ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И ТЕЛЕКОММУНИКАЦИИ
СПбГУТ))) Июнь 2017 Том 5 № 2 ISSN 2307-1303 http://www.itt.sut.ru
TELECOM IT
June 2017 Vol. 5 Iss. 2 ISSN 2307-1303 http://www.itt.sut.ru

ОБЗОР ПРОТОКОЛОВ ИНТЕРНЕТА ВЕЩЕЙ

Т. А. Москаленко[1], Р. В. Киричек[1*], А. Е. Кучерявый[1]

[1] СПбГУТ, Санкт-Петербург, 193232, Российская Федерация
* Адрес для переписки: kirichek@sut.ru

Аннотация
Принимая во внимание сегодняшнее разнообразие технологий и устройств, подключенных к сети, огромное количество производителей, возникает множество проблем их взаимодействия и необходимость в создании и принятии специализированных стандартов и протоколов связи. **Предмет исследования.** Статья посвящена описанию различных стандартов и протоколов, предлагаемых IEEE, IETF и МСЭ для использования в рамках концепции Интернета вещей. **Метод.** Статья включает в себя описание протоколов прикладного, транспортного, сетевого и канального уровней модели TCP/IP. Также рассмотрены протоколы, разработанные специально для удовлетворения требований IoT. **Основные результаты.** Приведено краткое описание протоколов и принцип работы каждого из них. **Практическая значимость.** Данная обзорная статья способствует более корректному выбору протоколов при построении сети, обнаружении и управлении устройствами, налаживания взаимодействия между объектами Интернета вещей.

Ключевые слова
Интернет вещей, протоколы IoT, стандарты, D2D, M2M.

Информация о статье
УДК 004.7
Язык статьи – русский.
Поступила в редакцию 21.04.17, принята к печати 02.06.17.

Ссылка для цитирования: Москаленко Т. А., Киричек Р. В., Кучерявый А. Е. Обзор протоколов Интернета вещей // Информационные технологии и телекоммуникации. 2017. Том 5. № 2. С. 1–12.

AN OVERVIEW OF IOT PROTOCOLS

T. Moskalenko[1], R. Kirichek[1*], A. Koucheryavy[1]

[1] SPbSUT, St. Petersburg, 193232, Russian Federation
* Corresponding author: kirichek@sut.ru

Abstract—Having taken for attention today's diversity of network technologies and devices, a variety of manufacturers, there are many problems of their simultaneous work and the necessity of creation and adoption of specialized standards and communication protocols. **Research subject.** The various

standards and protocols proposed by the IEEE, IETF and ITU for use within the concept of the Internet of things are considered in the article. **Method.** The article describes protocols of the application, transport, network and link layers of the TCP/IP model. Also, protocols developed specifically to meet IoT requirements are considered. **Core results.** A brief description of the protocols and the operation principle of each of them are given. **Practical relevance.** The review article leads to a more correct choice of protocols when building a network, detecting and managing devices, establishing interaction between objects of the Internet of things.

Keywords—Internet of things, IoT protocols, IoT standards, D2D, M2M.

Article info
Article in Russian.
Received 21.04.17, accepted 02.06.17.

For citation: Moskalenko T., Kirichek R., Koucheryavy A.: An Overview of IoT Protocols // Telecom IT. 2017. Vol. 5. Iss. 2. pp. 1–12 (in Russian).

Введение

Сегодня технологии Интернета вещей активно внедряются во все сферы жизни общества, позволяя использовать различные устройства, не обязательно физические, для создания конкретных решений, способных облегчить жизнь человечества. Устройства становятся способными слышать, видеть, думать, в некоторых случаях действовать [1, 2]. Для правильной и эффективной работы устройства должны корректно общаться и координировать свои действия с другими для того, чтобы принимать решения, которые могут быть столь критичны, как спасение жизней или зданий. Технологии распределенных вычислений, встроенные датчики, современные беспроводные технологии позволяют Интернету вещей выполнять поставленные задачи. Однако принимая во внимание сегодняшнее разнообразие данных технологий и устройств, огромное количество производителей, возникает множество проблем их взаимодействия и необходимость в создании и принятии специализированных стандартов и протоколов связи [3].

Разработка успешных приложений IoT включает в себя задачи обеспечения мобильности: при перемещении IoT устройства меняется IP-адрес, следовательно, необходима налаженная работа протоколов маршрутизации; надежности (система должна быть очень надежной и быстрой в плане сбора и передачи данных и принятия решений), масштабируемости, т.е. возможности расширения пользователей сети. Концепция Интернета вещей предполагает, что к сети будет подключены миллионы устройств. Также среди задач необходимо отметить обеспечение управления и доступности: отслеживание сбоев, конфигурации и производительности такого большого количества устройств, за что отвечают соответствующие протоколы управления. Кроме того необходимо обеспечить совместимость в сети: гетерогенные устройства и протоколы должны быть в состоянии работать друг с другом с учетом сохранения безопасности и конфиденциальности.

