УДК 004.075

Ле Тхань Тунг Нгуен, магистрант, e-mail: juno249@gmail.com,
А. Г. Кравец, д-р техн. наук, проф., e-mail: agk@gde.ru,
Нгок Зыонг Буй, аспирант, e-mail: ramsetii@gmail.com,
Волгоградский государственный технический университет

Анализ средств и моделей взаимодействия между компонентами в системе управления корпоративной мобильностью

Коммуникация M2M (Machine-2-Machine) является новой технологией, позволяющей осуществлять "бесшовное" взаимодействие большого числа устройств в различных сетях без необходимости вмешательства человека в компоненты распределенных компьютерных систем, в частности, систем управления корпоративной мобильностью (УКМ). В настоящее время появляется ряд специализированных энергоэффективных протоколов M2M на прикладном уровне с низким энергопотреблением и потерями (Low power and Lossy Networks, LLNs), среди которых наиболее популярны HTTP, MQTT, XMPP, CoAP. В данной работе анализируются эти протоколы, они обеспечивают взаимодействие между сервером и устройствами в системе УКМ. Предлагаются формальные средства и модели обмена управляющими сообщениями на основе технологии асинхронной передачи. Проведены эксперименты, направленные на тестирование латентности и джиттера при приеме-отправке сообщений между мобильным устройством и сервером УКМ на базе этих протоколов.

Ключевые слова: *управление корпоративной мобильностью, ресурс-ориентированная архитектура, службы Push-уведомлений, архитектура REST, модель публикация/подписка, модель запрос/ответ, обмен сообщениями, асинхронная передача, MQTT, CoAP, XMPP, HTTP, брокер сообщений*

Введение

В настоящее время существует несколько компаний-разработчиков решений для управления корпоративной мобильностью (УКМ), которые предоставляют новые инструменты для взаимодействия с корпоративными информационными ресурсами с помощью смартфонов и планшетов вне зависимости от времени или места, что позволяет пользователям работать более продуктивно и обеспечивает оперативность, гибкость и контроль затрат для корпорации [1—3]. При этом существует проблема конфиденциальности данных и информационной безопасности компании, особенно в процессах передачи информации между сервером и мобильными устройствами, имеющими доступ к ресурсам предприятия [4—7].

В системе УКМ существуют следующие процессы [8]: регистрация мобильного устройства; централизованное управление мобильными устройствами, приложениями и контентом; защита мобильного устройства; распределение ресурсов корпорации на устройствах; мониторинг мобильных устройств. На рис. 1 (см. четвертую сторону обложки) показана контекстная среда работы системы УКМ с участием служб *Push*-уведомлений — важных критических компонентов мобильных устройств, требующих обновления контекста пользователя [9, 10].

Для описания задач и результатов операций — обмена данными функциональных сервисов на *backend*-сервере, обмена управляющими сообщениями между сервером и устройствами — используются модели взаимодействия и современные протоколы, которые гарантируют повышение эффективности коммуникации в системе УКМ по таким параметрам, как масштабируемость, работоспособность, снижение потерь пакетов данных при передаче, экономия расхода заряда батареи, пропускной способности, обеспечение безопасности.

Таким образом, целью данной работы является анализ существующих протоколов для управления взаимодействием между шлюзами, сетью Интернет и конечными приложениями/устройствами, а также исследование моделей обмена сообщениями в системе УКМ.

1. Коммуникационные протоколы взаимодействия в корпоративной сети

Для коммуникации между системными компонентами в корпоративной сети на прикладном уроне используются ряд специализированных энергоэффективных протоколов M2M (Machine-2-Machine) с низким энергопотреблением и потерями (Low power and Lossy Networks, LLNs) [11]. Среди них наиболее популярны HTTP (HyperText Transfer Protocol), MQTT (Message Queuing Telemetry Transport), XMPP (The Extensible Messaging and Presence Protocol), CoAP (Constrained Application Protocol) и др. Протокол CoAP представляет собой модель взаимодействия запрос/ответ (клиент/сервер) для манипулирования ресурсами и передачи данных. Протоколы MQTT, XMPP основаны на брокере для обмена сообщениями на основе архитектуры публикация/подписка. У каждого из них есть свои преимущества и недостатки, приведенные ниже.

Протокол HTTP. HTTP (HyperText Transfer Protocol — протокол передачи гипертекста) был разработан Тимом Бернерсом в марте 1991 г. [12]. В настоящее время HTTP используется в сети Интернет для передачи и получения различных данных веб-приложений от удаленного сервера к клиентскому компьютеру. Протокол HTTP основан на модели взаимодействия запрос/ответ для информационных систем клиент/сервер. Одной из основных проблем, связанных с этой архитектурой, является то, что существует некоторая задержка: во-первых, при отправке клиентом своих данных на сервер, т. е. клиентам после отправки необходимо ожидать результата получения и обработки запроса сервером; во-вторых, при опросе клиентом новых доступных данных. Это повышает нагрузку на сеть и требования к пропускной способности, кроме того, протокол HTTP затрачивает больше энергии батареи.

