

# ИССЛЕДОВАНИЕ ПРОИЗВОДИТЕЛЬНОСТИ ПРОГРАММНЫХ ИНСТРУМЕНТОВ МЕЖСЕТЕВОГО ВЗАИМОДЕЙСТВИЯ ДЛЯ СЕМАНТИЧЕСКИХ ШЛЮЗОВ ИНТЕРНЕТА ВЕЩЕЙ

Л. А. Власенко<sup>1</sup>, В. Д. Фам<sup>1</sup>,  
В. А. Кулик<sup>1</sup>, Р. В. Киричек<sup>1\*</sup>

<sup>1</sup> СПбГУТ, Санкт-Петербург, 193232, Российская Федерация

\* Адрес для переписки: kirichek@sut.ru

## Аннотация

**Предмет исследования.** В данной статье исследуется производительность библиотек для разработки клиент-серверных приложений семантических шлюзов Интернет-Вещей на базе типового одноплатного компьютера Raspberry Pi rev. 3 Model B. В качестве серверных приложений для протоколов MQTT и XMPP были использованы готовые приложения mosquitto и Ejabberd соответственно. Для таких протоколов как CoAP и HTTP/2 серверные приложения были разработаны самостоятельно. В ходе выполнения работ для данной статьи был исследован следующий перечень библиотек: Paho-mqtt, hbmqtt, pyamtk для протокола MQTT, aiosoap, txThings для протокола CoAP, xmpppy, sleekxmpp, Twisted-xmpp для протокола XMPP, hyper, hyper, Hyper-h2 для протокола HTTP/2. Каждое из разработанных приложений было написано на своей версии языка программирования высокого уровня Python, в зависимости от поддержки библиотекой той или иной версии. **Метод.** Основным методом исследования производительности библиотек является натурный эксперимент. Для каждой из библиотек было проведено по двадцать испытаний, данные аппроксимированы, результаты выведены в таблицу. **Основные результаты.** В статье приведены показатели задержки, загрузки центрального процессора и оперативной памяти семантического шлюза при преобразовании протоколов для каждой из исследуемых библиотек. Данные преобразованы в диаграммы для наглядной демонстрации преимуществ одной библиотеки над другой. **Практическая значимость.** Разработана архитектура экспериментального стенда. Даны рекомендации по выбору библиотек разработки семантических шлюзов Интернет-Вещей, функционирующих на основе протоколов MQTT, CoAP, XMPP, HTTP/2. Исходя из этих рекомендаций при невозможности использования библиотеки по умолчанию, легко выбрать аналогичную, максимально близкую по задержке.

## Ключевые слова

Интернет Вещей, интернет вещь, библиотеки, задержки, производительность, MQTT, CoAP, XMPP, HTTP.

## Информация о статье

УДК 004.738

Язык статьи – русский.

Поступила в редакцию 17.01.17, принятая к печати 28.02.17.

**Ссылка для цитирования:** Власенко Л. А., Фам В. Д., Кулик В. А., Киричек Р. В. Исследование производительности программных инструментов межсетевого взаимодействия для семантических шлюзов Интернета вещей // Информационные технологии и телекоммуникации. 2017. Том 5. № 1. С. 44–53.