В общем случае принята следующая модель взаимодействия устройств в сети Интернета вещей. Оконечные устройства, датчики, сенсоры общаются друг с другом (так называемое взаимодействие D2D – *Device to Device*). Данные, собранные устройствами, отправляются на сервер для последующего анализа

и обработки (взаимодействие D2S – *Device to Server*). Этот сервер может включать в себя несколько вычислительных машин или объектов, которым также необходимо общаться между собой (взаимодействие S2S – *Server to Server*). Для выполнения различных задач необходимо использование различных протоколов. Далее приведены наиболее распространенные и перспективные протоколы на сегодняшний день, дано краткое описание каждому из них.

Обзор существующих протоколов

Asterisk. Стек различных протоколов (SIP, H.323, MGCP и др.) для мониторинга и управления системой компьютерной телефонии Asterisk путем отправки команд CLI и обработки ответов[1]. Среди функций Asterisk можно выделить голосовую почту, конференц-связь, интерактивное голосовое меню и т. д. Asterisk используется для управления и контроля качества обслуживания виртуальной инфраструктуры, беспроводных, голосовых сетей.

BACnet (*Building Automation and Control Networks*). Открытый протокол автоматизации и управления инженерными сетями. Поддерживает спецификации BACnet IP и BACnet MS/TP для чтения/записи свойств объектов, доступа к сервисам устройств и обработки оповещений[2]. Протокол BACnet является международным стандартом ISO 16484-5. Используется для домашней и промышленной автоматизации, управления процессами, тестирования и измерений, межмашинного взаимодействия (M2M).

CAP (*Common Alerting Protocol*). Протокол общего оповещения, предназначенный для обмена сообщений CAP[3]. Используется для приема и передачи сигналов об аварийных или нештатных ситуациях.

CoAP (*Constrained Application Protocol*). Веб-протокол передачи данных для использования в ограниченных узлах и сетях Интернета вещей [4]. В отличие от HTTP CoAP на транспортном уровне использует протокол UDP, клиент и сервер взаимодействуют без установления соединения. Основными преимуществами использования CoAP для IoT является простота, низкие затраты памяти и питания.

CORBA (*Common Object Request Broker Architecture*). Протокол для осуществления интеграции изолированных систем, который даёт возможность программам, написанным на разных языках программирования, работающим в разных узлах сети, взаимодействовать друг с другом так же просто, как если бы они находились в адресном пространстве одного процесса[4]. Протокол необходим для выполнения вызовов CORBA через IP сеть со спецификацией входных параметров и обработкой данных ответа.

CWMP (CPE WAN *Management Protocol*). Протокол управления и мониторинг абонентским оборудованием (CPE) в соответствии со спецификацией TR-069[5]. Протокол предназначен для автоконфигурации и динамической подготовки сервисов к работе, управления версиями программного обеспечения, мониторинга состояний и производительности, диагностики устройств и сети в целом.

[1] Официальный сайт Asterisk. URL: http://www.asterisk.org

[2] Официальный сайт BACnet. URL: http://www.bacnet.ru

[3] Рекомендация X.1303. Общий протокол оповещения (CAP 1,1). 2007.

[4] Официальный сайт CARBA. URL: http://www.corba.org

[5] Broadband forum TR-069 CPE WAN Management Protocol Issue. Ноябрь 2013.

DHCP (*Dynamic Host Configuration Protocol*). Протокол динамической настройки узла, позволяющий устройствам автоматически получать IP-адрес и другие параметры, необходимые для работы в сети TCP/IP [5]. В IoT служит для мониторинга работоспособности DHCP-сервера.

DLMS/COSEM (*Distribution Line Message Specification | COmpanion Specification for Energy Metering*). Протокол, базирующийся на концепциях модели OSI, регламентирующий обмен данными между приборами учета и системами сбора данных, в основе которого лежит клиент-серверная архитектура. Основополагающими спецификациями в этом стандарте являются DLMS и COSEM[6]. Используется для получения текущих показаний приборов учета и истории. Удаленный сбор данных экономит поставщикам коммунальных услуг расходы, связанные со сбором показаний счетчиков. Еще одним преимуществом является то, что выставленные счета будут соответствовать реальным показателям потребления, а не основываться на оценке.

