

Real Time Streaming Protocol

A. Δικτύωση πολυμέσων

Περιεχόμενα

1. Συνοπτική παρουσίαση του RTSP.....	4
1.1 Εισαγωγή.....	4
1.3 Οντότητες του RTSP.....	4
1.2 Βασικές λειτουργίες του RTSP.....	5
1.4 Ιδιότητες του RTSP.....	5
1.6 Ομοιότητες και διαφορές του RTSP με το HTTP.....	6
1.5 Μήνυμα RTSP.....	7
Μέθοδοι.....	7
2. Αναλυτική παρουσίαση του RTSP.....	11
2.1 Εισαγωγή.....	11
2.2 Λειτουργία, Δυνατότητες, Επεκτάσεις, Ασφάλεια.....	11
2.2.1 Λειτουργία.....	12
2.2.2 RTSP και HTTP.....	12
2.2.3 Δυνατότητες.....	13
2.2.4 Λειτουργικότητα ενός media server.....	14
2.2.5 Επέκταση του Πρωτοκόλλου.....	14
2.2.6 Θέματα ασφάλειας στο RTSP.....	14
2.3 RTSP Μηνύματα.....	15
2.3.1 Αίτηση.....	16
2.3.2 Απάντηση.....	17
2.4 Οι μέθοδοι στο RTSP.....	18
2.5 Κωδικοί κατάστασης.....	19
2.6 Καταστάσεις του RTSP.....	21
2.6.1 Μέθοδοι και Αλλαγή Κατάστασης.....	22
2.6.2 Μεταβάσεις Καταστάσεων.....	22
2.7 Ελάχιστη υλοποίηση του RTSP.....	24

2.7.1 Client.....	24
2.7.2 Server.....	24
2.8 Το RTSP σε λειτουργία.....	25
2.9 Παραδείγματα.....	26
2.9.1 Video on Demand (Unicast).....	27
2.9.2 Ζωντανή παρουσίαση με χρήση Multicast.....	30

1. Συνοπτική παρουσίαση του RTSP

Σημείωση: Μελετάτε την συνοπτική παρουσίαση και εφόσον απαιτείται ή επιθυμείτε να μελετήσετε λεπτομέρειες ανατρέχετε ή μελετάτε την αναλυτική παρουσίαση.

1.1 Εισαγωγή

Το RTSP (Real Time Streaming Protocol) είναι ένα πρωτόκολλο του επιπέδου εφαρμογής, υπεύθυνο για τον έλεγχο της μετάδοσης ροών δεδομένων σε πραγματικό χρόνο πάνω από IP δίκτυα. Η λειτουργία του βασίζεται στο μοντέλο πελάτη εξυπηρετητή και έχει χρησιμοποιήσει αρκετά στοιχεία από το HTTP. Δημιουργήθηκε στα πλαίσια των εργασιών της ομάδας εργασίας MMUSIC και τον Απρίλιο του 1998 δημοσιεύτηκε ως προτεινόμενο πρότυπο (RFC 2326) από την IETF.

Ο ρόλος του RTSP είναι η εγκατάσταση και ο έλεγχος μίας ή περισσότερων χρονικά συγχρονισμένων ροών συνεχόμενων δεδομένων όπως video και audio. Επίσης παρέχει ένα τρόπο επιλογής καναλιών διανομής όπως το UDP, TCP ή IP-multicast. Ο μηχανισμός διανομής βασίζεται αποκλειστικά στο RTP και αυτό γιατί το RTSP σχεδιάστηκε ώστε να βρίσκεται πάνω από το RTP και να ελέγχει και να παραδίδει δεδομένα πραγματικού χρόνου. Εκτός όμως από το RTP το RTSP μπορεί να χρησιμοποιηθεί μαζί με το RSVP για να εγκαταστήσει και να διαχειριστεί συνεδρίες ροών περιορισμένου εύρους.

1.3 Οντότητες του RTSP

- *RTSP Πελάτης:* ένας πελάτης στέλνει αιτήσεις σε έναν εξυπηρετητή (media server) προκειμένου να λάβει ροές δεδομένων. Θα θεωρήσουμε ότι ένας πελάτης μπορεί και να λαμβάνει αιτήσεις από έναν RTSP εξυπηρετητή προκειμένου να αρχίσει να παρουσιάζει τις ροές που λαμβάνει.
- *RTSP εξυπηρετητής:* σύμφωνα με το πρωτόκολλο είναι ο εξυπηρετητής που λαμβάνει αιτήσεις από πελάτες προκειμένου να επαναμεταδώσει αποθηκευμένες ροές δεδομένων ή να αναλάβει την εγγραφή ροών δεδομένων. Εκτός όμως από αυτή τη λειτουργία θα

Θεωρήσουμε ότι ένας RTSP εξυπηρετητής μπορεί και να στείλει αιτήσεις σε έναν πελάτη προκειμένου να αρχίσει να παρουσιάζει τις ροές που λαμβάνει.

1.2 Βασικές λειτουργίες του RTSP

Το πρωτόκολλο υποστηρίζει τις εξής λειτουργίες:

- *Ανάκτηση ροών από εξυπηρετητές (media servers):* Ο πελάτης μπορεί να ζητήσει την περιγραφή της παρουσίασης¹ μέσω HTTP ή με κάποια άλλη μέθοδο. Εάν η παρουσίαση είναι multicast, τότε η περιγραφή της παρουσίασης περιέχει τις multicast διευθύνσεις και τις θύρες που θα χρησιμοποιηθούν για τις συνεχόμενες ροές. Εάν η παρουσίαση είναι unicast, ο πελάτης παρέχει τις πληροφορίες του προορισμού για λόγους ασφάλειας.
- *Πρόσκληση εξυπηρετητή σε μία συνεδρία (conference):* Ένας εξυπηρετητής μπορεί να δεχτεί μία πρόσκληση προκειμένου να συμμετάσχει σε μία ήδη υπάρχουσα συνεδρία ή να αναμεταδώσει αποθηκευμένες ροές σε μία παρουσίαση ή και να κάνει εγγραφή ενός υποσυνόλου ροών ή και ολόκληρης της παρουσίασης.
- *Προσθήκη μίας ροής σε μία ήδη υπάρχουσα παρουσίαση:* Η λειτουργία αυτή είναι ιδιαίτερα χρήσιμη για ζωντανές παρουσιάσεις, εάν ο εξυπηρετητής μπορεί να ενημερώσει τον πελάτη για επιπλέον ροές που θα γίνουν διαθέσιμες.

1.4 Ιδιότητες του RTSP

Το RTSP έχει τις εξής ιδιότητες:

- *Επεκτάσιμο:* νέες μέθοδοι και παράμετροι μπορούν εύκολα να προστεθούν
- *Εύκολο στην ανάλυσή του:* μπορεί να αναλυθεί από HTTP ή MIME αναλυτές
- *Ασφαλές:* χρησιμοποιεί τους μηχανισμούς ασφάλειας που χρησιμοποιούνται στο διαδίκτυο. Οι μηχανισμοί digest ή basic πιστοποίησης που χρησιμοποιούνται στο HTTP είναι άμεσα εφαρμόσιμοι.

¹ Με τον όρο περιγραφή παρουσίασης εννοούμε την πληροφορία για μία ή περισσότερες ροές πληροφορίας που μεταδίδονται μέσα σε μία συγκεκριμένη παρουσίαση. Στο SDP χρησιμοποιείται ο όρος συνεδρία για μία ζωντανή παρουσίαση.

