## *Лекция 2.* Протоколы IoT

**Цель лекции** – ознакомить магистрантов с основами IoT и изучение протоколов для эффективного функционирования этой экосистемы.

### Введение

Интернет вещей (IoT) – это сеть физических устройств, подключенных к Интернету, которые могут обмениваться данными и взаимодействовать друг с другом. Протокол IoT играет ключевую роль в обеспечении эффективной связи между устройствами, управлении данными и их безопасности.

В этом процессе решающую роль играют протоколы, обеспечивающие стандартизированный метод обмена данными между устройствами. Они позволяют различным системам и компонентам взаимодействовать друг с другом независимо от производителя или архитектуры. Важно понимать, что каждый протокол имеет свои особенности, преимущества и недостатки, и выбор правильного решения может существенно повлиять на эффективность и безопасность системы IoT.

В этой лекции мы рассмотрим основные протоколы, используемые в MQTT, CoAP, HTTP и другом IoT, а также обсудим логическую схему системы IoT.

### Протокол IoT

Устройства IoT обычно подключаются к Интернету через сеть IP (Интернет-протокол).

Однако такие устройства, как Bluetooth и RFID, позволяют подключаться к устройствам Интернета вещей локально. В этих случаях будет разница в емкости, расстоянии и используемом элементе памяти. Подключение через IP-сети относительно сложное, требует увеличенной памяти и мощности IoT-устройств, а также нет проблем с расстоянием. С другой стороны, сети без IP требуют относительно небольшого энергопотребления и памяти, но существуют ограничения по расстоянию.

### Уровень канала (связи)

В компьютерных сетях канальный уровень – это самый низкий уровень набора протоколов Интернета, сетевой архитектуры Интернета. Канальный уровень – это группа методов и протоколов связи, которые работают только на том канале, к которому физически подключен хост. Канал – это физический и логический компонент сети, используемый для соединения хостов или узлов в сети, а протокол канала (связи) – это набор методов и стандартов, которые работают только между соседними сетевыми узлами локального сегмента сети или глобального сетевого соединения. В таблице 2.1 ниже показаны различные методы канального уровня с разными стандартами.

Таблица 2.1. Стандарты уровня канала

| Нет | Стандартный | Описание |
|---|---|---|
| **Стандарт Ethernet** | | |
| 1 | 802.3 | Коаксиальный кабель |
| 2 | 802.3.i | Медная витая пара |
| 3 | 802.3.j | Волоконная оптика |
| 4 | 802.3.ae | Оптоволокно, 10 Гбит/с |
| **Стандарт Wi-Fi** | | |
| 1 | 802.11a | 5 ГГц |

| | | |
|---|---|---|
| 2 | 802.11b, 802.11g | 2,4 ГГц |
| 3 | 802.11.n | 2,4/5 ГГц |
| 4 | 802.11.ac | 5 ГГц |
| 5 | 802.11.ad | 60 ГГц |
| **Стандарт WiMax** | | |
| **1** | 802.16m | Для мобильных станций - 100 Мбит/с, для фиксированных станций - 1 Гбит/с. |
| **Стандарты мобильной связи** | | |
| **1** | 2G | GSM-CDMA |
| **2** | 3G | UMTS, CDMA 2000 |
| **3** | 4G | LTE |
| **4** | 5G | LTE-Advanced Pro |

**Сетевой (Интернет) уровень**

Сетевой уровень отвечает за передачу данных и связь от одного хоста к другому сетевому хосту. Вместо описания того, как передаются данные, он реализует эффективный метод передачи. Протоколы используются на сетевом уровне для обеспечения эффективной связи. Данные группируются в пакеты или, в случае очень больших данных, делятся на более мелкие подпакеты. Каждый используемый протокол имеет свои особенности и преимущества.