DNP3 (*Distributed Network Protocol*). Протокол передачи данных между объектами сети IoT[7]: чтение/запись, выбор данных и управление процессами: прямое управление, управление событиями и т. д. Для безопасной аутентификации используется расширение Secure Authentication.

DNS (*Domain Name System*). Протокол для получения IP-адреса по имени устройства [6]. В рамках IoT используется для мониторинга работоспособности DNS-сервера.

Ethernet/IP. Открытый промышленный протокол, который поддерживает обмен сообщениями (обмен сообщениями ввода/вывода в реальном времени). Стандарт EtherNet/IP обеспечивает объединение в единое информационное пространство всех компонентов систем автоматизации IoT – от уровней средств ввода/вывода, контроллерного оборудования, серверов до уровня систем управления предприятием.

EVA-DTS. Протокол передачи данных для торговых автоматов. Стандарт определяет структуру общих элементов данных и описывает средства передачи данных. Используется также для мониторинга торговых автоматов (локальных файлов и папок), сбора доступных метрик, статистики и ошибок EVA-DTS, проверки контрольных сумм, загрузки содержимого файлов в ядро системы.

FTP (*File Transfer Protocol*). Протокол прикладного уровня стека TCP/IP, предназначенный для передачи файлов в сети [7]. В IoT используется для мониторинга атрибутов удаленных файлов, контроля работоспособности FTP-сервера (мониторинг доступности, загрузки процессора, использования дискового пространства и памяти, состояния процессов, а также нестандартных метрик для серверов, работающих на различных платформах и операционных системах).

GPS/GLONASS и M2M Data. Система для получения произвольных отчетов от любых спутниковых датчиков и других устройств M2M через протоколы TCP или UDP. Обработка команд осуществляется на основе бизнес-правил. В IoT данная система лежит в основе управления транспортом и служит для определения и отслеживания местоположения, сбора, хранения, обработки и визуали-

[6] Recommendation IEC 61334-4-41. Distribution automation using distribution line carrier systems. Part 4: Data communication protocols. Section 41: Application protocol - Distribution line message specification. 1996.

[7] IEEE Standard for Electric Power Systems Communications - Distributed Network Protocol (DNP3). 2010. URL: http://ieeexplore.ieee.org/servlet/opac?punumber=5518535

зации различных телеметрических данных. С ее помощью можно также осуществлять мониторинг текущего состояния автомобиля (скорость, топливо/уровень заряда батарей, текущий расход, уровень масла, и др.), эксплуатацию транспортных средств (удаленное выключение двигателя, отправка сообщений водителю или оператору и т. д.), мониторинг профилактических и ремонтных работ, настройку обмена данными с системой управления запасами в режиме реального времени.

HTTP/HTTPS (*HyperText Transfer Protocol*). Протокол прикладного уровня для передачи данных [8]. Основой HTTP является клиент-серверная архитектура. Для IoT более безопасная реализация представляет собой только HTTP клиента на устройстве, т. е. устройство может инициировать соединения, а не получать. Задачами протокола HTTP/HTTPS являются предоставление доступа к внутренним веб-серверам контроллеров/модулей, находящихся в сетях, защищенных брандмауэрами или NAT, загрузка содержимого веб-страниц в ядро системы, мониторинг работоспособности веб-сервера.

ICMP (*Internet Control Message Protocol*). Сетевой протокол, использующийся для оповещения об ошибках на устройствах в сети. Любое устройство имеет возможность отправлять, получать или обрабатывать сообщения ICMP. В большинстве случаев ICMP не используется между конечными пользователями, а используется администратором сети для устранения неполадок или проверки потерь пакетов на маршрутах [9]. Но существуют и пользовательские утилиты, такие как мониторинг доступности (*ping*), трассировка сетевых маршрутов (*traceroute*).

МЭК 60870-5-104. Протокол телемеханики, предназначенный для передачи данных в центры управления[8]. Передача осуществляется после установки TCP соединения. Используется в решениях домашней и промышленной автоматизации (системы кондиционирования, освещения, видеонаблюдения, получение данных от счетчиков, измерительных преобразователей), контроля и управления центрами обработки данных.

IMAP (*Internet Message Access Protocol*). Протокол прикладного уровня модели TCP/IP, предназначенный для доступа к электронной почте [10]. IMAP позволяет не только принимать сообщения, но и управлять электронной почтой прямо на почтовом сервере, т. е. при просмотре письма не скачиваются на устройство, а остаются на сервере. Также протокол необходим для мониторинга работоспособности IMAP-сервера.

