

# Исследование Технологий Повышения Эффективности Приложений для Систем Internet of Things

Владимир Саенко  
кафедра информационных управляющих систем  
Харьковский национальный университет  
радиоэлектроники Харьков, Украина  
vladimir.sayenko@nure.ua

Роман Руденко  
кафедра информационных управляющих систем  
Харьковский национальный университет  
радиоэлектроники Харьков, Украина  
rorudenko@gmail.com

## Study on Technologies for Increasing the Efficiency of Internet of Things Apps

Vladimir Sayenko  
Information Control System department  
Kharkov National University of Radioelectronics  
Kharkov, Ukraine  
vladimir.sayenko@nure.ua

Roman Rudenko  
Information Control System department  
Kharkov National University of Radioelectronics  
Kharkov, Ukraine  
rorudenko@gmail.com

**Аннотация**—Рассмотрены вопросы повышения эффективности систем Интернета вещей. Предлагается методология повышения эффективности функционирования системы Интернета вещей, основанная на улучшении отдельных компонентов, выборе протоколов коммуникаций, распределении трафика и формировании распределенной топологии обработки данных. Актуальность данной темы обусловлена стремительным развитием Интернета вещей и отсутствием методологии повышения эффективности функционирования системы.

**Abstract**—The issues of improving the efficiency of the Internet of Things are considered. A methodology is proposed for improving the performance of the Internet of Things system based on improving individual components, choosing communication protocols, distributing traffic, and creating a distributed data processing topology. The relevance of this topic is due to the rapid development of the Internet of Things and the lack of a methodology to improve the efficiency of the system.

**Ключевые слова**—Интернет вещей, туманные вычисления, протокол.

**Keywords**—Internet of Things, Fog computing, protocol.

### I. ВВЕДЕНИЕ

Человечество вступает в новую эпоху – Интернета вещей (Internet of Things - IoT). В современном мире с каждым днем появляется все больше и больше новостей, посвященных разработкам в этой области.

Интернет вещей - это современная технология, в которой интеллектуальные объекты вокруг нас взаимодействуют друг с другом для достижения общих целей и включают в себя такие объекты, как смарт-камеры, носимые датчики, датчики окружающей среды, интеллектуальные бытовые приборы, транспортные средства и т. д. [1].

Для функционирования таких систем актуальными становятся вопросы повышения их эффективности.

Системы IoT интегрируются в облачные сервисы.

К основным преимуществам облачных технологий можно отнести: отказоустойчивость; высокий уровень безопасности; высокая скорость обработки данных; экономичность инженерных затрат; возможность использования хранилищ данных.

В сфере облачных технологий все большую и большую популярность приобретает принципиально новое направление технологий, получивший название "туманные вычисления" (Fog computing). Ее основной особенностью является обработка данных в непосредственной близости от источников их получения, без необходимости их передачи в облако.

Туманные технологии не является альтернативой для облачных. Напротив, они плодотворно взаимодействует с облачными технологиями, особенно в администрировании и аналитике данных, и такое взаимодействие порождает новый класс приложений.



Основные архитектурные отличия Fog от Cloud:

- Обеспечение качества услуг (QoS, Quality of Service), что требует динамической адаптации приложений к состоянию сети.
- Отслеживание местоположения (Location Awareness) для того, чтобы поддерживать стабильность работы приложения в условиях мобильности терминала.
- Отслеживание контекстной информации (Context Awareness), т.е. способность обнаруживать наличие доступных ресурсов поблизости.

Эти технологии также способствуют повышению эффективности построения систем IoT.

## II. ОСНОВНЫЕ РЕЗУЛЬТАТЫ

Пусть система Интернета вещей представляется в виде компонентной структуры: сенсорный узел, брокер, сервер, приложение пользователя.

