty έως και 8 συμμετέχοντες
- Το Skype 25 συμμετέχοντες
- Το appear.in υποστηρίζει έως και 12 συμμετέχοντες σε ένα δωμάτιο

- Το Amazon Chime επιτρέπει έως 16 συμμετέχοντες στην επιφάνεια εργασίας και έως και 8 συμμετέχοντες σε iOS (δεν υπάρχει υποστήριξη για Android ακόμη)
- Atlassian Stride and Meet Jitsi έως και 50 συμμετέχοντες

Αυτό σημαίνει ότι δεν μπορούμε να ξεπεράσουμε τους 50 συμμετέχοντες;

Θεωρούμε ότι υπάρχει ένας αυξανόμενος βαθμός δυσκολίας καθώς αυξάνεται το μέγεθος της συνάντησης:



Εικόνα 18 Πολυπλοκότητα σε σχέση με αριθμό συμμετεχόντων

Όταν κοιτάζουμε τις πλατφόρμες CPaaS (Communications Platform as a Service Explained), εκείνοι που υποστηρίζουν βιντεοκλήσεις και ομαδικές κλήσεις έχουν συχνά όρια στο μέγεθος της συνάντησής τους. Στις περισσότερες περιπτώσεις, δίνουν έναν αυθαίρετο αριθμό με τον οποίο έχουν δοκιμάσει ή είναι άνετοι με αυτόν. Όπως έχουμε δει, αυτός ο αριθμός είναι κατάλληλος για ένα πολύ συγκεκριμένο σενάριο, το οποίο μπορεί να μην είναι αυτό που σκεφτόμαστε.

Στο CPaaS, αυτοί οι αριθμοί ποικίλλουν από 10 συμμετέχοντες έως 100+ σε ένα μόνο session. Συνήθως, εάν μπορούμε να πάμε σε μεγαλύτερο αριθμό, οι επιπλέον συμμετέχοντες θα είναι μόνο για προβολή.

Λίγα πράγματα που πρέπει να θυμόμαστε:

- Όσο μεγαλύτερο είναι το μέγεθος της ομάδας τόσο πιο περίπλοκο είναι να υλοποιηθεί και να βελτιστοποιηθεί
- Το πρόγραμμα περιήγησης πρέπει να εκτελεί πολλούς αποκωδικοποιητές, κάτι που αποτελεί από μόνο του βάρος
- Οι κινητές συσκευές, ειδικά οι παλαιότερες, μπορούν να γονατίσουν πολύ γρήγορα σε τέτοιες περιπτώσεις. Δοκιμάζουμε τις παλαιότερες, πιο αδύναμες συσκευές που σκοπεύουμε να υποστηρίξουμε προτού καθορίσουμε το μέγεθος της ομάδας που θα υποστηρίξουμε
- Μπορούμε να δημιουργήσουμε το SFU με τρόπο που δεν δρομολογεί όλα τα εισερχόμενα δεδομένα σε όλους, αλλά επιλέγει μερικά δεδομένα για αποστολή.

3.4 Χωρητικότητα ροών σε έναν media server

Λαμβάνοντας υπόψη όσα έχουμε δει παραπάνω σχετικά με τις ταχύτητες και τις τροφοδοσίες, είναι ασφαλές να πούμε ότι εξαρτάται πραγματικά από αυτό που κάνουμε.

Εάν αυτό που ψάχνουμε είναι ομαδική κλήση όπου όλοι είναι ενεργοί, θα πρέπει να στοχεύσουμε σε 100-500 συμμετέχοντες συνολικά σε έναν μόνο διακομιστή. Οι αριθμοί θα διαφέρουν ανάλογα με το μηχάνημα που επιλέγουμε για το διακομιστή πολυμέσων και τους ρυθμούς bit που σχεδιάζουμε ανά ροή κατά μέσο όρο.

Εάν αυτό που ψάχνουμε είναι μια μετάδοση ενός ατόμου σε ένα μεγαλύτερο κοινό, όλα γίνονται μέσω του WebRTC για τη διατήρηση χαμηλού λανθάνοντος χρόνου, τα 200-1.000msec είναι πιθανώς μια καλύτερη εκτίμηση, ίσως ακόμη περισσότερο.

Ένα άλλο πράγμα που θα πρέπει να αντιμετωπίσουμε είναι σε ποια μηχανήματα πρόκειται να φιλοξενήσουμε τον διακομιστή πολυμέσων μας. Θα είναι τα μεγαλύτερα δυνατόν μηχανήματα που θα είναι διαθέσιμα ή θα εξυπηρετούμαστε καλύτερα με μικρότερα;

Εάν επιλέξουμε μεγάλα μηχανήματα σημαίνει ότι θα έχουμε τη δυνατότητα να εξυπηρετούμε μεγαλύτερο κοινό και περιόδους σύνδεσης σε ένα μόνο μηχάνημα, επομένως η πολυπλοκότητα της υπηρεσίας μας θα είναι χαμηλότερη. Εάν κάτι καταρρεύσει, όπως συνήθως γίνεται με τους διακομιστές πολυμέσων, θα επηρεαστούν περισσότεροι χρήστες. Και όταν θα χρειαστεί να αναβαθμίσουμε τον διακομιστή πολυμέσων μας (που θα το κάνουμε κάποια στιγμή), αυτή η διαδικασία μπορεί να μας κοστίσει περισσότερο ή να γίνει κάπως πιο περίπλοκη επίσης.

Όσο μεγαλύτερο είναι το μηχάνημα, τόσο περισσότερους πυρήνες θα έχει. Αυτό οδηγεί σε διακομιστές πολυμέσων που πρέπει να εκτελούνται σε λειτουργία πολλαπλών νημάτων, οπότε θα είναι πιο περίπλοκοι στην κατασκευή, τον εντοπισμό σφαλμάτων και την επιδιόρθωση, αφού θα υπάρχουν περισσότερα κινούμενα μέρη.

Η επιλογή μικρών μηχανημάτων σημαίνει ότι θα αντιμετωπίσουμε προβλήματα κλίμακας νωρίτερα και θα απαιτηθούν αλγόριθμοι και ευρετικές μέθοδοι που είναι πιο περίπλοκες. Θα έχουμε περισσότερες ακραίες περιπτώσεις στο load balancing της υπηρεσίας μας.

