Основные принципы работы синхронных и асинхронных протоколов.

Синхронные протоколы.

Синхронные протоколы, такие как HTTP, работают по модели «запрос-ответ». Это означает, что клиент (например, браузер) отправляет запрос на сервер и ждёт ответа, прежде чем продолжить работу. Каждое новое действие требует нового запроса. По умолчанию HTTP использует одно соединение на запрос. Если заголовок Connection: keep-alive включён, TCP-соединение остаётся открытым для нескольких запросов, но всё равно может разрываться по таймауту. В отличие от WebSocket, HTTP не поддерживает автоматические ping-pong сообщения. У http-запроса есть заголовок (содержит, к примеру: метод, путь) и тело (содержит, например, какие-то данные в формате json). Может использоваться и используется в REST API, как мы работали с ним на парах.

Асинхронные протоколы.

WebSocket – это асинхронный протокол, который позволяет двустороннюю связь между клиентом и сервером. После установки соединения данные могут передаваться в реальном времени без постоянных перезапросов. Для поддержания соединения используются ping/pong пакеты: клиент и сервер кидают друг-другу маленькие пакеты, чтобы соединение поддерживалось. В отличие от HTTP, где клиент всегда запрашивает данные, WebSocket позволяет серверу отправлять сообщения без запроса (push-уведомления). Также клиент может пытаться пере подключиться с задержкой. При помощи асинхронных протоколов можно реализовать схему publish-subscribe, где есть,к примеру, клиент, который может подписаться на условный канал и сервер будет слать в канал сообщения, его задача - отправить.