УДК 519.718.3

ОБЪЕКТ ИНФРАСТРУКТУРЫ «УМНОГО ГОРОДА»: СИСТЕМА МОНИТОРИНГА ОКРУЖАЮЩЕЙ ОБСТАНОВКИ И ИНФОРМИРОВАНИЯ/ОПОВЕЩЕНИЯ НАСЕЛЕНИЯ

Е.А. Басыня

Новосибирский государственный технический университет,
Научно-исследовательский институт информационно-коммуникационных технологий
Россия, 630073, г. Новосибирск, пр. К. Маркса, 20

Аннотация. *Одним из возможных инструментов повышения уровня жизни горожан выступает автоматизация управления городской инфраструктурой с использованием информационно-коммуникационных решений. На обзор выносится оригинальная система мониторинга окружающей обстановки и информирования/оповещения населения в качестве объекта инфраструктуры «умного города». Излагается проектирование и программная реализация клиентской и серверной частей проекта на базе операционной системы Alpine Linux, языка программирования Python, веб-фреймворка Django и технологии контейнеризации Docker. Освещается одно из реализованных аппаратно-программных клиентских решений с использованием технологий беспроводной передачи данных Wi-Fi, 4G/LTE и LoRaWAN. Конечными потребителями данной системы выступают как государственные учреждения, так и частные предприятия. Способы использования жестко не регламентируются, рекомендовано и апробировано внедрение от «умных транспортных остановок» до заводских и городских распределенных систем информирования/оповещения населения с мониторингом окружающей обстановки. Данная работа является логическим продолжением авторского проекта распределенной системы информирования и экстренного оповещения населения. Научная новизна заключается в осуществлении дополнительной проверки подлинности управляющих воздействий на основе скрытой маркировки информационных потоков, автономно вычисляемой и производимой на стороне клиента и сервера через метаинформацию о предшествующих сетевых взаимодействиях (трассировке маршрута, командах, временных метках и т. д.). Для достижения данной цели используется функция свертки, осуществляется преобразование данных произвольной длины в выходную битовую строку установленной размерности с последующим сигнатурным сопоставлением с таблицей действий, которые не мешают сетевой коммуникации и не детектируются инструментами автомодельности сетевого трафика. Архитектурно предусмотрена возможность интеграции с авторскими наукоемкими решениями: системой интеллектуально-адаптивного управления сетевой инфраструктурой предприятия, а также модулем самоорганизующегося виртуального защищенного канала связи на основе стохастического многослойного шифрования и оверлейных технологий.*

Ключевые слова: *Smart Cities, умные/интеллектуальные объекты, системы информирования и оповещения населения, системы мониторинга окружающей среды, TCP/IP, беспроводная передача данных, Wi-Fi, 4G/LTE, LoRaWAN, управление информационными потоками, безопасность информационных ресурсов.*

Басыня Евгений Александрович (к.т.н.), доцент Новосибирского государственного технического университета, директор Научно-исследовательского института информационно-коммуникационных технологий.

Введение

Развитие концепции «умного города» становится одним из ключевых векторов социально-экономического развития населенных пунктов. Эффективное управление городской инфраструктурой может быть осуществлено посредством внедрения современных наукоемких технических решений для автоматизации информационных и рабочих процессов всех сфер жизнедеятельности общества. Интеллектуальное управление городскими ресурсами и услугами поможет оптимизировать загрузку транспортных магистралей и потребление ресурсов жилищно-коммунального хозяйства, улучшить экологическую ситуацию, создать единую гибкую экосистему сервисов для граждан. Конечной целью данных мероприятий выступает повышение качества жизни населения с оптимизацией финансовых издержек на ее сопровождение.

В Российской Федерации идея «умного города» закреплена в национальном проекте «Жилье и городская среда» и программе «Цифровая экономика». В республике Казахстан концепция нашла отражение в государственной программе «Цифровой Казахстан». Евросоюз закрепил ее в отдельном пункте стратегического плана развития Европы и инвестирует через различные фонды программы двойных дипломов, международный обмен опытом, совместные научно-технические изыскания в этой сфере. В качестве примера можно привести программу «SMARTCITY: Innovative Approach Towards a Master Program on Smart Cities Technologies» (проводимую с 15 ноября 2018 г. по 14 ноября 2021 г.), целью которой является создание нового поколения междисциплинарных инженеров информационно-коммуникационных технологий в сфере «интеллектуальных/умных городов». Проект соответствует принципам Болонского процесса и направлен на развитие Европейского пространства высшего образования.

Соответственно, научным сообществом в данной сфере производится множество научно-исследовательских и опытно-конструкторских работ. Труды О.А. Иващука, И.С. Константинова, О.П. Архипова, А.В. Коськина, О.В. Пилипенко направлены на развитие общих концепций формирования специализированных автоматизированных систем без учета и проработки прикладной реализации [1–3]. Изыскания S. Paiva, D. Santos, J.F. Rosaldo, M.A. Munizaga, C. Palma, V. Moustaka, A. Vakali, L.G. Anthopoulos носят прикладной характер в области определения городских показателей через компьютерное зрение, сбора и аналитики данных [4–7]. Однако предлагаемые методы не учитывают реального объема данных и не могут быть масштабируемы без архитектурных изменений.

Основная проблематика заключается в сложном междисциплинарном подходе к обустройству «умных городов», конвергенция отраслей и технологий лишь начинает развиваться. Даже в технической сфере нет единой модели стандартизации и унифицированного стека технологий для разработки программных и аппаратно-программных решений, которые выступили бы катализатором развития индустрии и позволили распараллеливать научные изыскания с гарантированной совместной взаимной интеграцией, в том числе с существующими инфраструктурными объектами.

Решение данной проблемы может быть выработано только коллаборацией ведущих международных научно-исследовательских институтов с последующим оформлением технических стандартов. Альтернативный путь развития – конкурентная война гигантов индустрии информационных технологий в попытке занять и монополизировать нишу рынка, что и происходит на текущий момент: Huawei, Apple и множество других компаний конкурируют в разработке опера-

ционных систем для интернета вещей (англ. internet of things, IoT) и собственных стеков технологий для «умных домов» и «умных городов».