Πώς λοιπόν θα αποφασίσουμε ότι ο διακομιστής πολυμέσων μας πέτυχε πλήρη χωρητικότητα; Πώς θα αποφασίσουμε εάν η επόμενη περίοδος σύνδεσης θα πρέπει να γίνει σε νέο μηχάνημα ή σε άλλο ή να τοποθετηθεί στον τρέχοντα διακομιστή πολυμέσων που χρησιμοποιούμε; Εάν χρησιμοποιούμε ήδη μια τρέχουσα συνεδρία και οι νέοι συμμετέχοντες θέλουν να συμμετάσχουν σε μια συνεδρία που εκτελείται ήδη σε αυτόν τον διακομιστή πολυμέσων, θα υπάρχει αρκετός χώρος για αυτούς; Δεν είναι εύκολο να απαντηθούν τέτοιες ερωτήσεις.

Υπάρχουν 3 διαφορετικές μέθοδοι μετρήσεων που καθορίζουν το πότε πρέπει να γίνει κλιμάκωση από έναν διακομιστή πολυμέσων σε άλλους. Ακολουθούν οι γενικές εναλλακτικές λύσεις:

1) Με βάση την CPU, δηλαδή όταν το ποσοστό χρήσης της CPU φτάσει μια συγκεκριμένη τιμή, αυτό σημαίνει ότι το μηχάνημα είναι "γεμάτο". Είναι μια καλύτερη λύση όταν χρησιμοποιούμε μικρότερα μηχανήματα, καθώς η CPU θα είναι ένας από τους πρώτους πόρους που θα εξαντληθούν.

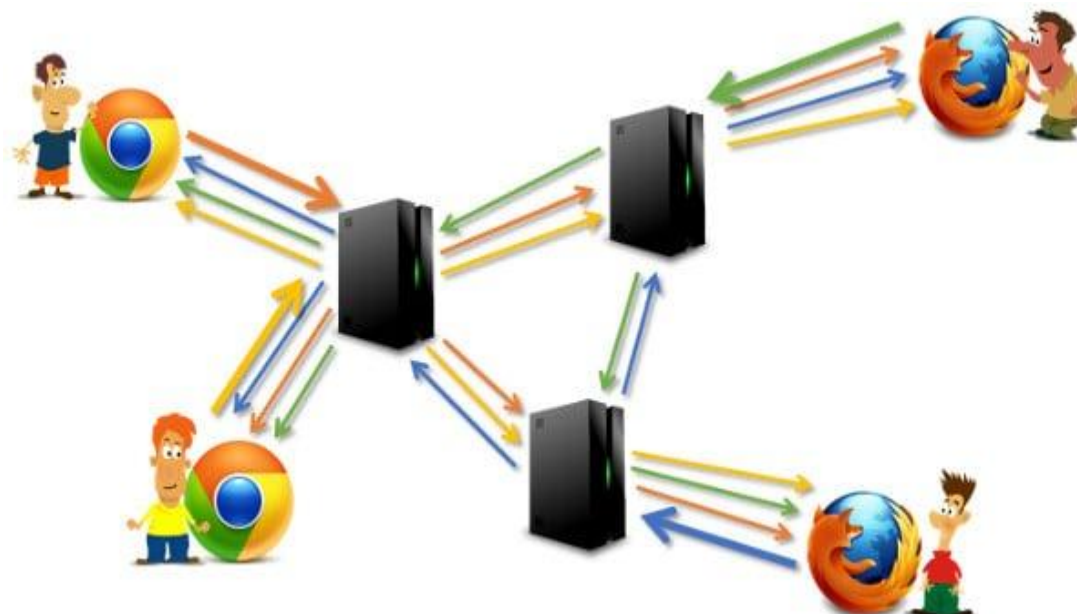
2) Με βάση το εύρος ζώνης, καθώς οι SFU καταναλώνουν πολλούς πόρους δικτύωσης. Εάν χρησιμοποιούμε μεγαλύτερα μηχανήματα, πιθανότατα δεν θα φτάσουμε το όριο της CPU, αλλά θα καταλήξουμε να τρώμε πάρα πολύ εύρος ζώνης. Έτσι, το αποτέλεσμα θα είναι να καθορίζουμε τη διαθέσιμη χωρητικότητα μέσω παρακολούθησης του εύρους ζώνης.

3) Με βάση τις ροές. Η πρόκληση μερικές φορές με τη CPU και το εύρος ζώνης είναι ότι ο αριθμός των περιόδων σύνδεσης και ροών που μπορούν να υποστηριχθούν ενδέχεται να διαφέρουν, ανάλογα με τις δυναμικές συνθήκες. Η στρατηγική κλιμάκωσής μας ενδέχεται να μην είναι σε θέση να το αντιμετωπίσει αυτό και μπορεί να θέλουμε μεγαλύτερο έλεγχο στους υπολογισμούς. Αυτό θα μας οδηγήσει στην αλλαγή του μεγέθους του μηχανήματος χρησιμοποιώντας ως κριτήριο τη CPU ή το εύρος ζώνης, αλλά θέτοντας κανόνες σε ισχύ που βασίζονται στον αριθμό των ροών που μπορεί να υποστηρίξει ο διακομιστής.

Η πρόκληση εδώ είναι ότι όποιο σενάριο κι αν επιλέξουμε, το μέγεθος είναι κάτι που πρέπει να αναλάβουμε μόνοι μας. Πολλοί αποφασίζουν να χρησιμοποιήσουν το testRTC όταν πρέπει να αντιμετωπίσουν αυτό το πρόβλημα.

3.5 Cascading σε μια συνεδρία

Το Cascading είναι η διαδικασία σύνδεσης ενός διακομιστή πολυμέσων με τον άλλο. Το παρακάτω διάγραμμα απεικονίζει την κατάσταση:



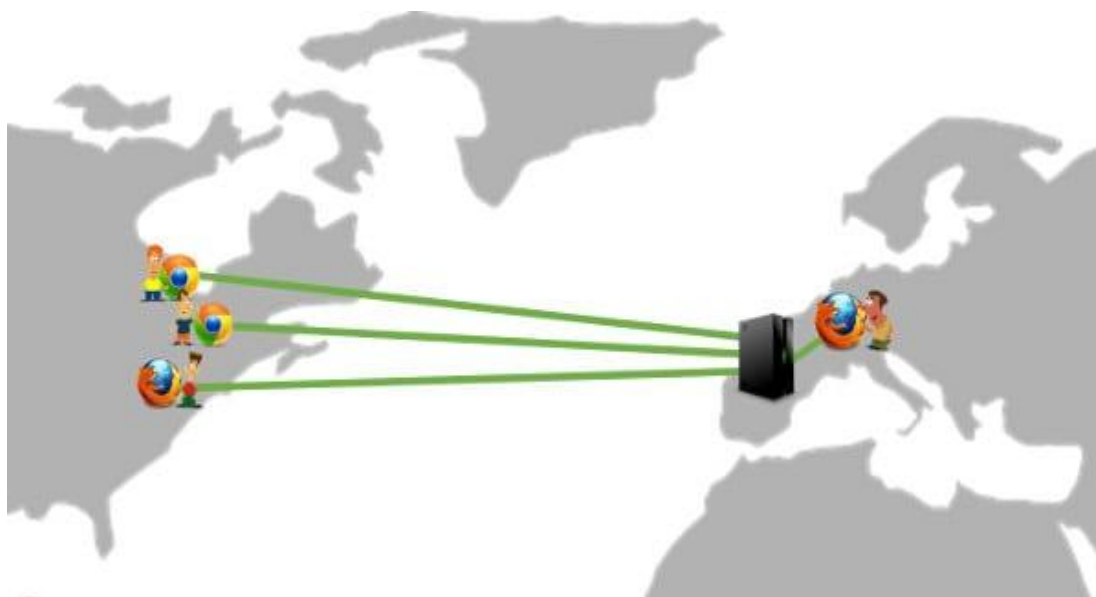
Εικόνα 19 Cascading

Έχουμε μια ομαδική βιντεοκλήση 4 κατευθύνσεων που εκτείνεται σε 3 διαφορετικούς διακομιστές μέσω. Οι διακομιστές δρομολογούν τα δεδομένα μεταξύ τους όπως χρειάζεται για να μπορέσουν να συνδεθούν. Το ερώτημα είναι γιατί θα θέλαμε να το κάνουμε αυτό.

1)Γεωγραφική Κατανομή

Όταν εκτελούμε μια καθολική υπηρεσία και έχουμε ένα SFU ως μέρος αυτής, το ερώτημα που τίθεται αμέσως είναι για μια νέα συνεδρία, ποιο SFU θα διαθέσουμε για αυτήν και σε ποιο από τα κέντρα δεδομένων. Δεδομένου ότι θέλουμε να έχουμε τους διακομιστές πολυμέσων όσο το δυνατόν πιο κοντά στους χρήστες, είτε έχουμε γνώση εκ των προτέρων για τη συνεδρία και ξέρουμε πού να την εκχωρήσουμε, είτε αποφασίζουμε με κάποια εύλογα κριτήρια, όπως η γεωγραφική τοποθεσία, οπότε επιλέγουμε το κέντρο δεδομένων που βρίσκεται πιο κοντά στο χρήστη που δημιουργήσε τη σύσκεψη.

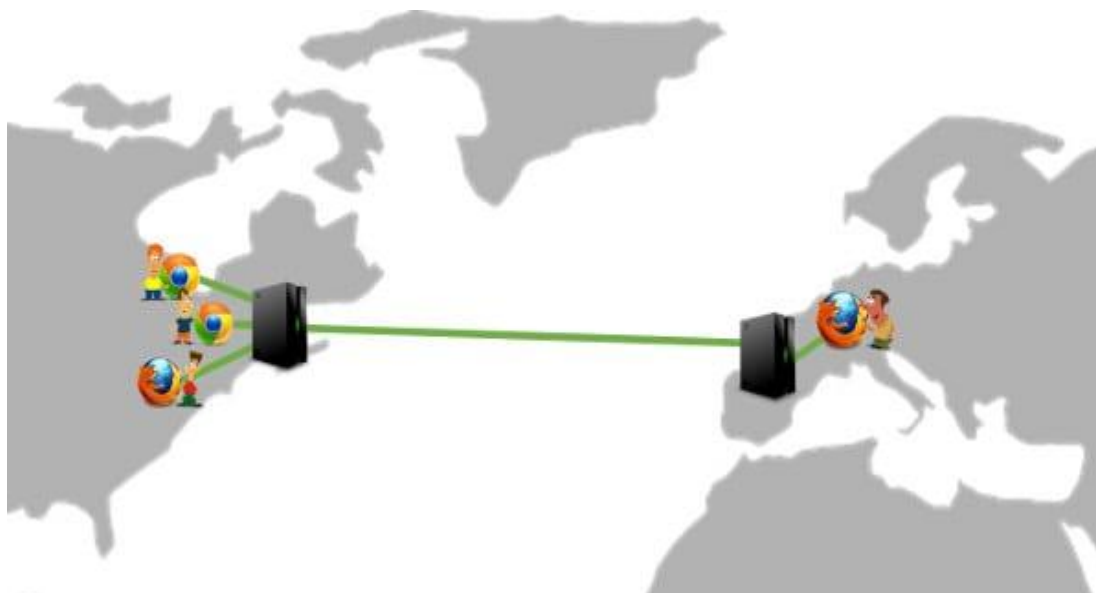
Ας υποθέσουμε ότι 4 άτομα είναι σε κλήση. 3 από αυτούς συμμετέχουν από τη Νέα Υόρκη, ενώ το 4ο από τη Γαλλία. Τι συμβαίνει εάν ο Γάλλος χρήστης ενταχθεί πρώτος;



Εικόνα 20 Εκδοχή 1

Ο διακομιστής θα φιλοξενηθεί στη Γαλλία, οπότε 3 από τα 4 άτομα θα βρίσκονται μακριά από το διακομιστή πολυμέσων, κάτι που δεν είναι και η καλύτερη προσέγγιση.

Μια λύση είναι η διεξαγωγή της συνάντησης με τη διάδοση σε διακομιστές που βρίσκονται πλησιέστερα σε καθέναν από τους συμμετέχοντες:



Εικόνα 21 Εκδοχή 2

Χρησιμοποιούμε περισσότερους πόρους για να εξυπηρετήσουμε αυτήν την περίοδο σύνδεσης, αλλά έχουμε πολύ περισσότερο έλεγχο στις διαδρομές

δεδομένων, ώστε να μπορούμε να τις βελτιστοποιήσουμε καλύτερα. Αυτό βελτιώνει την ποιότητα των δεδομένων για τη συνεδρία.

2)Κατακερματισμένες κατανομές

Ας υποθέσουμε ότι μπορούμε να συνδέσουμε έως και 100 συμμετέχοντες σε έναν διακομιστή πολυμέσων. Επιπλέον, κάθε συνάντηση μπορεί να φιλοξενήσει έως και 10 συμμετέχοντες. Στην ιδανική περίπτωση, δεν θα θέλουμε να εκχωρήσουμε περισσότερες από 10 συναντήσεις ανά διακομιστή πολυμέσων.