Новая крупная версия сетевого протокола HTTP/2 создана для решения этих проблем [13]. Протокол HTTP/2 является бинарным, т. е. по сравнению с предыдущим стандартом поток данных разделяется по фреймам с метаданными, имеющими фиксированную структуру и размер. В отличие от протокола HTTP, в протоколе HTTP/2 применяется эффективная технология сжатия данных HPACK, присваивающая имена заголовков и значения записей в таблицах и использующая только необходимый номер записи. Эта технология позволяет уменьшить размер заголовка протокола, в связи с чем снижается время обработки запроса.

Эффективность сетевых ресурсов повышается за счет применения мультиплексирования, которое позволяет клиентам использовать одно соединение TCP (Transmission Control Protocol) для множества одновременных потоков запросов от любой стороны — клиента или сервера. Каждый поток имеет приоритет, указанный клиентом самостоятельно на основе важности и зависимости одного потока от другого.

В протокол HTTP/2 внедрили Server Push, позволяющий серверу сразу же, не дожидаясь ответа клиента, добавить нужные файлы в кэш для быстрой выдачи. Более того, HTTP/2 поддерживает весь набор методов доступа протокола HTTP (GET, PUT, POST и т. п.), URI (Uniform Resource Identifier), при этом большее число заголовков. Для повышения безопасности соединения и защиты данных применяется TLS. Данные изменения позволили повысить производительность сервисов и мобильных устройств, оптимизировать трафик и уменьшить задержку доступа и нагрузку на сервер и клиента.

Протокол MQTT. Протокол MQTT (Message Queuing Telemetry Transport) [14] — простой, быстродействующий протокол, разработанный компанией IBM на основе модели взаимодействия публикации/подписки для Интернета вещей, он также может быть использован для надежной корпоративной системы передачи сообщений на мобильные устройства и между устройствами.

Характеристики протокола:

— он работает по принципу публикация/подписка, которая обеспечивает распределение сообщений и развязки приложений "один ко многим";

— транспорт обмена сообщениями не зависит от содержания данных;

— небольшие накладные расходы на транспортном уровне (фиксированный размер заголовка длиной 2 байт), протокол обмена также сведен к минимуму числа потоков данных для уменьшения сетевого трафика и контроля потери во время передачи сообщений;

— поддержка уровней качества сервиса (Quality of Service, QoS) [15] для доставки сообщений;

— для внедрения не требуется устанавливать большое количество программного обеспечения.

Протокол MQTT использует брокеры сообщений, через которые клиенты могут зарегистрировать подписки и публиковать свои данные. Брокеры обеспечивают маршрутизацию сообщений от публикаторов к подписчикам. Сообщения по определенной теме доставляются подписанным клиентам в зависимости от очереди маршрутизации.

Сообщения в протоколе MQTT доставляются с гарантией на основе QoS. Существуют три уровня QoS:

— *At most once* (QoS = 0, максимум однократная доставка) означает, что публикатор выполняет однократную отправку сообщения, но не контролирует доставку;

— *At least once* (QoS = 1, минимум однократная доставка): доставка сообщения контролируется, однако разрешается доставлять более одного раза;

— *Exactly once* (QoS = 2, однократная доставка): доставка сообщения гарантируется лишь один раз.

Для обеспечения безопасности передачи сообщений протокол MQTT использует TLS/SSL (Transport Layer Security/Secure Sockets Layer).

Протокол XMPP. XMPP (The Extensible Messaging and Presence Protocol) [16] — расширяемый протокол, стандартизирован IETF (Internet Engineering Task Force) для системы мгновенного обмена сообщениями и информацией о присутствии.

XMPP работает по TCP и обеспечивает модели взаимодействия публикация/подписка (асинхронная) и запрос/ответ (синхронная). Он предназначен для близких коммуникаций в реальном времени и таким образом поддерживает небольшой "след" сообщения (*message footprint*) и низкую задержку обмена сообщениями. Протокол XMPP обеспечивает простой метод адресации устройств вида адреса электронной почты (name@domain.com) для идентификации пользователей, по которому данные передаются в независимости от точек доступа. Схема адресации состоит из набора элементов, образующих доменный идентификатор и идентификатор ресурса.

XMPP обеспечивает безопасность связи с помощью криптографических протоколов TLS/SSL, встроенных в ядро спецификации протокола. Но он не использует сервис качества QoS, который делает его непрактичным для коммуникации, и только наследственные механизмы протокола TCP гарантируют надежность. В XMPP используется специализированный язык разметки XML (eXtensible Markup Language), который создает дополнительные накладные расходы на передачу данных вследствие ненужных тегов XML и требует дополнительных вычислительных мощностей для XML-парсинга, что увеличивает потребление энергии устройства.

Протокол CoAP. Протокол CoAP (Constrained Application Protocol) [17] был разработан IETF (Internet Engineering Task Force) с использованием расширенных методов HTTP для встроенных устройств. CoAP определяет заголовок сообщения, коды запроса/ответа, параметры сообщения, а также механизмы повторной передачи. В отличие от текстового протокола HTTP, CoAP транспортирует сообщения асинхронно в виде бинарных данных через UDP (User Datagram Protocol) и передает меньше данных в заголовках запроса, что уменьшает размер служебных данных, снижает требования к пропуской способности и повышает гибкость в моделях взаимодействия. Он создает альтернативу протокола HTTP для RESTful API (Representational State Transfer Application Programming Interface) на устройствах с ограниченными ресурсами и использует подмножество основных методов HTTP (GET, POST, PUT и DELETE) для представления ресурсов, ориентированных на взаимодействие в архитектуре клиент/сервер. Протокол CoAP также обеспечивает возможность многоадресной рассылки сообщений и обработку единого обмена сообщениями, выполняемого между конечными точками. Сообщения могут быть следующих видов:

- *Confirmable* — сообщения требуют подтверждения. Когда пакеты не теряются, каждое сообщение вызывает обратное сообщение типа *Acknowledgement* или Reset;
- *Non-confirmable* — сообщения не требуют подтверждения;
- *Acknowledgement* — подтверждает получение *Confirmable*-сообщения, но не указывает на успех или неудачу;
- *Reset* — указывает на то, что конкретное сообщение (*Confirmable* или *Non-confirmable*) было получено, но контекст, подлежащий обработке, отсутствует.

