



ОРГАНИЗАЦИЯ СЕРВИСОВ УМНОГО ГОРОДА НА ОСНОВЕ МИКРОСЕРВИСНОЙ АРХИТЕКТУРЫ

А. С. А. Мутханна^{1*}, Д. Н. Тараскин¹

¹Санкт-Петербургский государственный университет телекоммуникаций
им. проф. М. А. Бонч-Бруевича,
Санкт-Петербург, 193232, Российская Федерация

*Адрес для переписки: ammarexpress@gmail.com

Аннотация—Одним из актуальных направлений в области исследования сетей связи пятого поколения является обслуживание, интеграция, быстрое развертывание и предоставление различных сервисов. **Предмет исследования.** Предметом исследований является анализ методов организации сервисов умного города на базе сетей пятого поколения. **Метод.** Сбор, анализ и сравнение опубликованной информации по методам организации сервисов умного города. **Основные результаты.** Основным результатом данной работы является выбор наиболее подходящего метода организации сервисов в сетях пятого поколения. **Практическая значимость.** Практическая значимость работы состоит в возможности применения основных результатов для построения и реализации сервисной архитектуры.

Ключевые слова—архитектура, сервис, API, сети связи пятого поколения, FaaS, BaaS, SOA.

Информация о статье

УДК 004.732.

Язык статьи – русский.

Поступила в редакцию 11.11.2021, принята к печати 10.12.2021.

Ссылка для цитирования: Мутханна А. С. А., Тараскин Д. Н. Организация сервисов умного города на основе микро-сервисной архитектуры // Информационные технологии и телекоммуникации. 2021. Том 9. № 3. С. 48–55. DOI 10.31854/2307-1303-2021-9-3-48-55.



THE METHOD OF ORGANIZING "SMART CITY" SERVICES BASED ON THE 5G/IMT-2020 INFRASTRUCTURE

A. S. Muthanna^{1*}, D. Taraskin¹

¹The Bonch-Bruевич Saint-Petersburg State University of Telecommunications,
St. Petersburg, 193232, Russian Federation

*Corresponding author: ammarexpress@gmail.com

Abstract—One of the current directions in the field of research of fifth-generation communication networks is maintenance, integration, rapid deployment and provision of various services. **The subject of research** is the analysis of methods for organizing smart city services based on fifth-generation networks. **The research method** is the collection, analysis and comparison of published information on the methods of organizing smart city services. **Core results** of this work is the choice of the most appropriate method for organizing services in fifth-generation networks. **Practical relevance** of the work consists in the possibility of applying the main results for the construction and implementation of a service architecture.

Keywords—Architecture, service, API, 5G-communication networks, FaaS, BaaS, SOA.

Article info

Article in Russian.

Received 11.11.2021, accepted 10.12.2021.

For citation: Muthanna A. S., Taraskin D.: The Method of Organizing "Smart City" Services Based on the 5G/IMT-2020 Infrastructure // Telecom IT. 2021. Vol. 9. Iss. 3. pp. 48–55 (in Russian). DOI 10.31854/2307-1303-2021-9-3-48-55.



Введение

В результате развития информационных технологий помимо трафика, генерируемого людьми, появился также трафик, генерируемый устройствами – трафик Интернета вещей. Эту технологию было необходимо привести к одной концепции и провести стандартизацию. 5G/IMT-2020 является новейшим стандартом в области сетевой архитектуры. Данная технология очень тесно связано с «Интернетом вещей» из-за одного из сценариев её работы – mMTC (*Massive Machine Type Communications*).

Интернет вещей имеет несколько основных приложений. Одним из них является «Умный город». Умный город – интеграция инфокоммуникационных технологий, в том числе Интернета Вещей, для управления городскими системами [2]. Это приложение Интернета вещей, которое направлено на улучшение бытовых, коммуникационных, транспортных и других условий для горожан. Однако стоит отметить, что многие сервисы данного приложения можно реализовать и в сельской местности. Существуют некоторые проблемы по организации сети «Умного города», одну из которых предлагается решить в данной работе.