Τι γίνεται όμως αν υποθέταμε ότι το μέσο μέγεθος συνάντησης είναι 2 συμμετέχοντες; Μπορεί να μας οδηγήσει σε αυτόν τον τύπο κατανομής:



Εικόνα 22 Τύπος κατανομής

Αυτό προκαλεί μεγάλη σπατάλη πόρων διακομιστή. Πώς μπορούμε να το λύσουμε αυτό;

1. Με την υποχρέωση των ατόμων να δεσμεύονται εκ των προτέρων στο μέγιστο μέγεθος συνάντησης. Βέβαια αυτό είναι κάτι που πραγματικά δε θα θέλαμε να κάνουμε
2. Αναλαμβάνοντας ένα ρίσκο, υποθέτουμε ότι εάν διαθέσουμε το 50% της χωρητικότητας ενός διακομιστή, το υπόλοιπο της χωρητικότητας το αφήνουμε για υπάρχουσες συναντήσεις επιτρέποντάς τους να αυξηθούν. Εξακολουθούμε να σπαταλάμε πόρους, αλλά σε μικρότερο βαθμό. Θα υπάρχουν περιπτώσεις όπου δεν θα μπορούμε να συμπληρώσουμε τις συναντήσεις λόγω πόρων διακομιστή
3. Μετεγκατάσταση συνεδριών σε διακομιστές πολυμέσων σε μια προσπάθεια «ανασυγκρότησης» των διακομιστών. Είναι τόσο άσχημο όσο ακούγεται, και πιθανώς εξίσου ενοχλητικό για τους χρήστες
4. Συνεδρίες Cascade. Τα αφήνουμε να αναπτυχθούν στα μηχανήματα

Αυτό το τελευταίο συμβαίνει απότομα; Μπορούμε να το κάνουμε κάνοντας κράτηση ορισμένων πόρων ενός διακομιστή πολυμέσων για τη μεταφορά υφιστάμενων περιόδων σύνδεσης σε άλλους διακομιστές πολυμέσων.

3)Μεγαλύτερες συναντήσεις

Υποθέτοντας ότι θέλουμε να δημιουργήσουμε μεγαλύτερες συσκέψεις από αυτές που μπορεί να χειριστεί ένας μοναδικός διακομιστής πολυμέσων, η μόνη επιλογή μας είναι να χρησιμοποιήσουμε cascade.

Εάν ο διακομιστής πολυμέσων μας μπορεί να φιλοξενήσει 100 συμμετέχοντες και θέλουμε συσκέψεις στο μέγεθος των 5.000 συμμετεχόντων, τότε θα πρέπει να είμαστε σε θέση να προχωρήσουμε για να τους υποστηρίξουμε. Αυτό δεν είναι εύκολο, γεγονός που εξηγεί γιατί δεν υπάρχουν πολλές τέτοιες λύσεις, αλλά είναι σίγουρα δυνατό.

Υπενθυμίζουμε ότι σε τόσο μεγάλες συναντήσεις, η ροή των μέσων δεν θα είναι αμφίδρομη. Θα έχουμε λιγότερους συμμετέχοντες να στέλνουν πολυμέσα και πολύ περισσότερους μόνο να λαμβάνουν δεδομένα.

4.ΜΕΛΕΤΗ ΣΕΝΑΡΙΩΝ ΧΡΗΣΗΣ

Σε αυτό το κεφάλαιο θα μελετηθεί η περίπτωση της δευτεροβάθμιας εκπαίδευσης σε τρία διαφορετικά σενάρια.

- Σενάριο 1: Μονόδρομη επικοινωνία με αποστολή δεδομένων από τον καθηγητή στο server και από τον server προς όλους τους μαθητές δίχως οι μαθητές να στέλνουν βίντεο/ήχο συνεχώς.
- Σενάριο 2: Αμφίδρομη επικοινωνία καθηγητή με μαθητές
- Σενάριο 3: Αμφίδρομη επικοινωνία καθηγητή με 1 μαθητή, μονόδρομη με τους υπολοίπους

Να σημειώσουμε ότι κατά την εκτίμηση του αριθμού των εξυπηρετητών που απαιτούνται για ανάπτυξη και κλιμάκωση της υπηρεσίας σε μεγάλο αριθμό χρηστών, θεωρείται ότι οι εξυπηρετητές διαθέτουν τους υπολογιστικούς πόρους (CPU) και μνήμη προκειμένου να ανταπεξέλθουν στον απαιτούμενο φόρτο. Επομένως το κριτήριο που χρησιμοποιείται για την εκτίμηση του αριθμού εξυπηρετητών είναι η διαδικτυακή σύνδεση (1 Gbps, 10 Gbps).

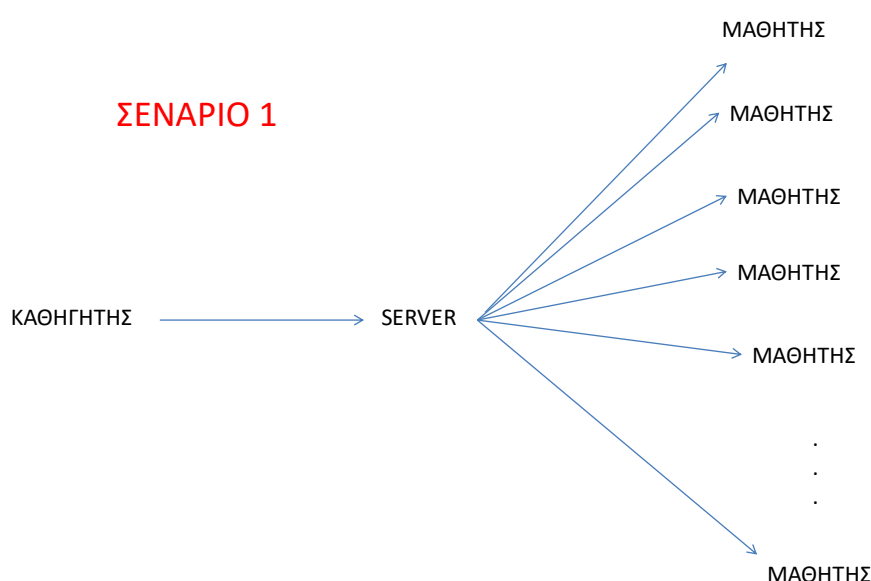
Σε αυτό το κεφάλαιο θα μελετηθεί η περίπτωση της δευτεροβάθμιας εκπαίδευσης σε τρία διαφορετικά σενάρια.

