

## МОДЕЛИ ВЗАИМОДЕЙСТВИЯ WEB-ПРИЛОЖЕНИЙ МЕЖДУ КЛИЕНТСКОЙ И СЕРВЕРНОЙ ЧАСТЯМИ

*Б.В. Сорокин, бакалавр*

*Харьковский национальный университет радиоэлектроники  
sorokin.bohdan@gmail.com*

*В статье рассматриваются существующие протоколы обеспечивающие взаимодействие web-приложений, их преимущества, недостатки и способы оптимизации.*

*Sorokin V. B. Models of interaction of WEB-applications between client and server. This article about existing protocols to ensure the interaction of web-applications, their advantages, disadvantages and methods of optimization.*

*Ключевые слова:* HTTP, WEBSOCKET, ПРОТОКОЛ.

*Keywords:* HTTP, WEBSOCKET, PROTOCOL.

HTTP - протокол прикладного уровня передачи данных. Основой HTTP является технология «клиент-сервер», то есть предполагается существование потребителей (клиентов), которые инициируют соединение и посылают запрос, и поставщиков (серверов), которые ожидают соединения для получения запроса, производят необходимые действия и возвращают обратно сообщение с результатом.

Когда HTTP был создан и выпущен в мир, он воспринимался как простой и достаточно прямолинейный протокол и перед ним стояли конкретны задачи – передача данных в виде гипертекстовых документов в формате HTML. Но время показало что это не так и в настоящее время протокол используется для передачи самых различных данных.

В настоящее время большинство сайтов работают с протоколом HTTP 1.1, который оказался немного более гибким и оптимизированным по сравнению с HTTP 1.0, но по прежнему не смог избавиться от многих недостатков, таких как:

- чувствительность к задержкам, частично из-за того, что в конвейерной передаче HTTP по-прежнему хватает проблем и она отключена у подавляющего числа пользователей;
- блокировка начала очереди;
- к каждому запросу обязательно добавляется заголовок, что приводит к увеличению трафика и затратам ресурсов сервера на чтение заголовка (может оказаться критичным при использовании Polling или Long Polling подходов).

Способы оптимизации:

- использование спрайтов (объединение мелких изображений для уменьшения количества запросов на сервер);
- использование data-url встроенных в CSS файл вместо картинок;
- объединение js и css файлов.
- шардинг – размещение ресурсов на различных серверах, горизонтальное масштабирование.

На базе протокола SPDY (разработан и инициирован в Google) была разработана и представлена в спецификации RFC 7540 новая версия HTTP протокола – HTTP 2.

Основные задачи, которые были достигнуты в новом протоколе:

- сжатия данных в заголовках HTTP;
- использования push-технологий на серверной стороне;
- конвейеризация запросов;
- устранения проблемы блокировки "head-of-line" протоколов HTTP 1/1.1;
- мультиплексирование множества запросов в одном соединении TCP.

Данные изменения позволили ускорить работы протокола, оптимизировать трафик и уменьшить нагрузку на сервер и клиент.

HTTP 2 хорошо подходит для статических приложений, но если появляется необходимость сделать динамическое приложение с возможностью

инициирования событий на клиенте со стороны сервера (Real-time Web Application) данный протокол не справляется с такой задачей. Для реализации подобных задач на HTTP протоколе можно использовать Polling либо Long Polling подходы, но они очень негативно скажутся на производительности приложения.

Для реализации Real-time Web Application рекомендуется использовать WebSocket. WebSocket – протокол полнодуплексной связи поверх TCP-соединения, предназначенный для обмена сообщениями между браузером и веб-сервером в режиме реального времени. Данный протокол позволяет получать информацию о каких-либо событиях на клиенте или на сервере практически мгновенно, и уменьшает расходы трафика, так как не добавляет к своим запросам заголовки. Стоит учитывать что WebSocket на время сессии держит соединение открытым.

### *Литература*

1. В.Г. Олифер, Н.А. Олифер. Компьютерные сети. Принципы, технологии, протоколы - СПб.: Питер, 2001 - 672 с.
2. С. Золотов. Протоколы Internet - СПб.: BHV - Санкт-Петербург, 1998 - 786 с.
3. Документы RFC-1945, RFC-2068, RFC-2069., RFC 2616.