# RESEARCH ON THE PERFORMANCE OF INTERWORKING SOFTWARE TOOLS FOR INTERNET SEMANTIC GATEWAYS

L. Vlasenko<sup>1</sup>, V. Fam<sup>1</sup>, V. Kulik<sup>1</sup>, R. Kirichek<sup>1\*</sup>

<sup>1</sup> SPbSUT, St. Petersburg, 193232, Russian Federation

\* Corresponding author: kirichek@sut.ru

**Abstract—Research Subject.** This article analyzes the productivity of libraries for developing client-server applications semantic gateways of the Internet of Things based on standard single-board computer Raspberry Pi rev. 3 Model B. As server applications for protocols MQTT and XMPP were used ready-made applications mosquitto and Ejabberd accordingly. For protocols CoAP and HTTP/2 server applications were developed independently. Processing works of this article the following libraries were investigated: Paho-mqtt, hbmqtt, nyamuk for MQTT, aiocoap, txThings for CoAP, xmpppy, sleekxmpp, Twisted-xmpp for XMPP, hyper, hyper, Hyper-h2 for HTTP / 2. Every developed application was written using its own library version of programming high-level language Python, depending on which version library supports. **Method.** The main method of researching the productivity of libraries is a full-scale experiment. There were done 20 experiments for every library. Data of experiments were approximated. The outcome was aggregated in a single table. **Core results.** The article contains delay indexes, loads of CPU and RAM of semantic gateways in converting protocols of different libraries. Data was converted in diagrams for the clear demonstration of predominance of one library over another. **Practical relevance.** Processing the works of this article there was developed architecture of the experimental stand. There were given some recommendations for choosing libraries for the development of semantic gateways of the IoT, based on the protocols MQTT, CoAP, XMPP, HTTP / 2.

**Keywords**—Internet Things, Internet thing, libraries, delays, performance, MQTT, CoAP, XMPP, HTTP.

## Article info

Article in Russian.

Received 17.01.17, accepted 28.02.17.

**For citation:** Vlasenko L., Fam V., Kulik V., Kirichek R.: Research on the Performance of Interworking Software Tools for Internet Semantic Gateways // Telecom IT. 2017. Vol. 5. Iss. 1. pp. 44–53 (in Russian).

## Введение

На сегодняшний день, информационные технологии (ИТ) охватывают значительную часть жизнедеятельности, как человека в частности, так и общества в целом. Будущее ИТ тесно связано развитием сетей связи пост-NGN (*next generation networks*), в том числе технологий Интернета Вещей (ИВ) [1, 2, 3]. Интернет Вещей – концепция описывающая взаимодействие устройств, подключенных к сети связи общего пользования (ССОП), друг с другом и пользователями сети. Все больше различных объектов подключаются к ССОП, что создает новые

возможности в сфере безопасности, медицины, аналитики, управления и экономики в целом. В настоящее время технологии ИВ чаще всего используются для мониторинга и управления реальными и виртуальными устройствами, подключенными к ССОП.

Для организации межсетевого взаимодействия между устройствами ИВ и облачным сервисом ИВ (или пользователем), применяемым для управления и контроля данной интернет вещью, используются различные протоколы ИВ [4, 5]. Одной из проблем организации систем взаимодействия устройств ИВ с облачным сервисом и конечным пользователем является преобразование пакетов, из формата, используемого в локальной сети, в формат, используемый облачным сервисом или конечным пользователем. Именно для решения этой проблемы предлагаются использовать специальное устройство, выполняющие преобразование пакетов для протоколов прикладного уровня – семантический шлюз [6, 7, 8].

На производительность межпротокольного преобразования влияет производительность программных инструментов разработки приложений (библиотека). В данной статье будут исследованы следующие показатели производительности семантического шлюза для различных библиотек разработки приложений: задержки преобразования протоколов, загрузка оперативной памяти, загрузка центрального процессора.

В данной работе проводится исследование производительности различных библиотек для протоколов MQTT, CoAP, XMPP и HTTP, на базе типового одноплатного компьютера Raspberry Pi rev. 3 Model B. В результате данной работы были получены показатели производительности для различных библиотек разработки клиентов для протоколов ИВ и были сформулированы выводы, в которых были описаны преимущества и недостатки исследуемых библиотек.

### **Архитектура экспериментального стенда**

Сетевой шлюз ИВ, представленный на рисунке (рис. 1), выполняет межпротокольное преобразование для сетей MQTT, CoAP, XMPP, HTTP/2. На канальном и сетевом уровне преобразование выполняет шлюз межсетевого взаимодействия, с помощью которого подключенные разнородные сети (например, сети ZigBee, 6LoWPAN, LoRaWAN, WiFi, Ethernet и др.) осуществляют обмен данными [9]. На прикладном уровне задачу преобразования протоколов прикладного уровня выполняет семантический шлюз Интернет Вещей, выполняющий коммутацию, захват и анализ пакетов, получаемых от клиентов каждой сети [10, 11]. Клиенты в свою очередь разработаны с помощью ряда библиотек на языке программирования Python.