- Σενάριο 1: Μονόδρομη επικοινωνία με αποστολή δεδομένων από τον καθηγητή στο server και από τον server προς όλους τους μαθητές δίχως οι μαθητές να στέλνουν βίντεο/ήχο συνεχώς.
- Σενάριο 2: Αμφίδρομη επικοινωνία καθηγητή με μαθητές
- Σενάριο 3: Αμφίδρομη επικοινωνία καθηγητή με 1 μαθητή, μονόδρομη με τους υπολοίπους

Να σημειώσουμε ότι κατά την εκτίμηση του αριθμού των εξυπηρετητών που απαιτούνται για ανάπτυξη και κλιμάκωση της υπηρεσίας σε μεγάλο αριθμό χρηστών, θεωρείται ότι οι εξυπηρετητές διαθέτουν τους υπολογιστικούς πόρους (CPU) και μνήμη προκειμένου να ανταπεξέλθουν στον απαιτούμενο φόρτο. Επομένως το κριτήριο που χρησιμοποιείται για την εκτίμηση του αριθμού εξυπηρετητών είναι η διαδικτυακή σύνδεση (1 Gbps, 10 Gbps).

Σενάριο 1: Μονόδρομη επικοινωνία καθηγητή με μαθητές

Στο πρώτο σενάριο, όπως φαίνεται και στην παρακάτω εικόνα έχουμε μονόδρομη επικοινωνία με την αποστολή δεδομένων από τον καθηγητή στο server και από τον server προς όλους τους μαθητές. Θεωρούμε ότι ο μέγιστος αριθμός μαθητών που συνδέονται σε μια συνεδρία είναι: 25 μαθητές/μαθήτριες.



Εικόνα 23 Μονόδρομη επικοινωνία από καθηγητή σε 25 μαθητές

Παίρνοντας λοιπόν ως δεδομένα τα στοιχεία του παρακάτω πίνακα, όσον αφορά την αποστολή video, default ανάλυσης από τον καθηγητή, την αποστολή μικρής ανάλυσης από τον καθηγητή καθώς και τη λήψη μικρής ανάλυσης από τους μαθητές:

Πίνακας 5 Απαιτήσεις σε εύρος ζώνης

Στοιχεία από webex	min (Mbps)	Max (Mbps)	Aver (Mbps)

Αποστολή default ανάλυσης video καθηγητή	1,6	2,2	1,9
Αποστολή μικρής ανάλυσης video καθηγητή	0,5	0,7	0,6
Λήψη μικρής ανάλυσης video μαθητή	0,2	0,26666	0,23333

Θα έχουμε τα παρακάτω αποτελέσματα όσον αφορά την κίνηση των δεδομένων ανά συνεδρία με τις τιμές να είναι σε Mbps:

Πίνακας 6 Μέση κίνηση ανά συνεδρία για το πρώτο σενάριο υλοποίησης

Σενάριο:	Σενάριο A
Μέση Κίνηση / συνεδρία	1
default ανάλυση	
Εξερχόμενη κίνηση καθηγητή	1,9
Εισερχόμενη κίνηση μαθητή	1,9
Εισερχόμενη κίνηση καθηγητή	0
Εισερχόμενη κίνηση server	1,9
Εξερχόμενη κίνηση server	47,5
Terabytes/μήνα out	2,9925
Μικρή ανάλυση	
Εξερχόμενη κίνηση καθηγητή	0,6
Εισερχόμενη κίνηση μαθητή	0,6
Εισερχόμενη κίνηση καθηγητή	0
Εισερχόμενη κίνηση server	0,6
Εξερχόμενη κίνηση server	15
Terabytes/μήνα out	0,945

Αν θεωρήσουμε ότι διαθέτουμε servers με δυνατότητες 1 και 10Gbps τότε ο αριθμός των servers ανά συνεδρία που θα προκύψουν θα είναι:

Πίνακας 7 Αριθμός servers ανά συνεδρία για το πρώτο σενάριο υλοποίησης με default ανάλυση

default ανάλυση		Σενάριο A
Σενάρια Servers		Αριθμός servers / συνεδρία
A. 1Gbps Εισερχόμενη		0,0019
A. 1Gbps Εξερχόμενη		0,0475
B. 10 Gbps Εισερχόμενη		0,00019
B. 10 Gbps Εξερχόμενη		0,00475

Βασιζόμενοι στα προηγούμενα μπορούμε να υπολογίσουμε τον αριθμό servers χωρητικότητας 1Gbps για διάφορες τιμές συνεδριών οπότε θα προκύψουν τα ακόλουθα

Πίνακας 8 Αριθμός servers 1Gbps για διαφορετικό αριθμό συνεδριών

default ανάλυση	
Αριθμός servers 1 Gbs/ συνεδρίες:	
1000 εισερ.	1,9
1000 εξερ.	47,5
10000 εισερ.	19
10000 εξερ.	475
50000 εισερ.	95
50000 εξερ.	2375
100000 εισερ.	190
100000 εξερ.	4750

Με παρόμοιο τρόπο μπορούμε να υπολογίσουμε τον αριθμό servers χωρητικότητας 10Gbps για διάφορες τιμές συνεδριών οπότε θα έχουμε

Πίνακας 9 Αριθμός servers 10Gbps για διαφορετικό αριθμό συνεδριών

Αριθμός servers 10 Gbs/ συνεδρίες:	
1000 εισερ.	0,19
1000 εξερ.	4,75
10000 εισερ.	1,9
10000 εξερ.	47,5
50000 εισερ.	9,5
50000 εξερ.	237,5

100000 εισερ.	19
100000 εξερ.	475

Αν θέλαμε να υπολογίσουμε ένα σχετικό κόστος για την περίπτωση των servers χωρητικότητας 1Gbps και 10Gbps και για 50000 συνεδρίες θα προέκυπταν τα ακόλουθα με την υπόθεση ότι ο ένας server 1Gbps έχει ένα κόστος γύρω στα 400 ευρώ το μήνα ενώ ένας server 10Gbps περίπου 1500 ευρώ το μήνα

Πίνακας 10 Κόστος υλοποίησης για το πρώτο σενάριο

1 Gbps / ευρώ μήνα	400,00 €
Συνολικό κόστος / μήνα	950.000,00 €
10 Gbps / ευρώ μήνα	1500,00€
Συνολικό κόστος / μήνα	357.000,00€

Στην περίπτωση της μικρής ανάλυσης θα είχαμε

Πίνακας 11 Αριθμός servers ανά συνεδρία

Σενάριο A	
Μικρή ανάλυση	
Σενάρια Servers	Αριθμός servers / συνεδρία
A. 1Gbps Εισερχόμενη	0,0006
A. 1Gbps Εξερχόμενη	0,015
B. 10 Gbps Εισερχόμενη	0,00006
B. 10 Gbps Εξερχόμενη	0,0015