Осуществляя декомпозицию выделенной проблематики, следует выделить сложность совмещения федеральных систем оповещения населения о чрезвычайных ситуациях с инструментами оповещения о функционировании городской инфраструктуры. Актуальность данной тематики обусловлена необходимостью своевременного таргетированного донесения информации до населения с целью оптимизации процессов различных сфер его деятельности.

Цель работы и постановка задач

Целью данной работы являлась разработка и исследование системы мониторинга окружающей обстановки и информирования/оповещения населения в качестве объекта инфраструктуры «умного города». Целевыми потребителями данной системы выступают как государственные учреждения, так и частные предприятия.

Способы использования жестко не регламентируются, рекомендовано и апробировано внедрение от «умных транспортных остановок» до заводских и городских распределенных систем информирования/оповещения населения (в том числе о чрезвычайных ситуациях) с мониторингом окружающей обстановки. Данная работа является логическим продолжением проекта распределенной системы информирования и экстренного оповещения населения с применением беспроводной передачи данных [8].

Научная новизна заключается в осуществлении дополнительной проверки подлинности управляющих воздействий на основе скрытой маркировки информационных потоков, автономно вычисляемой и производимой на стороне клиента и сервера через метаинформацию о предшествующих сетевых взаимодействиях.

Предлагаемое решение

На этапе проектирования системы мониторинга окружающей обстановки и информирования/оповещения населения было принято решение использовать распределенную клиент-серверную архитектуру, функционирующую в глобальной сети Интернет (рис. 1).

Клиентский модуль системы представляет собой комплекс аппаратно-программного обеспечения, производящего информирование/оповещение граждан в режиме реального времени или проигрывающего аудиотреки согласно установленному расписанию, а также осуществляющего сбор и передачу информации с проводных и беспроводных датчиков (от температуры до уровня веществ в воздухе) центральному серверу.

Для осуществления данных функций были задействованы технологии беспроводной передачи данных Wi-Fi (англ. Wireless Fidelity), 4G/LTE (англ. Fourth Generation / Long-Term Evolution) и энергоэффективной сети дальнего радиуса действия LoRaWAN.

Организация защиты процесса сетевого взаимодействия может осуществляться стандартными технологиями VPN (англ. Virtual Private Network) или с использованием авторской системы самоорганизующегося виртуального защищенного канала связи на основе стохастического многослойного шифрования и оверлейных технологий для обеспечения наивысшего уровня информационной безопасности [8].

Серверная часть представляет собой программную платформу, которая может быть интегрирована в авторскую систему интеллектуально-адаптивного управления сетевой инфраструктурой предприятия [9–10] или представлена отдельным сервисом операционной системы Linux. В первом случае применяются наукоемкие технические решения в области системного анализа, управления и обработки информации, повышающее безопасность, надежность и отказоустойчивость технической системы.

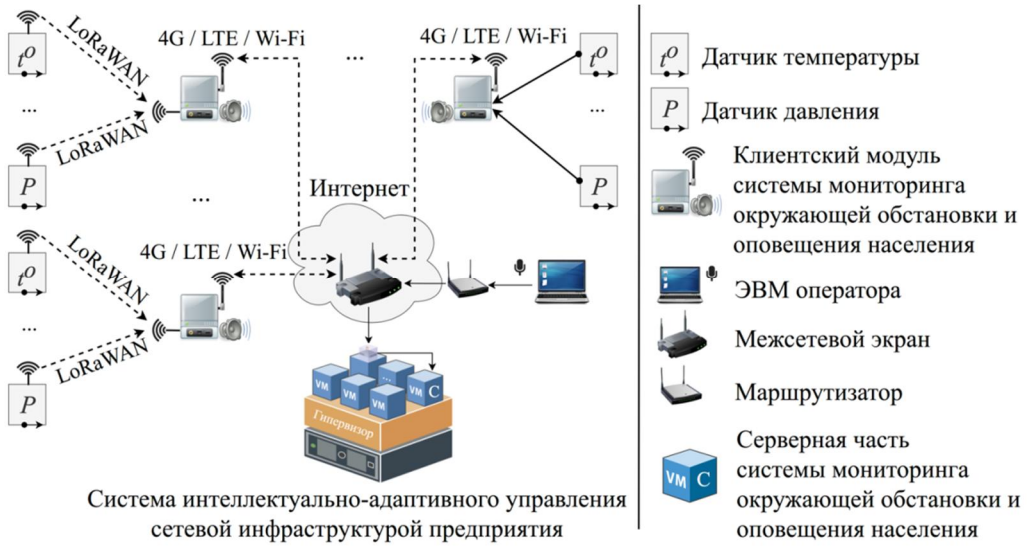

Рис. 1. Архитектура серверной части системы

Существует два общих подхода к организации управления информационной инфраструктурой предприятия: централизованный и децентрализованный. Исследованиями в данной области занимаются российские и зарубежные ученые. Работы И.Ю. Павлова, В.Л. Колоскова, Е.Б. Иванова, А. Бирюкова, А.Н. Зуева, А.А. Лукичева, Е.С. Басана, Н.В. Михайлова. предлагают методы централизованного управления и администрирования сетевых устройств и информационных систем [11–14].

Снижение временных издержек на системное и сетевое администрирование является преимуществом предлагаемых решений. К сожалению, с точки зрения информационной безопасности подобные подходы формируют единую точку входа в информационную инфраструктуру предприятия, представляющую для злоумышленника интерес по ее взлому или выводу из состояний доступности.

Другой интересной стратегией хакера может выступать атака имперсонацией и фальсификацией информационных потоков (в том числе управляющих воздействий), от которой не защищают предложенные методы. Данная угроза и возможность ее эксплуатации обусловлены уязвимостями стека протоколов TCP/IP, базовые протоколы которого даже не производят проверку подлинности субъекта взаимодействия. В качестве простых примеров можно привести ARP и DHCP спуфинг.