- *Ανεξάρτητο από τον μηχανισμό μεταφοράς:* το RTSP μπορεί να χρησιμοποιήσει τόσο ένα αναξιόπιστο πρωτόκολλο μεταφοράς όπως το UDP όσο και ένα αξιόπιστο (TCP), εφόσον παρέχει αξιοπιστία σε επίπεδο εφαρμογής.
- *Ικανότητα πολυ-εξυπηρετητών:* κάθε ροή δεδομένων μέσα σε μία παρουσίαση μπορεί να προέρχεται από διαφορετικό εξυπηρετητή. Ένας πελάτης επομένως μπορεί να εγκαταστήσει παράλληλα διαφορετικές συνεδρίες με διαφορετικούς εξυπηρετητές. Ο συγχρονισμός των ροών γίνεται στο επίπεδο μεταφοράς.
- *Έλεγχος συσκευών εγγραφής:* το πρωτόκολλο μπορεί να ελέγξει τόσο τις συσκευές εγγραφής όσο και συσκευές όπως οι VoD (Video on Demand) εξυπηρετητές.
- *Διαχωρισμός ελέγχου ροών και έναρξης της συνεδρίας:* ο έλεγχος των ροών σαν λειτουργία είναι ανεξάρτητη από την πρόσκληση ενός εξυπηρετητή σε μία συνεδρία. Βασική προϋπόθεση αποτελεί η παροχή ενός μοναδικού αναγνωριστικού της συνεδρίας από το πρωτόκολλο έναρξης συνεδρίας. Πιο συγκεκριμένα τόσο το H.323 όσο και το SIP μπορούν να χρησιμοποιηθούν προκειμένου να καλέσουν έναν εξυπηρετητή στην συνεδρία.
- *Έλεγχος του εξυπηρετητή:* εάν ένας πελάτης μπορεί να ξεκινήσει μία ροή, θα πρέπει να είναι σε θέση και να σταματήσει την ροή.

1.6 Ομοιότητες και διαφορές του RTSP με το HTTP

Το RTSP έχει βασιστεί αρκετά στη φιλοσοφία του HTTP. Τόσο οι πελάτες όσο και οι εξυπηρετητές στέλνουν αιτήσεις και λαμβάνουν αποκρίσεις. Παρά το γεγονός ότι τα δύο αυτά πρωτόκολλα έχουν πολλές ομοιότητες παρουσιάζουν τις εξής διαφορές:

- Το RTSP εισάγει ένα σύνολο από νέες μεθόδους.
- Ένας RTSP εξυπηρετητής πρέπει να διατηρεί κατάσταση εξ' ορισμού για όλες σχεδόν τις περιπτώσεις, κάτι το οποίο έρχεται σε αντίθεση με την stateless φύση του HTTP.
- Το RTSP χρησιμοποιεί κωδικοποίηση UTF-8 αντί για ISO 8859-1 που χρησιμοποιείται κατόπιν προσπαθειών διεθνοποίησης της HTML.
- Το RTSP URL είναι της μορφής `rtsp://media.example.com:554/twister/audiotrack`, όπου
 - `rtsp://` είναι το αναγνωριστικό για την TCP σύνδεση ενώ `rtspu://` για την UDP σύνδεση.
 - 554 είναι η θύρα που χρησιμοποιεί το πρωτόκολλο

- twister είναι το όνομα της παρουσίασης
- audiotrack είναι το όνομα μίας συγκεκριμένης ροής στην παρουσίαση

1.5 Μήνυμα RTSP

Η λειτουργία του πρωτοκόλλου βασίζεται στην ανταλλαγή RTSP μηνυμάτων μεταξύ πελατών και εξυπηρετητών. Ένα μήνυμα μπορεί να είναι είτε μία αίτηση είτε μία απόκριση. Κάθε μήνυμα αποτελείται από τρία μέλη. Εάν είναι αίτηση περιέχει μία μέθοδο, ένα σύνολο κεφαλίδων και ένα message body, ενώ εάν είναι απόκριση περιέχει αντί για μία μέθοδο έναν κωδικό κατάστασης συνοδευόμενο από μία φράση. Οι μέθοδοι που χρησιμοποιεί το RTSP είναι αναφορικά οι: DESCRIBE, ANNOUNCE, GET_PARAMETER, OPTIONS, PAUSE, PLAY, RECORD, REDIRECT, SETUP, SET_PARAMETER, TEARDOWN. Από τις μεθόδους αυτές θα αναλύσουμε τις πιο βασικές και αυτές που θα χρησιμοποιήσουμε στο επόμενο κεφάλαιο. Οι κεφαλίδες όπως και οι κωδικοί κατάστασης που χρησιμοποιούνται από το RTSP είναι αρκετές από αυτές που χρησιμοποιούνται από το HTTP.

Μέθοδοι

Θα περιγράψουμε περιληπτικά την κάθε μέθοδο δίνοντας ένα παράδειγμα για την καθεμία.

- **SETUP:** ένας πελάτης μπορεί να στείλει μία αίτηση SETUP σε έναν εξυπηρετητή προκειμένου να καθορίσει το μηχανισμό μεταφοράς που θα χρησιμοποιηθεί για τη μετάδοση της επιθυμητής ροής. Ο εξυπηρετητής σε απάντηση δημιουργεί ένα αναγνωριστικό για αυτή τη συνεδρία.

Παράδειγμα:

```
C->S: SETUP rtsp://example.com/foo/bar/baz.rm RTSP/1.0
      CSeq: 302
      Transport: RTP/AVP;unicast;client_port=4588-4589

S->C: RTSP/1.0 200 OK
      CSeq: 302
      Date: 23 Jan 1997 15:35:06 GMT
      Session: 47112344
      Transport: RTP/AVP;unicast;
```



```
client_port=4588-4589;server_port=6256-6257
```

Η αίτηση SETUP όπως έχει οριστεί στο πρότυπο έχει κατεύθυνση από έναν πελάτη σε έναν εξυπηρετητή.

- **PLAY:** Η μέθοδος αυτή δίνει εντολή σε έναν εξυπηρετητή να αρχίσει να στέλνει δεδομένα μέσω του μηχανισμού που καθορίστηκε από την αίτηση SETUP που προηγείται πάντα από μία αίτηση PLAY.

Παράδειγμα:

```
C->S: PLAY rtsp://example.com/foo/bar/baz.rm RTSP/1.0
```

```
CSeq: 303
```

```
Session: 47112344
```

```
Range: smpte=0:10:20-;time=19970123T153600Z
```

```
S->C: RTSP/1.0 200 OK
```

```
CSeq: 303
```

```
Date: 23 Jan 1997 15:35:06 GMT
```

```
Range: smpte=0:10:22-;time=19970123T153600Z
```

- **RECORD:** με την μέθοδο αυτή έχουμε έναρξη της εγγραφής ενός συνόλου ροών σύμφωνα με την περιγραφή παρουσίασης που έχει προηγηθεί. Ο εξυπηρετητής επιλέγει να αποθηκεύσει τα εγγεγραμμένα δεδομένα κάτω από το ίδιο Request-URI ή κάποιο διαφορετικό. Πριν από μία αίτηση RECORD πρέπει να έχει προηγηθεί μία SETUP.

Παράδειγμα:

Θεωρούμε ότι έχει προηγηθεί SETUP.

```
C->S: RECORD rtsp://example.com/foo/bar/baz.rm RTSP/1.0
```

```
CSeq: 304
```

```
Session: 47112344
```

Conference: 128.16.64.19/32492374

- **PAUSE:** σταματάει προσωρινά μία ροή χωρίς να αποδεσμεύει τους πόρους του εξυπηρετητή. Σύμφωνα με το πρότυπο, αίτηση PAUSE μπορεί να στείλει ένας πελάτης σε έναν εξυπηρετητή ο οποίος μεταδίδει δεδομένα.

