

Протокол RTMP - Real Time Messaging Protocol

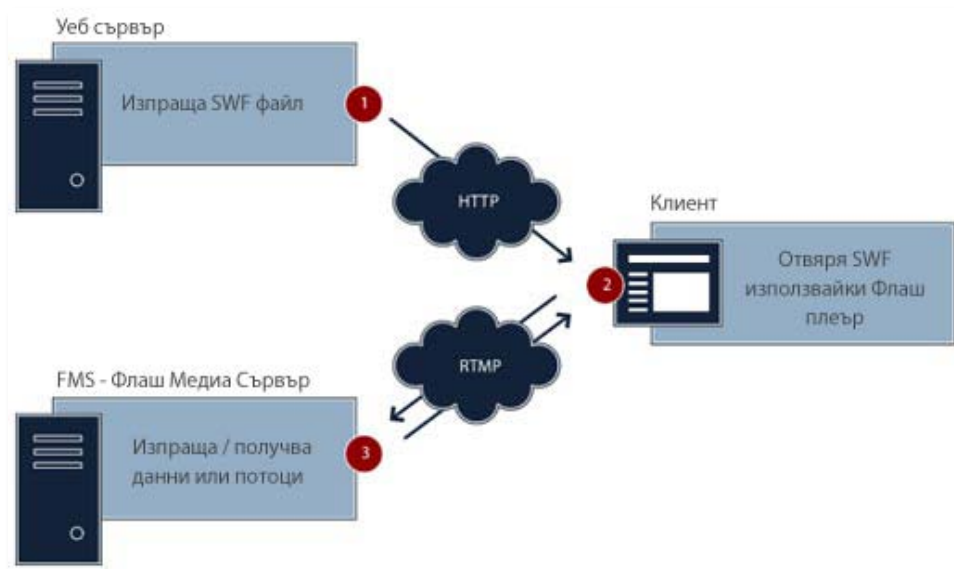
Протокола RTMP (Протокол за съобщения в реално време) е предназначен за предаване на голям обем от аудио, видео информация или данни между флаш технологии разработени от Адоб (Adobe). До 10 Януари 2009, протолът беше патентован и разработван от Адоб Системс (Adobe Systems). Вече RTMP е без патент и има отворена спецификация на следния адрес: <http://www.adobe.com/devnet/rtmp/>

Накратко RTMP е протокол създаден от Адоб за предаване на аудио и/или видео потоци или данни в реално време през Интернет. Като комуникацията се осъществява между флаш плъер-а (Flash Player), който се намира на клиентския слой и медия сървър (Media Server), който се намира на сървен слой.

Как се ползва

За да ползваме RTMP протокола, трябва да инсталираме на нашия уеб сървър софтуер, който да ни предоставя такава опция. Такива софтуери се наричат Media Server, такива са:

- Wowza Media Server
- Open Source Red5 server
- Adobe's Flash Media Server (FMS)



Как работи RTMP

Базовия RTMP (е TCP-базиран) протокол, който поддържа постоянна връзка между клиент и медия сървър (друг клиент), като по този начин дава възможност за връзка в реално време. За да се гарантира безпрепятственото предоставяне на видео и аудио потоци, докато все още се поддържа способността от предаване на по-големи парчета от информация, протоколът може да раздели потоците (видео, аудио, данни) във фрагменти. Размерът на фрагментите се определя динамично между клиента и сървъра (взема се предвид скоростта за обмен на данни между клиента и сървъра), макар и големината на фрагментите да имат стойности по подразбиране. Големината на аудио фрагмент е 64 байта, за видео и дата фрагмент е 128 байта. Фрагментите от един поток могат да бъдат **interleaved** и **multiplexed**. На практика обаче, отделните фрагменти не са interleaved. Вместо това обаче interleaved и multiplexed се правят на пакетно ниво, като от няколко активни канала едновременно се предават RTMP пакети. Всеки вид канал отговаря на определени свойства: скорост (bandwidth), латентност.

RTMP определя няколко канала, по които могат да бъдат изпратени / получени пакети. Каналите работят независимо една от друга. Например се определя канал за видео потока от данни, канал за аудио потока от данни, канал, който менажира RPC (Remote Procedure Call) заявки и отговори. По време на сесия на RTMP, няколко канала могат да бъдат едновременно активни. Когато се пакетират RTMP данните, към пакетите се добавят хедъри. Хедърите определят, идентификационен номер на канала на който пакета трябва да бъде изпратен, времето (timestamp), когато е било генериран пакета, големината на полезната информация, която пакета пренася. Една връзка до медия сървър, може да предава различни потоци от аудио/видео информация през различни канали.

На по-високо ниво, RTMP капсулира MP3 и Flash мултимедийни потоци, и може да прави RPC извиквания с помощта на AMF (Action Message Format) формат.