Αντίστοιχα με πριν όσον αφορά τους servers χωρητικότητας 1Gbps

Πίνακας 12 Αριθμός servers για διαφορετικό πλήθος συνεδριών

Μικρή ανάλυση	
Αριθμός servers 1 Gbs/ συνεδρίες:	
1000 εισερ.	0,6

1000 εξερ.	15
10000 εισερ.	6
10000 εξερ.	150
50000 εισερ.	30
50000 εξερ.	750
100000 εισερ.	60
100000 εξερ.	1500

Ενώ για την περίπτωση των servers 10Gbps

Πίνακας 13 Αριθμός servers 10Gbps για διαφορετικό πλήθος συνεδριών

Αριθμός servers 10 Gbs/ συνεδρίες:	
1000 εισερ.	0,06
1000 εξερ.	1,5
10000 εισερ.	0,6
10000 εξερ.	15
50000 εισερ.	3
50000 εξερ.	75
100000 εισερ.	6
100000 εξερ.	150

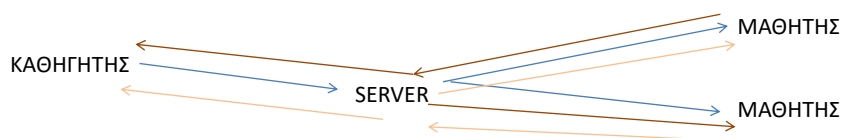
Τελικά όσον αφορά το κόστος για την περίπτωση της μικρής ανάλυσης προκύπτει ότι

Πίνακας 14 Κόστος υλοποίησης για την μικρή ανάλυση στο σενάριο A

1 Gbps / ευρώ μήνα	400,00 €
Συνολικό κόστος / μήνα	300.000,00 €
10 Gbps / ευρώ μήνα	1500,00€
Συνολικό κόστος / μήνα	112.500,00€

Στο δεύτερο σενάριο που θα μελετήσουμε υπάρχει αμφίδρομη επικοινωνία από τον καθηγητή προς όλους τους 25 μαθητές. Πιο κάτω φαίνεται μια ενδεικτική υλοποίηση για 1 καθηγητή και 2 μαθητές

ΣΕΝΑΡΙΟ 2



Εικόνα 24 Αμφίδρομη επικοινωνία μεταξύ καθηγητή και μαθητών

Παίρνοντας λοιπόν ως δεδομένα τα στοιχεία του παρακάτω πίνακα, όσον αφορά την αποστολή video, default ανάλυσης από τον καθηγητή, την αποστολή μικρής ανάλυσης καθώς και τη λήψη μικρής ανάλυσης από τους μαθητές:

Πίνακας 15 Μέση κίνηση ανά συνεδρία για το δεύτερο σενάριο υλοποίησης

Σενάριο:	Σενάριο B
Μέση Κίνηση / συνεδρία	25
default ανάλυση	
Εξερχόμενη κίνηση καθηγητή	1,9
Εισερχόμενη κίνηση μαθητή	7,5
Εισερχόμενη κίνηση καθηγητή	5,83
Εισερχόμενη κίνηση server	7,73
Εξερχόμενη κίνηση server	187,5
Terabytes in / μήνα	0,4872
Terabytes out / μήνα	11,8125
Μικρή ανάλυση	
Εξερχόμενη κίνηση καθηγητή	0,6
Εισερχόμενη κίνηση μαθητή	6,2

Εισερχόμενη κίνηση καθηγητή	5,83
Εισερχόμενη κίνηση server	6,43
Εξερχόμενη κίνηση server	155
Terabytes in /μήνα	0,4053
Terabytes out / μήνα	9,765

Αν θεωρήσουμε ότι διαθέτουμε servers με δυνατότητες 1 και 10Gbps τότε ο αριθμός των servers ανά συνεδρία που θα προκύψουν θα είναι:

Πίνακας 16 Αριθμός servers ανά συνεδρία για το δεύτερο σενάριο υλοποίησης

default ανάλυση		Σενάριο B
Σενάρια Servers		Αριθμός servers / συνεδρία
A. 1Gbps Εισερχόμενη		0,00773
A. 1Gbps Εξερχόμενη		0,1875
B. 10 Gbps Εισερχόμενη		0,000773
B. 10 Gbps Εξερχόμενη		0,01875

Βασιζόμενοι στα προηγούμενα μπορούμε να υπολογίσουμε τον αριθμό servers χωρητικότητας 1Gbps για διάφορες τιμές συνεδριών οπότε θα προκύψουν τα ακόλουθα

Πίνακας 17 Αριθμός servers 1Gbps για διαφορετικό πλήθος συνεδριών

default ανάλυση	
Αριθμός servers 1 Gbs/ συνεδρίες:	
1000 εισερ.	7,73
1000 εξερ.	187,5
10000 εισερ.	77,3
10000 εξερ.	1875
50000 εισερ.	386,5
50000 εξερ.	9375
100000 εισερ.	773
100000 εξερ.	18750

Με παρόμοιο τρόπο μπορούμε να υπολογίσουμε τον αριθμό servers χωρητικότητας 10Gbps για διάφορες τιμές συνεδριών οπότε θα έχουμε

Πίνακας 18 Αριθμός servers 10Gbps για διαφορετικό αριθμό συνεδριών

Αριθμός servers 10 Gbs/ συνεδρίες:	
1000 εισερ.	0,773
1000 εξερ.	18,75
10000 εισερ.	7,73
10000 εξερ.	187,5
50000 εισερ.	38,65
50000 εξερ.	937,5
100000 εισερ.	77,3
100000 εξερ.	1875

Αν θέλαμε να υπολογίσουμε ένα σχετικό κόστος για την περίπτωση των servers χωρητικότητας 1Gbps και 10Gbps και για 50000 συνεδρίες θα προέκυπταν τα ακόλουθα με την υπόθεση ότι ο ένας server 1Gbps έχει ένα κόστος γύρω στα 400 ευρώ το μήνα ενώ ένας server 10Gbps περίπου 1500 ευρώ το μήνα

Πίνακας 19 Κόστος υλοποίησης για το δεύτερο σενάριο

1 Gbps / ευρώ μήνα	400,00 €
Συνολικό κόστος / μήνα	3.750.000,00 €
10 Gbps / ευρώ μήνα	1500,00€
Συνολικό κόστος / μήνα	1.407.000,00€

