

ПЕТРОЗАВОДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНСТИТУТ МАТЕМАТИКИ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ
КАФЕДРА ИНФОРМАТИКИ И МАТЕМАТИЧЕСКОГО ОБЕСПЕЧЕНИЯ

01.04.02 - Прикладная математика и информатика

Отчет по выполнению проекта в рамках курса
«Оценивание производительности сетевых систем»

АНАЛИЗ СЕТЕВОГО ТРАФИКА ВЕБ-ПРИЛОЖЕНИЯ SPOTIFY

Выполнили:
студенты курса группы 22603

А. С. Гормакова
М.А. Мальцева
А.П. Воробьева

Руководитель:
к.т.н., доцент О. Ю. Богоявленская

подпись

Содержание

Постановка задачи	3
1 Описание приложения	3
2 Метрики анализа	4
2.1 План экспериментов	4
2.2 Сценарии экспериментов	5
2.3 Обработка файлов	7
2.4 Результаты экспериментов	9
3 Заключение	10

Постановка задачи

Spotify (рус. Спотифа́й) — шведский интернет-сервис потокового аудио, позволяющий легально и бесплатно прослушивать музыкальные композиции, аудиокниги и подкасты, не скачивая их на устройство.

Целью работы является проведение анализа сетевого трафика веб-приложения Spotify.

1 Описание приложения

Spotify доступен в виде веб-сайта, приложений для смартфонов, смарт-устройств и медиа-систем автомобилей. На сервис приходится более 36 % мирового аудиостриминга. Приложение доступно почти в 120 странах, включая Россию, Украину, Белоруссию. В его каталоге больше 50 миллионов песен и 4 миллиарда плейлистов с треками русских и зарубежных исполнителей.

В Spotify легко найти музыку или подкаст для любой ситуации, так как сервис доступен на телефоне, компьютере, планшете и других устройствах.

Spotify присутствует на множестве современных устройств, на операционных системах «Windows», «macOS», «Linux», «Android» и «iOS», а функция «Spotify Connect» (рус. Соединение по Спотифай) позволяет пользователям задействовать ряд устройств одновременно.

Более 70 % прослушиваний музыки происходит через плей-листы, а не по поисковому запросу или со страницы артиста. Также пользователи могут поискать что-нибудь интересное в плейлистах друзей.

Более того, для пользователя на основе его предпочтений формируется микс дня, который обновляется ежедневно. Такие плейлисты оставляют его в «зоне комфорта», так как не ищут для слушателя что-то новое, а группируют то, что ему точно нравится, иногда разбавляя чем-то похожим.

Spotify имеет ряд преимуществ над другими аналогичными сервисами:

- Алгоритмы подбора музыки практически идеально подстраиваются под предпочтения слушателей.
- Функция “Музыка нон-стоп”, которая анализирует песни в определённом плейлисте и пытается предсказать похожие треки по окончании плейлиста, которые могли бы быть в нём — как если бы человек, который создал плейлист, просто продолжал бы добавлять в него музыку.

- Есть полезная функция «радио»: его можно создать на основе любой песни, альбома или плейлиста.

2 Метрики анализа

- Средняя длина UDP/TCP пакета;
- Процент потери TCP пакета;
- среднее время подключения пользователя к серверу;
- Среднее время ответа сервера на запрос клиента;

CT (Connection Time) – время установления соединения с сервером. Этот параметр наглядно отражает задержки в сети, так как пакеты в этот момент передаются малой длины и с приоритетом, настроенным для данного приложения. Пакеты при установлении соединения обрабатываются сервером/клиентом также с высоким приоритетом. Некоторые системы мониторинга на этой стадии дают развернутую оценку, и делят время CT на составляющие: RTT и SRT и CD (Client Delay). Это позволяет оценить по времени SRT, занят ли сервер и на сколько в момент установления соединения.

RR (Retransmission rates) – наиболее наглядный индикатор, который показывает потери пакетов в каналах связи или очередях активного оборудования. Многие специалисты при анализе потери пакетов исключают факт проверки настроек активного оборудования, а на нем может быть отключена фрагментация пакетов, некорректно настроен MTU, время жизни и т.д.