Архитектура разрабатываемого сервиса

Для организации архитектуры мониторинга системы «Умный город» была выбрана микросервисная архитектура, она наиболее подходит для развертки этой системы и её мониторинга. Основной причиной выбора данного способа развертки и разработки стала необходимость постоянного масштабирования по горизонтальной и вертикальной оси (рис. 1, см. ниже).

Центральным устройством в сервисе станет MQTT брокер Eclipse Mosquitto. Для развёртки сервиса с брокером был взят сервер. На этом же сервере расположится база данных, бэк-энд, сервис-воркер и фронтэнд для мониторинга нагрузки брокера. Такое архитектурное решение в расположении сервисов было принято в связи с необходимостью получения минимальной круговой задержки между брокером и сервисом статистики.

Сервис IoTDM

Необходимый сервис для менеджмента и управления устройствами IoT. Прежде всего IoTDM использовался для создания модели «умных» устройств и их управления с последующей генерацией данных на умных устройствах средствами API. Руководствуясь документацией oneM2M, имеется возможность разработать собственное дерево ресурсов, которое удобно применять на архитектуре микро и макроуровня¹.

Сервис IoTDM может являться также и брокером «умных» устройств, предоставлять мониторинг и возможность настройки политики доступа до ресурсов Ин-

¹ OpenFlow Switch Specification. URL: <http://archive.openflow.org/documents/openflow-spec-v1.0.0.pdf>



тернета Вещей [3]. Однако при использовании микросервисной архитектуры, появилась необходимость разделять «умные устройства» и системы мониторинга и аналитики.

