МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное
образовательное учреждение высшего образования
«Самарский национальный исследовательский университет
имени академика С.П. Королева»

(Самарский университет)

Институт информатики, математики и электроники

Факультет информатики
Кафедра суперкомпьютеров и общей информатики

Отчет по лабораторной работе №4

Дисциплина: «Технологии Интернета вещей»

Выполнил: Мелешенко И.С.

Группа: 6133-010402D.

Дата: 16.05.2022

Самара 2022

СОДЕРЖАНИЕ

1 ОПРЕДЕЛЕНИЕ ПРОТОКОЛА COAP

Одним из широко используемых протоколов для взаимодействия между устройствами сети IoT (Internet of Things) или IoT-устройствами и внешней средой является протокол CoAP (Constrained Application Protocol). Протокол CoAP создан рабочей группой The Internet Engineering Task Force (IETF) Constrained RESTful Environments (CoRE) в июне 2014 г. Стандарт данного протокола описан в документе RFC 7252.

Протокол CoAP предназначен для взаимодействия простых устройств, например датчиков малой мощности, выключателей, клапанов, которые управляются или контролируются удаленно через сеть Интернет. Такие устройства используются в области Интернета вещей, а порождаемый ими информационный обмен называется межмашинным взаимодействием (M2M). Часто подобные устройства называют устройствами с ограниченными ресурсами. Они обычно имеют ограниченный энергоресурс, небольшой объем памяти и невысокую мощность, поэтому в работе с ними важно обеспечивать низкие энергозатраты, использовать передачу сообщений малого объема. Протокол CoAP обеспечивает взаимодействие этих устройств, соблюдая все необходимые требования.

Сеть, в которой работают такие устройства, называют сетью с ограниченными ресурсами. Существенная особенность протокола CoAP – это его совместимость с протоколом HTTP, что обеспечивает при его использовании взаимодействие совокупности устройств IoT, формирующих некую сеть, с всемирной паутиной Интернет.

Рассмотрим такое устройство с ограниченными ресурсами, как датчик. У пользователя в помещении может быть установлено некоторое количество датчиков, управлять которыми он может либо через сеть Интернет, либо непосредственно через сеть с ограниченными ресурсами в том случае, если на его устройстве установлено приложение, работающее по протоколу CoAP.

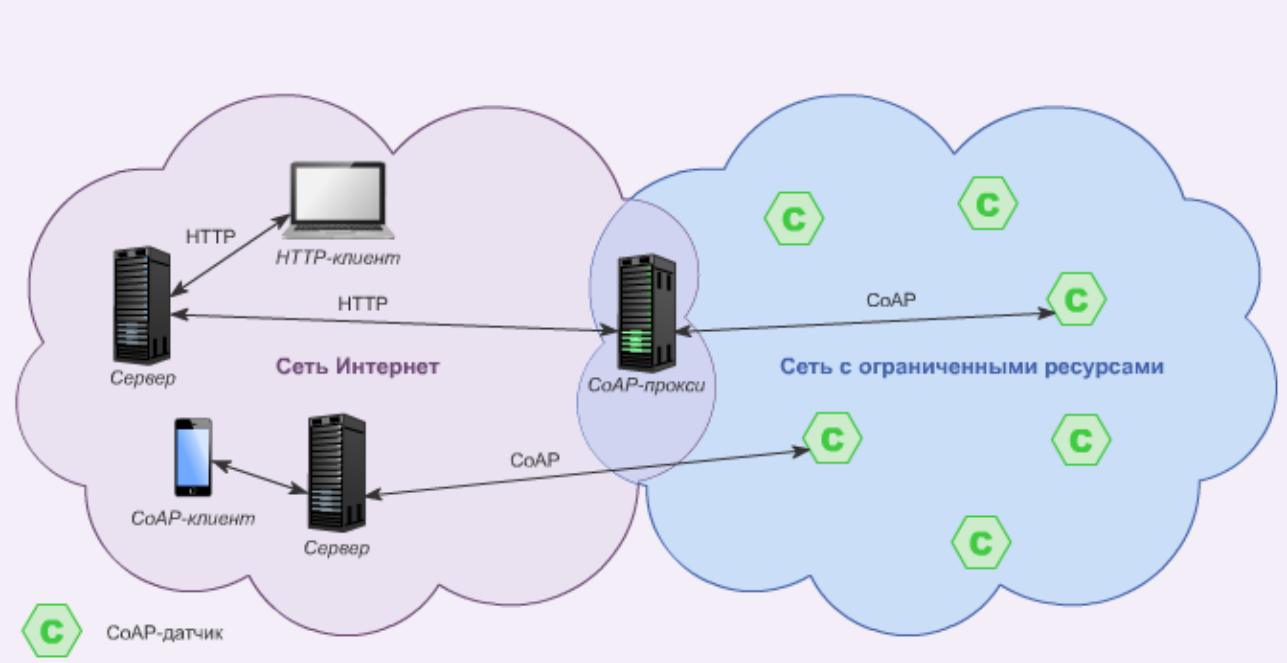
2 СТЕКИ ПРОКОЛОВ, КОТОРЫЕ ИСПОЛЬЗУЕТ СОАР