Παράδειγμα:

```
C->S: PAUSE rtsp://example.com/foo/bar/baz.rm RTSP/1.0
```

```
CSeq: 305
```

```
Session: 47112344
```

```
S->C: RTSP/1.0 200 OK
```

```
CSeq: 305
```

```
Date: 23 Jan 1997 15:35:06 GMT
```

- **TEARDOWN:** ελευθερώνει τους πόρους που σχετίζονται με μία ροή δεδομένων και η RTSP συνεδρία δεν υπάρχει πλέον στον εξυπηρετητή.

Παράδειγμα:

```
C->S: TEARDOWN rtsp://example.com/foo/bar/baz.rm RTSP/1.0
```

```
CSeq: 306
```

```
Session: 47112344
```

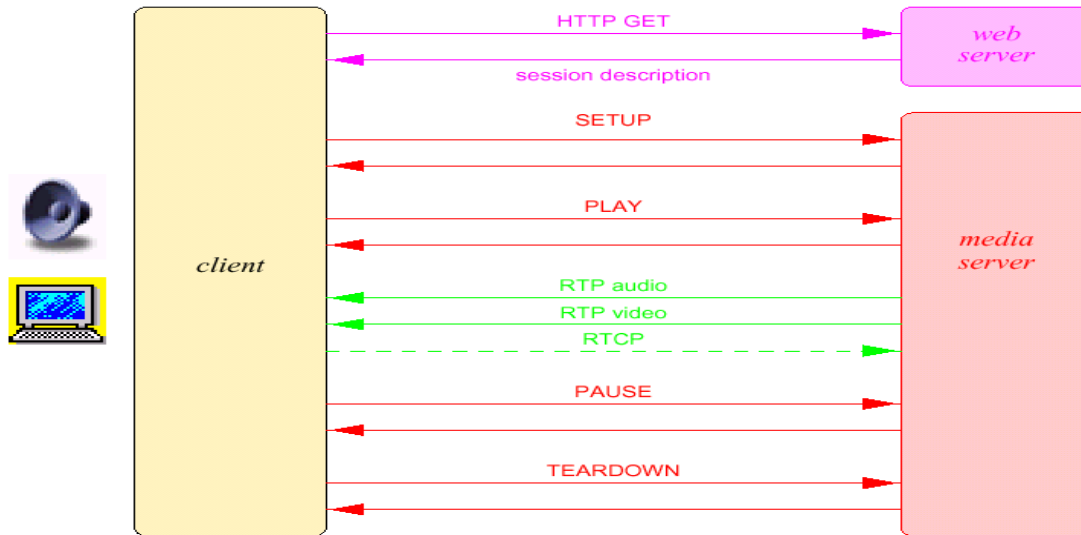
```
S->C: RTSP/1.0 200 OK
```

```
CSeq: 306
```

Από τα προηγούμενα παραδείγματα παρατηρούμε ότι κάθε φορά που στέλνουμε ένα νέο μήνυμα με διαφορετική μέθοδο αυξάνει το Cseq αλλά το αναγνωριστικό της συνεδρίας (Session) παραμένει το ίδιο.

Παράδειγμα λειτουργίας

Στο επόμενο σχήμα μπορούμε να διακρίνουμε συνοπτικά την λειτουργία του RTSP. Τα μηνύματα που ανταλλάσσονται έχουν τη μορφή των μηνυμάτων που παρουσιάστηκαν προηγουμένως.



2. Αναλυτική παρουσίαση του RTSP

Παρουσιάζονται η φιλοσοφία του πρωτοκόλλου, ο σκοπός, οι δυνατότητές του και οι λειτουργίες του και αναλύονται με περιορισμένη λεπτομέρεια τα χαρακτηριστικά της υλοποίησής του, με ορισμένη έμφαση στην επεκτασιμότητά του και τη συνεργασία του με άλλα πρωτόκολλα. Τέλος, παρουσιάζονται δύο παραδειγματικά σενάρια επικοινωνίας ενός client και ενός server μέσω RTSP μηνυμάτων.

2.1 Εισαγωγή

Το Real Time Streaming Protocol (RTSP) είναι ένα πρωτόκολλο του επιπέδου εφαρμογών για τον έλεγχο της μεταφοράς δεδομένων με ιδιότητες πραγματικού χρόνου (real-time). Προσφέρει ένα ευρύ φάσμα δυνατοτήτων, για να παρέχει έλεγχο σε κατ' απαίτηση (on-demand) μεταφορά δεδομένων πραγματικού χρόνου, όπως ήχο και βίντεο. Τα δεδομένα μπορεί να προέρχονται είτε από ζωντανές μεταδόσεις, είτε να είναι αποθηκευμένα. Το πρωτόκολλο είναι σχεδιασμένο ώστε να ελέγχει πολλαπλές συνόδους (sessions) μεταφοράς δεδομένων, να παρέχει τη δυνατότητα για επιλογή του καναλιού μεταφοράς μηνυμάτων, όπως UDP, multicast UDP ή TCP, καθώς και τη δυνατότητα για επιλογή του μηχανισμού μεταφοράς, όπως για παράδειγμα το RTP.

2.2 Λειτουργία, Δυνατότητες, Επεκτάσεις, Ασφάλεια

Εδώ θα αναφερθούν σε ενότητες ο βασικός προσανατολισμός του RTSP, ο τρόπος λειτουργίας του, η σχέση του με το HTTP, οι δυνατότητές του για υποστήριξη διαφορετικών εφαρμογών-υπηρεσιών, η επεκτασιμότητά του και θέματα ασφάλειας του πρωτοκόλλου.

2.2.1 Λειτουργία

Το RTSP εγκαθιδρύει και ελέγχει μία ή περισσότερες χρονικά συγχρονισμένες (time-synchronized) ροές (streams) μη στατικών μέσων, όπως ήχο και βίντεο. Τυπικά, δεν είναι αυτό που κάνει τη μεταφορά της συνεχούς ροής, ωστόσο είναι δυνατή η αλληλοπαρεμβολή της συνεχούς ροής του μέσου με τη ροή ελέγχου. Με άλλα λόγια, το RTSP παίζει το ρόλο του «δικτυακού τηλεχειριστηρίου» ("network remote control") από την πλευρά του χρήστη, για εξυπηρετητές πολυμέσων (multimedia servers). Το σύνολο των ροών που υπόκεινται σε έλεγχο από το RTSP ορίζεται από μια «περιγραφή παρουσίασης» (presentation description).

Δεν έχει νόημα η αναφορά σε μία RTSP σύνδεση. Στη πραγματικότητα ο server εγκαθιστά και διατηρεί μία σύνοδο με τον client, στην οποία προσάπτει ένα αναγνωριστικό (session identifier). Μία RTSP σύνοδος δεν είναι σε καμία περίπτωση συνυφασμένη με μία σύνδεση επιπέδου μεταφοράς όπως μία σύνδεση TCP. Κατά τη διάρκεια μιας συνόδου, ένας RTSP client μπορεί να ανοίξει και να κλείσει πολλές αξιόπιστες συνδέσεις μεταφοράς (TCP) προς το sever και να εκδώσει RTSP αιτήσεις. Εναλλακτικά, μπορεί να χρησιμοποιήσει ένα πρωτόκολλο μεταφοράς χωρίς σύνδεση (connectionless) όπως είναι το UDP.

Οι ροές που ελέγχονται από το RTSP είναι δυνατόν (και συνήθως αυτό συμβαίνει) να χρησιμοποιούν το RTP, αλλά η λειτουργία του δεν εξαρτάται σε καμία περίπτωση από τον μηχανισμό που χρησιμοποιείται για να μεταφέρει τα συνεχή μέσα. Τέλος, οι πιθανώς διαφορετικές ροές μέσων που αποτελούν μια παρουσίαση, μπορεί να προέρχονται από διαφορετικούς servers.