Στην περίπτωση της μικρής ανάλυσης θα είχαμε

Πίνακας 20 Αριθμός servers ανά συνεδρία

	Σενάριο B
Μικρή ανάλυση	
Σενάρια Servers	Αριθμός servers / συνεδρία
A. 1Gbps Εισερχόμενη	0,00643
A. 1Gbps Εξερχόμενη	0,155
B. 10 Gbps Εισερχόμενη	0,000643
B. 10 Gbps Εξερχόμενη	0,0155

Αντίστοιχα με πριν όσον αφορά τους servers χωρητικότητας 1Gbps

Πίνακας 21 Αριθμός servers για διαφορετικό πλήθος συνεδριών

Μικρή ανάλυση	
Αριθμός servers 1 Gbs/ συνεδρίες:	
1000 εισερ.	6,43
1000 εξερ.	155
10000 εισερ.	64,3
10000 εξερ.	1550
50000 εισερ.	321,5
50000 εξερ.	7750
100000 εισερ.	643
100000 εξερ.	15500

Ενώ για την περίπτωση των servers 10Gbps

Πίνακας 22 Αριθμός servers 10Gbps για διαφορετικό πλήθος συνεδριών

Αριθμός servers 10 Gbs/ συνεδρίες:	
1000 εισερ.	0,643
1000 εξερ.	15,5
10000 εισερ.	6,43
10000 εξερ.	155
50000 εισερ.	32,15
50000 εξερ.	775
100000 εισερ.	64,3
100000 εξερ.	1550

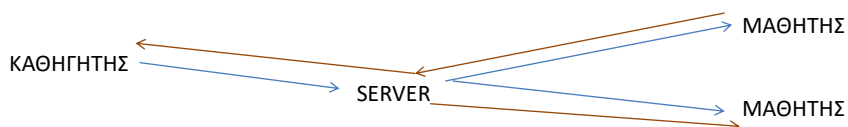
Τελικά όσον αφορά το κόστος για την περίπτωση της μικρής ανάλυσης προκύπτει ότι

Πίνακας 23 Κόστος υλοποίησης για τη μικρή ανάλυση στο σενάριο B

1 Gbps / ευρώ μήνα	400,00 €
Συνολικό κόστος / μήνα	3.100.000,00 €
10 Gbps / ευρώ μήνα	1500,00 €
Συνολικό κόστος / μήνα	1.162.500,00 €

Στο τρίτο σενάριο που θα μελετήσουμε υπάρχει μονόδρομη επικοινωνία από τον καθηγητή προς όλους τους μαθητές εκτός από έναν μαθητή όπου υπάρχει αμφίδρομη επικοινωνία. Πιο κάτω φαίνεται μια ενδεικτική υλοποίηση για 1 καθηγητή και 2 μαθητές

ΣΕΝΑΡΙΟ 3



Εικόνα 25 Μονόδρομη επικοινωνία σε όλους τους μαθητές εκτός από έναν

Παίρνοντας λοιπόν ως δεδομένα τα στοιχεία του παρακάτω πίνακα, όσον αφορά την αποστολή video, default ανάλυσης από τον καθηγητή, την αποστολή μικρής ανάλυσης καθώς και τη λήψη μικρής ανάλυσης από τους μαθητές:

Πίνακας 24 Μέση κίνηση ανά συνεδρία για το τρίτο σενάριο υλοποίησης

Σενάριο Γ	
Σενάριο:	2
Μέση Κίνηση / συνεδρία	
default ανάλυση	
Εξερχόμενη κίνηση καθηγητή	1,9
Εισερχόμενη κίνηση μαθητή	2,13
Εισερχόμενη κίνηση καθηγητή	0,23
Εισερχόμενη κίνηση server	2,13
Εξερχόμενη κίνηση server	53,1
Terabytes in / μήνα	0,1344
Terabytes out / μήνα	3,3453
Μικρή ανάλυση	
Εξερχόμενη κίνηση καθηγητή	0,6
Εισερχόμενη κίνηση μαθητή	0,83
Εισερχόμενη κίνηση καθηγητή	0,23
Εισερχόμενη κίνηση server	0,83
Εξερχόμενη κίνηση server	20,6
Terabytes in / μήνα	0,0525
Terabytes out / μήνα	1,2978

Αν θεωρήσουμε ότι διαθέτουμε servers με δυνατότητες 1 και 10Gbps τότε ο αριθμός των servers ανά συνεδρία που θα προκύψουν θα είναι:

Πίνακας 25 Αριθμός servers ανά συνεδρία για το τρίτο σενάριο υλοποίησης

Σενάριο Γ	
default ανάλυση	
Σενάρια Servers	Αριθμός servers / συνεδρία
A. 1Gbps Εισερχόμενη	0,00213
A. 1Gbps Εξερχόμενη	0,00213
B. 10 Gbps Εισερχόμενη	0,000213
B. 10 Gbps Εξερχόμενη	0,000213

Βασιζόμενοι στα προηγούμενα μπορούμε να υπολογίσουμε τον αριθμό servers χωρητικότητας 1Gbps για διάφορες τιμές συνεδριών οπότε θα προκύψουν τα ακόλουθα

Πίνακας 26 Αριθμός servers 1Gbps για διαφορετικό αριθμό συνεδριών

default ανάλυση	
Αριθμός servers 1 Gbs/ συνεδρίες:	
1000 εισερ.	2,13
1000 εξερ.	53,1
10000 εισερ.	21,3
10000 εξερ.	531
50000 εισερ.	106,5
50000 εξερ.	2655
100000 εισερ.	213
100000 εξερ.	5310

Με παρόμοιο τρόπο μπορούμε να υπολογίσουμε τον αριθμό servers χωρητικότητας 10Gbps για διάφορες τιμές συνεδριών οπότε θα έχουμε

Πίνακας 27 Αριθμός servers 10Gbps για διαφορετικό αριθμό συνεδριών

Αριθμός servers 10 Gbs/ συνεδρίες:	
1000	0,213
1000	5,31
10000	2,13
10000	53,1
50000	10,65
50000	265,5
100000	21,3
100000	531

Αν θέλαμε να υπολογίσουμε ένα σχετικό κόστος για την περίπτωση των servers χωρητικότητας 1Gbps και 10Gbps και για 50000 συνεδρίες θα προέκυπταν τα ακόλουθα με την υπόθεση ότι ο ένας server 1Gbps έχει ένα κόστος γύρω στα 400 ευρώ το μήνα ενώ ένας server 10Gbps περίπου 1500 ευρώ το μήνα