IPMI (*Intelligent Platform Management Interface*). IPMI представляет собой набор интерфейсов и протоколов для управления и контроля возможностей программного обеспечения и операционной системы вне зависимости от используемого устройства. Системные администраторы могут использовать IPMI сообщениями для мониторинга состояния серверов и устройств (например, значения температуры, напряжения, состояние вентиляторов, источников питания и т. д.)

JMS (*Java Message Service*). Стандарт для обмена сообщениями, широко использующийся для интеграции серверных приложений, таких как базы данных, аналитических системы и автоматизации бизнес-процессов. JMS используется в основном с Java-приложениями. JMS широко используется в центрах обработ-

[8] ГОСТ МЭК 60870-5-104. Устройства и системы телемеханики Часть 5. Протоколы передачи. 2005.

ки данных, что позволяет легко интегрировать в серверные приложения. Из недостатков для IoT можно отметить высокие требования к питанию, обработке и памяти устройств.

JMX (*Java Management Extensions*). JMX определяет стандарт для написания JMX-объектов, так называемых MBean'ов. Любой JMX-клиент (удаленное или локальное приложение) имеет возможность читать, записывать, вызывать методы и получать доступ к атрибутам этим MBean'ов с помощью контейнера, в котором они содержатся. Также JMX позволяет запрашивать конфигурационные установки серверов и изменять их во время выполнения приложения. Кроме этого JMX осуществляет мониторинг, формирует уведомления о событиях, таймер и выполняет динамическую загрузку классов из XML-файлов.

KNX. Протокол для взаимодействия с устройствами KNX через IP[9]. Система KNX служит для домашней и промышленной автоматизации (управление освещением, климатом, обеспечение безопасности и др.). Все устройства сети (в терминах KNX – датчики, актуаторы и системные устройства) подключаются в проводной шине, через которую и обмениваются информацией с использованием протокола KNX. Обмен информацией обязательно сопровождается подтверждением на каждое сообщение, поэтому данная технология не подходит сетям критическим к задержкам.

LDAP (*Lightweight Directory Access Protocol*). Протокол прикладного уровня для доступа к каталогам, а именно к системе управления базами данных. Является упрощенной версией протокола DAP. LDAP используется для формирования запросов на чтение, поиск, редактирование, установление и разрыва связи, загрузки результатов запросов в ядро системы. Система LDAP содержит в себе готовые схемы хранения данных, например, структуру учетных записей, справочника адресов, прав и привилегий. Таким образом, в каталогах LDAP содержатся данные, необходимые для аутентификации пользователя (имя пользователя, пароль), данные о правах пользователя, пользовательские настройки, данные о группах людей, например, списки рассылки уведомлений.

LON/LonTalk. Протокол для обмена данными, предназначенный для мониторинга и управления сетевыми устройствами, взаимодействующими через различные среды коммуникации такие, как витая пара, линии электропитания, оптоволокно, и беспроводную радиочастотную среду[10]. Протокол используется для задач автоматизации различных функций в промышленном управлении, домашней автоматизации, мониторинга транспортных средств, а также в системах автоматизации зданий таких, как системы управления освещением и системы отопления, вентиляции, кондиционирования, системы интеллектуального здания.

MDB (*Multi Drop Bus*). Протокол для взаимодействия устройств, подключенных к компьютерной шине MDB[11]. В любой точке шины можно определить, какое устройство посылает информацию. Ведомые устройства фильтруют данные в шине, которые им необходимо получить. MDB шины используются в торговых автоматах, где контроллеры обмениваются данными с компонентами торгового автомата, например, приемниками денежных средств. Возможна ра-

[9] Официальный сайт KNX. URL: http://www.konnex-russia.ru

[10] Официальный сайт LON/LonTalk. URL: http://www.lon.ru

[11] MDB / ICP, Supported by the Technical Members of: NAMA, EVA, EVMMA, Version 4.2. 2011.

бота в двух режимах: режим ведомого (мониторинг транзакций) и режим ведущего (обработка транзакций, инициируемых сервером). Также MDB используется для мониторинга входящих данных по последовательному порту или TCP/UDP соединению.

Meter-Bus. Протокол для взаимодействия устройств по шине M-Bus[12]. Технология M-bus преимущественно применяется в автоматизированных системах для снятия показаний с приборов учета электрической энергии (электросчётчики), тепловой энергии (теплосчётчики), расходомеров воды и газа. Данные передаются на компьютерную станцию (сервер) напрямую или через концентраторы шины M-Bus, а также усилители-повторители сигнала.