2.2.2 RTSP και HTTP

Το πρωτόκολλο είναι παρόμοιο στη σύνταξη και τη λειτουργία με το HTTP/1.1, έτσι ώστε οι μηχανισμοί επέκτασής του μπορούν στις περισσότερες περιπτώσεις να εφαρμοστούν και στο RTSP. Ωστόσο, το RTSP διαφέρει σε ορισμένα σημαντικά θέματα από το HTTP:

- Το RTSP εισάγει έναν μεγάλο αριθμό από νέες μεθόδους και έχει ένα διαφορετικό αναγνωριστικό πρωτοκόλλου.

- Ένας RTSP server χρειάζεται εξ ορισμού να διατηρεί την κατάστασή του (state) σε όλες σχεδόν τις περιπτώσεις, αντίθετα από την χωρίς-κατάσταση (stateless) φύση του HTTP.
- Οι RTSP server και client μπορούν και οι δύο να εκδώσουν αιτήσεις.
- Τα πραγματικά δεδομένα μεταφέρονται από ένα διαφορετικό πρωτόκολλο.
- Το Request-URI πάντα χρησιμοποιεί το απόλυτο URI. Λόγω της προς-τα-πίσω συμβατότητας (backward compatibility), το HTTP/1.1 μεταφέρει μόνο το απόλυτο μονοπάτι στην αίτηση και βάζει το host name σε ένα διαφορετικό πεδίο της επικεφαλίδας (header field).

Αυτό κάνει ευκολότερη την «εικονική φιλοξενία» (“virtual hosting”), όπου ένας μόνο host με μία IP διεύθυνση φιλοξενεί διάφορα δέντρα εγγράφων (document trees).

Οι RTSP αιτήσεις μπορεί να διαχειριστούν από proxies, tunnels και caches, όπως στο HTTP/1.1 .

2.2.3 Δυνατότητες

Το πρωτόκολλο έχει τη δυνατότητα για την υποστήριξη των παρακάτω λειτουργιών:

- Ανάκτηση μέσω από έναν media server:
Ο client μπορεί να ζητήσει μια περιγραφή της παρουσίασης μέσω HTTP ή κάποιας άλλης μεθόδου. Αν η παρουσίαση είναι multicast, η περιγραφή της παρουσίασης περιέχει τις multicast διευθύνσεις και τα ports που πρέπει να χρησιμοποιηθούν. Αν η παρουσίαση πρόκειται να σταλεί μόνο στον συγκεκριμένο client μέσω unicast, ο client παρέχει τον προορισμό για λόγους ασφάλειας.
- Πρόσκληση ενός media server σε μια σύνοδο (conference):
Ένας media server μπορεί να «προσκληθεί» να συμμετάσχει σε μία υπάρχουσα σύνοδο, είτε για να παίξει κάποιο μέσο στην παρουσίαση είτε για να εγγράψει (record) όλα ή ένα υποσύνολο από τα μέσα (βίντεο ή ήχο) μίας παρουσίασης. Αυτός ο τρόπος λειτουργίας είναι χρήσιμος για κατανεμημένες εκπαιδευτικές εφαρμογές. Τα διάφορα μέρη που συμμετέχουν στη παρουσίαση μπορούν να πάρουν σειρά πατώντας τα πλήκτρα του «τηλεχειριστηρίου».
- Εισαγωγή επιπλέον μέσω σε μία υπάρχουσα παρουσίαση:

Ειδικά για ζωντανές (live) παρουσιάσεις, είναι χρήσιμο αν ο sever μπορεί να δηλώσει στον client ότι επιπρόσθετα μέσα είναι ή γίνονται διαθέσιμα.

2.2.4 Λειτουργικότητα ενός media server

Επειδή δεν έχουν όλοι οι media servers την ίδια λειτουργικότητα, δεν υποστηρίζουν κατ' ανάγκη το ίδιο σύνολο αιτήσεων. Για παράδειγμα:

- Ένας server που έχει τη δυνατότητα μόνο να παίζει (playback) κάποια μέσα, δεν χρειάζεται να υποστηρίζει την αίτηση εγγραφής (RECORD).
- Ένας server που υποστηρίζει μόνο ζωντανές μεταδόσεις, δεν είναι απαραίτητο να κάνει seeking (μετακίνηση στη χρονική θέση της παρουσίασης).
- Κάποιοι servers μπορεί να μην υποστηρίζουν την αλλαγή των παραμέτρων ροής, οπότε δεν υλοποιούν τις αιτήσεις SET_PARAMETER και GET_PARAMETER.

Εναπόκειται στους δημιουργούς των περιγραφών παρουσίασης να μην απαιτούν το ανέφικτο από έναν server.

2.2.5 Επέκταση του Πρωτοκόλλου

Το RTSP μπορεί να επεκταθεί με τρεις τρόπους:

- Στις υπάρχουσες μεθόδους μπορούν να προστεθούν νέες παράμετροι, αρκεί αυτές να μπορούν χωρίς πρόβλημα να αγνοηθούν από τον παραλήπτη.
- Μπορούν να προστεθούν νέες μέθοδοι. Αν ο παραλήπτης του μηνύματος δεν κατανοεί την αίτηση, απαντά με τον κωδικό λάθους 501 (Μη Υλοποιημένο) και ο αποστολέας δε θα πρέπει να προσπαθήσει να χρησιμοποιήσει ξανά αυτή τη μέθοδο. Ο client μπορεί επίσης να χρησιμοποιήσει τη μέθοδο OPTIONS για να ρωτήσει το server ποιες μεθόδους υποστηρίζει.
- Μια νέα έκδοση του πρωτοκόλλου μπορεί να οριστεί, επιφέροντας αλλαγές σε όλες σχεδόν τις παραμέτρους (εκτός από τη θέση του αριθμού έκδοσης του πρωτοκόλλου).
-

2.2.6 Θέματα ασφάλειας στο RTSP

Γενικά, το RTSP επαναχρησιμοποιεί τους μηχανισμούς ασφάλειας του web. Οι βασικές μέθοδοι αυθεντικοποίησης του HTTP, όπως η βασική (basic) και η «συνοπτική» (digest) εφαρμόζονται

και στο RTSP. Βεβαίως, μπορούν να χρησιμοποιηθούν και μηχανισμοί ασφάλειας των επιπέδων δικτύου και μεταφοράς. Τέλος, οι υλοποιήσεις των servers θα πρέπει να λαμβάνουν υπόψη τους και μια σειρά ζητημάτων ασφάλειας -όμοιων στις περισσότερες περιπτώσεις με το HTTP-, όπως την αποφυγή επιθέσεων denial-of-service, την προστασία προσωπικών ή ευαίσθητων πληροφοριών που μεταφέρονται μέσω των RTSP μηνυμάτων, τις επιθέσεις στο σύστημα αρχείων (λόγω των διακινούμενων URIs), την υποκλοπή του αναγνωριστικού συνόδου. Πρέπει να σημειωθεί επίσης ότι, καθώς το RTSP παρέχει μόνο τον έλεγχο των ροών και όχι την μεταφορά τους (αυτή την αναλαμβάνουν άλλα πρωτόκολλα), δεν υπεισέρχεται σε θέματα ασφάλειας των διακινούμενων μέσων.

2.3 RTSP Μηνύματα

Το RTSP είναι ένα βασισμένο-σε-κείμενο (text-based) πρωτόκολλο και χρησιμοποιεί το σύνολο χαρακτήρων κατά ISO 10646 με κωδικοποίηση UTF-8. Η βασική μονάδα της επικοινωνίας μέσω του RTSP είναι το μήνυμα, που μπορεί να μεταδοθεί μέσω ενός πρωτοκόλλου με ή χωρίς σύνδεση. Οι γραμμές τερματίζονται με CRLF, αλλά οι αποδέκτες θα πρέπει να είναι ικανοί να ερμηνεύουν τα CR (Carriage Return) και LF (Line Feed) και από μόνα τους ως τερματισμούς γραμμών.

