

這是 Google 對 http://dslab.ee.ncku.edu.tw/~lily/learning/learning_ch2-1.html 的快照。這是該網頁於 2012年7月19日 23:06:46 GMT 顯示時的快照。在此期間，[目前網頁](#) 可能已經變更。[瞭解更多資訊](#)

這些搜尋字詞已反白標明：**rtsp streaming**

[純文字版](#)

Chapter 2-1: 何謂 RTSP (Real Time Streaming Protocol)?

1. [簡介: 什麼是 RTSP?](#)
2. [RTSP 和 HTTP 有何不同?](#)
3. [RTSP 的運作方式](#)

簡介: 什麼是 RTSP

RTSP 可以用來控制一到數個 audio 或 video 的 media streams. 它負責的是 streams 的控制, 但傳輸時所用的 protocol 或 mechanism 卻不在它定義的範圍內. 也就是說, server 的 maintainer 可以選擇用 tcp 或 udp 來 implement 它的傳輸協定 RTP.

RTSP 有什麼特點呢?

1. 它的語法和運作跟 HTTP/1.1 類似.
2. 它並不是個很強調 time synchronized 的 protocol, 所以在現今的網路上, 它可以容忍偶而的網路延遲.
3. **RTSP** 還支援 Multicast, 所以, 除了降低網路的 load 外, 還能提供 video conference 的功能.
4. 尤其與 HTTP/1.1 運作方式相似, 所以 proxy 的 cache 功能也適用於 **RTSP**. 在 load sharing 上, 因 **RTSP** 具有 redirect 功能, 所以可視 load 轉換提供服務的 server.

RTSP 和 HTTP 有何不同?

1. **RTSP** 新增了一些 methods, 而且跟 HTTP 有不同的 protocol identifier.
2. **RTSP** server 需要隨時 maintain state, 簡單的說, 它不像 HTTP 一樣是 stateless.
3. 除了 client 有 Request 外, server 也有可能發出 Request.
4. Request-URI 要用 absolute URI.

RTSP 的運作方式

• RTSP 的表示語法:

1. URL

`rtsp_URL = ("rtsp:" | "rtspu:" | "rtsps:") "://" host [":"port] [abs_path]`

rtsp: 使用tcp之類的reliable protocol.

rtspu: 使用udp之類的unreliable protocol.

rtsp: 使用tcp connection secured by TLS.

2. Time

SMPTE Relative Timestamps

smpte-range = "smpte" "=" smpte-time "-" [smpte-time]

smpte-time = 2DIGIT ":" 2DIGIT ":" 2DIGIT [":" 2DIGIT] [":" 2DIGIT]

time code format: hours:minutes:seconds:frames.subframes

example:

smpte=10:12:33:20-

smpte=10:07:00-10:07:33:05.01

Normal Play Time(NPT)

npt-time = npt-sec | npt-hhmmss

npt-sec = 1*DIGIT ["." *DIGIT]

npt-hhmmss = npt-hh ":" npt-mm ":" npt-ss ["." *DIGIT]

npt-hh = 1*DIGIT ; 任何正數

npt-mm = 2DIGIT ; 00-59

npt-ss = 2DIGIT ; 00-59

example:

npt=123.45-125

npt=12:05:35.3

Absolute Time

utc-range = "clock" "=" utc-time "-" [utc-time]

utc-time = utc-date "T" utc-time "Z"

utc-date = 8DIGIT ; <YYYYMMDD>

utc-time = 6DIGIT ["." fraction] ; < HHMMSS.fraction >

example:

19961108T143720.25Z

• RTSP 的運作流程:

首先要定義兩個名詞: Presentation, Presentation description.

Presentation: 表示一到數個可被 Client 端同時開啟的 streams.

Presentation description: 是一個用來描述 Presentation 的檔案, 裡頭包含了 network address 之類的, 跟 Presentation 相關的 information. 這種檔案可以用多種 formats 表示, 比如 session description format [SDP](#).

一開始, 使用者會先在網路上找到他想要的 presentation description, 點選後, 懂得 RTSP 的 browser 就會照著 presentation description file 中指定的 address, port, directory, 到 提供服務的 Server 開啟 presentation. 要注意的是, 一個 presentation 中包含的數個 session, 不一定要在同一個 Server 上. 因此, 這可以是一種減輕 Server Load 的方法.

[Example](#)

- **RTSP States:**

RTSP 的 control message 跟 data 是可以分開的, 也就是說, 我們可以用 tcp 來傳送 control message, 而以 udp 傳送 data. 所以, Server 端必須對每一個 stream 都 maintain session state.

簡單的來說, 我們可以將 **RTSP Server** 的 State 分成以下數個:

- **SETUP**: Server 將會 allocate resources, 好開始一個被 request 的 stream.
 - **PLAY** and **RECORD**: 經由 **SETUP** allocate 的 stream, 開始傳送資料.
 - **PAUSE**: 暫時 halt 住一個 stream, 但不將 Server 端的 resource free 掉.
 - **TEARDOWN**: 將一個 stream 佔用的 resource free 掉, 並關閉這個 **RTSP session**.
-

- 提供的 **Methods** :

RTSP 為了要能達到控制 stream 的目的, 提供了許多的 methods, 列舉如下:

- [OPTIONS](#)
 - [DESCRIBE](#)
 - [ANNOUNCE](#)
 - [SETUP](#)
 - [PLAY](#)
 - [PAUSE](#)
 - [TEARDOWN](#)
 - [GET_PARAMETER](#)
 - [SET_PARAMETER](#)
 - [REDIRECT](#)
 - [RECORD](#)
-

[[上一頁](#)] [[回到首頁](#)] [[下一頁](#)]