CoAP представляет собой модель взаимодействия запрос/ответ, который использует как синхронную, так и асинхронную технологии. Протокол CoAP поддерживает унифицированный идентификатор ресурса (URI) в запросах для идентификации интерфейса, экземпляра объекта или ресурса.

В протоколе CoAP использован DTLS (Datagram Transport Layer Security) для обеспечения высокого уровня безопасности связи и передачи данных по протоколу UDP. Режимы безопасности DTLS включают в себя как предварительный, так и открытый ключи для поддержки встраиваемых устройств.

Сравнительный анализ протоколов для обмена сообщениями. На основании проведенного анализа можно сделать следующие выводы:

- большинство рассмотренных протоколов используют транспорт TCP, что обеспечивает необходимый уровень надежности;
- протокол MQTT доставляет сообщения между устройствами по принципу взаимодействия публикация/подписка через центральный брокер сообщений с использованием QoS для проверки обмена;
- протокол XMPP обеспечивает быстрый асинхронный обмен в качестве метода адресации и поддерживает огромное число пользователей и разработчиков сети Интернет;
- протокол CoAP работает поверх UDP, что уменьшает объем служебных данных в передаваемом пакете для сетей с низким энергопотреблением;

Сравнительный анализ протоколов для обмена сообщениями в системе УКМ

Критерии	HTTP	XMPP	CoAP	MQTT
Модель	Запрос/ответ	Публикация/подписка, Запрос/ответ	Запрос/ответ	Публикация/подписка, Запрос/ответ
Качество сервиса	1 вид	1 вид	4 вида	3 вида
Тип распределения данных	Один к одному	Один ко многим	Один ко многим	Один ко многим
Обеспечение безопасности	SSL/TLS	SASL/TLS	DTLS	SSL/TLS
Формат сообщения	JSON/XML	XML	JSON/XML/CBOR	JSON/XML
Объем сообщения	Большой	Маленький	Маленький	Маленький
Низкие накладные расходы протокола	Нет	Да	Да	Да
Низкое энергопотребление	Нет	Да	Да	Да
Миллионы подключенных клиентов	Нет	Да	Да	Да
Разнообразие клиентских платформ	Да	Да	Да	Да
Push-сообщения	Да	Да	Да	Да

- протокол HTTP не оптимизирован для низкого энергопотребления или сведения к минимуму числа потоков данных [18].

Критерии и результаты сравнительного анализа протоколов для обмена сообщениями между мобильными устройствами и серверами представлены в таблице.

Протокол MQTT использует QoS, которое позволяет увеличить надежность работы системы и снизить вероятность потерь данных при передаче между компонентами системы, но повышает время задержек сообщений. Вместе с тем MQTT имеет короткую длину заголовка сообщения, которая влияет на снижение затрат энергии батарей и пропускной способности (*brandwidth*).

В рамках анализа указанных выше преимуществ протоколов и требований к решению определенных проблем организации обмена сообщениями между сервером и мобильными устройствами, для реализации процесса обмена сообщениями в системе УКМ выбран протокол MQTT.

2. Модель взаимодействия для управления мобильностью в корпоративной сети

В существующих MEAP (Mobile Enterprise Application Platform — платформа для корпоративных мобильных приложений) решениях [1] устройства и приложения имеют возможность обращаться к ресурсам вычислительных серверов, которые связываются с этими приложениями/устройствами через IP-адрес сервера и управляют доступом к ресурсам. Подход к построению коммуникационной модели с использованием рассмотренных эффективных протоколов для обеспечения взаимодействия и распространения данных является необходимостью [4, 19]. В системе УКМ агент на мобильном устройстве взаимодействует с различными клиент-сервисными компонентами.

Существуют следующие виды взаимодействия:

— синхронное (запрос/ответ) – клиент делает запрос на сервер и поток, делающий запрос может быть заблокирован во время ожидания своевременного ответа;

— асинхронное (уведомление, публикация/подписка) – клиент отправляет запрос на сервис, отвечающий асинхронно. Клиент не блокируется во время ожидания и работает с предположением о том, что ответ не может быть получен в течение некоторого времени.

Модель взаимодействия запрос/ответ. Эта модель является основным способом связи между компонентами в системе УКМ. При использовании модели запрос/ответ клиент отправляет запрос на сервис, в котором запрос обрабатывается и ответ отправляется обратно. У многих клиентов поток блокирует запросы во время ожидания ответа. Другие клиенты для управления событиями могут использовать асинхронный клиентский код, который может инкапсулироваться. Однако в отличие от использования обмена сообщениями клиент предполагает, что ответ поступит своевременно.