Программная архитектура семантического шлюза представляет собой приложение, функционирующее на основе архитектуры микросервисов и разработанное с помощью каждой из рассматриваемых библиотек. Микросервисы взаимодействуют друг с другом через локальный сервер HTTP. Данный экспериментальный стенд был разработан на базе модельной сети лаборатории Интернета Вещей Санкт-Петербургского государственного университета телекоммуникаций им. проф. М. А. Бонч-Бруевича [12, 13, 14].

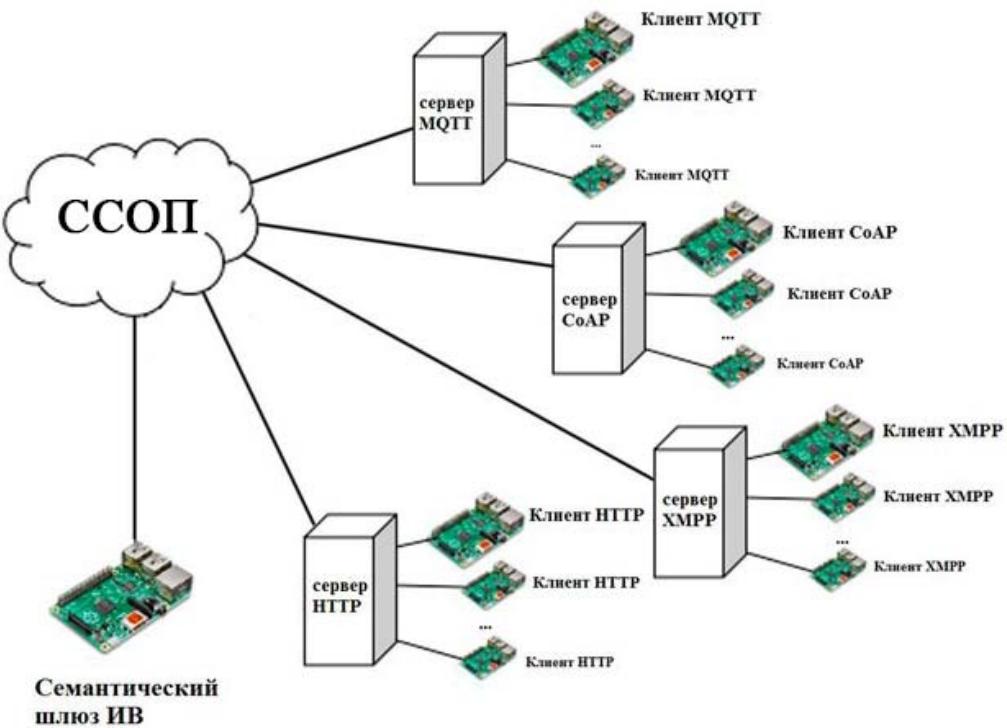


Рис. 1. Модель взаимодействия сегментов ССОП с помощью semanticического шлюза Интернета Вещей

### Библиотеки программирования

В работе исследованы показатели производительности semanticического шлюза при преобразовании пакетов различных прикладных форматов. А именно: задержки, загрузки процессора и оперативной памяти [15]. Объектами исследования являлись библиотеки `raho-mqtt`, `hbmqtt`, `nyamuk` для MQTT, `aio.coap`, `txThing` для CoAp, `xmpppy`, `sleekxmpp`, `Twisted-xmpp` для XMPP, `hyper`, `Hyper-h2` для HTTP/2. С их использованием были разработаны клиентские приложения:

#### MQTT

– Клиентское приложение, разработанное с помощью библиотеки `Paho-mqtt`, позволило подключаться к MQTT брокеру и подписаться на топики для получения сообщений. Клиент поддерживает Python 2.7/3. В качестве сервера MQTT был использован серверное приложение (брокер) MQTT – `mosquitto`.

– Клиент библиотеки `Nyamuk` также позволяет подписываться на топики брокера (издателя). Библиотека основана на библиотеке `libmosquitto`. Реализует 3.1 и 3.1.1 версии протокола MQTT. Поддерживает Python 2.7.

– Клиент `hbmqtt` осуществляет функции, аналогичные клиентам остальных библиотек. Библиотека реализует полный набор спецификаций протокола MQTT 3.1.1. Поддерживает Python 3.

#### CoAP

– `txThings` основан на наборе библиотек (фреймворк) `Twisted`. Поддерживает Python 2.7/3.