В целях совмещения преимуществ централизованного и децентрализованного подходов с нивелированием рисков информационной безопасности было решено ввести дополнительную проверку подлинности управляющих воздействий.

Для этого был разработан оригинальный алгоритм скрытой маркировки информационных потоков, автономно вычисляемой и производимой на стороне клиента и сервера через метаинформацию о предшествующих сетевых взаимодействиях (трассировке маршрута, командах, временных метках и т. д.).

Данная информация передается хеш-функции, осуществляющей преобразование данных произвольной длины в выходную битовую строку установленной размерности. Далее идет сигнатурное сопоставление с таблицей действий, которые не вызывают подозрений у глобального наблюдателя, не мешают сетевой коммуникации. Например, работа с параметрами заголовков дейтаграмм – от модификации простых полей IP пакетов DSCP (англ. Differentiated Services Code Point) до связанной обработки идентификаторов и флагов. Аналогичные действия можно производить и в теле дейтаграмм, включая непосредственные управляющие воздействия сервера. Вариант хеш-функции администратор системы выбирает самостоятельно, как и производит составление таблицы действий.

Таким образом, у заказчиков присутствует возможность применения ноу-хау для повышения уровня безопасности информационных ресурсов. Очередной интересной опцией предлагаемого решения выступает подсчет проходимости пешеходного трафика в локациях через статистическую обработку устройств в зоне действия Wi-Fi сети. Далее рассматриваются проектирование и разработка аппаратно-программных компонентов системы.

Программная реализация предлагаемого решения

Для реализации серверной части использовался язык программирования Python и веб-фреймворк Django для веб-интерфейса и API (англ. application programming interface). Вместо Django ORM (англ. Object-Relational Mapping) для доступа к базе данных (БД) используются SQL-запросы. Для выполнения периодических и отложенных задач используется Celery. Postgresql задействована в качестве системы управления базами данных (СУБД). Роль сервера приложения играет Uwsgi, роль веб-сервера – Nginx. Код сервера, а также различные сервисы (Nginx, OpenVPN server и др.) запускаются внутри Docker-контейнеров.

Архитектура серверной части системы мониторинга окружающей обстановки и информирования/оповещения населения в качестве объекта инфраструктуры «умного города» представлена на рис. 2.

Одним из ключевых элементов архитектуры является контроллер платформы (англ. Platform controller) – средство достижения инкапсуляции, изоляции и управления информационными потоками между модулями системы.

Администрирование предлагаемого решения осуществляется через веб-интерфейс и включает следующие функции для оператора системы (рис. 3):

– авторизация, настройка, конфигурирование учетных записей пользователей/групп;
– загрузка аудиотреков;
– создание, редактирование, удаление плей-листов;
– конфигурирование расписаний и режимов работы;
– вещание в режиме онлайн;
– управление устройствами;
– мониторинг и аналитика функционирования устройств, показаний датчиков (телеметрия);
– вывод устройств на карте;
– управление информационной безопасностью;

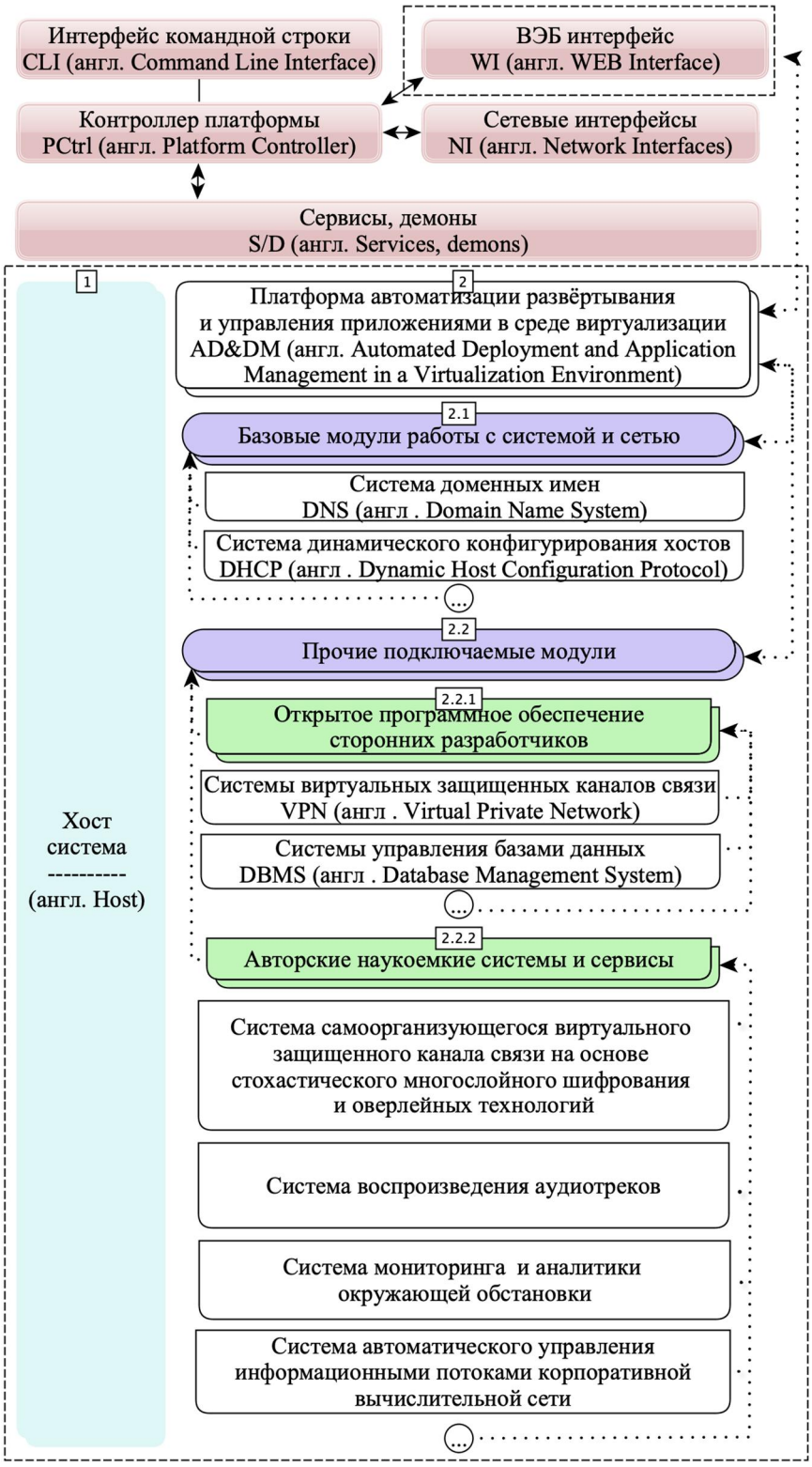

Рис. 2. Архитектура серверной части системы

– оповещение и информирование в режиме реального времени;
– обновление прошивки устройств;
– техническая поддержка;
– справочная информация.

