Хотелось бы рассказать о протоколе **CoAP**. Это сокращение от Constrained Application Protocol, что в переводе значит Ограниченный протокол приложений. Данный протокол используют при передачи данных через Интернет, он работает на прикладном уровне и требуется там, где сильно ограничены ресурсы и используемого устройства, и сети.

СЛАЙД

Он работает с Интернетом Вещей.  Интернет вещей это такая единая сеть, которая объединяет окружающие нас объекты реального мира и виртуальные объекты. На самом деле, люди стремятся сделать всё «умным», именно при помощи Интернета Вещей. Но многие путают понятие автоматизации с реально Умным прибором. По-настоящему Умным считается то, что само знает когда и зачем оно должно выполнять работу, без команды человека. Требуется только результат. Человек не должен переживать, как это должно происходить. Например, при пробуждении человека утром, чайник сам начинает кипятить воду, а плита разогревать завтрак, с учетом того, что это может произойти в любое время.

СЛАЙД

Данный протокол разработан специально для того, чтобы напрямую связать электронные интеллектуальные устройства с приложениями. По сути, он контролирует различные датчики и приборы. Например, электрические лампы, которые включаются, когда замечают звук или движение. Это наиболее простая разработка с использованием датчиков

СЛАЙД

**Безопасность**. Интернет Вещей был бы никому не нужен, если бы он не имел достойную защиту от взлома и вмешательства. А эта проблема действительно существует, поскольку работа происходит в локальных сетях. Но протокол CoAP на самом деле обеспечивает надежную защиту. И имеет мощные алгоритмы шифрования, которые способны работать в данной системе.

СЛАЙД

О внутреннем устройстве протокола.

URI - это символьная строка, которая позволяет идентифицировать какой-либо ресурс. В данном протоколе она делится на: компоненты хоста, порта, пути и запроса, при этом использует значения, которые даны по умолчанию, для повышения эффективности.

СЛАЙД

CoAP имеет две основные функции: обнаружение ресурсов и защищенное обнаружение. Первому соответствует порт с номером 5683, второму – 5684.

СЛАЙД

**Модель REST**. Протоколы HTTP и CoAP подчиняется особо популярной модели REST. **REST** это стиль взаимодействия компонентов веб-приложения в сети. Выглядит это так: серверы хранят данные по адресам URL, и клиент может обращаться к ним стандартными методами, такими как GET – получение ресурса, PUT – обновление ресурса, POST – создание ресурса и DELETE – удалении ресурса.

СЛАЙД

Многие коды ответов аналогичны кодам HTTP. 2.хх говорит об успешном проведении запроса, 4.хх говорит о том, что клиент совершил какую-либо ошибку, 5.хх говорит о том, что ошибка произошла на сервере.

СЛАЙД

Для передачи данных CoAP использует запросы и ответы. Для этого он использует простой, двоичный и базовый форматы заголовков. При чем, после самого заголовка могут идти дополнительные данные(опции) в формате «Тип-Длина-Значение».

Поля заголовка:

1. Версия – номер версии CoAP
2. Тип – указывает тип сообщения (0 – подтверждаемый, 1 – неподтверждаемый, 2 – признан, 3 - сброс)
3. Длина токена
4. Код
5. Идентификатор сообщения – требуется для избегания дублирования сообщений и подтверждения соответствия типа

СЛАЙД

CoAP может интегрировать различные форматы данных. Например, CBOR, JSON, XML.

СЛАЙД

Протокол CoAP работает на двух уровнях:

Request/Responses – этот уровень работает по модели «клиент/сервер» и управляет ресурсами и передачей

Messages – организовывает обмен сообщений между конечными точками.

СЛАЙД

CoAP разработан на основе протокола HTTP, и представляет его двоичную версию. Но, в отличие от протокола HTTP, который использует транспортный протокол ТСР для обмена данными, а CoAP транспортируется через UDP, что позволяет ему работать гораздо быстрее. TCP это транспортный протокол передачи данных в сетях TCP/IP, который сперва устанавливает соединение с сетью, а после чего передает данные и ждет отклика об успешной доставке этих данных, за счет чего отличается особой надежностью. UDP это транспортный протокол, который может передавать сообщения, не требуя при этом установки соединения с сетью, за счет чего некоторые данные могут быть утеряны.

Однако, при использовании протокола UDP для передачи данных, мы можем передавать сообщения небольших размеров, которые не делятся на части, или делятся совсем незначительно. Многие сервера работают так, что им вообще не нужно сохранять свое состояние.

Так как протокол HTTP и протокол CoAP используют одну модель REST, то они могут быть просто соединены друг с другом при помощи специальных proxy.

СЛАЙД

При передачи данных CoAP может использовать «Подтверждающие» сообщения. Суть в том, что после отправки сообщения, при успешной его отправке, возвращается одно подтверждающее сообщение.

СЛАЙД

Пример работы:

Сперва выполняется поиск доступных ресурсов и попытка подключиться к ним. И, после успешного подключения, можно пробовать отправлять запросы при помощи методов GET, POST, PUT, DELETE.

СЛАЙД

GET запрос завершился с кодом 2,05, т.е. показал нам содержание ресурса, к которому мы обращаемся

СЛАЙД

POST запрос завершился с кодом 4,05, что значит, что метод не разрешен

СЛАЙД

DELETE запрос завершился с кодом 2,02, что говорит об успешном удалении ресурса