Существуют многочисленные архитектуры для реализации обмена сообщениями. На ресурс-ориентированной архитектуре RESTful [20] основан один из наиболее популярных протоколов.

В этой архитектуре уникальный URI-стандартный адрес используется для описания

ресурсных данных каждого устройства и/или каждой операции.

В качестве примера приведем принцип конкретной операции *{operationID}*, управляемой администратором, для конкретного устройства *{deviceID}* с помощью командной строки cURL (по типу HTTP-взаимодействия):

```
curl -X GET -H "Content-Type: application/json"
     -H "Authorization: Bearer <TOKEN>"
     -k -v baseIP/{deviceID}/{operationID}
```

URI-стандарт включает в себя части *host (полное имя домена в системе)* и *path (иерархический путь к каталогу)*. В представленной выше команде адрес *baseIP/{deviceID}/{operationID}* также разделяется на части host *baseIP* и part */{deviceID}/{operationID}*.

Масштабируемая модель системы УКМ. Операции в системе УКМ отличаются и поэтому реализуются различными сервисами с конструкцией использования *{deviceID}* и *{operationID}* для адресуемых устройств и операций, на основе следующих предположений:

— есть *m* типов сервисов, выполняющих *m* типов операций $\{operationID_j\}$;

— есть *n* устройств, имеющих *n* идентификаторов $\{deviceID_i\}$.

Моделирование операции управления мобильностью происходит в координатной плоскости *OYZ* масштабируемого куба AKF (The Scale Cube, авторы M. Abbott, T. Keeven и M. Fisher) [21] по правилу "трех" (концепция Gartner) [22] (рис. 2).

Адресные ресурсы операций на сервере описываются кортежем:

$$resource\_data = \{interaction\_type, deviceID, operationID\}$$

где *resource_data* — ресурсные данные операции (задачи и/или результаты), описываемые ресурсной моделью;

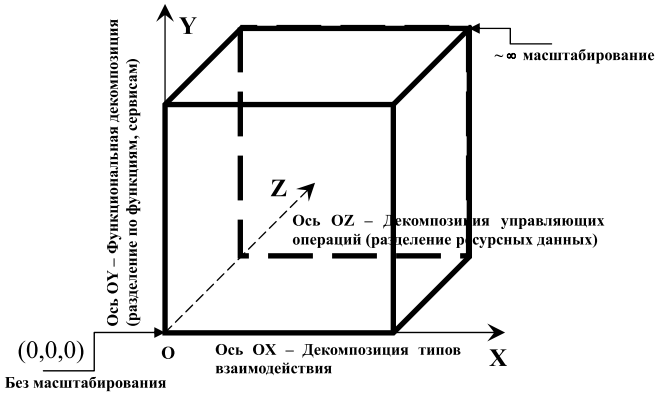

Рис. 2. Масштабируемая модель системы УКМ на основе шкалы куба AKF

interaction_type — тип взаимодействия при обработке операции;

deviceID — идентификатор устройства — параметр, использующийся для идентификации применения управляющих операций на конкретном устройстве;

operationID — идентификатор процедуры на агенте, обрабатывающей операцию.

Например, операция УКМ с использованием ресурс-ориентированной архитектуры, в которой управляющая операция выполняется методами HTTP в архитектурном стиле REST, описывается следующим образом: *GET/PUT/POST/DELETE baseIP/a590g62f8b5c/changelockcode*.

Таким образом, разработанная масштабируемая модель системы УКМ на основе декомпозиции управляющего пространства мобильности позволяет уточнить измерения куба AKF, специфичные для данного класса систем, а именно: тип взаимодействия, функция/сервис и операция управления.

Модель взаимодействия публикация/подписка. В отличие от традиционной модели взаимодействия запрос/ответ новая асинхронная технология, реализующая механизм доставки данных с гарантией в виде сообщения, использует модель публикация/подписка для коммуникации между компонентами [24]. В модели публикация/подписка подписчики обычно получают лишь часть общего числа опубликованных сообщений. Публикаторы отправляют сообщения на промежуточный брокер сообщений или сервис событий, которые действуют как посредники и обеспечивают доставку сообщений между системными компонентами [25]. На этих брокерах клиенты могут зарегистрировать подписки и публиковать свои данные. Брокеры обеспечивают маршрутизацию сообщений от публикаторов к подписчикам и выполняют определение очереди сообщений. Отправителям не нужно указывать получателя сообщений, и получатели могут не беспокоиться о том, кто присылает им сообщения. Сообщения доставляются подписанным клиентам в зависимости от очереди маршрутизации по определенной теме.

Архитектура публикация/подписка включает в себя масштабируемые свойства — временно-пространственная развязка (*time-space decoupling*) и синхронизационная развязка (*synchronization decoupling*), и предоставляет асинхронное распространение информации от публикаторов к подписчикам без определения конечных адресов получателей [23, 24]. Эти свойства позволяют системам публикация/подписка стать коммуникационным компонентом между системными сущностями в крупномасштабных динамических

Рис. 3. Концепция коммуникационной модели публикация/подписка

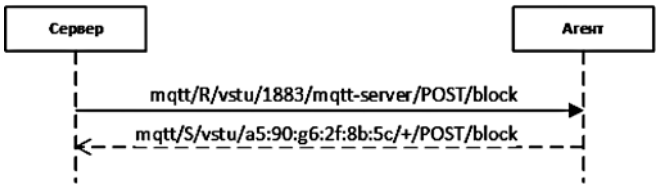

Рис. 4. Пример выполнения операции "изменение кода блокировки экрана устройства

сетях. На рис. 3 показана концепция коммуникационной модели публикация/подписка.