Основные сервисы сетевого уровня:

- **Маршрутизация**: сетевой уровень определяет оптимальный способ передачи данных через сеть. Он использует различные протоколы маршрутизации, такие как RIP, OSPF или BGP, для поиска эффективных маршрутов.
- **Адресация**: каждое устройство в сети должно иметь уникальный адрес, чтобы другие устройства могли его идентифицировать и отправлять на него данные. Интернет вещей часто использует IP-адреса, хотя некоторые протоколы могут использовать свои собственные схемы адресации.
- **Управление потоком данных**: на сетевом уровне контролируется скорость передачи данных, чтобы избежать перегрузки сети. Это важно в средах с ограниченными ресурсами, типичных для многих устройств Интернета вещей.
- **Обработка ошибок**. Сетевой уровень должен иметь возможность обнаруживать и обрабатывать ошибки, возникающие во время передачи данных. Это может включать перераспределение потерянных пакетов или корректировку маршрутов.

На сетевом уровне используются различные протоколы для обеспечения надежной передачи данных.

**IP (Интернет-протокол)**. Протокол IP помогает однозначно идентифицировать каждое устройство в сети. Протокол IP отвечает за передачу данных от одного узла к другому в сети. Протокол IP является протоколом без установления соединения, поэтому он не гарантирует доставку данных. Протоколы высокого уровня, такие как TCP, используются для успешной передачи данных. Протокол IP делится на два типа:

- **IPv4:** IPv4 обеспечивает 32-битную схему адресации. Адрес IPv4 состоит из четырех числовых полей, разделенных точками. IPv4 можно настроить через DHCP или вручную. Поскольку IPv4 не поддерживает методы аутентификации или шифрования, он не предоставляет дополнительных функций безопасности. IPv4 далее делится на пять классов: класс A, класс B, класс C, класс D и класс E.
- **IPv6:** IPv6 – это последняя версия IP. Он имеет 128-битную схему адресации. IP-адрес состоит из восьми полей, разделенных двоеточиями, и эти поля являются буквенно-цифровыми. Адрес IPv6 задается в шестнадцатеричном формате. IPv6 предоставляет дополнительные функции безопасности, такие как аутентификация и шифрование. IPv6

обеспечивает целостность связи. IPv6 предоставляет более широкий диапазон IP-адресов, чем IPv4.

**6LoWPAN (IPv6 over Low-Power Wireless Personal Area Networks)** – позволяет передавать пакеты IPv6 по маломощным сетям, таким как Zigbee.

**ARP (протокол разрешения адресов).** ARP используется для преобразования логического адреса (IP-адреса) в физический адрес (MAC-адрес). При общении с другими узлами необходимо знать MAC-адрес или физический адрес целевого узла. Если какой-либо из узлов сети хочет узнать физический адрес другого узла в той же сети, хост отправляет пакет запроса ARP. Этот пакет запроса ARP состоит из IP-адреса и MAC-адреса исходного хоста и только IP-адреса целевого хоста. Пакет ARP принимается каждым узлом сети. Узел с собственным IP-адресом распознает его и отправляет MAC-адрес запрашивающему узлу. Но передача и получение таких пакетов для выяснения MAC-адреса целевого узла увеличивает нагрузку на трафик. Поэтому, чтобы уменьшить этот трафик и повысить производительность, системы, использующие ARP, кэшируют последние полученные привязки IP-адреса к MAC-адресу.

**Транспортный уровень**

В компьютерных сетях транспортный уровень представляет собой концептуальное разделение методов многоуровневой архитектуры протоколов сетевого стека в наборе протоколов Интернета и модели OSI. Протоколы на этом уровне предоставляют приложениям услуги связи между хостами. Он предоставляет такие услуги, как транспорт, ориентированный на соединение, надежность, управление потоком и мультиплексирование.

Самым популярным транспортным протоколом набора протоколов Интернета является протокол управления передачей (TCP). Он используется для транспорта с установлением соединения, а протокол пользовательских дейтаграмм (UDP) без установления соединения используется для упрощения обмена сообщениями.