Протокол CoAP - это протокол прикладного уровня, который использует UDP (User Datagram Protocol), в качестве транспортного протокола по умолчанию, что позволяет уменьшить размер служебных данных и увеличить эффективность работы. В редких случаях могут также использоваться TCP или SCTP.

3 ОСНОВНЫЕ ЧЕРТЫ COAP

- web-протокол, отвечающий требованиям межмашинного взаимодействия M2M в сети с ограниченными ресурсами.
- использование в качестве транспортного протокола – UDP.
- обмен сообщениями происходит по принципу «клиент-сервер»
- асинхронный обмен сообщениями
- поддержка URI и Content-Type.
- совместимость с протоколом HTTP.

4 АРХИТЕКТУРА COAP



5 ПРИМЕР ВЗАИМОДЕЙСТВИЯ ПОЛЬЗОВАТЕЛЬСКОГО УСТРОЙСТВА (HTTP-КЛИЕНТА) С COAP-ДАТЧИКОМ ЧЕРЕЗ СЕТЬ ИНТЕРНЕТ

Рассмотрим пример взаимодействия пользовательского устройства (HTTP-клиента) с CoAP-датчиком через сеть Интернет. HTTP-клиент генерирует HTTP-запросы и отправляет их на сервер, который при отсутствии необходимой информации на полученный запрос обращается к CoAP-прокси.

CoAP-прокси – это устройство, соединяющее сеть Интернет на основе протокола HTTP и сеть с ограниченными ресурсами, поддерживающую протокол CoAP. Прокси преобразует сообщения одной сети (протокол HTTP) в сообщения, понятные для другой (протокол CoAP).

В случае если запрос от пользовательского устройства сразу попадает в сеть с протоколом CoAP, то он поступает на CoAP-датчик. При таком взаимодействии сервер получает все запросы и далее индивидуально однонаправленно посылает соответствующему устройству относящийся к нему запрос и ожидает от него ответ.

6 ОБМЕН СООБЩЕНИЯМИ COAP

Как говорилось ранее, обмен сообщениями CoAP происходит по типу «клиент-сервер».

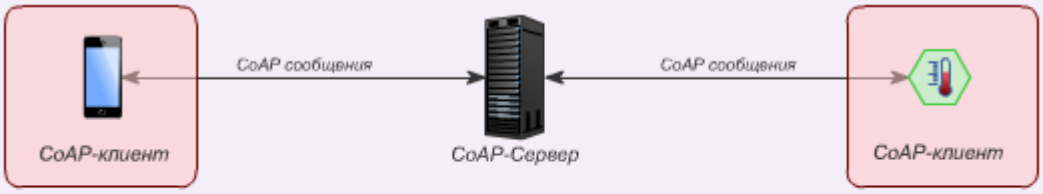
В рассматриваемом примере термостат выступает в роли клиента CoAP, регистрирует состояние и изменение ресурса на заранее известном CoAP-сервере, используя методы PUT и POST.

Приложение (контроллер), запущенное на смартфоне выступает также в роли CoAP-клиента, отслеживает состояние термостата с помощью метода GET и управляет им с помощью метода POST.

Сервер CoAP выполяет действия запрошенные термостатом и контроллером.

Сервер CoAP уведомляет термостат, когда управление ресурсом обновлено контроллером и уведомляет контроллер, когда состояние температуры обновлено термостатом.