В рамках работы УКМ по модели публикация/подписка клиенты и сервер должны подписаться на тему для обеспечения доставки сообщений следующего вида:

mqtt/ + /<название-предприятия>/<ид-устройства>/<ид-приложения>/<ид-операции>/#,

где + тема для отправки запроса (Request): *mqtt/R/... /...,*
тема для отправки ответа (Response): *mqtt/S/.../...*

Пример на рис. 4 показывает выполнение операции "изменение кода блокировки экрана устройства" на предприятии с названием vstu по порту 1883. Запрос инициирует: *mqtt/R/vstu/1883/mqtt-server/POST/block*. Ответ определен следующим образом: *mqtt/S/vstu/1883/a590g62f8b5c/POST/block*.

3. Вычислительный эксперимент

Проведем сравнение изменения латентности (*latency*) и джиттера (*jitter*) при приеме-отправке сообщений для каждого из исследуемых протоколов для системы УКМ. Тестовый прототип исследования описан далее.

Испытания проводили с использованием сервера со следующей конфигурацией: процессор Intel Core i3 3220 3,30 ГГц, 2 Гбайт ОЗУ под операционной системой Linux Ubuntu 12.04.5 x64 с предустановленными программами WireShark, Tshark и TCPDump для анализа трафика, подключенного через проводную сеть к сети Интернет со скоростью загрузки/скачивания в ~ 10 Mbps (Мбит/с). Серверные части написаны на языке PHP и работают на веб-сервере стека решений XAMPP.

Мобильным устройством, рассмотренным в экспериментах, является LG Nexus 5 под операционной системой Android 5.0, 16 Гбайт внутренняя память, 2 a/b/g/n Гбайт ОЗУ и поддержка Wi-Fi 802.11. Устройство подключено через Wi-Fi к сети со скоростью загрузки 2 Mbps и со скоростью скачивания 4 Mbps.

На мобильное устройство было установлено приложение, реализующее сетевое взаимодействие с использованием различных протоколов (MQTT, XMPP и CoAP). Приложение написано на языке Java для Android-устройств на основе следующих библиотек для разработки: Paho (для MQTT) [26], Smack (для XMPP) [27], nCoAP (для CoAP) [28].

Для каждого протокола были проведены восемь измерений по увеличению числа сообщений с использованием показателей латентности и джиттера для оценки эффективности использования протокола в системе УКМ. Сообщения генерируются размером 254 байт в формате JSON и отправляются последовательно каждую миллисекунду в течение нескольких часов. Результаты проведенных испытаний показаны на рис. 5 (см. четвертую сторону обложки).

По сравнению с остальными протоколами, латентность при приеме-отправке сообщений протокола MQTT меньше, чем протоколов CoaAP/ XMPP в ~ 2,5 раза и джиттер: в ~ 2 раза с протоколом CoAP и в 5,7 раз с XMPP.

На графике показано, что при увеличении числа сообщений, латентность и джиттер возрастают.

Таким образом, проведенный вычислительный эксперимент показал хорошую эффективность использования протокола MQTT для связи между мобильными устройствами и сервером с небольшим объемом информации в сообщении.

Заключение

В целом, можно сделать заключение, что использование M2M протоколов, и прежде всего протокола MQTT, дает более эффективную коммуникацию в системе УКМ. Предложенный подход повышает эффективность за счет уменьшения расхода заряда батареи, меньших накладных расходов на доставку данных между мобильными устройствами и обеспечение безопасности обмена сообщениями на базе трехуровневой архитектуры. Кроме того, совместное использование протоколов M2M на основе модели взаимодействия публикация/подписка повышает гиб-

кость системы. В связи с этим в данный момент протокол MQTT для MEAP-решений позволяет устранить проблемы обеспечения работоспособности при увеличении числа устройств в системе и экономить ресурсы системы.

Исследование выполнено при финансовой поддержке РФФИ (проект № 15-07-06254).

Список литературы

1. **Kravets A. G., Bui N. D., Al-Ashval M. S.** Mobile Security Solution for Enterprise Network // Knowledge-Based Software Engineering. Proc. of 11th Joint Conf., JCKBSE 2014. Volgograd, Russia, September 17—20, 2014 / ed. by A. Kravets, M. Shcherbakov, M. Kultsova, Tadashi Iijima. Volgograd State Technical University [etc.]. Springer International Publishing, 2014. V. 466. P. 371—382. (Series: Communications in Computer and Information Science).

2. **Аль-Ашваль М. С., Кравец А. Г.** Анализ показателей качества мобильных корпоративных сетей // Современные проблемы науки и образования: электрон. науч. журнал / РАЕ. 2014. № 6. URL: http://www.science-education.ru/120-16129.

3. **Аль-Ашваль М. С., Кравец А. Г.** Система критериев и показателей качества мобильных корпоративных сетей // Инновации на основе информационных и коммуникационных технологий. Инфо 2014. Матер. XI междунар. науч.-практ. конф., г. Сочи, 1—10 окт. 2014 г. М.: Национальный исследовательский ун-т "Высшая школа экономики" [и др.], 2014. С. 501—504.