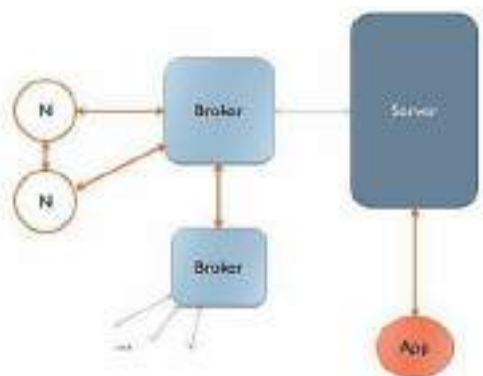


Рис. 1. Структура системы Интернета вещей

Сенсорный узел объединяет информацию от нескольких датчиков и направляет ее либо по запросу, либо самостоятельно с определенным интервалом времени или по происшествии какого-либо события на сервер. Брокер - это специализированный "сервер", который осуществляет пересылку сообщений, их хранение и фильтрацию. Приложение пользователя установлено на персональном устройстве, и служит для графического представления обработанной сервером информации и управления системой.

Предлагается методология повышения эффективности функционирования системы Интернета вещей, основанная на:

- улучшении отдельных компонентов,
- выборе протоколов коммуникаций,
- распределении трафика и формировании распределенной топологии обработки данных (туманные вычисления).

Улучшение отдельных компонентов предполагает выбор определенного типа компонента с определенными

характеристиками. К таким характеристикам относятся стоимостные, энергетические, физические (размеры).

Выбор протокола коммуникаций подразумевает рассмотрение всех участков передачи данных. Предлагается следующая структура связей в системы интернета вещей:

- между сенсорными узлами/датчиками,
- сенсорный узел – брокер,
- брокер – сервер,
- сервер - приложение пользователя.

Поддержка связей реализуется с помощью специальных прикладных протоколов. Такая классификация позволяет абстрагироваться от конкретного решения по применению протокола и сфокусироваться на его целевом назначении.

Для обеспечения связи между сенсорными узлами/датчиками используется протокол DDS (Data Distribution Service). Этот протокол распределяет данные между устройствами. DDS реализует прямую шинную связь между устройствами на базе реляционной модели данных. Протокол DDS реализует многоадресную систему, используя UDP.

На участке сенсорный узел - брокер реализуются такие задачи, как регистрация сенсорного узла; конфигурация и настройки узлов; передача и распределение информации и т.д. На этом сегменте сети могут использоваться два следующих протокола: XMPP (Extensible Messaging and Presence Protocol) и CoAP (Constrained Application Protocol). Выбор конкретного протокола зависит от условий реализуемой сети.

В относительно небольших персональных сетях предлагается использовать протокол XMPP. Он обеспечивает простой способ адресации устройств и поддерживает различные коммуникационные модели (запрос-ответ, публикация-подписка и другие). Сильными сторонами этого протокола являются удобная передача данных между удаленными, чаще всего независимыми точками, безопасность и масштабируемость [2].

Для сетей с ограниченными ресурсами, низким энергопотреблением предлагается использовать протокол CoAP. Это протокол прикладного уровня, работающий на основе REST (Representational State Transfer). В отличие от протокола HTTP CoAP – это бинарный протокол, который работает поверх UDP, что значительно уменьшает общий размер передаваемых данных и повышает гибкость взаимодействия.

На участке брокер - сервер решаются следующие задачи: сбор и агрегация данных; организация очереди сообщений; распределение и хранение информации "до востребования".



Для загруженных сетей с большим количеством устройств предлагается применять протокол, снижающий нагрузку на канал за счет организации очередей, – протокол MQTT (Message Queue Telemetry Transport). Он предназначен для телеметрии и дистанционного мониторинга. Используется для обмена сообщениями между устройствами по принципу "издатель-подписчик" и позволяет устройствам посылать и получать данные при возникновении некоторого события.

Для сетей, использующих оборудование различных платформ и допускающих применение простого протокола передачи сообщений, предлагается использовать протокол STOMP (Simple Text Oriented Message Protocol). Это простой протокол обмена сообщениями, предполагающий широкое взаимодействие со многими языками, платформами и брокерами. Протокол в целом похож на HTTP и является простым текстовым протоколом, что позволяет клиентам STOMP общаться с любым брокером сообщений, поддерживающим данный протокол.