7 ОСНОВНЫЕ ХАРАКТЕРИСТИКИ COAP-КЛИЕНТОВ



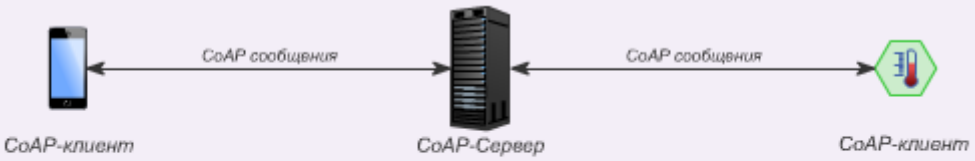
- ✔ CoAP клиенты регистрируют ресурсы на заранее известном CoAP сервере. Клиенты CoAP также могут использовать обнаружение с помощью многоадресной рассылки для динамического поиска серверов, что наиболее подходит для решений, работающих в локальной сети.
- ✔ Клиенты CoAP могут найти ресурсы, доступные на сервере, с помощью функции обнаружения ресурсов.
- ✔ «Умные устройства» и контроллеры могут выступать в качестве CoAP клиентов и выполнять необходимые операции, используя методы.
- ✔ Клиенты CoAP могут отслеживать изменения интересуемых ресурсов с помощью функции наблюдения протокола CoAP.
- ✔ Решения CoAP могут использовать проксирование и кэширование для ограничения сетевого трафика, повышения производительности и по ряду других причин.

8 МОДЕЛЬ REST

Подобно протоколу HTTP, протокол CoAP следует широко распространенной модели REST (Representational State Transfer): серверы предоставляют свои ресурсы по адресам URL, и клиенты обращаются к ним посредством стандартных методов, таких как GET, PUT, POST и DELETE.

В апреле 2017 г. вышел документ RFC 8132, расширяющий протокол CoAP. В данном стандарте добавлены и описаны два новых метода: FATCH и PATCH. Протокол CoAP был дополнен этими методами, так как возникла необходимость некоторым приложениям получать доступ или изменять ресурс не полностью, а взаимодействовать только с определенной его составляющей.

9 МЕТОДЫ COAP И ИХ НАЗНАЧЕНИЕ



Метод	Назначение
GET	выполняет поиск ресурсов и предоставляет информацию, которая соответствует ресурсу, указанному в ссылке URI (Uniform Resource Identifier), где ресурс – это источник данных, которым, например, может являться датчик
PUT	задает новое действие над ресурсом
POST	отвечает за изменения действия над ресурсом
DELETE	служит для удаления активированных возможностей ресурса
FETCH	выполняет предоставление частичной информации о ресурсе по параметрам в запросе
PATCH	отвечает за частичное изменение действия над ресурсом

10 СХЕМА COAP-URI

Доступ к ресурсам выполняется по URI-последовательности символов, идентифицирующей физический или абстрактный ресурс.

Протокол CoAP использует "coap" и "coaps" URI схемы для идентификации ресурсов и определения способа обнаружения ресурсов.

Пользовательское устройство запрашивает данные с датчика через URI-ссылку, но чтобы данный запрос был понятен датчику, он будет преобразован на пользовательском устройстве в один из методов REST и занесен в CoAP-сообщение. Например, при вводе URI-ссылки приложение на пользовательском устройстве создает запрос GET на датчик, где в опциях сообщения протокола CoAP будет содержаться URI.

```
Схема coap-URI  = "coap:" "//" host [ ":" port ] path-abempty [ "?" query
```

Любая схема coap-URI начинается с последовательности "coap:". Далее следует host, который указывает адрес или имя CoAP-сервера. Port содержит номер UDP порта CoAP-сервера. Если используется UDP порт по умолчанию (5683), то в схеме coap-URI он не указывается.

path-abempty определяет ресурс в области хоста и порта. Он состоит из последовательности сегментов пути, разделенных символом "/".

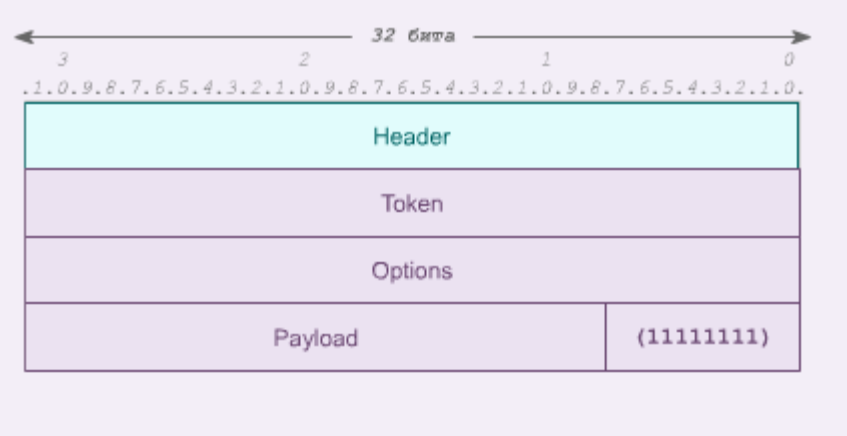
query служит для дальнейшей параметризации ресурса. Он состоит из последовательности аргументов, разделенных символом амперсанда "&". Аргумент часто имеет формат пары "ключ = значение".