– `aio.coap`, основана на библиотеке `txThings`. Библиотека поддерживает Python 3, использует асинхронный ввод\вывод для обеспечения параллельного выполнения операций.

### XMPP

- xmpppy – библиотека Python, которая предназначена для разработки клиентов XMPP. Библиотека наследует часть кода из jabberpy. Поддерживает Python 2.7.
- SleekXMPP-библиотека XMPP для Python 2.7/3. Поддерживает Python 2.7/3.
- Twisted-xmpp. Движок библиотеки написан на Python и поддерживает версии 2.7/3.

### HTTP/2

HTTP/2 – новый стандарт протокола HTTP. Существуют следующие библиотеки для разработки клиент-серверных приложений для языка Python:

- hyper — библиотека Python для разработки клиентов HTTP. Обеспечивает поддержку спецификаций 14, 15, и 16 протокола HTTP 2.
- hyper-h2 — более современная версия библиотеки hyper. Поддерживает протокол HTTP/2.

Для каждой библиотеки было проведено по 20 экспериментов преобразования протоколов. Были зафиксированы показатели задержки в мс, загрузки оперативной памяти и центрального процессора в %. Апроксимированные данные занесены в таблицы (табл. 1, табл. 2).

Таблица 1.

Результаты производительности библиотек при преобразовании протоколов

Протокол	Используемая библиотека	Среднее время, мс		
		MQTT--> CoAP	MQTT--> XMPP	MQTT--> HTTP
MQTT	Paho-mqtt	<b>2,8</b>	<b>5,96</b>	<b>167</b>
	hbmqtt	<b>80</b>	<b>70</b>	<b>200</b>
	nyamuk	<b>50</b>	<b>60</b>	<b>180</b>
CoAP		CoAP--> MQTT	CoAP--> XMPP	CoAP--> HTTP
	aiocoap	<b>3,3</b>	<b>5,8</b>	<b>212</b>
	txThings	<b>2,75</b>	<b>7,95</b>	<b>259</b>
XMPP		XMPP--> MQTT	XMPP--> CoAP	XMPP--> HTTP
	xmpppy	<b>6,6</b>	<b>7,5</b>	<b>279</b>
	sleekxmpp	<b>8,06</b>	<b>15,45</b>	<b>220</b>
	Twisted-xmpp	<b>3,5</b>	<b>7</b>	<b>200</b>
HTTP/2		HTTP--> MQTT	HTTP--> CoAP	HTTP--> XMPP
	hyper	<b>7,5</b>	<b>7</b>	<b>6,88</b>
	Hyper-h2	<b>10</b>	<b>20</b>	<b>15</b>

Таблица 2.

Загрузка оперативной памяти и центрального процессора при преобразовании протоколов

Протокол	Используемая библиотека	% CPU	% RAM
MQTT	Paho-mqtt	5,1	4,6
	hbmqtt	12	6,2
	nyamuk	10	7
CoAP	aiocoap	5,4	5
	txThings	4	5,2
XMPP	xmpppy	4,6	5,2
	sleekxmpp	7,5	5
	Twisted-xmpp	4	5,5
HTTP/2	hyper	4,3	5,7
	Hyper-h2	6	8,2

Результат эксперимента для протокола MQTT (рис. 2).