Разновидности на RTMP

Защо има нужда от разновидности на протокола? Идеята е firewall-ите на компютрите блокират портове които не разпознават, затова се създават различни видове на протокола. Те се използват в среди, където поради предпазни мерки RTMP протоколът е блокиран.

За пример ще вземем как флаш плеър създава връзка. Идеята е да видите приложение на разновидностите и нуждата от тях.

Както разбрахме по стандарт RTMP използва порт 1935 по подразбиране. Ако при създаване на връзка през порт 1935, се окаже че е неуспешна, флаш плеъра опитва отново да създаде връзка през 443 (RTMPS) или 80(RTMPT) в опит да преодолее, ограниченията наложени от firewall-а.

RTMPT (RTMP Tunneled)

Този протокол "обвива" RTMP данните във валидни HTTP, и по подразбиране комуникира през порт 80. Докато RTMPT изисква незначително по-висока скорост на обмен (сравнено с RTMP), дължаща се на добавените HTTP хедъри, протокола може да се използва успешно в среди, където мерките за сигурност ще блокират RTMP протокола.



RTMPS (RTMP Secure)

Използва се идеята на RTMPT. Този протокол пак "обвива" RTMP данните във валидни HTTPS, и по подразбиране комуникира през порт 443.

RTMPE (RTMP Encrypted)

В този протокол е усъвършенствана криптирана версия на RTMP. RTMPE е по-бърз, отколкото SSL и не изисква SSL сертификати за да се използва.

RTMPTE (RTMP Tunneled Encrypted)

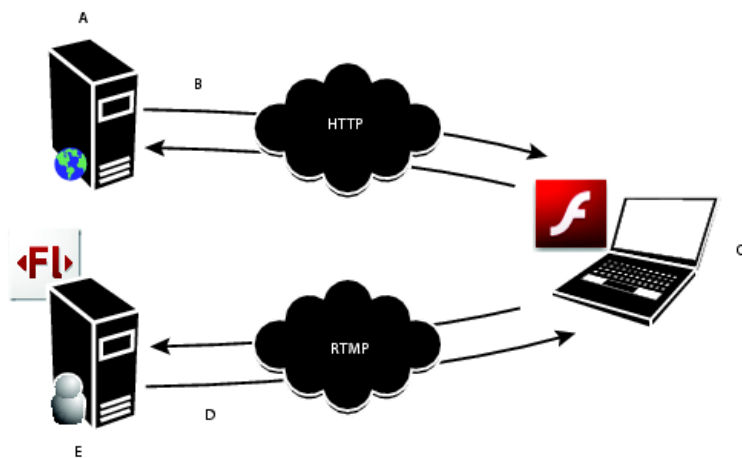
Усъвършенствана версия на RTMPT и RTMPS, като пак се използва криптирана връзка за комуникация, т.е. RTMP данните са криптирани и отново са "обвити" във HTTP. Порт подразбиране е 80. Може да използва и HTTPS, като е усъвършенствана бързината и лекотата на извършване на обвиването на пакетите - стресът върху сървъра е значително намален.

RTMFP (Real-Time Media Flow Protocol)

RTMFP е базиран на протокола UDP (User Datagram Protocol). това е значително нова версия на протокола. Идеята на този протокол е да се ползва Media Server, само, за да се създаде връзка, но предаването на данни от потребител до потребител.



В тази връзка нека видим как става самата комуникация:



- A. Уеб сървър
- B. Клиента се свързва към сървър, който му връща HTML страница със SWF в нея.
- C. Флаш плеър изобразява SWF файла.
- D. SWF файла се свързва към медия сървър, в нашия случай FMS. Сървъра връща поток от данни през установената връзка
- E. Flash Media Server

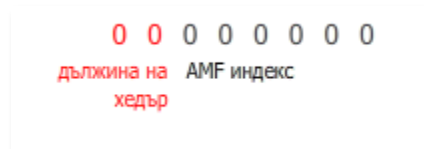
Обмен на информация

Протоколът е контейнер на пакети наречени AMF или аудио/видео потоци. Една връзка до медия сървъра, може да предава различни потоци от аудио/видео информация през различни канали.