Τα βασισμένα-σε-κείμενο πρωτόκολλα διευκολύνουν την εισαγωγή προαιρετικών παραμέτρων με έναν αυτοπεριγραφικό τρόπο. Επειδή ο αριθμός των παραμέτρων είναι μικρός και η συχνότητα των εντολών είναι χαμηλή, δεν μας ανησυχεί η αποδοτικότητα της επεξεργασίας των μηνυμάτων κειμένου. Επίσης, ένα πλεονέκτημα των βασισμένων-σε-κείμενο πρωτοκόλλων αποτελεί η εύκολη επεκτασιμότητά τους.

Τα RTSP μηνύματα μπορούν να μεταφερθούν πάνω από οποιοδήποτε χαμηλού επιπέδου πρωτόκολλο που είναι 8-bit clean.

Τα RTSP μηνύματα είναι είτε αιτήσεις (συνήθως από τον client στον server, αλλά πιθανώς και αντίστροφα), είτε απαντήσεις (συνήθως από τον server στον client, αλλά σε ορισμένες

περιπτώσεις και αντίστροφα). Επειδή στην παρούσα εργασία μας απασχολεί και η υλοποίηση του πρωτοκόλλου, θα πρέπει να σημειώσουμε ότι το κυρίως βάρος στην υλοποίηση RTSP clients και servers αποτελεί η κατασκευή και η συντακτική ανάλυση (parsing) των μηνυμάτων από τις δύο πλευρές. Η δομή και η σύνταξή τους παρουσιάζεται συνοπτικά στη συνέχεια.

2.3.1 Αίτηση

Ένα μήνυμα αίτησης από τον client στο server ή αντίστροφα, περιλαμβάνει στην πρώτη γραμμή του (Request-Line) τη μέθοδο που θα εφαρμοστεί και το αντικείμενο στο οποίο ζητείται να εφαρμοστεί η μέθοδος και την έκδοση του πρωτοκόλλου. Στις επόμενες γραμμές περιλαμβάνονται επικεφαλίδες που καθορίζουν κάποιες επιθυμητές παραμέτρους του αντικειμένου και του τρόπου εξυπηρέτησης της μεθόδου. Η δομή ενός μηνύματος αίτησης, όπως διατυπώνεται στο RFC 2326, είναι η εξής:

```
Request = Request - Line
        *( general - header
          | request - header
          | entity - header )
        CRLF
        [ message - body ]
```

Η δομή της γραμμής αίτησης, όπως διατυπώνεται στο RFC 2326, είναι η εξής:

```
Request-Line = Method SP Request-URI SP RTSP-Version CRLF
```

Με "SP" δηλώνεται ο χαρακτήρας κενό και με "CRLF" ο τερματισμός γραμμής.

Παράδειγμα μηνύματος αίτησης:

```
SETUP rtsp://video.example.com/twister/video RTSP/1.0
CSeq: 1
Transport: RTP/AVP/UDP;unicast;client_port=3058-3059
```

2.3.2 Απάντηση

Αφού λάβει και ερμηνεύσει ένα μήνυμα αίτησης, ο παραλήπτης απαντάει με ένα μήνυμα απάντησης, η πρώτη γραμμή του οποίου (Status-Line) αποτελείται από την έκδοση του πρωτοκόλλου, ακολουθούμενη από έναν αριθμητικό κωδικό κατάστασης (Status-Code) και μια φράση κειμένου που επεξηγεί τον κωδικό κατάστασης, δηλαδή αν διαχειρίστηκε επιτυχώς την αίτηση ή όχι και τον λόγο της αποτυχίας. Ακολουθούν επικεφαλίδες που καθορίζουν παραμέτρους του αντικειμένου και λεπτομέρειες του τρόπου διαχείρισής του. Η δομή ενός μηνύματος απάντησης, όπως διατυπώνεται στο RFC 2326, είναι η εξής:

```
Response = Status - Line
          *( general - header
            | response - header
            | entity - header  )
          CRLF
          [ message - body ]
```

Η δομή της γραμμής κατάστασης, όπως διατυπώνεται στο RFC 2326, είναι η εξής:

```
Status-Line = RTSP-Version SP Status-Code SP Reason-Phrase CRLF
```

Με "SP" δηλώνεται ο χαρακτήρας κενό και με "CRLF" ο τερματισμός γραμμής.

Παράδειγμα μηνύματος απάντησης:

```
RTSP/1.0 200 OK

CSeq: 3

Session: 0456804596
```

2.4 Οι μέθοδοι στο RTSP

Οι μέθοδοι ορίζουν την ενέργεια που θα πραγματοποιηθεί πάνω στον πόρο που υποδεικνύεται στην RTSP αίτηση. Ως «πόρος» συνήθως εννοείται ένα αρχείο μέσου (π.χ. βίντεο, ήχος).

Μέθοδος	Κατεύθυνση	Αντικείμενο	Υλοποίηση
DESCRIBE	C→S	P, S	προτεινόμενη
ANNOUNCE	C→S, S→C	P, S	Προαιρετική
GET_PARAMETER	C→S, S→C	P, S	Προαιρετική
OPTIONS	C→S, S→C	P, S	Υποχρεωτική (S→C: προαιρετική)
PAUSE	C→S	P, S	προτεινόμενη
PLAY	C→S	P, S	Υποχρεωτική
RECORD	C→S	P, S	Προαιρετική
REDIRECT	S→C	P, S	Προαιρετική
SETUP	C→S	S	Υποχρεωτική
SET_PARAMETER	C→S, S→C	P, S	Προαιρετική
TEARDOWN	C→S	P, S	Υποχρεωτική

Πίνακας 2.1: Οι RTSP μέθοδοι, η κατεύθυνσή τους (S: server, C: client), τα αντικείμενα (P: presentation, S: stream) πάνω στα οποία εφαρμόζονται και η απαίτηση για υλοποίησή τους.

Οι μέθοδοι αποτελούν την κύρια έκφραση της επικοινωνίας στο RTSP, αφού δηλώνουν ρητά τις ενέργειες χειρισμού που γίνονται πάνω σε συγκεκριμένα αντικείμενα (π.χ. αρχεία πολυμέσων). Μια καταγραφή των μεθόδων γίνεται στον πίνακα 3.1.

Παρακάτω ακολουθεί μια σύντομη περιγραφή -κυρίως της λειτουργικότητας- των RTSP μεθόδων.

- OPTIONS: Παρουσιάζει τις υποστηριζόμενες RTSP μεθόδους.
- DESCRIBE: Παρέχει την περιγραφή (συνήθως με SDP σύνταξη) της παρουσίασης ή του μέσου που έχει ζητηθεί στον server.
- ANNOUNCE: Μεταβάλλει την περιγραφή της παρουσίασης, κατά τη διάρκεια της RTSP συνόδου.
- SETUP: Προσδιορίζει και εγκαθιστά το μηχανισμό και τις παραμέτρους μεταφοράς των ροών των μέσων.
- PLAY: Δίνει εντολή στον server να ξεκινήσει τη μετάδοση του μέσου που έχει ζητηθεί. Πρέπει οπωσδήποτε να έχει προηγηθεί το SETUP.
- PAUSE: Διακόπτει προσωρινά τη μεταφορά της ροής του μέσου ή των μέσων που αποτελούν την παρουσίαση, ο server όμως δεν αποδεσμεύει τους σχετικούς πόρους.
- TEARDOWN: Τερματίζει τη μεταφορά των ροών και ελευθερώνει όλους τους σχετικούς πόρους. Στην ουσία διακόπτεται η υπάρχουσα RTSP σύνοδος.
- GET_PARAMETER: Ζητάει τις τιμές μιας ή περισσότερων παραμέτρων που σχετίζονται με μια παρουσίαση ή ένα μέσο.
- SET_PARAMETER: Θέτει την τιμή μιας παραμέτρου.
- REDIRECT: Ανακατευθύνει τον client προς έναν άλλο server.