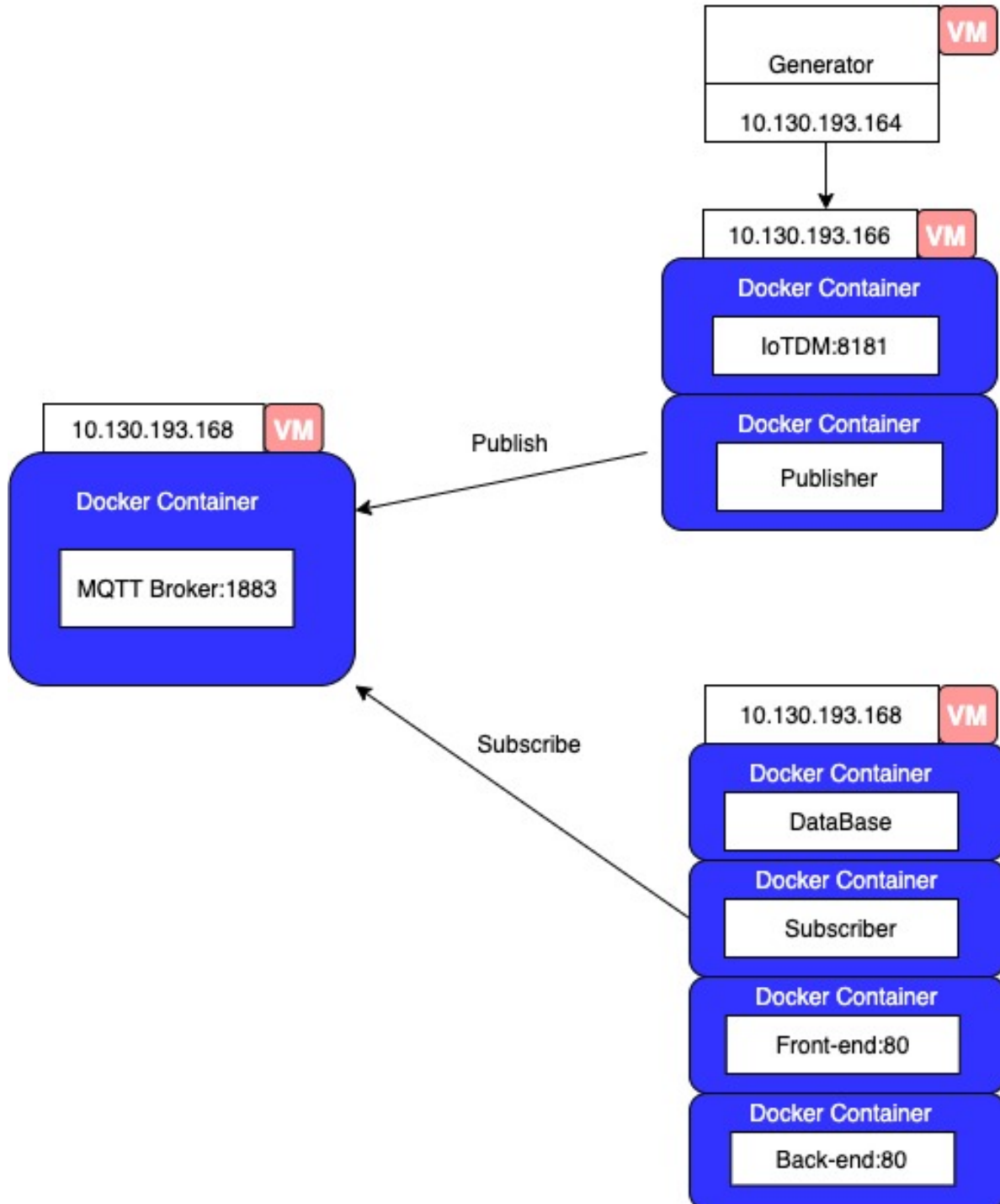


Рис. 1. Общая архитектура микросервисов

При использовании брокера MQTT «умные» устройства могут быть подключены напрямую к брокеру, однако при большом количестве устройств разного типа возникают проблемы с разверткой, так как необходимо подключать каждое



устройство вручную, настраивать публикацию данных с устройства в «топик» брокера.

В случае использования сервиса IoTDM в первую очередь можно группировать «умные» устройства. Группировать устройства можно как по их типу – например группировать температурные, сенсорные и другие датчики вместе². При такой структуре появляется возможность публиковать данные сразу с группы устройств. IoTDM имеет возможность хранить данные с устройств, что важно при временном отсутствии связи между устройством и MQTT брокером, после восстановления соединения публикация данных пройдет в нужный канал.

Как показано на рис. 1, сервис IoTDM был развернут внутри Docker-контейнера. Также был выделен порт 8181, который используется для API запросов до IoTDM сервиса, необходимых для манипуляции, сбора данных и других функций.

Помимо контейнера с IoTDM на данном сервере также расположено небольшое приложение для публикации данных из IoTDM в топик MQTT брокера. Данное приложение при помощи API IoTDM забирает данные из хранилища IoTDM и публикует их в топике.

Сервис статистики

Помимо необходимых обычно приложений для мониторинга – бэкэнд сервиса, фронтэнд сервиса и базы данных для мониторинга, на сервере статистики было необходимо развернуть вспомогательный сервис для сбора системных данных о брокере. Задача данного приложения – подписаться на системную «ленту» брокера, забирать данные о нагрузке и сохранить их в базе данных [1].

Общая архитектура разрабатываемого ПО

При разработке модели для проведения нагрузочного тестирования необходимо учесть возможность уменьшения нагрузки путем реализации дополнительных сервисов. В качестве архитектурного решения, которое уменьшит нагрузку на MQTT брокер возможно реализовать второй MQTT брокер с мостовым соединением к первому. Такое решение уменьшает количество публикаций, сделанные «умными устройствами» в топик брокера [4].

Помимо всего прочего можно разграничивать доступ пользователей к получаемой информации на машинном уровне – давая подписаться только на второстепенные брокеры и публикуя в них информацию нужную конкретной группе пользователей, что значительно увеличит безопасность сервиса.

Результаты тестирования

На рис. 2 представлен график зависимости количества переданных или полученных пакетов от публикаций – это максимально возможное количество байт при данной конфигурации оборудования (модель без распределения нагрузки)

² OpenFlow Switch Specification. URL: <http://archive.openflow.org/documents/openflow-spec-v1.0.0.pdf>



между микросервисами). Как видно по графику, максимальное количество байт составило – 1000 байт/мин.