Структура на пакетите



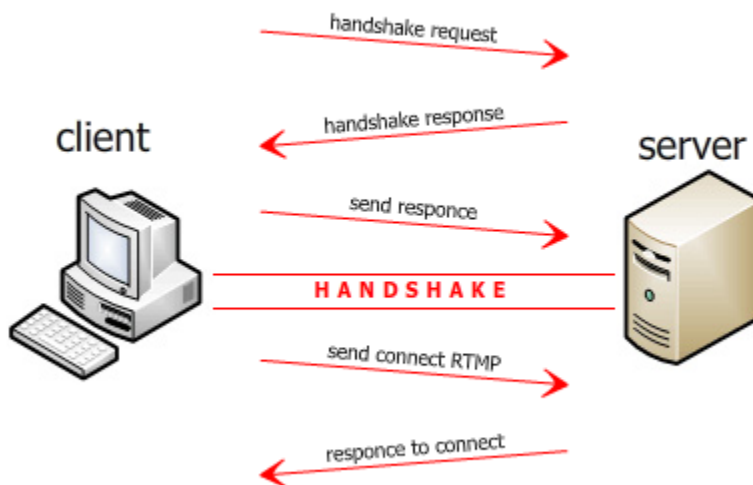
Пакетите могат да бъдат големи до 140 байта. Дължината на хедъра на пакета е с фиксиран размер и може да бъде: 1,4,8,12 байта, докато големината на полезната информация може да е до 128 байта. Дължината на хедъра се определя от първия байт на съобщението. При първа двойка:



- 00 - 12 байта
- 01 - 8 байта
- 10 - 4 байта
- 11 - 1 байт

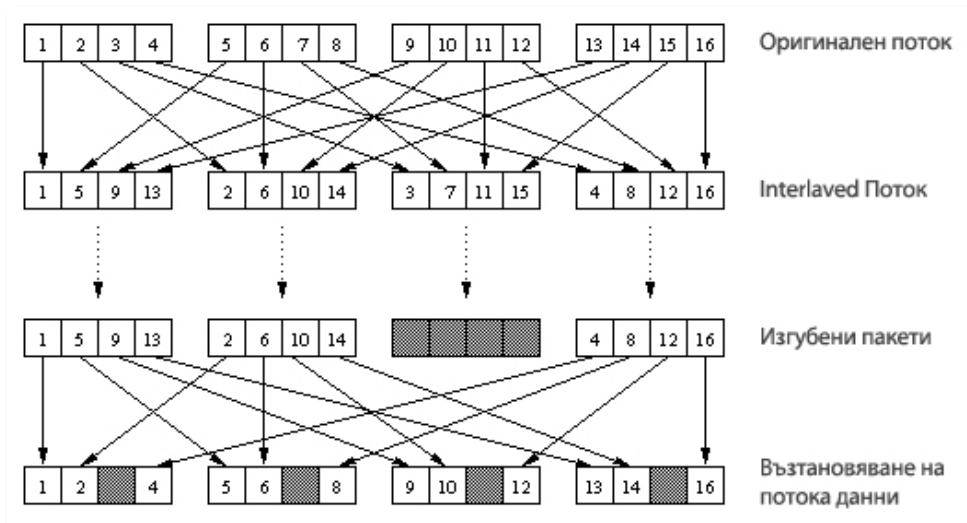
определят големината на хедъра, като останалите 6 цифри от първия байт е индекс на AMF пакета.

HandShake – или как става връзката със сървъра.



- C → S : Изпраща искане за Handshake. Този молба не е пакет от RTMP протокола, а единичен байт (0x03) следван от 1536 байта. Съдържанието на тези байтове не е важно за протокола, но не се генерират случайно.
- S → C : Изпраща отговор на Handshake искането. И този отговор не е RTMP пакет, а единичен байт (0x03) следван от две парчета по 1536 байта (или общо от 3072 байта). Първото парче са байтове генерирани от сървъра, а второто парче са байтовете, който клиента е изпратил първия път в искането за Handshake.
- C → S : Проверява дали второто парче си съответства със изпратеното. Ако да изпраща до сървъра генерираните от него байтове.
- Когато сървъра получи байтовете и са съответстващи на изпратените се получава HANDSHAKE. И всички пакети от тук нататък са RTMP.
- C → S : изпраща искане за връзка със RTMP (медия) сървъра.
- S → C : Сървъра отговаря на искането, като може да одобри или откаже искането

Метод interleaving



Ползата от Interleaving метода, за предаване на пакети се ползва, когато предаваната информация е видео или аудио поток.

Идеята е преди да се предат пакетите по мрежата, те да бъдат разместени в произволен характер или наредени така че, първоначално съседни пакети помежду си, след разместването да са отделени на равно разстояние по-между си. Когато пакетите се получат от приемника, той ги размества отново така че да се получат в оригиналната поредица. При този метод се получава разпределена загуба на пакети. Така получена единична загуба на пакет във потока дава минимална загуба на качество, докато не получаването на цели поредици от пакети би се усетило значително повече от клиента.

Въпреки че Interleaving метода не намалява размера на загубите, той значително подобрява качеството на аудио/видео потока. Очевидният недостатък на Interleaving е, че тя увеличава латентността. Това ограничава използването на тази техника за работа с високо интерактивни приложения, въпреки че той работи много добре за не-интерактивни положения. Главното предимство на метода е, че той не води до увеличаване на обема на данните за предаване.

Латентност (latency)

Времето, от изпращането на една заявка до получаване на отговора и.