11 СХЕМА COAPS-URI

```
Схема coaps-URI = "coaps:" "//" host [ ":" port ] path-abempty [ "?" query ]
```

Схема coaps-URI используется для случаем обеспечения безопасности с помощью протокола DTLS (Datagram Transport Layer Security).

12 ФОРМАТ СООБЩЕНИЙ ПРОТОКОЛА COAP. НАЗНАЧЕНИЕ ПОЛЕЙ ЗАГОЛОВКА



Сообщение протокола CoAP начинается с заголовка, с фиксированным размером в 4 байта, за ним следует маркер, опции и полезная нагрузка.

Все сообщения кодируются в бинарном виде.

Заголовок сообщения протокола CoAP содержит следующие поля.

Ver (Version), Версия (длина 2 бита)

Содержит номер версии протокола CoAP. На данный момент существует только одна версия протокола.

T (Type), Тип (длина 2 бита)

Поле указывает тип сообщения: Confirmable (0), Non-confirmable (1), Acknowledgement (2) или Reset (3).

TKL (Token Length), Длина маркера (длина 4 бита)

Поле указывает длину поля Token (маркера). Длина может быть переменной: 0–8 байт, длины от 9 до 15 зарезервированы.

Code, Код (длина 8 бит)

Поле "код" записывается, как "c.dd", где "c" – это цифра от 0 до 7 (поле длиной 3 бита) и "dd" – две цифры в диапазоне от 00 до 31 (поле длиной 5 бит).

Message ID, Идентификатор сообщения (длина 16 бит)

Содержит уникальный идентификатор сообщения. Message ID идентифицирует сообщение, чтобы определить, на какой запрос пришла информация или ошибка.

CoAP-устройство, отправляя сообщение Confirmable или Non-confirmable, случайным образом генерирует значение идентификатора для этого сообщения. В ответных сообщениях Acknowledgement- или Reset-идентификаторы будут те же, что и в сообщении, на которое они отвечают.

Token, Маркер (длина от 0 до 8 байт, указывается в поле TKL)

Поле содержит значение маркера, которое случайным образом генерируется устройством CoAP и используется в пределах одной сессии для установления соответствия запроса с ответом, т.е. происходит группировка сообщений по цепочке "запрос-ответ".

Нулевое значение маркера используется, когда никакие другие маркеры не используются в устройстве, на которое отправляется запрос, или если запросы делаются последовательно и в малом количестве.

Options, Опции

Далее следуют опции Options, в которых описываются различные параметры сообщений. Например, есть Max-Age Option, которая устанавливает максимальное время хранения информации во временной памяти, по умолчанию этот параметр равен 60 сек. Другая опция, Content-Format Option, задается в виде числа, которое устанавливает формат представления полезной нагрузки: значение 0 указывает на то, что полезная нагрузка будет текстовой, а значение 41 указывает на то, что полезная нагрузка будет в формате xml. Для адресации по URI в протоколе CoAP предусмотрены опции Uri-Host (определяет адрес интернет-хоста запрашиваемого ресурса), Uri-Port (определяет номер порта транспортного уровня запрашиваемого

ресурса), Uri-Path (определяет часть пути до ресурса) и Uri-Query (определяет параметр, который запрашивает ресурс).

Например:

Uri-Host = "example.net"

Uri-Port = 5683

Uri-Path = ".well-known"

Uri-Path = "core"

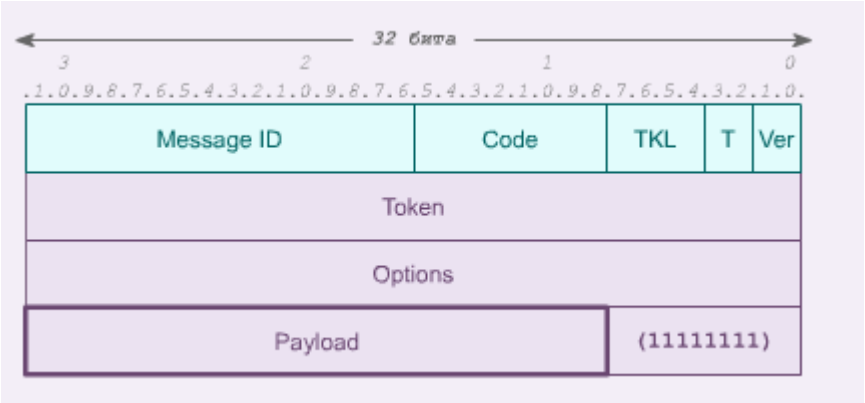
Uri-Query = "login?"

coap://example.net:5683/.wellknown/core/ login?