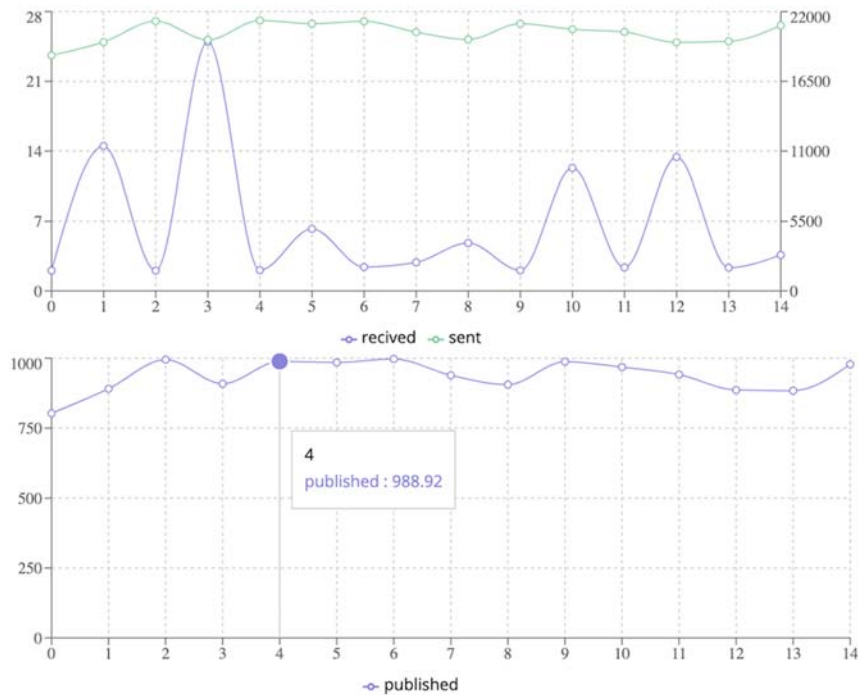


Рис. 2. Максимальное значение публикаций в байтах без распределения нагрузки

На рис. 3 (см. ниже) изображен тот же самый график, но на этот раз рассматривается модель с распределением нагрузки между микросервисами.

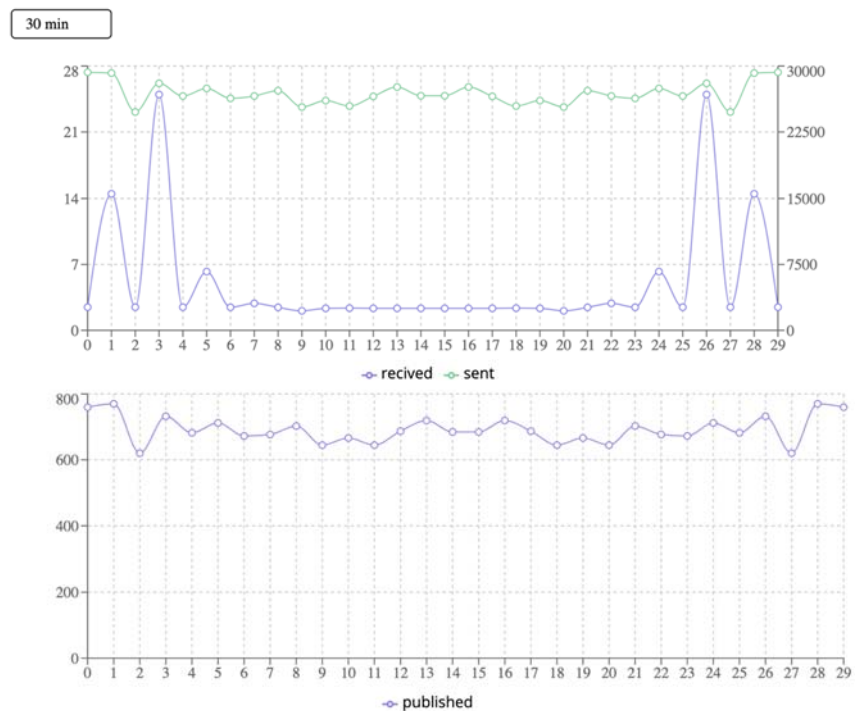


Рис. 3. Максимальная нагрузка при использовании второго брокера в режиме моста (модель с распределением нагрузки)