Modbus. Открытый коммуникационный протокол для обмена данными между сетевыми устройствами, основанный на архитектуре ведущий-ведомый (*master-slave*) Обмен данными представляет собой транзакции, состоящие из запросов и ответов. Преимуществами протокола Modbus являются открытость и массовость использования (множество производителей выпускают различные типы устройств, датчиков, поддерживающих данный протокол). Задачами протокола Modbus являются контроль и управление сетями, чтение и запись данных в регистры хранения, доступ к файлам, диагностика состояния устройств.

MQTT. Сетевой протокол для обмена сообщениями в сетях с низкой пропускной способностью между устройствами, реализующих модель ведущий-ведомый. Работает поверх TCP/IP, который обеспечивает простой и надежный поток данных[13]. Основной целью MQTT является удаленный мониторинг данных, собираемых из большого количества устройств, и их телеметрия в IT-инфраструктуру. Поскольку данные предоставляются для ИТ-инфраструктуры, вся система имеет возможность легко транспортировать данные в корпоративные технологии, такие как ActiveMQ и ESBS. Протокол нацелен на большую сеть небольших устройств, которые необходимо контролировать или управлять из облака. Также он предназначен для «многоадресной передачи» данных для многих приемников. MQTT чрезвычайно прост, предлагая несколько вариантов управления. Примером работы является мониторинг нефтепровода на наличие утечек или вандализма. Это информация от тысячи датчиков, которая должны быть сконцентрирована в одном месте для анализа. Когда система обнаруживает проблему, она может принять меры, чтобы исправить эту проблему.

NetFlow. Открытый протокол, разработанный компанией Cisco для мониторинга трафика в сети, и поддерживается не только оборудованием данной компании, но и устройствами от других производителей. Главными задачами протокола Netflow являются декомпозиция и глубокий анализ сетевого трафика [11]. Netflow предоставляет возможность анализа сетевого трафика на уровне сеансов, делая запись о каждой транзакции TCP/IP. Протокол Netflow содержит в себе три компонента: сенсор, коллектор и анализатор. Сенсор представляет собой устройство, фиксирующее данные о конкретном сеансе связи, которые проходят через него. Иными словами, сенсор фиксирует потоки, проходящие через него. Под потоками подразумеваются все пакеты, имеющие оди-

[12] Официальный сайт Meter-Bus. URL: http://www.m-bus.com
[13] Официальный сайт MQTT. URL: http://mqtt.org/

наковые поля адреса, порта, кода сообщений ICMP, версии протокола IP, сетевого интерфейса. Коллектор необходим для сбора информации от сенсоров и размещении ее в определенных базах данных. Анализатор в свою очередь осуществляет обработку данных и их визуализацию.

NMEA 0183 (*National Marine Electronics Association*). Протокол для передачи сообщений между устройствами, в частности транспортными средствами, оснащенными приемопередатчиками GPS/ГЛОНАСС. Все сообщения передаются в текстовом виде, и каждое сообщение заканчивается значением контрольной суммы, для проверки целостности на приемной стороне. Протокол NMEA 0183 используется в основном для отслеживания местонахождения устройств, времени местоопределения, может содержать данные скорости и направлении движения, количестве используемых спутников для определения координат, соотношении сигнал/шум каждого видимого спутника. В протоколе NMEA 0183 есть ограничение для максимальной длины сообщения, она не должна превышать 80 символов.

ODBC (*Open Database Connectivity*). Программный интерфейс, основанный на SQL, для доступа к базам данных. Предназначен для унификации взаимодействия между приложением и различными источниками данных. Поставщики данных разрабатывают драйвера для взаимодействия со стандартными функциями ODBC API с учетом своего продукта, что позволяет прикладным программистам разрабатывать приложения с возможностью доступа к данным, не изучая особенности работы с базами данных. Пример использования – Integration Manager. Программа, позволяющая объединить несколько сложных баз данных в одну структуру с возможностью упрощенного сбора, представления и передачи информации.

OPC (*OLE for Process Control*). Стандарт интерфейсов для совместной работы средств автоматизации, функционирующих на разных аппаратных платформах, в разных промышленных сетях и производимых разными фирмами. Стандарт работает с таким операционными системами как Windows, Linux и Mac OS. Основной частью стандарта является спецификация OPC DA для протокола взаимодействия между клиентом и аппаратной частью устройства (контроллерами, модулями ввода/вывода и т. д.) в режиме реального времени. Существует четыре режима работы сервера OPC DA: синхронный, асинхронный, режим подписки и режим обновления данных[14]. Сервер OPC DA активно применяется в решениях промышленной и домашней автоматизации.