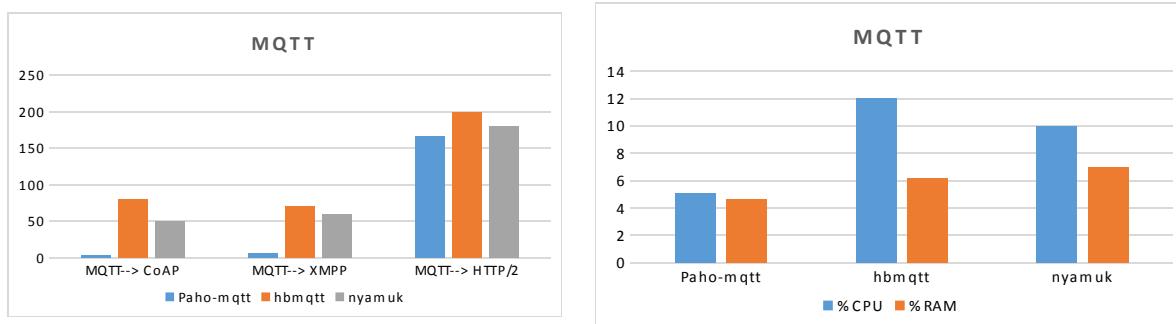


Рис. 2. Сравнение загрузок ЦП и ОП при использовании различных библиотек MQTT

Для протокола MQTT предпочтительной по загрузке ОП и ЦП, а также по значению задержки является библиотека Paho-mqtt (библиотека по умолчанию). При использовании библиотеки nyamuk значения выше, но приемлемые для систем критичных к показателям задержек. Библиотека hbmqtt имеет худшие результаты относительно остальных. Стоит обратить внимание на загрузку центрального процессора, она составляет 12 %, что может оказаться критичным для системы, чувствительной к загрузкам.

Результаты эксперимента для протокола CoAP (рис. 3).

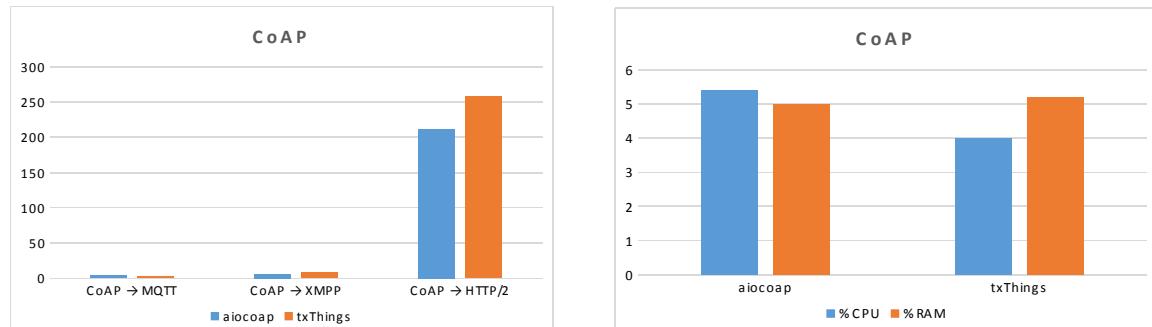


Рис. 3. Сравнение загрузок ЦП и ОП при использовании различных библиотек CoAP

Для протокола СоАР библиотека txThings имеет лучшие показатели загрузки, но отличие от библиотеки aiosoap в данном пункте минимальны. Так как показатели задержки aiosoap меньше, чем txThings, aiosoap является предпочтительным вариантом. Обе библиотеки подходят для систем критичных к показателям задержки и загрузок.

Результаты эксперимента для протокола XMPP (рис. 4).

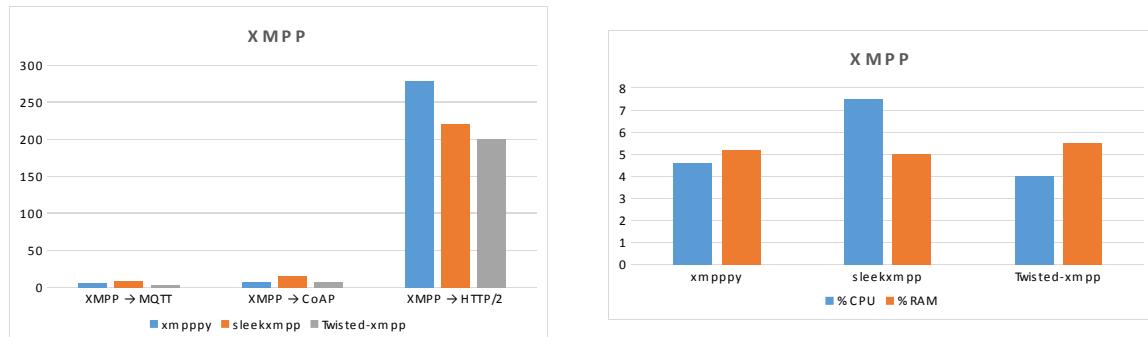


Рис. 4. Сравнение загрузок ЦП и ОП при использовании различных библиотек XMPP

Для протокола XMPP лучшей является библиотека xmpppy, так как имеет минимальные показатели задержки и загрузки ЦП и ОП. Библиотека Twisted-xmpp незначительно превышает уровень задержки xmpppy, но также может использоваться для систем критичных к уровням загрузок и задержки. Sleekxmpp имеет большой показатель загрузки ЦП, поэтому нуждается в предварительном анализе перед использованием в определенных системах.