Отличаются данные протоколы тем, что протокол MQTT обеспечивает "сквозную" связь, как от брокера к сенсорным узлам, так и от брокера к серверу, тогда как протокол STOMP ориентирован только на взаимодействие брокера с сервером.

На участке Сервер - Приложение выполняются задачи, связанные с взаимодействием пользователя и системы: получение информации с сервера; конфигурация пользователем параметров (частоты получения информации, активация/деактивация датчиков и узлов и т.д.) и другие.

Для обеспечения связи приложения пользователя с другими элементами сети Интернета вещей предлагается использовать протокол SOAP (Simple Object Access Protocol), который поддерживает обслуживание Web-сервисов и обеспечивает совместную работу платформы и интернет-приложений. Преимуществом этого протокола является выделенный механизм доступа RPC (Remote Procedure Call), который отвечает за удаленный вызов функций. SOAP использует базовую модель соединения, обеспечивающую согласованную передачу сообщения от отправителя к получателю, потенциально допускающую наличие посредников, которые могут обрабатывать часть сообщения или добавлять к нему дополнительные элементы.

Ключевые особенности протоколов зависят от их предполагаемого применения. Поэтому к выбору оптимального протокола для своего приложения нужно подходить основательно, объективно взвешивать все положительные и отрицательные свойства каждого из них, исходя из конкретных потребностей.

Традиционная система связи включает в себя клиентское устройство и центр обработки данных.

Однако, за период поступления данных в центр обработки данных, они могут существенно устареть и потерять свою актуальность. Задержка в передаче данных может быть несущественной, а в некоторых случаях достаточно критической.

Для решения этой проблемы предлагается использовать технологию "туманных вычислений". Туманные вычисления могут выполняться в любом месте, куда могут поступить данные - в центрах обработки данных, на крайних узлах сети (edge of network) и даже где-то между ними. Туманные технологии распределяют вычислительные ресурсы, сервисы, средства коммуникации, средства хранения и управления, перемещая их ближе к самим устройствам, различным системам или непосредственно к пользователям [3].

Преимуществом туманных вычислений является снижение объема данных, передаваемых в облако, что уменьшает требования к пропускной способности сети, увеличивает скорость обработки данных и снижает задержки в принятии решений.

Для разработки структуры туманных вычислений выделено три цели – они должны быть горизонтально масштабируемыми; должны быть в состоянии работать через облако; и быть общесистемной технологией, которая простирается от границ сети до облака и различных сетевых протоколов.

Самый большой потенциал развития технологии туманных вычислений имеют следующие отрасли: энергетика, коммунальные службы, и транспорт, сельское хозяйство, торговля, а также здравоохранение и промышленное производство [4].

### III. ВЫВОДЫ

Предложена методология повышения эффективности функционирования системы Интернета вещей, основанная на улучшении отдельных компонентов, выборе протоколов коммуникаций, распределении трафика и формировании распределенной топологии обработки данных. Методология получила апробацию на кафедре Информационных управляющих систем Харьковского национального университета радиоэлектроники (специальность «Компьютерные науки»).

### ЛИТЕРАТУРА REFERENCES

- [1] Грингард С. Интернет вещей. Будущее уже здесь / Сэмюэл Грингард., 2017. – 188 с.
- [2] Протоколы «Интернета вещей»: основные сведения. [Электронный ресурс] - Режим доступа: <https://rtsoft.ru/press/articles/detail.php?ID=2718>.
- [3] Черников А . Новое в Cloud Computing: репликация и туман. [Электронный ресурс] / А . Черников – Режим доступа [https://ko.com.ua/novoe\\_v\\_cloud\\_computing\\_replikaciya\\_i\\_tuman\\_115420](https://ko.com.ua/novoe_v_cloud_computing_replikaciya_i_tuman_115420).
- [4] Туманные вычисления. [Электронный ресурс] - Режим доступа: [http://www.tadviser.ru/index.php/Статья:Туманные\\_вычисления\\_\(Fog\\_computing\)](http://www.tadviser.ru/index.php/Статья:Туманные_вычисления_(Fog_computing)).