TCP – один из основных протоколов набора протоколов Интернета. Он находится между уровнями приложений и сети, используемыми для предоставления надежных транспортных услуг. Это протокол связи, который помогает обмениваться сообщениями между различными устройствами по сети. Протокол IP, определяющий метод распределения пакетов данных между компьютерами. Работает с TCP. TCP имеет несколько ключевых особенностей. Он отслеживает передаваемые или получаемые сегменты , присваивая каждому уникальный номер . Такой подход позволяет реализовать управление потоками, ограничивающее скорость передачи данных от дистрибьютора, что обеспечивает надежную доставку информации. Кроме того, TCP реализует механизм управления ошибками, который гарантирует надежную передачу данных, а также учитывает уровень перегрузки в сети.

Приложения TCP охватывают множество областей. Например, во Всемирной паутине (WWW) протокол обеспечивает надежную связь между браузерами и веб-серверами. Он также используется для отправки и получения электронной почты, где такие протоколы, как SMTP, управляют доставкой сообщений между серверами. Передача файлов по FTP также опирается на TCP, чтобы обеспечить целостность данных во время загрузки и скачивания для безопасной передачи больших файлов. Сеансы SSH, которые часто используются для удаленного управления, полагаются на TCP для зашифрованной связи между клиентом и сервером. Потоковые сервисы, такие как Netflix, YouTube и Spotify, используют TCP для обеспечения плавного воспроизведения видео и музыки за счет управления сегментами данных и повторной передачи.

Среди преимуществ TCP – надежность в поддержании связи между отправителем и получателем, а также возможность отправлять данные в определенном порядке. Операции TCP не зависят от операционной системы и поддерживают различные протоколы

маршрутизации. Кроме того, он может регулировать скорость передачи данных в зависимости от скорости получателя.

Однако TCP также имеет свои недостатки. Он медленнее и требует большей пропускной способности, чем UDP. Кроме того, передача файлов запускается медленно, что не всегда подходит для локальных и частных сетей. TCP не поддерживает многоадресную и широковещательную рассылку и не загружает всю страницу, если какой-либо элемент страницы отсутствует.

Протокол пользовательских дейтаграмм (UDP) – это протокол транспортного уровня. UDP является частью набора протоколов IP, известного как набор UDP/IP. В отличие от TCP, это протокол, не требующий доверия и установления соединения. Тогда нет необходимости устанавливать соединение перед передачей данных. UDP помогает установить соединения с низкой задержкой и устойчивостью к потерям в сети. UDP обеспечивает связь между процессами.

UDP это известный один приложения для подходящий что делает несколько основной по спецификациям имеет Это протокол запросы я ответов простой обмен особенно для данные объем маленький был когда и транслировался управлять и не тот человек строго требования не существовало когда используется . Он также бок о бок идеально подходит для многоадресной рассылки, поскольку обеспечивает распределение пакетов. UDP используется в некоторых протоколах обновления маршрутизации, таких как RIP, и часто используется в реальных приложениях, где важна минимальная задержка между частями полученного сообщения.

Среди его основных приложений – потоковая передача мультимедиа в реальном времени. Благодаря низкой задержке UDP обеспечивает плавное воспроизведение аудио и видео даже в случае потери данных. В онлайн-играх многие разработчики выбирают UDP для быстрой и эффективной связи между игроками. Этот протокол также обрабатывает запросы к системе доменных имен (DNS), обрабатывая преобразование доменных имен в IP- адреса. Инструменты сетевого мониторинга используют UDP для быстрого обмена легкими данными. Его способность поддерживать несколько передач делает UDP подходящим, когда данные необходимо отправить нескольким получателям одновременно. Некоторые протоколы маршрутизации, такие как RIP, используют UDP для обмена информацией о маршрутизации.

Среди преимуществ UDP можно отметить отсутствие необходимости устанавливать соединение для передачи данных, поддержку вещания и ретрансляции, а также возможность работы в различных сетевых условиях. Он подходит для передачи реальных данных, а также может передавать информацию в неполном виде.