Результаты эксперимента для протокола HTTP/2 (рис. 5).

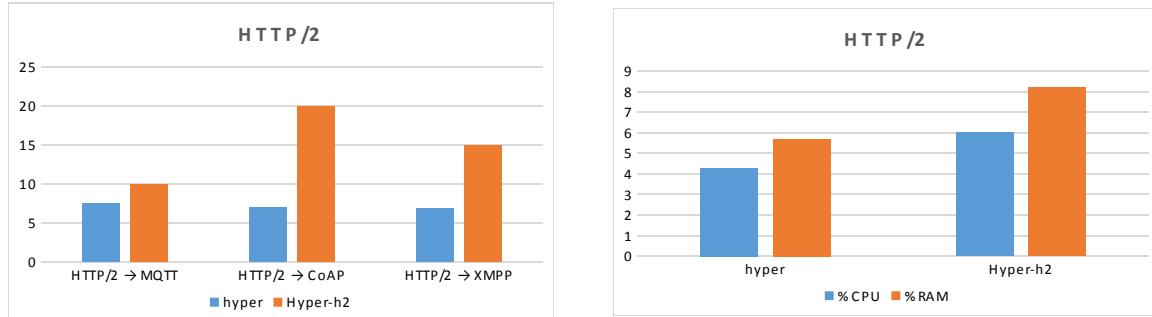


Рис. 5. Сравнение загрузок ЦП и ОП при использовании различных библиотек HTTP/2

Протокол HTTP/2 изначально не предназначен к использованию в технологиях ИВ, поэтому задержки при преобразовании протоколов MQTT, СоАР, XMPP в HTTP/2 очень высокие вне зависимости от выбранной библиотеки. Обратное же преобразование показывает удовлетворительные для критичных к задержкам систем значения. Наиболее предпочтительной к использованию библиотекой HTTP/2, является HTTP/2 – Hyper. Hyper-h2 показывает более высокие задержки и показатели загрузки шлюза, но также может использоваться в системах критичным к уровням загрузок и задержек.

Согласно результатам эксперимента, процент загрузки ОП и ЦП семантического шлюза при преобразовании пакетов незначителен для следующих библиотек raho-mqtt, aiosoap, xmpppy и Hyper. Данные библиотеки показывают лучшие

показатели задержки, следовательно, для достижения максимальной производительности, используя данную конфигурацию программно-аппаратной платформы, стоит выбирать их. Библиотека `hbmqtt` для протокола MQTT, библиотека `sleekxmpp` для протокола XMPP и библиотека `Hyper-h2` для HTTP/2 нуждаются в предварительном анализе, так как загружают от 7 до 12 % от производительности ЦП и могут не подойти для систем критичных к показателю загрузки.

Протокол HTTP/2 изначально не предназначен к использованию в технологиях ИВ, поэтому задержки при преобразовании протоколов MQTT, CoAP, XMPP в HTTP/2 высокие вне зависимости от выбранной библиотеки. В остальных случаях задержки при преобразовании протоколов с использованием библиотек по умолчанию являются наименьшими, и, следовательно, могут использоваться в системах критичных к показателю задержки.

## Заключение

В результате данной работы был проанализирован ряд библиотек для разработки клиентов для протоколов MQTT, CoAP, XMPP и HTTP-протоколов Интернет Вещей, на базе типового одноплатного компьютера Raspberry Pi rev. 3 Model B.

Исследованы особенности каждой из библиотек, разработаны клиентские приложения. Были получены показатели производительности семантических шлюзов ИВ, проведен ряд экспериментов, при которых были исследованы: сетевая задержка в мс, загрузка центрального процессора в % и загрузка оперативной памяти в %.

Полученные результаты показали, каких рекомендаций стоит придерживаться для организации семантических шлюзов для архитектуры, предложенной в рамках научно-исследовательской работы.