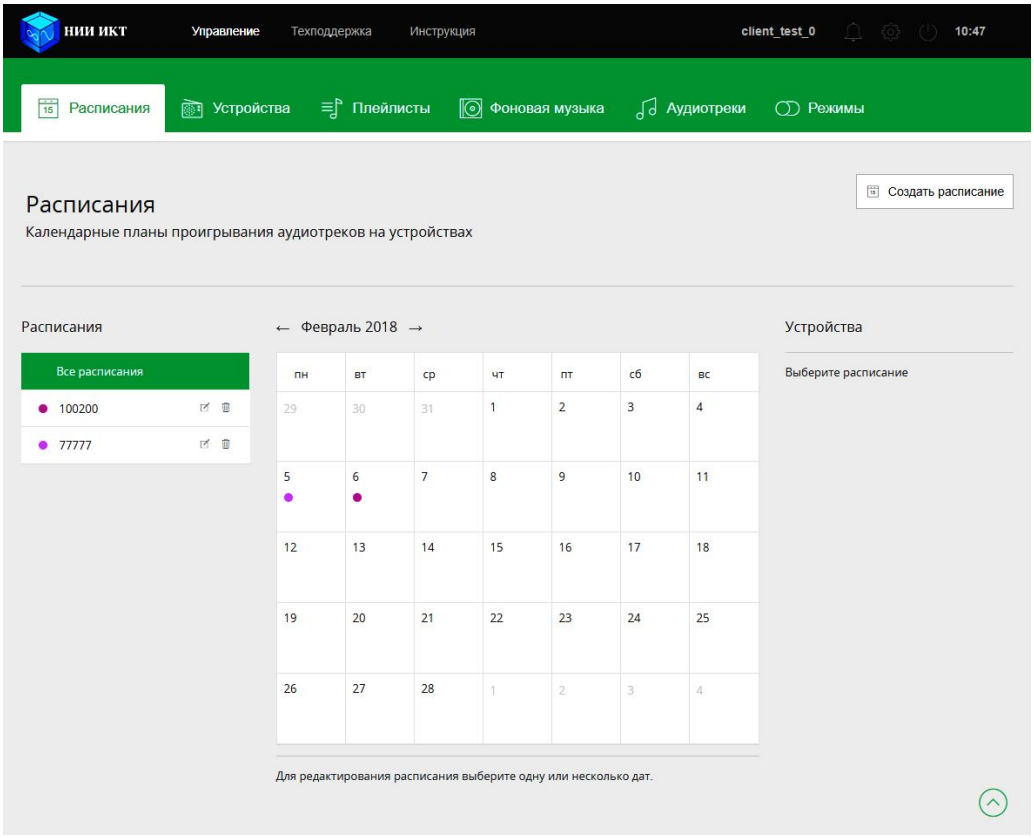

Рис. 3. Веб-интерфейс панели администрирования

Через веб-интерфейс редактируется содержимое базы данных, на сервер загружаются аудиотреки. Информация БД задействуется API для генерации ответов клиентам. Отправка клиентам системы аудиотреков осуществляется по протоколу HTTP внутри виртуальных защищенных каналов связи веб-сервером Nginx.

API предоставляет функционал обновления прошивок, регистрации новых хостов, получения конфигурации клиентов, получения списка аудиотреков для воспроизведения, вещания в режиме онлайн, отправки сведений о функционировании устройства и показании датчиков (телеметрия).

Клиентской частью системы является многопоточное приложение, ведущее трансляцию в режиме реального времени либо получающее от API список аудиотреков для воспроизведения и сами файлы от веб-сервера Nginx с дальнейшим воспроизведением согласно настройкам (в том числе по расписанию).

Дополнительный функционал заключается в сборе информации о состоянии всех аппаратно-программных компонентов хоста и датчиков мониторинга окружающей обстановки/среды для передачи на головной сервер. Таким образом осуществляется телеметрия. При реализации клиентской части использовался язык программирования Python. Аудиоплеер Mpg321 задействуется для воспро-

изведения треков, управление громкостью осуществляется Alsa mixer. Клиенты виртуальных защищенных каналов связи, код клиента и другие сервисы контейнеризованы технологией Docker.

Для автоматизации процесса системного администрирования была реализована функция удаленного обновления прошивки устройств и написаны следующие Ansible скрипты для операционной системы Alpine Linux: конфигурирование сетевых настроек, Apk репозитория (для обновления пакетов) и межсетевого экрана, включение модуля tun ядра (требуется для технологий VPN), установка и настройка OpenNTP клиента, сервисов виртуальных защищенных каналов связи, OpenSSH сервера, Docker, Gitlab Runner (для обновления серверного ПО через Gitlab Continuous Integration). Целевое назначение данных скриптов – автоматическая развертка, настройка и управление конфигурациями системы.

Одно из аппаратных решений клиентской части системы

В ходе выполнения работы было спроектировано и реализовано несколько аппаратно-программных решений под разных заказчиков. В качестве примера на обзор выносится один из возможных вариантов (рис. 4), аппаратная часть которого состоит из нескольких модулей:

– одноплатный микрокомпьютер Raspberry Pi 3B;
– источник питания (ИП);
– источник бесперебойного питания (ИБП);
– усилитель низких частот (УНЧ);
– громкоговоритель;
– микроконтроллерная плата Arduino Micro с проводными датчиками;
– беспроводные датчики (технология LoRaWAN);
– USB-модем 4G/LTE.