4. **Кравец А. Г., Аль-Ашваль М. С.** Mobile corporate networks security control // International Siberian Conference on Control and Communications (SIBCON—2016) (Russia, Moscow, May 12—14, 2016): Proceedings / Tomsk IEEE Chapter & Student Branch, National Research University "Higher School of Economics". Moscow, 2016. С. 1—6 (DOI: 10.1109/SIBCON.2016.7491811).

5. **Буй Нгок Зыонг, Кравец А. Г., Ле Тхань Тунг Нгуен.** Безопасная аутентификация в системе управления корпоративной мобильностью // Известия ВолгГТУ. Сер. Актуальные проблемы управления, вычислительной техники и информатики в технических системах. Волгоград, 2015. № 13 (177). С. 45—51.

6. **Буй Нгок Зыонг, Кравец А. Г., Ле Тхань Тунг Нгуен.** Информационная безопасность процесса регистрации Android-устройства в системе управления корпоративной мобильностью // Вестник компьютерных и информационных технологий. 2016. № 8. С. 44—51.

7. **Буй Нгок Зыонг, Кравец А. Г., Ле Тхань Тунг Нгуен.** Проблема проверки root-прав на Android-устройстве в системе управления мобильными приложениями // Информационные технологии в науке, образовании и управлении: матер. XLIV междунар. конф. и XIV междунар. конф. молодых ученых IT + S&E`15 (Гурзуф, 22 мая — 1 июня 2015 г.). / под ред. Е. Л. Глориозова; М.: ООО "Институт новых информационных технологий", 2015. С. 420—426.

8. **Буй Н. З.** Архитектура системы управления мобильностью в корпоративной сети // Современная наука: актуальные проблемы теории и практики. Сер. Естественные и технические науки. 2016. № 09-10. С. 30—35.

9. **Push** Technology: A Key Ingredient of Application Interactivity. URL: http://www.seven.com/downloads/pdf/SEVEN_Push_Whitepaper.pdf (дата обращения: 20.01.2017).

10. **Буй Н. З., Нгуен Л. Т. Т., Кравец А. Г.** Разработка службы Push-уведомлений в системе управления корпоративной мобильностью с использованием Google Cloud Messaging // Юность и Знания — Гарантия Успеха — 2015: сб. науч. тр. 2-й междунар. науч.-практ. конф. (1—2 окт. 2015 г.). В 2 т. Т. 2. Курск: Юго-Западный гос. ун-т, ЗАО "Университетская книга" [и др.]. 2015. С. 28—30.

11. **Jung M., Kim J. H., Wi H. W., Kim S.** Things-to-cloud communication: technology overview and design considerations // 2015 5th International Conference on the Internet of Things (IoT), October 26—28, 2015 in Seoul, S. Korea.

12. **HTTP.** URL: https://en.wikipedia.org/wiki/Quality_of_service (дата обращения: 26.1.17).

13. **Разъяснения HTTP2.** URL: http://habrahabr.ru/post/221427 (дата обращения: 05.02.2017).

14. **IBM. MQTT V3.1.** Protocol Specification // International Business Machines Corporation Eurotech. 2015.

15. **Quality** of service. URL: https://en.wikipedia.org/wiki/Quality_of_service (дата обращения: 26.01.17).

16. **ITU-T. Extensible Messaging and Presence Protocol (XMPP):** Core // RFC-3920. 2004.

17. **Shelby Z., Hartke K., Bormann C.** Constrained Application Protocol. URL: https://datatracker.ietf.org/doc/rfc7252 (дата обращения 18.01.2017).

18. **Stephen N.** Power Profiling: HTTPS Long Polling vs. MQTT with SSL, on Android. URL: http://stephendnicholas.com/archives/1217 (дата обращения: 28.01.17).

19. **Rodríguez-Domínguez C., Benghazi K., Noguera M., Garrido J. L., Rodríguez M. L., Ruiz-López T.** A Communication Model to Integrate the Request-Response and the Publish-Subscribe Paradigms into Ubiquitous Systems // E. T. S. I. I. T. University of Granada, C/Periodista Daniel Saucedo Aranda S/N, 18071 Granada, Spain, 2012.

20. **Jansen G.** Thoughts on RESTful API design / Geert Jansen [Электронный ресурс]. URL: https://media.readthedocs.org/pdf/restful-api-design/latest/restful-api-design.pdf (дата обращения 18.01.2017).

21. **Abbott M. L., Fisher M. T.** The art of scalability: Scalable Web Architecture, Processes, and Organizations for the Modern Enterprise. 2 ed., Pearsonn Education, Inc., 2015.

22. **Платформа** для корпоративных мобильных приложений. URL: https://ru.wikipedia.org/wiki/ (дата обращения: 20.1.2017).

23. **Bui N. D., Кравец А. Г., Nguyen T. A., Nguyen L. T. T.** Tracking events in mobile device management system // IISA 2015 — 6th International Conference on Information, Intelligence, Systems and Applications (Corfu, Greece, 6 July 2015 — 8 July 2015): Conference Proceeding / Ionian University, Institute of Electrical and Electronics Engineers (IEEE) [Piscataway, USA]. 2015. 6 p. DOI: 10.1109/IISA.2015.7388127.