2.5 Κωδικοί κατάστασης

Ο κωδικός κατάστασης (Status-Code) είναι ένας τριψήφιος ακέραιος που απεικονίζει το αποτέλεσμα της προσπάθειας του server να κατανοήσει και να ικανοποιήσει μία αίτηση που

δέχτηκε από τον client (ή αντίστροφα σε ορισμένες περιπτώσεις). Η επεξηγηματική φράση (Reason-Phrase) παρέχει μια επεξήγηση του κωδικού κατάστασης με μορφή κειμένου. Ο client δεν είναι υποχρεωμένος να εξετάσει, να ερμηνεύσει ή να δείξει στο χρήστη την επεξηγηματική φράση.

Το πρώτο ψηφίο του κωδικού κατάστασης καθορίζει την κλάση της απάντησης, ενώ τα δύο τελευταία ψηφία δεν προσδίδουν κάποια κατηγοριοποίηση. Υπάρχουν πέντε τιμές για το πρώτο ψηφίο:

- 1xx: Πληροφοριακό – Η αίτηση παρελήφθη, συνεχίζεται η επεξεργασία.
- 2xx: Επιτυχία – Η ενέργεια παρελήφθη επιτυχώς, κατανοήθηκε και έγινε αποδεκτή.
- 3xx: Ανακατεύθυνση – Πρέπει να γίνουν περαιτέρω ενέργειες ώστε να ολοκληρωθεί η αίτηση.
- 4xx: Λάθος του client – Η αίτηση περιέχει λάθος σύνταξη ή δεν μπορεί να περατωθεί.
- 5xx: Λάθος του server – Ο server απέτυχε να περατώσει μια έγκυρη αίτηση.

Παρακάτω (πίνακας 3.2) παρουσιάζεται ένα ενδεικτικό σύνολο ορισμένων κωδικών κατάστασης για το RTSP/1.0, καθώς και οι αντίστοιχες επεξηγηματικές φράσεις, οι οποίες είναι μόνο ενδεικτικές και μπορούν να αντικατασταθούν από άλλες χωρίς να επηρεάσουν το πρωτόκολλο. Σημειώνεται ότι το RTSP υιοθετεί τους περισσότερους κωδικούς κατάστασης του HTTP/1.1 και προσθέτει κάποιους επιπλέον αρχίζοντας την αρίθμηση από το x50, ώστε να αποφευχθούν επικαλύψεις με νέους κωδικούς κατάστασης του HTTP.

Κωδικός Κατάστασης	Επεξηγηματική Φράση
100	Continue
200	OK
302	Moved Temporarily

400	Bad Request
404	Not Found
415	Unsupported Media Type
455	Method Not Valid in This State
500	Internal Server Error
501	Not Implemented
505	RTSP Version not supported

Πίνακας 2.2: Μερικοί ενδεικτικοί Κωδικοί Κατάστασης και οι αντίστοιχες Επεξηγηματικές Φράσεις στο RTSP.

Βεβαίως οι κωδικοί κατάστασης είναι επεκτάσιμοι. Οι εφαρμογές που υλοποιούν το RTSP δεν είναι απαραίτητο να κατανοούν τη σημασία όλων των καταχωρημένων κωδικών κατάστασης. Ωστόσο, θα πρέπει να καταλαβαίνουν την κλάση κάθε κωδικού κατάστασης, όπως αυτή υποδεικνύεται από το πρώτο ψηφίο, και να χειρίζονται οποιαδήποτε μη αναγνωρίσιμη απάντηση ως ισοδύναμη προς τον x00 κωδικό κατάστασης της ίδιας κλάσης.

2.6 Καταστάσεις του RTSP

Οι καταστάσεις σε έναν RTSP server περιγράφουν κατά κάποιον τρόπο τη συμπεριφορά του πρωτοκόλλου, από την αρχικοποίηση μιας RTSP συνόδου μέχρι τον τερματισμό της. Το RTSP ελέγχει μια ροή, η οποία μπορεί να σταλεί μέσω ενός διαφορετικού πρωτοκόλλου, ανεξάρτητα

από το κανάλι ελέγχου. Για παράδειγμα, ο RTSP έλεγχος μπορεί να συμβαίνει μέσω μιας TCP σύνδεσης, ενώ η ροή των δεδομένων να γίνεται μέσω UDP. Έτσι, η μεταφορά των δεδομένων συνεχίζεται, ακόμη κι αν ο media server δεν λαμβάνει RTSP αιτήσεις. Επίσης, κατά τη διάρκεια της ζωής της, μια ροή μέσου μπορεί να ελεγχθεί από RTSP αιτήσεις που εκδίδονται ακολουθιακά από διαφορετικές TCP συνδέσεις. Έτσι, ο server χρειάζεται να διατηρεί «κατάσταση συνόδου» (session state), για να είναι ικανός να συσχετίζει RTSP αιτήσεις με μία ροή. Αντιστοίχιση καταστάσεων είναι δυνατόν να γίνει και για τον client, όμως εξαρτάται πολύ από την κάθε υλοποίηση και δεν επηρεάζει τόσο τη λειτουργία και τη συμβατότητά του.

2.6.1 Μέθοδοι και Αλλαγή Κατάστασης

Πολλές μέθοδοι στο RTSP δεν συμβάλλουν στην αλλαγή κατάστασης. Ωστόσο, οι ακόλουθες παίζουν κεντρικό ρόλο στην εύρεση και τη δέσμευση πόρων του server και οδηγούν στην μεταβολή της κατάστασής του:

- SETUP: Κάνει το server να βρει και να καθορίσει πόρους για μια ροή και να ξεκινήσει μια RTSP σύνοδο.
- PLAY και RECORD: Ξεκινά την μετάδοση δεδομένων σε μια ροή που έχει καθοριστεί μέσω του SETUP.
- PAUSE: Σταματάει προσωρινά μια ροή χωρίς να ελευθερώσει τους πόρους του server.
- TEARDOWN: Ελευθερώνει τους πόρους που σχετίζονται με μια ροή. Η RTSP σύνοδος παύει να υπάρχει στο server.

Οι RTSP μέθοδοι που συμβάλλουν στην αλλαγή της κατάστασης χρησιμοποιούν το πεδίο επικεφαλίδας Session, ώστε να καθορίσουν την RTSP σύνοδο της οποίας η κατάσταση μεταβάλλεται. Ο server παράγει αναγνωριστικά συνόδου όταν απαντά σε αιτήσεις SETUP.

Οι αιτήσεις OPTIONS, ANNOUNCE, DESCRIBE, GET_PARAMETER, SET_PARAMETER δεν έχουν καμία επίδραση στην αλλαγή της κατάστασης του client ή του server.

2.6.2 Μεταβάσεις Καταστάσεων

Γενικά, ο server αλλάζει κατάσταση όταν λαμβάνει αιτήσεις. Μπορούν να θεωρηθούν οι παρακάτω καταστάσεις:

- Init: Η αρχική κατάσταση, όπου δεν έχει παραληφθεί ακόμη αίτηση SETUP.
- Ready: Ένα SETUP παραλήφθηκε επιτυχώς και στάλθηκε απάντηση ή ένα PAUSE παραλήφθηκε επιτυχώς και στάλθηκε απάντηση.
- Playing: Ένα PLAY παραλήφθηκε επιτυχώς και στάλθηκε απάντηση. Άρχισε αποστολή δεδομένων.
- Recording: Ο server εγγράφει δεδομένα.