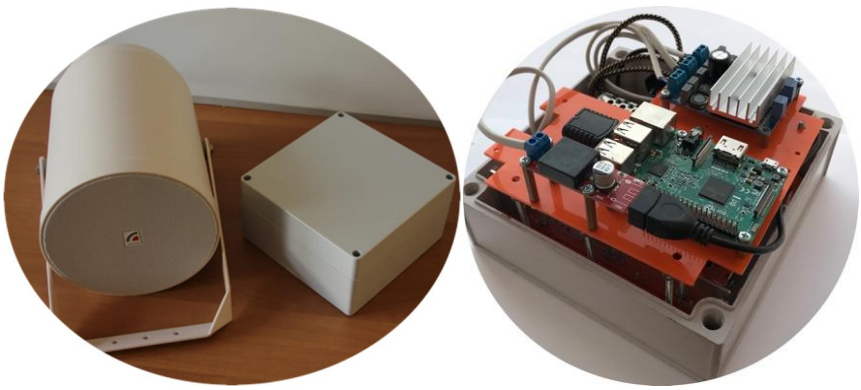

Рис. 4. Пример клиентского аппаратного решения

Сердцем системы является микрокомпьютер Raspberry Pi 3B – одна из последних моделей из линейки популярных одноплатных компьютеров Raspberry Pi. На его плате находятся процессор, оперативная память, разъем HDMI, композитный видео/аудио выход, 4 порта USB 2.0, интерфейсы связи Ethernet, Wi-Fi и Bluetooth. Также на плате расположены 40 контактов ввода/вывода общего назначения (GPIO).

Штатной операционной системой для Raspberry Pi является Linux, устанавливаемая на microSD карту, подключаемую в специальный слот на плате.

На Raspberry Pi 3 установлен 64-битный четырехъядерный процессор ARM Cortex-A53 с тактовой частотой 1,2 ГГц на ядро в составе однокристальной платформы Broadcom BCM2837. Микрокомпьютер имеет 1 ГБ оперативной памяти, которая делится с графической подсистемой. Ядром графической подсистемы является графический двухъядерный процессор VideoCore IV®, поддерживающий стандарты OpenGL ES 2.0, OpenVG, MPEG-2, VC-1 и способный кодировать, декодировать и выводить Full HD-видео (1080p, 60 FPS, H.264 High-Profile).

Источником питания является импульсный преобразователь напряжения 220 В переменного тока в напряжение 24 В (по умолчанию, может меняться в зависимости от условий потребителя) и 5 В постоянного тока для питания подсистемы вывода звука и микрокомпьютера соответственно. В качестве ИП используются промышленно выпускаемые блоки компании Mean Well серии G3. Данные блоки питания характеризуются доступной ценой, неплохим типовым ПД (73–86 %), удовлетворительным уровнем помех (до 80–200 мВ в зависимости от выходного напряжения), обладают встроенной защитой от короткого замыкания, перегрузки и перенапряжения, с гарантией от производителя 3 года.

Для предотвращения потери информации и выхода из строя microSD карты памяти микрокомпьютера Raspberry Pi в случае потери питания, а также для обеспечения «мягкого» отключения устройства используется источник бесперебойного питания с литий-ионным аккумулятором.

В случае исчезновения питающего напряжения на входе устройства аудиосистема отключается. ИБП продолжает питать микрокомпьютер Raspberry Pi от аккумулятора, одновременно сигнализируя по одному из портов GPIO («сигнальный») о потере питания. Питание прекращается по отрицательному фронту сигнала на другом порту GPIO («управляющий»), сигнализирующем о завершении работы операционной системы микрокомпьютера. В случае если на момент завершения работы внешнее питание системы восстановлено, ИБП на короткий промежуток времени отключает и снова подключает микрокомпьютер для инициализации повторного запуска его операционной системы.

Программное обеспечение клиентского устройства запрограммировано следить за сигналом на определенном порту GPIO и по отрицательному фронту сигнала запускать процедуру «мягкого» отключения. Эта процедура выглядит следующим образом:

- после включения устройства, запуска и инициализации необходимых служб сигнал на «управляющем» порту GPIO устанавливается в значение «Истина». Это означает, что запуск устройства произошел нормально и оно готово к мягкому отключению;
- по отрицательному фронту сигнального порта GPIO происходит запуск таймера на 1 секунду. Если в течение этого времени питание было восстановлено – таймер сбрасывается и дальнейших действий не предпринимается;
- если таймер завершил свою работу и питание не было восстановлено – запускается процедура остановки компонентов системы;
- после остановки компонентов происходит остановка системы. Микрокомпьютер Raspberry Pi устроен таким образом, что при остановке системы состояние портов GPIO сбрасывается – этот сброс порождает отрица-

тельный фронт сигнала на управляющем порту GPIO (переход из состояния «Истина» в состояние «Ложь»).

Усилитель низких частот представлен блоком устройства, от которого зависят громкость и чистота звука системы оповещения, количество и характеристики подключаемых громкоговорителей, а также мощность и преобразованное напряжение используемого ИП.

Устройством преобразования электрического сигнала в акустические волны выступает громкоговоритель. Стандартным решением принято использовать Roxton SW-20T, если другое не оговорено с потребителем.

Система мониторинга окружающей обстановки и информирования/оповещения населения предоставляет возможность подключения как проводных, так и беспроводных LoRaWAN датчиков. Устройство Raspberry Pi 3B не имеет аналого-цифрового преобразователя, коммутация проводных датчиков осуществляется посредством платы Arduino Micro на шину SPI микрокомпьютера Raspberry Pi 3B. В зависимости от целевого назначения системы возможно подключение различных типов датчиков: температуры, давления, влажности, кислотности, освещенности, уровня различных веществ, вибраций и многих других. Первичная обработка и фильтрация сигнала осуществляются мощностями Arduino. Полученные данные отправляются на сервер для дальнейшей обработки и аналитики.