Время ответа сервера на запрос клиента SRT (Server Response Time) – время, необходимое серверу на подготовку первого пакета с данными в ответ на запрос клиента. Основной параметр, который показывает задержку сервера. Далее необходимо анализировать, чем занят сервер, используя и другие решения, например, на основе SNMP или SCOM.

2.1 План экспериментов

Для проведения экспериментов использовалась операционная система Windows 10 и программная система Wireshark. В проведении экспериментов принимало участие три человека. Все эксперименты были проведены в вечернее время (после 18.00), их исполнение соответствовало нижеописанному алгоритму:

План экспериментов:

1. Запустить Spotify веб-сервер;
2. Запустить Wireshark;
3. Включить захват трафика;
4. Включить фильтрацию;
5. Выполнять различные действия:
 - поиск музыки в строке поиска;
 - поиск конкретного исполнителя;
 - прослушивание музыки;
 - поиск друзей;
 - получение радио по треку;
 - получение радио по исполнителю.

2.2 Сценарии экспериментов

- Средняя длина UDP/TCP пакета:
 1. Запустить клиент.
 2. Соблюдать алгоритм действий.
 3. Повторить вышеописанные шаги 50 раз.
 4. Экспортировать данные в CSV формат.
 5. Обработать данные.
 6. Вычислить среднюю длину TCP/UDP пакета.
- Средний процент потери TCP пакета:

Необходимо предварительно добавить в таблицу в качестве столбца сообщение `tcp.analysis.lost_segment`, указывающее на потерю сегмента в tcp пакете.

 1. Запустить клиент.
 2. Соблюдать алгоритм действий.
 3. Повторить вышеописанные шаги 50 раз.

4. Экспортировать данные в CSV формат.
 5. Обработать данные.
 6. Вычислить средний процент потери TCP пакета.
- Время подключения пользователя к серверу:
 1. Запустить клиент.
 2. Соблюдать алгоритм действий.
 3. Повторить вышеописанные шаги 50 раз.
 4. В Wireshark настроить фильтр TCP порта для выборки пакетов, которые относятся к данной TCP сессии.
 5. Выбрать сегмент с установленным контрольным флагом SYN, начальный номер последовательности, который относится к началу отслеживания потока данных от клиента на сервер для этой сессии.
 6. Отследить от сервера подтверждение ACK получения сегмента с флагом SYN от клиента.
 7. Оценить время ответа сервера сегментом, содержащим ACK.
 8. Экспортировать данные в CSV формат.
 9. Обработать данные.
 10. Вычислить среднее время подключения пользователя к серверу.

Для оценки времени ответа сервера необходимо расшифровать tls-трафик и добавить в качестве столбца таблицы время с момента запроса `http_time`.

- Время ответа сервера на запрос клиента:
 1. Запустить клиент.
 2. Соблюдать алгоритм действий.
 3. Повторить вышеописанные шаги 50 раз.
 4. Сохранить данные в CSV формат.
 5. Обработать данные.
 6. Вычислить среднее время ответа сервера на запрос клиента.

2.3 Обработка файлов

В данном разделе продемонстрированы примеры CSV файлов, которые были экспортированы из Wireshark, для дальнейшей обработки данных в MS excel.

Topic / Item	Count	Average	Min Val	Max Val	Rate (ms)	Percent	Burst Rate	Burst Start
Packet Lengths	34471	892,59	54	38014	0,0313	100%	1,8100	161,925
0-19	0	-	-	-	0,0000	0,00%	-	-
20-39	0	-	-	-	0,0000	0,00%	-	-
40-79	12944	56,22	54	79	0,0118	37,55%	0,6800	622,916
80-159	2431	109,84	80	159	0,0022	7,05%	0,5900	626,620
160-319	925	218,07	160	319	0,0008	2,68%	0,0500	588,160
320-639	1611	471,88	320	639	0,0015	4,67%	0,1500	641,046
640-1279	968	925,73	640	1279	0,0009	2,81%	0,0800	161,940
1280-2559	13627	1508,32	1281	1514	0,0124	39,53%	1,0400	66,358
2560-5119	1744	3204,15	2874	4434	0,0016	5,06%	0,4300	417,620
5120 and greater	221	8025,90	5774	38014	0,0002	0,64%	0,0700	308,258