Однако у UDP есть свои недостатки. Протокол не предоставляет механизма подтверждения успешной передачи данных и не контролирует последовательность пакетов. Поскольку UDP – это протокол без установления соединения, он не очень надежен для передачи данных. Пакеты UDP могут быть отброшены маршрутизаторами в случае коллизий, а протокол может потерять пакеты в случае обнаружения ошибок.

**Уровень приложения**

Уровень приложений – это уровень абстракции, который определяет общие протоколы связи и методы интерфейса, используемые хостами в сети связи. Абстракция прикладного уровня используется в обеих стандартных моделях компьютерных сетей: в наборе протоколов Интернета (TCP/IP) и в модели OSI. Хотя в обеих моделях для обозначения более высокого уровня используется один и тот же термин, определения и цели различаются. Уровень приложений – это интерфейс между устройством IoT и сетью, с которой оно взаимодействует. Он служит мостом между работой, которую выполняет устройство IoT, и данными, которые оно производит, которые распространяются по сети.

У инженеров имеется множество протоколов прикладного уровня Интернета вещей, охватывающих широкий спектр функций. Совместимый протокол для конкретного

приложения IoT зависит от ряда факторов, основанных на типе подключенного устройства и выполняемой им функции: **задержка данных, надежность, пропускная способность, транспортировка.**

Ниже описаны наиболее важные протоколы для Интернета вещей и их возможности.

**CoAP – Constrained Application Protocol.** Организации используют CoAP с ограниченным оборудованием и низкой скоростью передачи данных из-за простоты использования. Протокол совместим с HTTP и использует два основных типа сообщений: запрос и ответ. Сообщения могут быть подтвержденными или неподтвержденными. Пакеты данных небольшие, поэтому потери сообщений невелики. Обратной стороной является отсутствие безопасности в протоколе, которое инженеры обычно могут исправить с помощью безопасности транспортного уровня датаграмм, но DTLS имеет ограниченное применение в IoT.

**MQTT – Message Queue Telemetry Transport.** MQTT — это эффективный протокол публикации/подписки без потерь для упрощенного межмашинного взаимодействия (M2M ) по TCP . Для Интернета вещей публикация/подписка означает, что клиентам не нужно запрашивать обновления, что снижает сетевой трафик и нагрузку на обработку. Протокол также охватывает ряд уровней обеспечения качества, от подтверждения установления связи до требований подтверждения .

**XMPP – Extensible Message and Presence Protocol.** XMPP — это язык разметки кодирования документов, известный своей удобочитаемостью. На основе XML. Будучи расширением HTML, XMPP полезен для общения в реальном времени, включая взаимодействие, распространение контента и обмен мгновенными сообщениями.

**AMQP – Advanced Message Queuing Protocol.** AMQP - синхронный протокол . Как и MQTT , он использует подход публикации/подписки. Инженеры используют этот протокол в основном через TCP, но он поддерживает и другие способы транспортировки. AMQP обеспечивает безопасность посредством Transport Layer Security (TLS) и Secure Sockets Layer (SSL) .

**REST – Representational State Transfer.** REST — наиболее распространенный протокол, обеспечивающий синхронный ответ IoT на запросы через HTTP. HTTP делает его многофункциональным, а также обеспечивает аутентификацию и кэширование. Оба полезны в сложных средах, но их сложно реализовать в IoT. Протокол совместим с XML и JSON, которые можно использовать для связи M2M с планшетами и смартфонами.

**Контрольные вопросы:**

1. Что такое Интернет вещей (IoT) и какую роль играют протоколы в этой экосистеме?
2. Опишите основные характеристики, которым должны соответствовать протоколы IoT.
3. Какие протоколы канального уровня вы знаете и какие из них наиболее подходят для IoT-устройств?
4. Каковы основные отличия между протоколами IPv4 и IPv6, и почему IPv6 считается более подходящим для IoT?
5. Объясните, почему MQTT становится стандартом для IoT-приложений. Как он отличается от HTTP?
6. Каковы преимущества и недостатки использования TCP и UDP на транспортном уровне для IoT?
7. Какое значение имеет безопасность при проектировании протоколов для IoT?