Беспроводной датчик – это специализированное устройство с автономным питанием, преобразующее входное физическое воздействие в аналоговый или цифровой сигнал, агрегирующий полученные данные в информационные пакеты малого размера и передающий их серверу короткими сессиями через относительно большие временные интервалы. Типовые варианты применения беспроводных датчиков:

- подсчет событий – с периодическим уведомлением (например, счетчики водо- и газоснабжения, счетчики пассажиропотока);
- сбор информации – с периодическим уведомлением (например, погодные станции, датчики уровня жидкости);
- уведомление о событии – со срочным уведомлением (например, противопожарные датчики задымления, тревожные кнопки, сигнализации).

В конфигурации устройства с возможностью подключения беспроводных датчиков присутствует дополнительный модуль LoRaWAN шлюза, основанный на чипах модулирующего процессора Semtech SX1308 и приемопередатчика Semtech SX1272, работающий в нелицензируемом радиочастотном диапазоне 868 МГц. Мощность сигналов в сети LoRaWAN не превышает установленных российским законодательством ограничений. Информация с беспроводных датчиков перенаправляется на основной сервер через глобальную сеть Интернет для дальнейшего анализа.

Устройства LoRaWAN характеризуются низким энергопотреблением: срок службы автономного датчика от одного элемента питания формата AA (пальчиковая батарейка) может составлять более одного года. Они обладают большой дальностью связи и высокой устойчивостью к помехам, обеспечивая прием сигнала до 10 км на открытой местности и до 3 км в условиях городского ландшафта и естественного для города уровня радиозашумления. В случае необходимости увеличить зону покрытия или повысить качество связи с датчиками в местах с высокой плотностью застройки можно разместить несколько клиентских устройств, разделяющих одну беспроводную сеть.

Одним из ограничений применения технологии LoRaWAN является значительное ухудшение качества приема сигнала при установке шлюза в непосредственной близости с базовыми станциями операторов сотовой связи [15], а также законодательный запрет на использование радиоаппаратуры, работающей в частотном диапазоне 868 МГц, на территории аэропортов.

Таким образом, одно из реализованных аппаратно-программных решений клиентской части системы, представленное в настоящей работе, является надежным, отказоустойчивым объектом и может быть воспроизведено читателем.

Тестирование и исследование предлагаемого решения

Для проверки работоспособности аппаратно-программного продукта выполнялось ручное и автоматизированное тестирование на всех этапах разработки (пример представлен на рис. 5). В ходе данного процесса использовалось модульное тестирование, написанное на технологии Testinfra. Задействовались docker-py и другие модули языка Python, подходящие для тестирования инфраструктуры. Использовалась связка Vagrant/Virtualbox и контейнеризации компонентов системы. Продукт полностью покрыт инфраструктурными тестами. Была осуществлена автоматизация их запуска: при каждом внесении изменений в репозиторий Gitlab запускались юнит-тесты, а при сборке релизов инициализировались интеграционные тесты.

```
============================= test session starts ==============================
platform linux -- Python 3.6.8, pytest-5.0.0, py-1.8.0, pluggy-0.12.0
rootdir: /home/         /workspace/radiolka/client
collected 1440 items

tests/api/test_helpers.py ..........                                     [  4%]
tests/api/test_http_base.py ..........                                   [  8%]
tests/api/test_http_rest_protocol.py ..........                          [ 13%]
tests/api/test_protobuf_base.py ..........                               [ 18%]
tests/api/test_protobuf_protocol.py ..........                           [ 24%]
tests/api/test_serializers.py ..........                                 [ 27%]
tests/modules/audio_playback/test_alsa_mixer_controller.py ..........   [ 31%]
tests/modules/audio_playback/test_codec.py ..........                    [ 37%]
tests/modules/audio_playback/test_player.py ..........                   [ 41%]
tests/modules/audio_playback/test_scheduler.py ..........                [ 47%]
tests/modules/camera/test_codec.py ..........                            [ 51%]
tests/modules/video_playback/test_codec.py ..........                    [ 52%]
tests/modules/video_playback/test_framebufer.py ..........              [ 56%]
tests/modules/video_playback/test_player.py ..........                   [ 59%]
tests/modules/video_playback/test_scheduler.py ..........                [ 62%]
tests/sensors/test_controller.py ..........                              [ 64%]
tests/sensors/test_noise.py ..........                                   [ 65%]
tests/sensors/test_processor.py ..........                               [ 69%]
tests/sensors/sensors/test_aux_1.py ..........                           [ 70%]
tests/sensors/sensors/test_aux_2.py ..........                           [ 72%]
tests/sensors/sensors/test_aux_3.py ..........                           [ 74%]
tests/sensors/sensors/test_carbon_dioxide.py ..........                  [ 79%]
tests/sensors/sensors/test_electromagnetic.py ..........                 [ 80%]
tests/sensors/sensors/test_heat.py ..........                            [ 86%]
tests/sensors/sensors/test_humidity.py ..........                        [ 90%]
tests/sensors/sensors/test_microphone.py ..........                      [ 96%]
tests/sensors/sensors/test_pressure.py ..........                        [100%]

========================== 1440 passed in 753.88 seconds ===========================
```

Рис. 5. Результат примера выполнения части юнит-тестов проекта с использованием фреймворка pytest

Далее производилось ручное экспертное тестирование всех компонентов системы со стороны автора и заказчиков в неблагоприятных погодных условиях города Новосибирска более года. Инструменты сетевого анализа и идентификации автомодельности сетевого трафика не идентифицировали подозрительную сетевую активность или закономерность в информационных потоках со скрытой маркировкой. Таким образом, реализованная техническая система соответствует всем заявленным требованиям, является надежным и отказоустойчивым решением, обеспечивающим высокий уровень информационной безопасности. При этом важно отметить, что при использовании сконфигурированного самоорганизующегося виртуального защищенного канала связи на основе стохастического многослойного шифрования и оверлейных технологий скорость сетевого взаимодей-

ствия может составлять 2 Мбит/с (цена за повышение уровня безопасности). В случае необходимости данный момент нивелируется переключением на стандартные технологии VPN.

