

Общество с ограниченной ответственностью «Информационные горные технологии»

ОКПД2 26.30.11.150



Комплекс многофункциональной связи «СМС ИСЕТЬ»

Руководство пользователя

ИГТ.091000.000.00.000РП

Екатеринбург 2024

СОДЕРЖАНИЕ

| | | |
|-------|--|----|
| 1 | Назначение и область применения | 8 |
| 2 | Состав и описание комплекса | 9 |
| 2.1 | Возможности применяемой технологии | 9 |
| 2.2 | Аппаратное обеспечение | 9 |
| 2.2.1 | Сервер | 9 |
| 2.2.2 | Автоматизированные рабочие места | 10 |
| 2.3 | Программное обеспечение | 11 |
| 2.3.1 | Серверное ПО | 11 |
| 2.3.2 | Клиентское ПО..... | 13 |
| 3 | Клиентское программное обеспечение | 14 |
| 3.1 | Среда исполнения Java..... | 14 |
| 3.2 | Инженерный клиент..... | 15 |
| 3.2.1 | Внешний вид основного окна программы | 15 |
| 3.2.2 | Настройка устройств | 16 |
| 3.2.3 | Обновление встроенного ПО..... | 18 |
| 3.2.4 | Модель шахты | 19 |
| 3.2.5 | Отчеты | 20 |
| 3.2.6 | Настройка | 21 |
| 3.3 | АРМ Ламповщик | 23 |
| 3.3.1 | Внешний вид..... | 23 |
| 3.3.2 | Подготовка к использованию программы | 24 |
| 3.3.3 | Выдача светильников | 25 |
| 3.3.4 | Прием светильника..... | 26 |
| 3.3.5 | Временная замена светильника..... | 26 |
| 3.3.6 | Установка и настройка | 27 |
| 3.3.7 | NFC Reader | 30 |
| 3.4 | ПО Вентиляция 2..... | 33 |
| 4 | Серверное программное обеспечение..... | 34 |
| 4.1 | Установка и настройка ОС..... | 34 |
| 4.1.1 | Установка операционной системы | 34 |
| 4.1.2 | Базовая настройка операционной системы | 34 |
| 4.1.3 | Сервер DHCP | 37 |
| 4.2 | Установка серверного ПО | 40 |

| | | |
|-------|--|----|
| 4.2.1 | Установка сервера Voiloc | 40 |
| 4.2.2 | Настройка сервера Voiloc | 42 |
| 4.2.3 | Настройка журналов Voiloc..... | 60 |
| 4.2.4 | Дополнительные настройки системы | 62 |
| 4.2.5 | Резервирование сервера Voiloc..... | 62 |
| 4.2.6 | Создание долговременных архивов и чистка БД..... | 66 |

Настоящее Руководство пользователя (РП) содержит техническое описание программной части Комплекса системы многофункциональной связи Исеть (далее по тексту – СМС Исеть, Комплекс).

Настоящее Руководство распространяется на все возможные исполнения (модификации) Комплекса. При необходимости, для каждого исполнения (модификации) выпускается Дополнение к настоящему Руководству, включаемое в ведомость эксплуатационных документов для данного исполнения (модификации).

Цель настоящего Руководства заключается в предоставлении информации, необходимой для эксплуатации Комплекса по назначению.

Аппаратная часть Комплекса рассматривается в документе ИГТ.091000.000.00.000РЭ СМС Исеть Руководство по эксплуатации, руководство пользователя по ПО АРМ «Оператор шахты» рассматривается в документе ИГТ.091000.401.00.000РП АРМ Оператор шахты.

Термины, определения и сокращения

В настоящем документе применяют нижеперечисленные термины.

Эксплуатационные документы, паспорт (ПС) – документ по ГОСТ 2.601-2006.

Руководство по эксплуатации (РЭ) – документ по ГОСТ 2.601-2006 и ГОСТ 2.610-2006.

Структурная схема – документ по ГОСТ 2.701-84.

Техническое обслуживание (ТО) – комплекс операций или операция по поддержанию работоспособности или исправности системы и его составных частей при использовании ее по назначению, при хранении и транспортировании.