Πίνακας 28 Κόστος υλοποίησης για το τρίτο σενάριο

1 Gbps / ευρώ μήνα	400,00 €
Συνολικό κόστος / μήνα	1.062.000,00 €
10 Gbps / ευρώ μήνα	1500,00 €
Συνολικό κόστος / μήνα	399.000,00 €

Στην περίπτωση της μικρής ανάλυσης θα είχαμε

Πίνακας 29 Αριθμός servers ανά συνεδρία

Σενάριο Γ	
Μικρή ανάλυση	
Σενάρια Servers	Αριθμός servers / συνεδρία
A. 1Gbps Εισερχόμενη	0,00083
A. 1Gbps Εξερχόμενη	0,0206
B. 10 Gbps Εισερχόμενη	0,000083
B. 10 Gbps Εξερχόμενη	0,00206

Αντίστοιχα με πριν όσον αφορά τους servers χωρητικότητας 1Gbps

Πίνακας 30 Αριθμός servers 1Gbps για διαφορετικό πλήθος συνεδριών

Μικρή ανάλυση	
Αριθμός servers 1 Gbps/ συνεδρίες:	
1000	0,83
1000	20,6
10000	8,3
10000	206
50000	41,5
50000	1030
100000	83
100000	2060

Ενώ για την περίπτωση των servers 10Gbps

Πίνακας 31 Αριθμός servers 10Gbps για διαφορετικό πλήθος συνεδριών

Αριθμός servers 10 Gbps/ συνεδρίες:	
1000	0,083
1000	2,06
10000	0,83
10000	20,6
50000	4,15
50000	103
100000	8,3
100000	206

Τελικά όσον αφορά το κόστος για την περίπτωση της μικρής ανάλυσης προκύπτει ότι

Πίνακας 32 Κόστος υλοποίησης για τη μικρή ανάλυση στο σενάριο Γ

1 Gbps / ευρώ μήνα	400,00 €
Συνολικό κόστος / μήνα	412.000,00 €
10 Gbps / ευρώ μήνα	1500,00 €
Συνολικό κόστος / μήνα	154.500,00 €

5.ΣΥΜΠΕΡΑΣΜΑΤΑ

Όπως φαίνεται και από το τελευταίο κεφάλαιο όπου μελετούμε τις διαφορετικές περιπτώσεις τηλεδιάσκεψης για το σενάριο της δευτεροβάθμιας εκπαίδευσης, αυτό που προκύπτει είναι ότι όσο μεγαλύτερη ποσότητα δεδομένων θέλουμε να διακινήσουμε με τη μορφή περισσότερων δυνατοτήτων, για παράδειγμα αν θέλουμε να υπάρχει αμφίδρομη επικοινωνία μεταξύ καθηγητή και μαθητών και όχι μονόδρομη από τον καθηγητή, τόσο περισσότερο ανεβαίνουν οι ανάγκες μας σε υλικές υποδομές άρα τόσο περισσότερο αυξάνεται και το κόστος που θα πρέπει να καλύψουμε. Ανάλογα με τις δυνατότητες που θέλουμε να έχει το σύστημα που θα επιλέξουμε για υλοποίηση πρέπει να γίνει μια σωστή στάθμιση μεταξύ κόστους υλοποίησης και χαρακτηριστικών.

ΠΙΝΑΚΑΣ ΟΡΟΛΟΓΙΑΣ

Ξενόγλωσσος όρος	Ελληνικός Όρος
Peers	Ομότιμοι
Firewall	Τείχος προστασίας
Session	Συνεδρία
Signaling	Σηματοδοσία
Candidates	Υποψήφιοι
Handler	Χειριστής
Server	Διακομιστής
Relay	Αναμετάδοση
Metadata	Μεταδεδομένα
Router	Δρομολογητής
Bandwidth	Εύρος ζώνης

ΣΥΝΤΜΗΣΕΙΣ-ΑΡΚΤΙΚΟΛΕΞΑ-ΑΚΡΩΝΥΜΙΑ

RTC	Real Time Communication
API	Application Programming Interface
W3C	World Wide Web Consortium
NAT	Network Address Translation
IP	Internet Protocol
SIP	Session Initiation Protocol
XMPP	Extensible Messaging and Presence Protocol
P2P	Peer-To-Peer
ICE	Interactive Connectivity Establishment
SDP	Session Description Protocol
IETF	Internet Engineering Task Force
JSEP	Javascript Session Establishment Protocol
AGC	Automatic Gain Control
STUN	Session Traversal Utilities for NAT
TURN	Traversal Using Relay NAT
UDP	User Datagram Protocol
TCP	Transmission Control Protocol
MCU	Multipoint Control Unit
PSTN	Public Switched Telephone Network
VOIP	Voice Over IP
DTLS	Datagram Transport Layer Security

ΑΝΑΦΟΡΕΣ

- [1] Sam Dutton, “Getting started with webrtc”, February 2014; <https://www.html5rocks.com/en/tutorials/webrtc/basics/>
- [2] Kavirajan ST, “What is webrtc and how to setup stun/turn server for webrtc communication”, March 2020; <https://medium.com/av-transcode/what-is-webrtc-and-how-to-setup-stun-turn-server-for-webrtc-communication-63314728b9d0>
- [3] Sam Dutton, “WebRTC in the real world: STUN, TURN and signaling”, November 2013; <https://www.html5rocks.com/en/tutorials/webrtc/infrastructure/>
- [4] Malte Ubl and Eiji Kitamura, “Introducing Web Sockets: Bringing sockets to the web”, October 2010; <https://www.html5rocks.com/en/tutorials/websockets/basics/>
- [5] Jeff Tyson, “How Network Address Translation works”; <https://computer.howstuffworks.com/nat.htm>
- [6] Badri Rajasekar, “Mantis: Next-generation Cloud Technology for webrtc”, April 2013; <https://static.opentok.com/blog/mantis-next-generation-cloud-technology-for-webrtc/>
- [7] Wikipedia, Η Ελεύθερη Εγκυκλοπαίδεια; <https://www.wikipedia.org/>
- [8] J. Uberti, C. Jennings, “Javascript Session Establishment Protocol”, February 2013; <https://tools.ietf.org/html/draft-ietf-rtcweb-jsep-03#section-3.4.1>
- [9] Tsahi Levent-Levi, “How many users can fit in a webrtc call?”, January 2018; <https://bloggeek.me/how-many-users-webrtc-call/>
- [10] Jenny Liang, “What is CPaaS? Communications Platform as a Service Explained”, January 2020; <https://www.onsip.com/voip-resources/voip-fundamentals/what-is-cpaas-communications-platform-as-a-service-explained>