**TCP-порт:** TCP-порт — это числовой идентификатор, который используется для однозначного определения конечной точки в сетевой коммуникации. В рамках протокола TCP (Transmission Control Protocol), порты позволяют множеству приложений на одном устройстве обмениваться данными через сеть. Каждый TCP-порт привязан к определенному приложению или службе на устройстве. Например, веб-сервер может прослушивать запросы на порту 80, а почтовый сервер — на порту 25.

**Сетевой сокет:** Сетевой сокет — это программный интерфейс для сетевого взаимодействия между процессами на разных устройствах. Он представляет из себя комбинацию IP-адреса и порта, обеспечивая точку соединения для обмена данными между приложениями через сеть.

**WebSocket:** WebSocket — это протокол передачи данных поверх TCP, предназначенный для обеспечения полнодуплексного взаимодействия между веб-браузерами и веб-серверами через одно установленное соединение. WebSocket обеспечивает более эффективную и мгновенную двустороннюю связь по сравнению с традиционными методами, такими как HTTP.

**Процедура установки соединения между WS-сервером и WS-клиентом:**

Клиент отправляет запрос на установление соединения.

Сервер принимает запрос и, если он готов установить соединение, отправляет ответное подтверждение.

Соединение устанавливается, и дальнейшие данные могут передаваться в обоих направлениях.

**Широковещательное сообщение:** Широковещательное сообщение — это сообщение, которое отправляется одним участником сети и предназначено для всех устройств в данной сети. Это позволяет эффективно распространять информацию по всему сегменту сети без необходимости точечного адресации каждого устройства.

**Принцип организации потокового ввода/вывода через WS-соединение:** В WebSocket соединении данные передаются в виде потока, позволяя браузерам и серверам обмениваться информацией в реальном времени. Это достигается через отправку данных в виде фреймов, которые могут содержать текст, бинарные данные и другую информацию.

**Принцип действия и назначение механизма "ping/pong":** Механизм "ping/pong" используется для поддержания активности соединения. Одна сторона посылает пинг (ping), а другая отвечает понгом (pong). Если сторона не получает ответа на пинг в определенное время, она может считать соединение разорванным и принять соответствующие меры.

**RPC (Remote Procedure Call):** RPC — это механизм взаимодействия между удаленными компьютерами, который позволяет вызывать процедуры (функции) на удаленной машине, как если бы они вызывались локально.

**Принцип работы RPC-механизма, предоставляемого пакетом rpc-websockets:** RPC-механизм, реализованный в пакете rpc-websockets, позволяет вызывать удаленные процедуры через WebSocket. Клиент и сервер могут обмениваться сообщениями, содержащими запросы и ответы на эти запросы, а библиотека обеспечивает прозрачное выполнение удаленных процедур.

**Принцип действия и назначение механизма "subscriber/publisher":** Механизм "subscriber/publisher" используется для реализации паттерна "издатель-подписчик". Один компонент (издатель) отправляет сообщения, а другие компоненты (подписчики) получают уведомления об этих сообщениях. Это позволяет эффективно реализовывать асинхронное взаимодействие между компонентами системы.

**Принцип действия и назначение механизма уведомлений:** Механизм уведомлений позволяет устройствам или приложениям получать автоматические оповещения о событиях или изменениях, происходящих в системе. Это обеспечивает эффективный способ обновления информации и реагирования на изменения в режиме реального времени.