Рис. 1: Пример получения длины TCP пакетов

Topic / Item	Count	Average	Min Val	Max Val	Rate (ms)	Percent	Burst Rate	Burst Start
Packet Lengths	2072	816,13	43	1466	0,0481	100%	0,4500	10,461
0-19	0	-	-	-	0,0000	0,00%	-	-
20-39	0	-	-	-	0,0000	0,00%	-	-
40-79	854	64,18	43	76	0,0198	41,22%	0,2100	10,462
80-159	50	141,80	92	157	0,0012	2,41%	0,0200	15,412
160-319	56	204,25	174	310	0,0013	2,70%	0,0400	24,364
320-639	11	380,00	331	527	0,0003	0,53%	0,0100	4,418
640-1279	1	896,00	896	896	0,0000	0,05%	0,0100	25,789
1280-2559	1100	1466,00	1466	1466	0,0256	53,09%	0,2600	10,384
2560-5119	0	-	-	-	0,0000	0,00%	-	-
5120 and greater	0	-	-	-	0,0000	0,00%	-	-

Рис. 2: Пример получения длины UDP пакетов

No.	Time	Source	Destination	Protocol	Length	Info	TCP Delta Time	Stream Index	Lost package
110	0.000	192.168.1.6	35.186.224.25	TCP	74	3246 > 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1 TSval=40424431 TSecr=0	10.000000000	1	
111	0.032	35.186.224.25	192.168.1.6	TCP	74	443 > 3246 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1430 SACK_PERM=1 TSval=348033 TSecr=40424431	0.032554000	1	
112	0.000	192.168.1.6	35.186.224.25	TCP	66	3246 > 443 [ACK] Seq=1 Ack=1 Win=66560 Len=0 TSval=40424464 TSecr=3480337148	0.000066000	1	
115	0.008	192.168.1.6	35.186.224.25	TLSv1.3	648	Client Hello	0.008576000	1	
116	0.039	35.186.224.25	192.168.1.6	TCP	66	443 > 3246 [ACK] Seq=1 Ack=583 Win=66816 Len=0 TSval=3480337197 TSecr=40424472	0.039598000	1	
117	0.000	35.186.224.25	192.168.1.6	TLSv1.3	278	Server Hello, Change Cipher Spec, Application Data	0.000914000	1	
118	0.001	192.168.1.6	35.186.224.25	TLSv1.3	130	Change Cipher Spec, Application Data	0.001730000	1	
119	0.000	192.168.1.6	35.186.224.25	TLSv1.3	1484	Application Data	0.000577000	1	
120	0.000	192.168.1.6	35.186.224.25	TLSv1.3	332	Application Data	0.000009000	1	
121	0.030	35.186.224.25	192.168.1.6	TLSv1.3	646	Application Data, Application Data	0.030602000	1	
122	0.014	35.186.224.25	192.168.1.6	TLSv1.3	97	Application Data	0.014062000	1	
123	0.000	192.168.1.6	35.186.224.25	TLSv1.3	97	Application Data	0.000039000	1	
124	0.011	35.186.224.25	192.168.1.6	TCP	66	443 > 3246 [ACK] Seq=824 Ack=2331 Win=72448 Len=0 TSval=3480337252 TSecr=40424515	0.011162000	1	
125	0.018	35.186.224.25	192.168.1.6	TCP	66	443 > 3246 [ACK] Seq=824 Ack=2362 Win=72448 Len=0 TSval=3480337275 TSecr=40424560	0.018768000	1	
126	0.250	35.186.224.25	192.168.1.6	TLSv1.3	834	Application Data	0.250043000	1	
127	0.001	192.168.1.6	35.186.224.25	TLSv1.3	1484	Application Data	0.001705000	1	
128	0.000	192.168.1.6	35.186.224.25	TCP	66	3246 > 443 [ACK] Seq=2362 Ack=3010 Win=66560 Len=0 TSval=40424841 TSecr=3480337521	0.000055000	1	
129	0.000	35.186.224.25	192.168.1.6	TLSv1.3	1484	Application Data	0.000105000	1	
130	0.003	35.186.224.25	192.168.1.6	TLSv1.3	1401	Application Data	0.003688000	1	
131	0.000	35.186.224.25	192.168.1.6	TLSv1.3	1484	Application Data	0.000001000	1	
132	0.000	192.168.1.6	35.186.224.25	TCP	66	3246 > 443 [ACK] Seq=2362 Ack=7181 Win=66560 Len=0 TSval=40424845 TSecr=3480337522	0.000076000	1	
133	0.000	35.186.224.25	192.168.1.6	TLSv1.3	1484	Application Data	0.000303000	1	
134	0.000	192.168.1.6	35.186.224.25	TCP	66	3246 > 443 [ACK] Seq=2362 Ack=8599 Win=66560 Len=0 TSval=40424846 TSecr=3480337523	0.000021000	1	
135	0.001	35.186.224.25	192.168.1.6	TLSv1.3	1401	Application Data	0.001105000	1	

