1. **Поясните понятие «TCP-порт».**

TCP-порт — для обмена пакетами между приложениями

1. **Поясните понятие «сетевой сокет».**

Совокупность IP-адреса и номера порта называется сокетом. Сокет однозначно идентифицирует прикладной процесс в сети TCP/IP.

1. **Поясните понятие «WebSocket».**

WebSocket представляет собой альтернативу HTTP, его можно применять для организации обмена данными в веб-приложениях. Этот протокол позволяет создавать долгоживущие двунаправленные каналы связи между клиентом и сервером. После установления соединения канал связи остаётся открытым, что даёт в распоряжение приложения очень быстрое соединение, характеризующееся низкими задержками и небольшой дополнительной нагрузкой на систему.

Протокол WebSocket поддерживают все современные браузеры.

HTTP и WebSocket — это очень разные протоколы, в которых используются различные подходы к обмену данными. HTTP основан на модели «запрос — ответ»: сервер отправляет клиенту некие данные после того, как они будут запрошены. В случае с WebSocket всё устроено иначе. А именно:

* Сервер может отправлять сообщения клиенту по своей инициативе, не дожидаясь поступления запроса от клиента.
* Клиент и сервер могут обмениваться данными одновременно.
* При передаче сообщения используется крайне малый объём служебных данных. Это, в частности, ведёт к низким задержкам при передаче данных.

Протокол WebSocket очень хорошо подходит для организации связи в режиме реального времени по каналам, которые долго остаются открытыми. HTTP, в свою очередь, отлично подходит для организации эпизодических сеансов связи, инициируемых клиентом. В то же время надо отметить, что, с точки зрения программирования, реализовать обмен данными по протоколу HTTP гораздо проще, чем по протоколу WebSocket.

1. **Поясните процедуру установки соединения между WS-сервером и WS-клиентом.**

**Server:** new WebSocket.Server({port: 5000, host: HOST, path: '/download'});

**Client:** new WebSocket('ws:/localhost:5000/download');

1. **Поясните понятие «широковещательное сообщение».**

Широковещательные сообщения - тип сообщений, не имеющих адресата. Рассылаются на все устройства, находящиеся в сети.

1. **Поясните принцип организации потокового ввода/вывода через WS-соединение.**

**Server:** duplex = WebSocket.createWebSocketStream(ws, {encoding: 'utf8'});

uf = fs.createWriteStream(\_\_dirname +`/files/${k++}.txt`);

duplex.pipe(uf);

**Client:** duplex = WebSocket.createWebSocketStream(ws, {encoding: 'utf8'});

uf = fs.createReadStream(\_\_dirname + `/files/file.txt`);

uf.pipe(duplex);

1. **Поясните принцип действия и назначение механизма «ping/pong».**

В протокол встроена проверка связи при помощи управляющих фреймов типа PING и PONG.

Тот, кто хочет проверить соединение, отправляет фрейм PING с произвольным телом. Его получатель должен в разумное время ответить фреймом PONG с тем же телом.

Эта функциональность встроена в браузерную реализацию, так что браузер ответит на PING сервера, но управлять ей из JavaScript нельзя.

**Иначе говоря, сервер всегда знает, жив ли посетитель или у него проблема с сетью.**

1. **Поясните аббревиатуру «RPC».**

**Remote Procedure Call** — вызов удалённых процедур.

Приложение предоставляет доступ к части своей функциональности посредством удаленного вызова процедуры. Взаимодействие между приложениями осуществляется синхронно в режиме реального времени.

1. **Поясните принцип работы RPC-механизма, предоставляемого пакетом rpc-websockets.**

Для работы RPC-протокола необходимо выполнение следующих условий:

1. Уникальная идентификации всех удаленно вызываемых процедур на данном хосте. RPC-запросы содержат три поля идентификаторов - номер удаленной программы (сервиса), номер версии удаленной программы и номер удаленной процедуры указанной программы. Номер программы назначается производителем сервиса, номер процедуры указывает на конкретную функцию данного сервиса

2. Идентификация версии RPC-протокола. RPC-сообщения содержат поле версии RPC-протокола. Она используется для согласования форматов передаваемых параметров при работе клиента с различными версиями RPC.

3. Предоставление механизмов аутентификации клиента на сервере. RPC-протокол обеспечивает процедуру аутентификации клиента в сервисе, и, в случае необходимости, при каждом запросе или отправке ответа клиенту. Кроме того, RPC позволяет использовать различные дополнительные механизмы безопасности.

RPC может использовать четыре типа механизмов аутентификации:

- AUTH\_NULL - без использования аутентификации

- AUTH\_UNIX - аутентификация по стандарту UNIX

- AUTH\_SHORT - аутентификация по стандарту UNIX с собственной структурой кодирования

- AUTH\_DES - аутентификация по стандарту DES

4. Идентификация сообщений ответа на соответствующие запросы. Ответные сообщения RPC содержат идентификатор запроса, на основании которого они были построены. Этот идентификатор можно назвать идентификатором транзакции вызова RPC. Данный механизм особенно необходим при работе в асинхронном режиме и при выполнении последовательности из нескольких RPC-вызовов.

5. Идентификация ошибок работы протокола. Все сетевые или серверные ошибки имеют уникальные идентификаторы, по которым каждый из участников соединения может определить причину сбоя в работе.

1. **Поясните принцип действия и назначение механизма «subscriber/publisher».**

Издатель отправляет сообщение в топик (тема) откуда оно рассылается всем подписчикам

1. **Поясните принцип действия и назначение механизма уведомлений.**

Создание-подписка-отправка-уведомление