Στον πίνακα 2.3 φαίνονται οι μεταβάσεις καταστάσεις που συμβαίνουν ανάλογα με τις αιτήσεις που λαμβάνονται. Η στήλη «Επόμενη Κατάσταση» δείχνει την κατάσταση στην οποία μεταβαίνει ο server αφού στείλει απάντηση επιτυχίας (2xx). Αν μια αίτηση έχει ως αποτέλεσμα έναν κωδικό 3xx, η κατάσταση γίνεται Init, ενώ ένας κωδικός 4xx δεν μεταβάλλει την κατάσταση.

Κατάσταση	Λαμβανόμενο Μήνυμα	Επόμενη Κατάσταση
Init	SETUP	Ready
	TEARDOWN	Init
Ready	PLAY	Playing
	SETUP	Ready
	TEARDOWN	Init
	RECORD	Recording
Playing	PLAY	Playing
	PAUSE	Ready
	TEARDOWN	Init
	SETUP	Playing
Recording	RECORD	Recording
	PAUSE	Ready
	TEARDOWN	Init
	SETUP	Recording

Πίνακας 3.3: Μεταβάσεις καταστάσεων στον RTSP server ανάλογα με το λαμβανόμενο μήνυμα

2.7 Ελάχιστη υλοποίηση του RTSP

Παρουσιάζονται τα ελάχιστα χαρακτηριστικά που πρέπει να έχουν οι RTSP clients και servers, ώστε να θεωρείται ότι υλοποιούν το RTSP.

2.7.1 Client

Η υλοποίηση ενός RTSP client θα πρέπει να περιλαμβάνει τα ακόλουθα:

- Να παράγει τις αιτήσεις SETUP, TEARDOWN, και μία τουλάχιστον από τις PLAY και RECORD. Αν υλοποιείται η RECORD, θα πρέπει να υλοποιείται και η ANNOUNCE.
- Να περιλαμβάνει τις ακόλουθες επικεφαλίδες στις αιτήσεις: Cseq, Connection, Session, Transport. Αν υλοποιείται η ANNOUNCE, θα πρέπει να υπάρχει η δυνατότητα να περιλαμβάνει και τις επικεφαλίδες: Content-Language, Content-Encoding, Content-Length και Content-Type.
- Να μπορεί να αναλύσει συντακτικά τις εξής επικεφαλίδες στις απαντήσεις του server: Cseq, Connection, Session, Transport, Content-Language, Content-Encoding, Content-Length, Content-Type. Αν υλοποιείται η RECORD, η επικεφαλίδα Location θα πρέπει επίσης να είναι κατανοητή. Τέλος, οι υλοποιήσεις που είναι συμβατές με το RTP θα πρέπει να κατανοούν την επικεφαλίδα RTP-Info.
- Να κατανοεί την κλάση κάθε κωδικού λάθους που λαμβάνεται και να ενημερώνει το χρήστη για τους κωδικούς λάθους των κλάσεων 4xx και 5xx.
- Να μπορεί να απαντήσει σε ασύγχρονες αιτήσεις από το sever, όπως είναι η ANNOUNCE. Αυτό δεν σημαίνει ότι θα πρέπει να την υλοποιεί, αλλά ότι θα είναι σε θέση να απαντήσει θετικά ή αρνητικά σε οποιαδήποτε αίτηση λαμβάνει από το server.

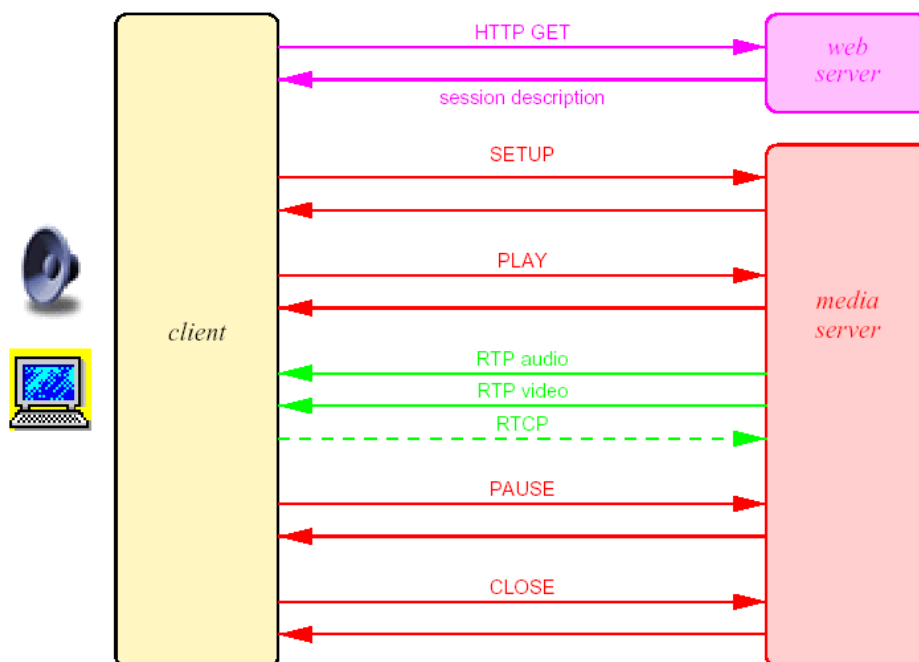
2.7.2 Server

Η υλοποίηση ενός RTSP server θα πρέπει να περιλαμβάνει τα ακόλουθα:

- Να υλοποιεί τις μεθόδους SETUP, TEARDOWN, OPTIONS και μία τουλάχιστον από τις PLAY και RECORD. Αν υλοποιείται η RECORD, θα πρέπει να υλοποιείται και η ANNOUNCE.
- Να περιλαμβάνει τις ακόλουθες επικεφαλίδες στις απαντήσεις: Connection, Content-Language, Content-Encoding, Content-Length, Content-Type, Public, Transport. Αν υλοποιείται η RECORD, θα πρέπει περιλαμβάνει και την επικεφαλίδα Location. Οι RTP-συμβατές υλοποιήσεις θα πρέπει να υλοποιούν και την επικεφαλίδα RTP-Info.
- Να μπορεί να αναλύσει συντακτικά και να απαντήσει κατάλληλα στις εξής επικεφαλίδες των αιτήσεων: Connection, Session, Transport, Require.

2.8 Το RTSP σε λειτουργία

Στην εικόνα 2.1 επιδεικνύεται η στοιχειώδης τυπική λειτουργία και τα διαδοχικά βήματα της επικοινωνίας ενός media server και ενός client/player, με το RTSP να δρα ως «δικτυακό τηλεχειριστήριο».



Εικόνα 2.1: Η διαδικασία ανάκτησης ενός μέσου από έναν media server, οι κατευθύνσεις των αιτήσεων και των ροών των μέσων (Μοβ: HTTP μηνύματα, Κόκκινο: RTSP μηνύματα, Πράσινο: RTP-RTCP μηνύματα)

Παρακάτω περιγράφεται αναλυτικά το σενάριο και τα βήματα της επικοινωνίας:

1. Ο client ζητά από έναν server (εδώ υποθέτουμε ότι υπάρχει ένα link σε μια web page) την περιγραφή του μέσου ή της παρουσίασης και ο web server του απαντά παρέχοντάς του την κατάλληλη περιγραφή (session description).
2. Έχοντας τα απαραίτητα δεδομένα (π.χ. τη διεύθυνση του media server και το όνομα αρχείου της παρουσίασης), ο client απευθύνει μια αίτηση SETUP για τη συγκεκριμένη παρουσίαση (πιθανώς ορίζοντας και κάποιες παραμέτρους μεταφοράς) στον media server, ο οποίος εγκαθιδρύει τη σύνοδο και εξασφαλίζει τους απαραίτητους πόρους.
3. Ο client ζητάει με τη μέθοδο PLAY από τον media server να ξεκινήσει τη μετάδοση του μέσου.
4. Στη συνέχεια, ο server παρέχει στον client το βίντεο και/ή τον ήχο μέσω RTP ροών, στα προσημωμηθέντα ports.
5. Ο client πλέον μπορεί να επιδράσει πάνω στη ροή με κάποια RTSP μέθοδο (π.χ. PAUSE).
6. Τέλος, ο client είναι υπεύθυνος να τερματίσει τη ροή και την επικοινωνία, εκδίδοντας μια αίτηση TEARDOWN.