Заключение

В рамках представленного проекта была спроектирована, разработана и аппаратно-программно реализована распределенная система мониторинга окружающей обстановки и информирования/оповещения населения в качестве объекта инфраструктуры «умного города». Использовался следующий стек технологий: язык программирования Python, веб-фреймворк Django, операционная система Alpine Linux, СУБД Postgresql, веб-сервер Nginx, виртуальные защищенные каналы связи (OpenVPN и др.), а также различные другие инструменты. Программная инженерия системы обеспечила ее надежное, безопасное и отказоустойчивое функционирование посредством технологий контейнеризации Docker и многослойной изоляции сервисов и потоков. Функционал системы включает информирование/оповещение граждан в режиме реального времени или проигрывание аудиотреков согласно установленному расписанию. Дополнительно выполняется сбор информации с различных типов проводных и беспроводных датчиков и их передача серверному модулю для последующей обработки и анализа. Через веб-интерфейс панели администрирования предоставляется возможность мониторинга и управления всеми функциями и объектами с произведением аналитики. На обзор вынесено одно из разработанных аппаратных решений клиентской части системы на базе одноплатного микрокомпьютера Raspberry Pi 3B, микроконтроллерной платы Arduino Micro с проводными датчиками, шлюза LoRaWAN с беспроводными датчиками и модемом 4G / LTE.

Научная новизна заключается в осуществлении дополнительной проверки подлинности управляющих воздействий на основе скрытой маркировки информационных потоков, автономно вычисляемой и производимой на стороне клиента и сервера через метаинформацию о предшествующих сетевых взаимодействиях (трассировке маршрута, командах, временных метках и т. д.). Используется функция свертки. Осуществляется преобразование данных произвольной длины в выходную битовую строку установленной размерности с последующим сигнатурным сопоставлением с таблицей действий, которые не мешают сетевой коммуникации и не детектируются инструментами автомодельности сетевого трафика.

Архитектурно предусмотрена возможность интеграции с авторскими наукоемкими решениями: системой интеллектуально-адаптивного управления сетевой инфраструктурой предприятия, а также модулем самоорганизующегося виртуального защищенного канала связи на основе стохастического многослойного шифрования и оверлейных технологий.

Рекомендовано и апробировано внедрение проекта от «умных транспортных остановок» до заводских и городских распределенных систем информирования/оповещения населения (в том числе о чрезвычайных ситуациях) с мониторингом окружающей обстановки.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Иващук О.А., Константинов И.С., Архипов О.П.* Подходы к созданию автоматизированной системы управления «умными фермами» // Вестник Орловского государственного аграрного университета. – 2012. – № 5 (38). – С. 15–20.
2. *Коськин А.В., Архипов О.П, Иващук О.А., Пилипенко О.В.* и др. Базовые принципы построения автоматизированной системы управления безопасным «умным городом» и механизмы их реализации // Строительство и реконструкция. – 2012. – № 2 (40). – С. 63–68.
3. *Архипов О.П., Иващук О.А., Константинов И.С., Коськин А.В. и др.* Рынок электронных услуг населению в России: проблемы и перспективы // Информационные ресурсы России. – 2011. – № 4 (122). – С. 2–5.
4. *Paiva S., Santos D., Rosaldo J.F.* A Methodological Approach for Inferring Urban Indicators Through Computer Vision // Proceedings of the 2018 IEEE International Smart Cities Conference (ISC2), Kansas City, MO, USA. 2018. P. 1–7.
5. *Munizaga M.A., Palma C.* Estimation of a disaggregate multimodal public transport origin-destination matrix from passive smartcard data from Santiago Chile // Transportation Research Part C: Emerging Technologies. – 2012. – Vol. 24. – P. 9–18.
6. *Moustaka V., Vakali A., Anthopoulos L.G.* A Systematic Review for Smart City Data Analytics // ACM Computing Surveys (CSUR). 2019. V. 51, no. 5. P. 1–41.
7. *Razumov D., Aleshin V.* Simulation modelling as a tool for design and development in large-scale automated systems: smart city application in terms of lack of statistical information // Advances in systems science and applications. 2018. V. 18, no. 3. P. 79–89.
8. *Французова Г.А., Гунько А.В., Басыня Е.А.* Самоорганизующаяся система управления трафиком вычислительной сети: метод противодействия сетевым угрозам // Программная инженерия. – 2014. – № 3. – С. 16–20.
9. *Басыня Е.А.* Система самоорганизующегося виртуального защищенного канала связи // Защита информации. Инсайд. – 2018. – № 5 (83). – С. 10–15.
10. *Басыня Е.А.* Распределенная система информирования и экстренного оповещения населения с применением беспроводной передачи данных // Вестник компьютерных и информационных технологий. – 2019. – № 1. – С. 37–47.
11. *Павлов И.Ю., Колосков В.Л., Иванов Е.Б.* Анализ централизованных и децентрализованных систем автоматизированного управления «интеллектуальным» домом // Новые информационные технологии в автоматизированных системах. – 2016. – № 19. – С. 338–340.
12. *Бирюков А.* Решения по централизованному управлению сетевыми устройствами // Системный администратор. – 2015. – № 11 (156). – С. 42–46.
13. *Зуев А.Н., Лукичев А.А.* Централизованная автоматизированная система управления инцидентами информационной безопасности для малых предприятий // Современные информационные технологии. теория и практика: Тр. IV всероссийской научно-практической конференции. – Череповец: ЧГУ, 2017. – С. 177–179.
14. *Басан Е.С., Михайлов Н.В.* Исследование, поиск и устранение уязвимостей для сети мобильных роботов с централизованным управлением // Математические методы и информационно-технические средства: Тр. XIV всероссийской научно-практической конференции. – Краснодар: КУ МВД РФ, 2018. – С. 26–28.
15. *Yang X., Karampatzakis E., Doerr C., Kuipers F.* Security Vulnerabilities in LoRaWAN // Proceedings of the IEEE/ACM Third International Conference on Internet-of-Things Design and Implementation (IoTDI), Orlando, FL, USA. 2018. P. 129–140.