## Литература

1. Кучеряный А. Е. Интернет Вещей // Электросвязь. 2013. № 1. С. 21–24.
2. Кучеряный А. Е., Кучеряный Е. А., Прокопьев А. В. Самоорганизующиеся сети. СПб.: Любавич. 2011. 312 с.
3. Кучеряный А. Е. Самоорганизующиеся сети и новые услуги // Электросвязь. 2009. № 1. С. 19–23.
4. Хакимов А. А., Мутханна А. С., Кулик В. А., Киричек Р. В. Анализ производительности шлюза умного дома на базе облачной платформы AllJoin // Информационные технологии и телекоммуникации. 2016. Т. 4. № 2. С. 77–85. URL: <http://www.sut.ru/doc/nauka/review/20162/77-85.pdf>
5. Мутханна А. С., Хакимов А. А. Сравнение протоколов Web-вещей // Информационные технологии и телекоммуникации. 2015. № 4 (12). С. 97–107. URL: <http://www.sut.ru/doc/nauka/review/4-15.pdf>
6. Desai P., Sheth A., Anantharam P. Semantic Gateway as a Service Architecture for IoT Interoperability, in Mobile Services (MS) // IEEE International Conference. 2015. p. 313–319. DOI: 10.1109/MobServ.2015.51.
7. Kotis K., Katasonov A. Semantic Interoperability on the Web of Things // Sixth International Conference on Complex, Intelligent and Software Intensive Systems (CISIS). 2012. pp. 630–635. DOI: 10.1109/CISIS.2012.200.
8. Gyrard A., Kanti Datta S., Bonnet C., Boudaoud K. A Semantic Engine for Internet of Things // 9<sup>th</sup> International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS). 2015. pp. 336–341. DOI: 10.1109/IMIS.2015.83.
9. Кулик В. А., Киричек Р. В., Бондарев А. Н. Методы исследования беспроводных каналов связи Интернета Вещей в условиях совместной работы // Информационные технологии и телекоммуникации. 2015. № 1 (9). С. 106–114. URL: <http://www.sut.ru/doc/nauka/review/1-15.pdf>

10. Кулик В. А., Киричек Р. В., Кучеряный А. Е. Программно-аппаратный комплекс для тестирования устройств Интернета Вещей // Информационные технологии и телекоммуникации. 2015. № 4 (12). С. 67–76. URL: <http://www.sut.ru/doci/nauka/review/4-15.pdf>
11. Кулик В. А., Выборнова А. И. Методы комплексного тестирования Интернета Вещей // Распределенные компьютерные и телекоммуникационные сети: управление, вычисление и связь (DCCN). 2016. С. 305–312.
12. Kirichek R., Golubeva M., Kulik V., Koucheryavy A. The Home Network Traffic Models Investigation // 18<sup>th</sup> International Conference on Advanced Communication Technology (ICACT). 2016. pp. 97–100. DOI: 10.1109/ICACT.2016.7423280.
13. Kirichek R., Koucheryavy A. Internet of Things Laboratory Test Bed // Lecture Notes in Electrical Engineering. 2016. Vol. 348. pp. 485–489. DOI: 10.1007/978-81-322-2580-5\_44.
14. Kirichek R., Vladyko A., Zakharov M., Koucheryavy A. Model Networks for Internet of Things and SDN // 18<sup>th</sup> International Conference on Advanced Communication Technology (ICACT). 2016. pp. 76–79. DOI: 10.1109/ICACT.2016.7423280.
15. Киричек Р. В., Владыко А. Г., Захаров М. В., Кучеряный А. Е. Модельные сети для Интернета Вещей и Программируемых Сетей // Информационные технологии и телекоммуникации. 2015. № 3 (11). С. 17–26. URL: <http://www.sut.ru/doci/nauka/review/3-15.pdf>