24. **Pongthawornkamol T., Nahrstedt K., Wang G.** The analysis of publish/subscribe systems over mobile wireless ad hoc networks // Proc. of ACM MobiQuitous. 07 Aug 2007, pp. 1—8.

25. **Message Broker**. URL: https://en.wikipedia.org/wiki/Message_broker (дата обращения: 26.08.16).

26. **Eclipse** Paho MQTT. URL: https://eclipse.org/paho/ (дата обращения 18.01.2017).

27. **Smack 4.1** Readme and Upgrade Guide. URL: https://github.com/igniterealtime/Smack/wiki/Smack-4.1-Readme-and-Upgrade-Guide (дата обращения 18.01.2017).

28. **CoAP** Client for Android based on nCoAP. URL: https://github.com/okleine/spitfirefox (дата обращения 18.01.2017).

Le Thanh Tung Nguyen, Post-bachelor, e-mail: juno249@gmail.com,
A. G. Kravets, Professor, e-mail: agk@.ru, **Ngoc Duong Bui**, Postgraduate, e-mail: ramsetii@gmail.com,
Volgograd State Technical University, Volgograd, 400131, Russia

Analysis of Interaction Method and Models Between Components of Enterprise Mobility Management System

M2M (Machine-2-Machine) communication is the new technology that allows seamless interaction of a wide various devices over different networks without the need of human intervention between the components in distributed computing systems, especially in the enterprise mobility management system (EMM). Nowadays, there are a large of specialized energy-efficient protocols (M2M protocols) in Application layer with Low power and Losssy Networks (LLNs). Most popular of them is HTTP, MQTT, XMPP, CoAP and others. This article analyzes these protocols that enable communication between the server and devices in EMM system and also new formal messaging model based on asynchronous transfer technology is present. Experiments aimed at testing the latency and jitter receiving-sending messages between mobile device and EMM server based on these protocols.

Keywords: *enterprise mobility management, resource-oriented architecture, push notification service, REST architecture, publish/subscribe model, request/response model, messaging model, asynchronous transfer, MQTT, CoAP, XMPP, HTTP, message broker*

References

1. **Kravets A. G., Bui N. D., Al-Ashval M. S.** Mobile Security Solution for Enterprise Network, *Knowledge-Based Software Engineering: Proc. of 11th Joint Conf., JCKBSE 2014,* Volgograd, Russia, September 17—20, 2014, ed. by A. Kravets, M. Shcherbakov, M. Kultsova, Tadashi Iijima; Volgograd State Technical University [etc.]. Springer International Publishing, 2014, vol. 466, pp. 371—382. (Series: Communications in Computer and Information Science).

2. **Al'-Ashval' M. S., Kravets A. G.** Analiz pokazatelej kachestva mobil'nyh korporativnyh setej, *Sovremennye problemy nauki i obrazovanija*, RAE, 2014, no. 6, available at: http://www.science-education.ru/120-16129 (in Russian).

3. **Al'-Ashval' M. S., Kravets A. G.** Sistema kriteriev i pokazatelej kachestva mobil'nyh korporativnyh setej, Innovacii na osnove informacionnyh i kommunikacionnyh tehnologij. Info 2014: mater. XI mezhdunar. nauch.-prakt. konf. (g. Sochi, 1—10 okt. 2014 g.), Moscow, Nacional'nyj issledovatel'skij un-t Vysshaja shkola jekonomiki, 2014, pp. 501—504 (in Russian).

4. **Kravec A. G., Al'-Ashval' M. S.** Mobile corporate networks security control, International Siberian Conference on Control and Communications (SIBCON—2016), Russia, Moscow, May 12—14, 2016: Proceedings, Moscow, 2016, pp. 1—6. DOI: 10.1109/SIBCON.2016.7491811 (in Russian).

5. **Buj Ngok Zyong, Kravets A. G., Le Than' Tung Nguen.** Bezopasnaja autentifikacija v sisteme upravlenija korporativnoj mobil'nost'ju, *Izvestija VolgGTU. Ser. Aktual'nye problemy upravlenija, vychislitel'noj tehniki i informatiki v tehnicheskih sistemah*, 2015, no. 13 (177), pp. 45—51 (in Russian).

6. **Buj Ngok Zyong, Kravets A. G., Le Than' Tung Nguen.** Informacionnaja bezopasnost' processa registracii Android-ustrojstva v sisteme upravlenija korporativnoj mobil'nost'ju, *Vestnik komp'juternyh i informacionnyh tehnologij*, 2016, no. 8, pp. 44—51.

7. **Buj Ngok Zyong, Le Than' Tung Nguen, Kravets A. G.** Problema proverki root-prav na Android-ustrojstve v sisteme upravlenija mobil'nymi prilozhenijami, *Informacionnye tehnologii v nauke, obrazovanii i upravlenii: mater. XLIV mezhdunar. konf. i XIV mezhdunar. konf. molodyh uchjonyh* IT + S&E`15, Gurzuf, 22 maja — 1 ijunja 2015 g., vesennjaja sessija, ed. E. L. Gloriozov; Institut novyh informacionnyh tehnologij, Moscow, 2015, pp. 420—426 (in Russian).