Полю полезной нагрузки (в случае ее наличия) предшествует фиксированный однобайтный маркер полезной нагрузки Payload Marker (0xFF), который указывает на окончание опции и начало полезной нагрузки.

Payload, Полезная нагрузка

Поле, завершающее формат сообщения, – это полезная нагрузка Payload, в которой содержится запрашиваемая информация.



13 ТИПЫ СООБЩЕНИЙ COAP И ИХ НАЗНАЧЕНИЕ. СПОСОБЫ ДОСТАВКИ COAP (И ДИАГРАММЫ ОБМЕНА СООБЩЕНИЯМИ ЭТИХ СПОСОБОВ)

В протоколе CoAP определено всего четыре типа сообщений: Confirmable, Non-confirmable, Acknowledgement, Reset.

Confirmable (CON) – сообщение, содержащее запрос или ответ, требующее подтверждения и считающееся надежным. Каждое сообщение Confirmable вызывает одно ответное сообщение подтверждения или сброса.

Non-confirmable (NON) – сообщение, содержащее запрос или ответ, не требующее подтверждения и надежной передачи, так как передается регулярно (например, показания от датчика).

Acknowledgement (ACK) – сообщение, подтверждающее, что пришло сообщение Confirmable. При этом само сообщение Acknowledgement не означает успех или неудачу любого из запросов, содержащегося в сообщении Confirmable.

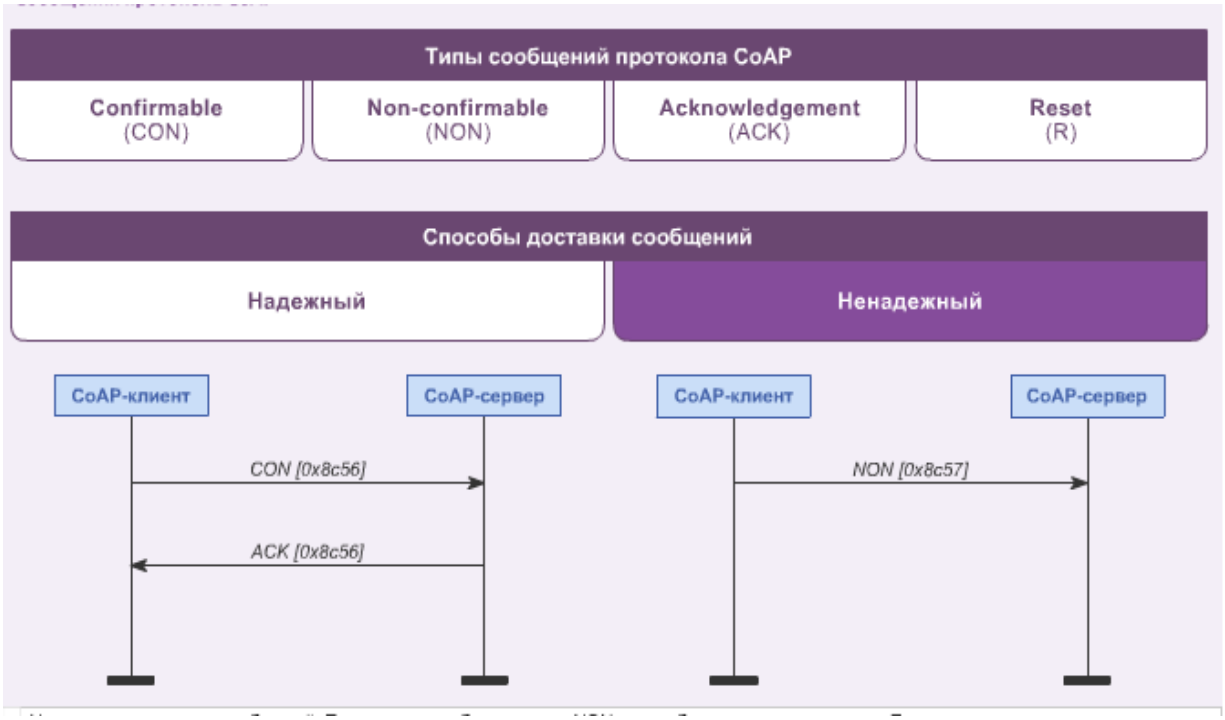
Reset (R) – сообщение сброса, указывающее на то, что конкретное сообщение (Confirmable или Non-confirmable) было получено, но некоторая часть текста отсутствует и невозможно правильно его обработать. Такая ситуация обычно возникает, когда принимающий узел перегружен. Вызов этого сообщения (например, путем отправления пустого сообщения Confirmable) также полезен в качестве проверки доступности узла (CoAP ping).