OPC UA (*OPC Unified Architecture*). Стек протоколов OPC UA является частью рассмотренного выше стандарта OPC. В отличие от протокола OPC DA OPC UA устанавливает взаимодействие между сервером и клиентом, не зависящее от программной и аппаратной части устройств и от типа сети. Основой архитектуры OPC UA является SOA (архитектура, ориентированная на услуги). Под услугой понимается набор функций, прописанных в конкретном ПО. Данный набор может передаваться от сервера клиенту и наоборот. Сообщения передаются в виде XML текста или бинарного файла. Кейсы применения OPC UA аналогичны OPC DA.

POP3 (*Post Office Protocol Version* 3). Протокол для приема сообщений электронной почты [12]. При просмотре почты с использованием протокола POP3

[14] Официальный сайт OPC. URL: https://opcfoundation.org/

все электронные письма сохраняются на устройство пользователя и автоматически удаляются с сервера. Все дальнейшие действия с письмами будут производиться именно на устройстве. Преимуществами использования данного протокола является возможность получения доступа к почтовому ящику даже при отсутствии Интернета.

Radius (*Remote Authentication Dial In User Service*). Протокол службы дистанционной аутентификации пользователей по коммутируемым линиям. Протокол RADIUS имеет клиент-серверную архитектуру и представляет собой службу без установления соединения. Сервер RADIUS может поддерживать множество методов аутентификации пользователя. Для проверки подлинности имени пользователя и пароля, предоставляемых серверу, могут использоваться протоколы PPP, PAP, CHAP, вход UNIX и другие механизмы аутентификации.

SIP (*Session Initiation Protocol*). Протокол сигнальной информации, предназначенный для управления сеансами мультимедийной связи. Наиболее распространенным вариантом применения протокола SIP является осуществление IP-телефонии (передача голоса, видео, мгновенных сообщений по IP сети). Стандарт протокола SIP описан в RFC 3261.

SMB/CIFS (*Server Message Block*). Протокол с клиент-серверной архитектурой, для удалённого доступа к сетевым ресурсам (файлам, принтерам и др.). Common Internet File System – первая версия протокола. Актуальная версия SMB 2.0 реализована Microsoft и используется в Microsoft Windows Network. Клиенты соединяются с сервером, используя протокол NetBIOS через TCP/IP, NetBEUI или IPX/SPX. После установки соединения, клиенты могут посылать SMB-команды серверу, который даёт им доступ чтения и записи к файлам, и позволяет выполнять весь перечень действий, которые можно выполнять с файловой системой. Однако в случае использования SMB эти действия совершаются через сеть. Протокол используется для получения доступа и мониторинг файлов и папок по технологии Microsoft Windows Network (SBM/CIFS).

SMI-S (*Storage Management Initiative Specification*). Стандарт управления дисковыми хранилищами. SMI-S является ANSI и ISO стандартом[15]. Актуальная версия SMI-S 1.5. Основная идея стандарта – унификация управления дисковыми хранилищами через веб-запросы. Более 800 различных аппаратных и 75 программных решений поддерживают данный стандарт. System Center Virtual Machine Manager 2012 позволяет подключить SMI-S хранилища, так чтобы администратор имел возможность из консоли VMM получать информацию о LUN, группах RAID, свободном месте на хранилище и так далее.

SMPP (*Short Message Peer-to-Peer*). Открытый стандарт в телекоммуникационной отрасли, разработанный специально для обеспечения гибкого интерфейса передачи коротких сообщений (ESME) между внешними устройствами или приложениями, маршрутизаторами и центрами сообщений (SMSC). Из-за своей универсальности и поддержкой SMS-протоколов без GSM, SMPP является наиболее широко используемым протоколом для короткого обмена сообщениями. SMPP реализован на Java в проекте jSMPP, на Python в проекте python-SMPP и на PHP в PHP-SMPP.

SMTP. Протокол, предназначенный только для отправки сообщений на почтовый сервер [13]. В рамках IoT протокол SMTP используется для отправ-

[15] Recommendation ISO/IEC 24775. Information technology – Storage management. 2011.

ки динамически сгенерированных сообщений по требованию, в ответ на события или в соответствии с расписанием.

SNMP (*Simple Network Management Protocol*). Протокол для управления устройствами в IP-сетях на основе архитектур TCP/UDP. Все управляемые коммутаторы и маршрутизаторы предоставляют информацию о состоянии своих портов/интерфейсов в реальном времени по протоколу SNMP. Эта информация доступна и используется встроенными средствами обработки данных (тревогами, отчетами, диаграммами и пр.). Протокол имеет три версии, из которых SNMPv1 и SNMPv2 считаются устаревшими, а SNMPv3 – полный интернет стандарт с максимальным уровнем готовности для RFC. На практике поддерживаются все три версии.