## References

1. Koucheryavy, A. Internet of Things // Electrosvyaz'. 2013. No. 1. pp. 21–24.
2. Koucheryavy, A., Koucheryavy, E., Prokopiev, A. Self-Organizing Networks. SPb.: Lubavich. 2011. 312 p.
3. Koucheryavy A. Self-Organizing Networks and New Services // Electrosvyaz'. 2009. № 1. pp. 19–23.
4. Hakimov, A., Muthanna, A., Kulik, V., Kirichek, R. Analysis of the Smart Home Gateway Performances based on Cloud Platform AllJoin // Telecom IT. 2016. Vol. 4. No. 2. pp. 77–85 (in Russian). URL: <http://www.sut.ru/doci/nauka/review/20162/77-85.pdf>
5. Muthanna, A., Hakimov, A. Comparison of Protocols for Web of Things // Telecom IT. 2015. No. 4 (12), pp. 97–107. URL: <http://www.sut.ru/doci/nauka/review/4-15.pdf>
6. Desai, P., Sheth, A., Anantharam, P. Semantic Gateway as a Service Architecture for IoT Interoperability, in Mobile Services (MS) // IEEE International Conference. 2015. p. 313–319. DOI: 10.1109/MobServ.2015.51.
7. Kotis, K., Katasonov, A. Semantic Interoperability on the Web of Things // Sixth International Conference on Complex, Intelligent and Software Intensive Systems (CISIS). 2012. pp. 630–635. DOI: 10.1109/CISIS.2012.200.
8. Gyrard, A., Kanti Datta, A., Bonnet, C., Boudaoud, K. A Semantic Engine for the Internet of Things // 9<sup>th</sup> International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS). 2015. pp. 336–341. DOI: 10.1109/IMIS.2015.83
9. Kulik, V., Kirichek, R., Bondarev, A. Research Methods Wireless Channel of Internet of Things in Other Networks Interference // Telecom IT. 2015. No. 1 (9). pp. 106–114. URL: <http://www.sut.ru/doci/nauka/review/1-15.pdf>
10. Kulik, V., Kirichek, R., Koucheryavy, A. Hardware-Software Complex for Testing Internet of Things Devices // Telecom IT. 2015. No. 4 (12). pp. 67–76. URL: <http://www.sut.ru/doci/nauka/review/4-15.pdf>
11. Kulik, V., Vybornova, A. Methods of Complex Testing the Devices of the Internet of Things // Distributed Computer and Communication Networks (DCCN). 2016. pp. 305–312.
12. Kirichek, R., Golubeva, M., Kulik, V., Koucheryavy, A. The Home Network Traffic Models Investigation // 18<sup>th</sup> International Conference on Advanced Communication Technology (ICACT). 2016. pp. 97–100. DOI: 10.1109/ICACT.2016.7423280.
13. Kirichek, R., Koucheryavy, A. Internet of Things Laboratory Test Bed // Lecture Notes in Electrical Engineering. 2016. Vol. 348. pp. 485–49. DOI: 10.1007/978-81-322-2580-5\_44.
14. Kirichek, R., Vladyko, A., Zakharov, M., Koucheryavy, A. Model Networks for the Internet of Things and SDN // 18th International Conference on Advanced Communication Technology (ICACT). 2016. pp. 76–79. DOI: 10.1109/ICACT.2016.7423280.

15. Kirichek, R., Vladyko, A., Zakharov, M., Koucheryavy, A. Model Networks for Internet of Things and SDN // Telecom IT. 2015. No. 3 (11). pp. 17–26. URL: <http://www.sut.ru/doc/nauka/review/3-15.pdf>

**Власенко Лидия Александровна**

– студентка, СПбГУТ, Санкт-Петербург, 193232, Российская Федерация, milaboky.liduska.v@gmail.com

**Фам Ван Даи**

– студент, СПбГУТ, Санкт-Петербург, 193232, Российская Федерация, daipham93@gmail.com

**Кулик Вячеслав Андреевич**

– ассистент, СПбГУТ, Санкт-Петербург, 193232, Российская Федерация, vaklicr@gmail.com

**Киричек Руслан Валентинович**

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

**Vlasenko Lydia**

– student, SPbSUT, St. Petersburg, 193232, Russian Federation, milaboky.liduska.v@gmail.com

**Fam Van Dai**

– student, SPbSUT, St. Petersburg, 193232, Russian Federation, daipham93@gmail.com

**Kulik Viacheslav**

– assistant, SPbSUT, St. Petersburg, 193232, Russian Federation, vaklicr@gmail.com

**Kirichek Ruslan**

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