Статья поступила в редакцию 2 августа 2019 г.

THE SMART CITY INFRASTRUCTURE OBJECT: A SYSTEM FOR ENVIRONMENTAL MONITORING AND PUBLIC INFORMATION / WARNING

E.A. Basinya

Novosibirsk State Technical University,
Research Institute of Information and Communication Technologies

20, Prospekt K. Marksa, Novosibirsk, 630073, Russian Federation

Abstract. *One of the possible tools to improve the living standards of citizens is automation of urban infrastructure management using information and communication solutions. An original system for environmental monitoring and public informing / warning as an infrastructure object of the smart city is reviewed. The design and software implementation of the client and server parts of the project based on the Alpine Linux operating system, the Python programming language, the Django web framework, and the Docker containerization technology are described. One of the implemented hardware and software client solutions using Wi-Fi, 4G / LTE and LoRaWAN wireless data transfer technologies is reviewed. The final consumers of this system are both government institutions and private enterprises. The methods of use are not strictly regulated, the implementation from smart transport stops to factory and city distributed public information / warning systems with environmental monitoring is recommended and tested. This work is an extension of the author's project of a distributed information and emergency public warning system. The scientific novelty lies in the implementation of additional authentication of control actions based on hidden marking of information flows, independently calculated and produced on the client and server side through meta-information about previous network interactions (trace route, commands, timestamps, etc.). To achieve this, the convolution function is used. Data of arbitrary length is converted into an output bit string of a set dimension, followed by signature matching with a table of actions that do not interfere with network communication and are not detected by network traffic self-similarity tools. The architecture provides for the possibility of integration with author's knowledge-intensive solutions: a system for intelligent adaptive management of the enterprise network infrastructure, as well as a self-organizing virtual secure communication channel module based on stochastic multilayer encryption and overlay technologies.*

Keywords: *Smart Cities, smart / intelligent objects, public information and warning systems, environmental monitoring systems, TCP / IP, wireless data transmission, Wi-Fi, 4G / LTE, LoRa, information flow management, information resources security.*

REFERENCES

1. *Ivashhuk O.A., Konstantinov I.S., Arxipov O.P.* Approaches to creating an automated management system for smart farms // Vestnik Orlovskogo gosudarstvennogo agrarnogo universiteta. 2012. no. 5 (38). Pp. 15–20 (in Russian).
2. *Kos`kin A.V., Arxipov O.P., Ivashhuk O.A., Pilipenko O.V. and oth.* Basic principles of building an automated control system for a safe "smart city" and mechanisms for their implementation // Stroitel`stvo i rekonstrukciya. 2012. № 2 (40). Pp. 63–68 (in Russian).
3. *Arxipov O.P., Ivashhuk O.A., Konstantinov I.S., Kos`kin A.V. and oth.* The market of electronic services to the population in Russia: problems and prospects // Informacionny`e resursy` Rossii. no. 4 (122). Pp. 2–5 (in Russian).
4. *Paiva S., Santos D., Rosaldo J.F.* A Methodological Approach for Inferring Urban Indicators Through Computer Vision // Proceedings of the 2018 IEEE International Smart Cities Conference (ISC2), Kansas City, MO, USA. 2018. Pp. 1–7.

Evgeny A. Basinya (Ph.D. (Techn.)), Associate Professor.

5. *Munizaga M.A., Palma C.* Estimation of a disaggregate multimodal public transport origin–destination matrix from passive smartcard data from Santiago Chile // Transportation Research Part C: Emerging Technologies. 2012. Vol. 24. Pp. 9–18.
6. *Moustaka V., Vakali A., Anthopoulos L.G.* A Systematic Review for Smart City Data Analytics // ACM Computing Surveys (CSUR). 2019. V. 51, no. 5. Pp. 1–41.
7. *Razumov D., Aleshin V.* Simulation modelling as a tool for design and development in large-scale automated systems: smart city application in terms of lack of statistical information // Advances in systems science and applications. 2018. V. 18, no. 3. Pp. 79–89.
8. *Franczuzova G.A., Gun`ko A.V., Basinya E.A.* Self-organizing computer network traffic management system: a method to counteract network threats // Programmnaya inzheneriya. 2014. no. 3. Pp. 16–20 (in Russian).
9. *Basinya E.A.* The System of Self-Organizing Virtual Secure Communication Channel // Zaŝita informacii. Inside. 2018. no. 5 (83). Pp. 10–15 (in Russian).
10. *Basinya E.A.* Distributed Public Alert and Warning System with the Use of Wireless Data Transmission / Vestnik komp`yuterny`x i informacionny`x texnologij. 2019. no. 1. Pp. 37–47 (in Russian).
11. *Pavlov I.Yu., Koloskov V.L., Ivanov E.B.* Analysis of centralized and decentralized systems of automated control of the "smart" house // Novy`e informacionny`e texnologii v avtomatizirovanny`x sistemax. 2016. № 19. C. 338–340.
12. *Biryukov A.* Centralized Network Device Management Solutions // Sistemny`j administrator. 2015. № 11 (156). C. 42–46.
13. *Zuev A.N., Lukichev A.A.* Centralized automated information security incident management system for small enterprises // Proceedings of the IV Conference on Modern information technology. theory and practice, Cherepovets: CSU, 2017. Pp. 177–179.
14. *Basan E.S., Mixajlov N.V.* Research, troubleshooting and vulnerabilities for a network of mobile robots with centralized management // Proceedings of the XIV Conference on Mathematical methods and information technology tools, Krasnodar: KU MVD RF, 2018. Pp. 26–28.
15. *Yang X., Karampatzakis E., Doerr C., Kuipers F.* Security Vulnerabilities in LoRaWAN // Proceedings of the IEEE/ACM Third International Conference on Internet-of-Things Design and Implementation (IoTDI), Orlando, FL, USA. 2018. Pp. 129–140.