Максимальное количество опубликованной информации на первом брокере, при переносе публикаций данных с датчиков на второй составила ~ 800 байт/мин. Также стоит отметить, что нагрузка не была распределена одинаково, связано это с тем, что задача была показать возможность уменьшения количества публикаций. Второй брокер не передавал информацию первому из топиков данных датчиков, что никак не повлияло на количество полученных байт на первом брокере.

Заключение

Выбранная архитектура в ходе развертки, разработки и тестирования оказалась весьма удачной. Связанно это с тем, что развертка и разработка каждого микросервиса происходит независимо от других сервисов. При масштабировании также не возникло никаких проблем.

Несмотря на правильный выбор архитектуры, выбор сервисов и некоторых библиотек нельзя назвать удачным.

1. IoTDM включает в себя все необходимое, однако общение между датчиками и брокером происходит по протоколу HTTP, из-за «тяжести» протокола в сравнении с MQTT. Также сервис достаточно тяжел и имеет множество функций, которые не нужны при такой архитектуре.

2. MQTT брокер Eclipse Mosquitto – хорошее решение, однако требует большое количество работы и разработки для внедрения в сеть «Умного города».

3. Postgresql – не совсем удачный выбор базы данных, как центральной для сервиса статистики, база становится тяжелой и со временем требует большого количества памяти.

Исследование выполнено в рамках выполнения ПНИ по государственному заданию СПбГУТ на 2021 год.

Литература

1. Бородин А. С., Волков А. Н., Мутханна А. С. А., Кучерявый А. Е. Искусственный интеллект в сетях связи пятого и последующих поколений // Электросвязь. 2021. № 1. С. 17–22.
2. Мутханна А. С. Интеллектуальная распределенная архитектура сети связи для поддержки беспилотных автомобилей // Электросвязь. 2020. № 7. С. 29–34.
3. Adel Nadjaran Toosi, Redowan Mahmud, Qinghua Chi, Rajkumar Buyya. Management and Orchestration of Network Slices in 5G, Fog, Edge and Clouds. pp. 5–16.
4. Rajkumar Buyya, Satish Narayana Srirama. Fog and Edge Computing: Principles and Paradigms. 2019. 471 p.

References

1. Borodin A. S., Volkov A. N., Mutkhanna A. S. A., Kucheryavyi A. E. Artificial intelligence for telecommunication networks // Electrosvyaz. 2021. No 1. pp. 17–22.
2. Mutkhanna A. S. Distributed intelligent communication network architecture for unmanned vehicles // Electrosvyaz. 2020. No 7. pp. 29–34.
3. Adel Nadjaran Toosi, Redowan Mahmud, Qinghua Chi, Rajkumar Buyya. Management and Orchestration of Network Slices in 5G, Fog, Edge and Clouds. pp. 5–16.
4. Rajkumar Buyya, Satish Narayana Srirama. Fog and Edge Computing: Principles and Paradigms. 2019. 471 p.

**Мутханна Аммар Салех Али**

кандидат технических наук, доцент кафедры Санкт-Петербургского государственного университета телекоммуникаций им. проф. М. А. Бонч-Бруевича,

ammarexpress@gmail.com

Тараскин Дмитрий Николаевич

магистрант Санкт-Петербургского государственного университета телекоммуникаций им. проф.

М. А. Бонч-Бруевича, taraskindima@mail.ru

Muthanna Ammar S.

Candidate of Engineering Sciences, Associate Professor at the Department, The Bonch-Bruevich Saint-Petersburg State University of Telecommunications,

ammarexpress@gmail.com

Taraskin Dmitry N.

Undergraduate, The Bonch-Bruevich Saint-Petersburg State University

of Telecommunications, taraskindima@mail.ru