8. **Bui N. D.** Architecture of Mobility Management system in Corporation Nextwork, *Sovremennaja nauka: aktual'nye problemy teorii i praktiki*, *Ser. Estestvennye i tehnicheskie nauki*, 2016, no. 09—10, pp. 30—35 (in Russian).

9. **Push** Technology: A Key Ingredient of Application Interactivity. Available at: http://www.seven.com/downloads/pdf/SEVEN_Push_Whitepaper.pdf (accessed: 20.1.2017).

10. **Bui N. D., Nguyen L. T. T., Kravets A. G.** Development of Push notification service in Enterprise Mobility Management system using Google Cloud Messaging. *Junost' i Znanija — Garantija Uspeha — 2015: sb. nauch. tr. 2-j mezhdunar. nauch.-prakt. konf. 1—2 okt. 2015 g.* V 2 t. T. 2. Jugo-Zapadnyj gos. un-t, ZAO "Universitetskaja kniga" [i dr.], Kursk, 2015, pp. 28—30 (in Russian).

11. **Jung M., Kim J. H., Wi H. W., Kim S.** Things-to-cloud communication: technology overview and design considerations, *2015 5th International Conference on the Internet of Things (IoT), October 26—28, 2015, Seoul, S. Korea.*

12. **HTTP.** Available at: https://en.wikipedia.org/wiki/Quality_of_service (accessed: 26.01.17).

13. **HTTP2.** Available at: http://habrahabr.ru/post/221427 (accessed: 05.02.2017).

14. **IBM. MQTT V3.1.** *Protocol Specification*, International Business Machines Corporation Eurotech. 2015.

15. **Quality** of service. Available at: https://en.wikipedia.org/wiki/Quality_of_service, accessed: 26.01.17.

16. **ITU-T.** Extensible Messaging and Presence Protocol (XMPP): Core, RFC-3920. 2004.

17. **Shelby Z., Hartke K., Bormann C.** Constrained Application Protocol. Available at: https://datatracker.ietf.org/doc/rfc7252 (accessed: 18.01.2017).

18. **Stephen N.** Power Profiling: HTTPS Long Polling vs. MQTT with SSL, on Android. Available at: http://stephendnicholas.com/archives/1217 (accessed: 28.01.17).

19. **Rodríguez-Domínguez C., Benghazi K., Noguera M. et al.** *A Communication Model to Integrate the Request-Response and the*

Publish-Subscribe Paradigms into Ubiquitous Systems, E. T. S. I. I. T. University of Granada, C/Periodista Daniel Saucedo Aranda S/N, 18071 Granada, Spain, 2012.

20. **Jansen G.** Thoughts on RESTful API design. Available at: https://media.readthedocs.org/pdf/restful-api-design/latest/restful-api-design.pdf (accessed: 18.01.2017).

21. **Abbott M. L., Fisher M. T.** *The art of scalability: Scalable Web Architecture, Processes, and Organizations for the Modern Enterprise*. 2 ed., Pearsonn Education, Inc., 2015.

22. **Platforma** dlya korporativnykx mobil'nykx prilozhenij. Available at: https://ru.wikipedia.org/wiki/Platforma_dlya_korporativnyx_mobil'nyx_prilozhenij (accessed: 20.01.2017).

23. **Bui N. D., Kravets A. G., Nguyen T. A., Nguyen L. T. T.** Tracking events in mobile device management system, *IISA 2015 — 6th International Conference on Information, Intelligence, Systems and Applications, Corfu, Greece, 6 July 2015 — 8 July 2015: Conference Proceeding*, Ionian University, Institute of Electrical and Electronics Engineers (IEEE), Piscataway, USA. 2015. 6 p. DOI: 10.1109/IISA.2015.7388127.

24. **Pongthawornkamol T., Nahrstedt K., Wang G.** The analysis of publish/subscribe systems over mobile wireless ad hoc networks, *Proc. of ACM MobiQuitous*, *07 Aug 2007*, pp. 1—8.

25. **Message** Broker. Available at: https://en.wikipedia.org/wiki/Message_broker (accessed: 26.08.16).

26. **Eclipse** Paho MQTT. Available at: https://eclipse.org/paho/ (accessed: 18.01.2017).

27. **Smack 4.1** Readme and Upgrade Guide. Available at: https://github.com/igniterealtime/Smack/wiki/Smack-4.1-Readme-and-Upgrade-Guide (accessed: 18.01.2017).

28. **CoAP** Client for Android based on nCoAP. Available at: https://github.com/okleine/spitfirefox (accessed: 18.01.2017).

Адрес редакции:

107076, Москва, Стромынский пер., 4

Телефон редакции журнала **(499) 269-5510**
E-mail: it@novtex.ru

Технический редактор *Е. В. Конова*.
Корректор *Е. В. Комиссарова*.
Сдано в набор 09.11.2017. Подписано в печать 25.12.2017. Формат 60×88 1/8. Бумага офсетная.
Усл. печ. л. 8,86. Заказ IT118. Цена договорная.

Журнал зарегистрирован в Министерстве Российской Федерации по делам печати, телерадиовещания и средств массовых коммуникаций.
Свидетельство о регистрации ПИ № 77-15565 от 02 июня 2003 г.

Оригинал-макет ООО "Адвансед солюшнз". Отпечатано в ООО "Адвансед солюшнз".
119071, г. Москва, Ленинский пр-т, д. 19, стр. 1.