Рис. 3: Пример получения времени подключения пользователя к серверу

Protocol	Length	Info	Delta time	Stream index	Lost package	Time since first frame in this TCP stream
TCP	66	62134 > 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1	0	0		0
TCP	66	62137 > 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1	0	1		0
TCP	66	443 > 62134 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1430 SACK_PERM=1 WS=256	0,017212	0		0,017212
TCP	54	62134 > 443 [ACK] Seq=1 Ack=1 Win=131328 Len=0	0,00007	0		0,017282
TLSv1.3	571	Client Hello	0,001174	0		0,018456
TCP	66	443 > 62137 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1430 SACK_PERM=1 WS=256	0,015191	1		0,015191
TCP	54	62137 > 443 [ACK] Seq=1 Ack=1 Win=131328 Len=0	0,000051	1		0,015242
TCP	60	443 > 62134 [ACK] Seq=1 Ack=518 Win=66816 Len=0	0,022507	0		0,040963
TLSv1.3	1484	Server Hello, Change Cipher Spec	0,009083	0		0,050046
TLSv1.3	1288	Continuation Data	0,007643	0		0,057689
TCP	54	62134 > 443 [ACK] Seq=518 Ack=2665 Win=131328 Len=0	0,000058	0		0,057747
TLSv1.3	571	Client Hello	0,059459	1		0,074701
TLSv1.3	1288	[TCP Spurious Retransmission], Continuation Data	0,054634	0		0,112381
TCP	66	[TCP Dup ACK 16#1] 62134 > 443 [ACK] Seq=518 Ack=2665 Win=131328 Len=0 SLE=1431 SRE=	0,000043	0		0,112424
TLSv1.3	118	Change Cipher Spec, Finished	0,007612	0		0,120036
TCP	60	443 > 62137 [ACK] Seq=1 Ack=518 Win=66816 Len=0	0,042964	1		0,117665
HTTP2	1718	HEADERS[1]: POST /ListAccounts?gpsia=1&source=ChromiumBrowser&json=standard	0,002032	0		0,122068
TLSv1.3	1484	Server Hello, Change Cipher Spec	0,013061	1		0,130726
TLSv1.3	1484	Continuation Data	0,013203	1		0,143929

Рис. 4: Пример получения процента потери TCP пакетов

2.4 Результаты экспериментов

В данном разделе представлены полученные в ходе анализа результаты.

Средняя длина TCP пакетов	1059.368
Средняя длина UDP пакетов	274.375
Средний процент потери TCP пакетов	0.046%
Среднее время подключения пользователя к серверу	48.4 мс
Среднее время ответа сервера на запрос клиента	57.6 мс

Ниже на рисунках 5 и 6 представлены графики средней длина для TCP и UDP пакетов.



Рис. 5: Гистограмма средней длины TCP пакетов



Рис. 6: Гистограмма средней длины UDP пакетов

3 Заключение

По результатам проведенных экспериментов можно сделать вывод, что сервер веб-приложения Spotify справляется с поступающей нагрузкой. При проведении анализа трафика не было обнаружено сбоев. Также было вычислено, что средний процент потери TCP пакетов близок к 0. Кроме того, время ответа сервера на запрос клиента незначительно больше времени подключения пользователя к серверу и является приемлемым, так как составляет 57.6мс.

Таким образом, данный сервис можно порекомендовать в качестве сервиса прослушивания музыкальных композиций.