2.9 Παραδείγματα

Παρακάτω παρουσιάζονται δύο σενάρια επικοινωνίας μεταξύ ενός client και ενός media server που υλοποιούν το RTSP. Παρατίθενται τα μηνύματα που ανταλλάσσονται μέσω του πρωτοκόλλου, από την έναρξη της επικοινωνίας (αίτηση του client) έως τον τερματισμό της συνόδου.

2.9.1 Video on Demand (Unicast)

Ο client C ζητάει μια ταινία από τους media servers A (audio.example.com) και V (video.example.com). Η περιγραφή των μέσων είναι αποθηκευμένη σε έναν web server W. Η περιγραφή των μέσων περιέχει περιγραφές της παρουσίασης και όλων των ροών της, συμπεριλαμβανόμενων των κωδικοποιητών/αποκωδικοποιητών που είναι διαθέσιμοι, δυναμικών RTP payloads, της στοίβας πρωτοκόλλων, και πληροφορίες περιεχομένου, όπως γλώσσας και περιορισμών πνευματικών δικαιωμάτων. Μπορεί επίσης να δίνει μια ένδειξη του χρονισμού της ταινίας.

```
C->W: GET /twister.sdp HTTP/1.1
```

```
Host: www.example.com
```

```
Accept: application/sdp
```

```
W->C: HTTP/1.0 200 OK
```

```
Content-Type: application/sdp
```

```
v=0
```

```
o=- 2890844526 2890842807 IN IP4 192.16.24.202
```

```
s=RTSP Session
```

```
m=audio 0 RTP/AVP 0
```

```
a=control:rtsp://audio.example.com/twister/audio.en
```

```
m=video 0 RTP/AVP 31
```

```
a=control:rtsp://video.example.com/twister/video
```

C->A: SETUP rtsp://audio.example.com/twister/audio.en RTSP/1.0

CSeq: 1

Transport: RTP/AVP/UDP;unicast;client_port=3056-3057

A->C: RTSP/1.0 200 OK

CSeq: 1

Session: 12345678

Transport: RTP/AVP/UDP;unicast;client_port=3056-3057;
server_port=5000-5001

C->V: SETUP rtsp://video.example.com/twister/video RTSP/1.0

CSeq: 1

Transport: RTP/AVP/UDP;unicast;client_port=3058-3059

V->C: RTSP/1.0 200 OK

CSeq: 1

Session: 23456789

Transport: RTP/AVP/UDP;unicast;client_port=3058-3059;
server_port=5002-5003

C->V: PLAY rtsp://video.example.com/twister/video RTSP/1.0

CSeq: 2

Session: 23456789

Range: smpte=0:10:00-

V->C: RTSP/1.0 200 OK

CSeq: 2

Session: 23456789

Range: smpte=0:10:00-0:20:00

RTP-Info: url=rtsp://video.example.com/twister/video;

seq=12312232;rtptime=78712811

C->A: PLAY rtsp://audio.example.com/twister/audio.en RTSP/1.0

CSeq: 2

Session: 12345678

Range: smpte=0:10:00-

A->C: RTSP/1.0 200 OK

CSeq: 2

Session: 12345678

Range: smpte=0:10:00-0:20:00

RTP-Info: url=rtsp://audio.example.com/twister/audio.en;

seq=876655;rtptime=1032181

C->A: TEARDOWN rtsp://audio.example.com/twister/audio.en RTSP/1.0

CSeq: 3

Session: 12345678

A->C: RTSP/1.0 200 OK

CSeq: 3

C->V: TEARDOWN rtsp://video.example.com/twister/video RTSP/1.0

CSeq: 3

Session: 23456789

V->C: RTSP/1.0 200 OK

CSeq: 3

Παρόλο που τα κομμάτια (tracks) του βίντεο και του ήχου είναι σε δύο διαφορετικούς servers, και ίσως ξεκινάνε σε ελαφρώς διαφορετικούς χρόνους, ο μπορεί να τα συγχρονίσει χρησιμοποιώντας μεθόδους του RTP, και ειδικότερα με τη βοήθεια συγκεκριμένων παραμέτρων των αναφορών ελέγχου του RTCP.

2.9.2 Ζωντανή παρουσίαση με χρήση Multicast

Ο media server M επιλέγει τη multicast διεύθυνση και port. Εδώ υποθέτουμε ότι ο web server W περιέχει απλά ένα δείκτη στην πλήρη περιγραφή, ενώ ο media server έχει αποθηκευμένη την περιγραφή.

C->W: GET /concert.sdp HTTP/1.1

Host: www.example.com

W->C: HTTP/1.1 200 OK

Content-Type: application/x-rtsl

<session>

<track src="rtsp://live.example.com/concert/audio">

</session>

C->M: DESCRIBE rtsp://live.example.com/concert/audio RTSP/1.0

CSeq: 1

M->C: RTSP/1.0 200 OK

CSeq: 1

Content-Type: application/sdp

Content-Length: 44

v=0

o=- 2890844526 2890842807 IN IP4 192.16.24.202

s=RTSP Session

m=audio 3456 RTP/AVP 0

a=control:rtsp://live.example.com/concert/audio

c=IN IP4 224.2.0.1/16

C->M: SETUP rtsp://live.example.com/concert/audio RTSP/1.0

CSeq: 2

Transport: RTP/AVP;multicast

M->C: RTSP/1.0 200 OK

CSeq: 2

Transport:RTP/AVP;multicast;destination=224.2.0.1;

port=3456-3457;ttl=16

Session: 0456804596

C->M: PLAY rtsp://live.example.com/concert/audio RTSP/1.0

CSeq: 3

Session: 0456804596

M->C: RTSP/1.0 200 OK

CSeq: 3

Session: 0456804596

Σημειώματα σχετικά με τα δικαιώματα Πνευματικής Ιδιοκτησίας

Σημείωμα Αναφοράς

Copyright Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών 2019. Παντελής Μπαλαούρας. «Εργαστήριο Δικτύων Επικοινωνίας II, Α. Δικτύωση Πολυμέσων, Α1. Αποθηκευμένο βίντεο συνεχούς ροής, Α1.1 VoD/RTSP/RTP/UDP» Έκδοση: 1.0. Αθήνα 2019.

Σημείωμα Αδειοδότησης

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά Παρόμοια Διανομή 4.0 [1] ή μεταγενέστερη, Διεθνής Έκδοση.



Η άδεια αυτή ανήκει στις άδειες που ακολουθούν τις προδιαγραφές του Ορισμού Ανοικτής Γνώσης [2], είναι ανοικτό πολιτιστικό έργο [3] και για το λόγο αυτό αποτελεί ανοικτό περιεχόμενο [4].

[1] <http://creativecommons.org/licenses/by-sa/4.0/deed.el>

[2] <http://opendefinition.org/okd/ellinika/>

[3] <http://freedomdefined.org/Definition/EI>

[4] <http://opendefinition.org/buttons/>

Διατήρηση Σημειωμάτων

Οποιαδήποτε αναπαραγωγή ή διασκευή του υλικού θα πρέπει να συμπεριλαμβάνει:

- Το Σημείωμα Αναφοράς
- Το Σημείωμα Αδειοδότησης
- Τη δήλωση διατήρησης Σημειωμάτων
- Το σημείωμα χρήσης έργων τρίτων (εφόσον υπάρχει)

μαζί με τους συνοδευόμενους υπερσυνδέσμους.