SOAP (*Simple Object Access Protocol*). Протокол технологий веб-служб, предназначенный для обмена произвольными XML сообщениями и вызова процедур. Последняя версия 1.2, по которой SOAP является расширением протокола XML-RPC[16]. SOAP может использоваться с любым протоколом прикладного уровня, однако его взаимодействие с каждым из этих протоколов имеет свои особенности, которые должны быть определены отдельно. Чаще всего SOAP используется с HTTP. Примером использования могут служить запросы на сервер интернет-магазина, содержащие id товара, для получения ответа с подробным описанием.

SQL (*Structured query language*). Структурированный язык запросов для работы с базами данных различных размеров. Основной задачей SQL является формирование и отправка динамически сгенерированных запросов SELECT/UPDATE/INSERT/DELETE и последующая загрузка результатов запросов в ядро системы.

SSH (*Secure Shell*). Протокол прикладного уровня, обеспечивающий безопасное соединение и передачу данных между устройствами. Это достигается благодаря используемому шифрованию трафика. Также возможно создание SSH туннеля между пользователями. В этом случае незашифрованные данные шифруются на одной стороне туннеля и расшифровываются на другой. В общем случае протокол SSH используется для удаленного доступа к устройствам, выполнения скриптов и приложений на удаленных компьютерах.

Syslog (*System log*). Стандарт и сетевой протокол отправки и регистрации сообщений о событиях (логов), происходящих в компьютерных сетях, основанных на протоколе IP. Механизм Syslog остается неизменным с незначительными вариациями: источники формируют простые текстовые сообщения небольшого размера (до 1024 байт) о происходящих в них событиях и передают их на обработку серверу «syslog server», используя один из сетевых протоколов UDP или TCP. Формирование сообщений о событиях и их передача происходит по определенным правилам, называемым протоколом Syslog. Как правило, сообщение отсылается в открытом виде, но возможно шифрование сообщений и отправка их по SSL/TLS. Источники сообщений и сервер Syslog могут располагаться на разных машинах, что позволяет организовать сбор и хранение сообщений от множества географически разнесенных разнородных источников в едином хранилище, что дает доступ сразу ко всем устройствам и компьютерам в сети.

[16] Официальный сайт SOAP. URL: https://www.w3.org/TR/soap/

Telnet (*Terminal network*). Протокол прикладного уровня, предназначенный для осуществления удаленного доступа к устройствам. Принцип работы аналогичен протоколу SSH с отличием лишь в том, что данные передаются в незашифрованном виде. Задачами протокола Telnet является выполнение скриптов и приложений на удаленных компьютерах, обработка данных, поступающих от устройства, и приведение их к стандартному виду. Telnet предоставляет полный доступ к функциональным возможностям устройства и может использоваться для удаленного управления робототехническими системами в IoT.

VMware SOAP API Программное обеспечение фирмы VMware предоставляющее интерфейс прикладного программирования (API) Lab Manager system[17]. Используя безопасный API, можно подключиться к серверу Lab Manager для выполнения или автоматизации различных операций. Lab Manager SOAP API использует XML-технологии, включая SOAP в качестве протокола связи, и описания веб-служб (WSDL) в качестве языка описания интерфейса. Используя предпочтительную среду разработки с поддержкой Web, вы можете создать клиент Web-сервиса приложений, использующих стандартные протоколы веб-сервисов, для программного выполнения задач: запрос информации о виртуальной машине и конфигурации, выполнение действия на машинах и конфигурациях, захват, проверка, клонирование, удаление и развертывание конфигураций, создание URL-адреса конфигурации LiveLink, который вы можете отправить по электронной почте другим членам команды.

WMI (*Windows Management Instrumentation*). Расширенная и адаптированная под Windows реализация стандарта WBEM, принятого многими компаниями, в основе которого лежит идея создания универсального интерфейса мониторинга и управления различными системами и компонентами распределенной информационной среды предприятия с использованием объектно-ориентированных идеологий и протоколов HTML и XML. Используется для мониторинга свойств объектов, выполнение WQL запросов и методов объектов, обработка событий.

Вывод

Сегодня в сети Интернет используются сотни протоколов. С процессом внедрения Интернета вещей в нашу жизнь, ростом устройств, подключенных к сети и расширением спектра возлагаемых на них задач, количество протоколов будет только расти. Каждый из рассмотренных протоколов широко применяется в сетях настоящего поколения, и у каждого из них есть, по крайней мере, по три реализации. В связи с этим возникает проблема выбора того или иного протокола. Для решения этой проблемы важно более детально изучить каждый из них, понимать принцип работы и возможные области применения.