В протоколе CoAP организовано два способа доставки сообщений для обеспечения качества обслуживания QoS: надежная доставка сообщений и ненадежная.

Надежная доставка сообщений. Поддержка повторной передачи сообщения типа CON, пока не будет получено подтверждение ACK с таким же

Message ID (в данном примере Message ID: 0x8c56). Если получателю не удается обработать сообщение, то он отвечает сообщением RST.

Ненадежная доставка сообщений. Передается сообщение типа NON, не требующее подтверждения. Если получателю не удается обработать сообщение, то он отвечает сообщением RST.



14 ПРИВЕДИТЕ ПРИМЕР СЦЕНАРИЯ ВЗАИМОДЕЙСТВИЯ УСТРОЙСТВ ПО ПРОТОКОЛУ COAP И ОПИШИТЕ ЕГО.

Рассмотрим различные примеры сценариев взаимодействия устройств по протоколу CoAP.

Сценарий 1 - Запрос списка ресурсов, данные о которых имеются на сервере.

CoAP-клиент запрашивает у сервера список ресурсов, данные о которых имеются на сервере. Такой запрос будет отправлен по CoAP протоколу методом GET с URI /.well-known/core (т.к. по умолчанию точкой входа в каталог ресурсов на сервере является URI /"Well-Known/Core"). В ответ сервер передает список, имеющихся у него ресурсов (в данном случае температуры "TemperatureC" и освещенности "LightLux").

Далее после получения информации о всех ресурсах, данные о которых имеются на сервере, CoAP-клиент запрашивает текущее значение температуры. Новый запрос будет отправлен по CoAP-протоколу методом GET с URI "/s/t". На полученный запрос сервер передает ответ, содержащий значение температуры 23,5°C.

15 МНОГОАДРЕСНАЯ РАССЫЛКА

Протокол CoAP поддерживает возможность отправлять запрос сразу группе устройств, реализуя таким образом многоадресную рассылку (multicast).

CoAP-устройства с ограниченными ресурсами могут объединятся в группы либо по функциональности устройств (снятие показаний света или температуры), либо по местоположению (снятие показаний в определенной комнате, этаже здания) и т.д.. Существует несколько способов создания групп.

В первом случае задание группы происходит по групповому IP-адресу или по полному имени хоста, т.е. FQDN (Fully Qualified Domain Name), которому в соответствии с системой DNS сопоставляется групповой IP-адрес.

Во втором случае задание группы происходит через каталог ресурсов (RD – Resource Directory). RD хранит URI, по которым предоставляется доступ к датчикам. При помощи методов REST датчики регистрируются в каталоге и периодически обновляют информацию о себе.

В третьем случае датчик определяется в группу пользовательским устройством, с которого отправляется запрос при помощи формата обмена данными Java Script Object Notation (JSON).

Рассмотрим сценарий отправки запроса протокола CoAP с использованием многоадресной рассылки.

Многоадресная рассылка всегда осуществляется с помощью сообщения типа Non-confirmable. Пользовательское устройство (CoAP-клиент) отправляет запрос на широковещательный IP-адрес, назначенный для группы. Такой запрос достигает всех датчиков, которые настроены на этот IP-адрес.

В сценарии CoAP-датчики 1, 3 и 5 настроены на IP-адрес группы, поэтому они принимают запрос и отвечают:

- CoAP-датчик 1 посылает успешный ответ (2.05), и он достигает сервера;

- CoAP-датчик 3 посылает успешный ответ, но при передаче он теряется;

- CoAP-датчик 5 не может дать ответ, так как временно недоступен (4.04);

- оставшиеся CoAP-датчики 2 и 4 не настроены на получение данного запроса и просто его отбрасывают.