Литература

1. Кучерявый А. Е. Интернет Вещей // Электросвязь. 2013. № 1. С. 21–24.
2. Кучерявый А. Е., Кучерявый Е. А., Прокопьев А. В. Самоорганизующиеся сети. СПб.: Любавич. 2011. 312 с.

[17] Руководство пользователя API-интерфейса VMware vCenter Lab Manager 4.0 для разработки приложений, использующих Lab Manager Web. URL: https://www.vmware.com/pdf/lm40_soap_api_guide.pdf

3. Гольдштейн Б. С., Кучерявый А. Е. Сети связи пост-NGN. СПб.: БХВ-Петербург. 2014. 160 с.
4. Shelby Z., Hartke K., Bormann C. RFC 7252 Constrained Application Protocol. 2014.
5. Droms R. RFC 2131 Dynamic Host Configuration Protocol. 1997.
6. Mockapetris P. RFC 1035 Domain names – Implementation and specification. 1987.
7. Postel J., Reynolds J. RFC 959 FILE TRANSFER PROTOCOL (FTP). 1985.
8. Fielding R., Gettys J., Mogul J., Frystyk H., Masinter L., Leach P., Berners-Lee T. RFC 2616 Hypertext Transfer Protocol – HTTP/1.1. 1999.
9. Postel J. RFC 792 INTERNET CONTROL MESSAGE PROTOCOL. 1981.
10. Crispin M. University of Washington RFC 3501 INTERNET MESSAGE ACCESS PROTOCOL. 2003.
11. Claise B. Cisco Systems RFC 3954 Cisco Systems NetFlow Services Export Version 9. 2004.
12. Myers J. RFC 1939 Post Office Protocol – Version 3. 1996.
13. Klensin J. RFC 5321 Simple Mail Transfer Protocol. 2008.

References

1. Koucheryavy A. Internet of Things // Elektrosvyaz'. 2013. No. 1. pp. 21–24.
2. Koucheryavy A., Koucheryavy E., Prokopiev A. Self-Organizing Networks. SPb.: Lyubavich. 2011. 312 p.
3. Goldstein B., Koucheryavy A. Post-NGN Communication Networks. SPb.: BHV-Peterburg. 2014. 160 p.
4. Shelby Z., Hartke K., Bormann C. RFC 7252 Constrained Application Protocol. 2014.
5. Droms R. RFC 2131 Dynamic Host Configuration Protocol. 1997.
6. Mockapetris P. RFC 1035 Domain names – Implementation and specification. 1987.
7. Postel J., Reynolds J. RFC 959 FILE TRANSFER PROTOCOL (FTP). 1985.
8. Fielding R., Gettys J., Mogul J., Frystyk H., Masinter L., Leach P., Berners-Lee T. RFC 2616 Hypertext Transfer Protocol – HTTP/1.1. 1999.
9. Postel J. RFC 792 INTERNET CONTROL MESSAGE PROTOCOL. 1981.
10. Crispin M. University of Washington RFC 3501 INTERNET MESSAGE ACCESS PROTOCOL. 2003.
11. Claise B. Cisco Systems RFC 3954 Cisco Systems NetFlow Services Export Version 9. 2004.
12. Myers J. RFC 1939 Post Office Protocol – Version 3. 1996.
13. Klensin J. RFC 5321 Simple Mail Transfer Protocol. 2008.

Москаленко Татьяна Андреевна – студентка, СПбГУТ, Санкт-Петербург, 193232, Российская Федерация, moskalenko.ta@spbgut.ru

Киричек Руслан Валентинович – кандидат технических наук, доцент, СПбГУТ, Санкт-Петербург, 193232, Российская Федерация, kirichek@sut.ru

Кучерявый Андрей Евгеньевич – доктор технических наук, профессор, СПбГУТ, Санкт-Петербург, 193232, Российская Федерация, akouch@mail.ru

Moskalenko Tatyana – Student, SPbSUT, St. Petersburg, 193232, Russian Federation, moskalenko.ta@spbgut.ru

Kirichek Ruslan – Candidate of Engineering Sciences, Associate Professor, SPbSUT, St. Petersburg, 193232, Russian Federation, kirichek@sut.ru

Koucheryavy Andrey – Doctor of Engineering Sciences, Full Professor, SPbSUT, St. Petersburg, 193232, Russian Federation, akouch@mail.ru