Запасная часть – составная часть системы, предназначенная для замены находившейся в эксплуатации такой же части с целью поддержания работоспособности или восстановления исправности системы.

Хранение – содержание используемого по назначению системы и ее составных частей в заданном состоянии в отведенном для его размещения месте с обеспечением сохранности в течение заданного срока.

Транспортирование - перемещение системы и ее составных частей в заданном состоянии с применением, при необходимости, транспортных и грузоподъемных средств, начинающееся с погрузки и кончающееся разгрузкой на месте назначения.

Автоматизированное рабочее место оператора, диспетчера (АРМ оператора, диспетчера) – ЦЭВМ (наземный компьютер), на которой отображается информация, получаемая от ШТД-РТД, ИКН, МИВ, СТП, положение ИКН, МИВ и СТП на мнемосхемах и с которой осуществляется диспетчерская голосовая связь с ИКН.

Сервер – ЦЭВМ (компьютер), которая обеспечивает обмен данными с ШТД-РТД, ИКН, МВИ, СТП, осуществляет обработку и хранение получаемой от них информации, подготавливает результаты локального позиционирования, обеспечивает доступ с АРМ к текущим и архивным данным.

Медиаконвертеры (МК) – устройства, обеспечивающие преобразование вида сигнала для различных сред передачи (медный проводник, оптоволокно, радиоэфир, ...).

Барьеры искробезопасности (БИ) – устройства, обеспечивающие гальваническое разделение искроопасных и искробезопасность электрических цепей.

Источники питания (ИП) – искробезопасные источники питания с аккумуляторной поддержкой для электропитания элементов системы.

Высокоскоростная магистраль – линия связи, формируемая устройствами СПИН между сервером и ШТД-РТД.

Ethernet 10BaseT – это обозначение искробезопасной версии Ethernet 10BaseT, которая используется в устройствах СПИН и ШТД-РТД и которая может не обеспечивать полной совместимости с устройствами с интерфейсом Ethernet 10T других производителей;

Ethernet 10BaseT, Ethernet 100BaseTX – технология Ethernet на основе использования медных витых пар в качестве среды передачи информации.

Ethernet 100BaseFX – технология Ethernet на основе использования оптоволокна в качестве среды передачи информации.

Локальная связь – голосовая связь только в зоне ведения горных работ без выхода на диспетчера.

Местная голосовая связь – голосовая связь в зонах ведения горных работ с выходом на диспетчера.

Глобальная голосовая связь – местная голосовая связь с выходом в телефонные сети общего пользования.

Локальное позиционирование – определение положения работника, транспортного средства (другого подвижного оборудования), средств КИПиА, оборудованных радиоустройствами, относительно неподвижных точек в фиксированных местах горных выработок;

Радиосегмент – совокупность горных выработок, в которых расположены ретрансляторы ШТД-РТД, которые образуют непрерывную цепочку устройств, взаимодействующих друг с другом в радиоэфире с единым трафиком. Оконечным устройством радиосегмента является шлюз ШТД-РТД. Один радиосегмент образует простейшую радиозону.

Радиогруппа (канал ИКН) – совокупность интеркомов с единым голосовым трафиком. ИКН одной радиогруппы могут находиться в различных радиосегментах. Один интерком может быть включен в несколько радиогрупп;

VoilocPhone (софт-телефон) – прикладное ПО, функционирующее на АРМ диспетчера и служащее для связи с ИКН.

Инженерный клиент (ИК) – прикладное ПО, функционирующее на АРМ, предназначенное для настройки и диагностики устройств СМС Исеть.

Редактор связей и справочников (Линковщик) – прикладное ПО, функционирующее на АРМ и служащее для создания списков должностей, подразделений, сотрудников, светильников и связей между ними.

АРМ Ламповщика – прикладное ПО, функционирующее на АРМ, предназначенное для учета и управления выдачей и приемом светильников в ламповой.

Базовая станция (БС), точка доступа (ТД) – устройство ШТД-РТД, подключенное к сети СМС Исеть.

Геосегмент – номер радиосегмента, присвоенный всем входящим в него базовым станциям через Инженерный клиент, служит для организации трафика между радиосегментами.

В настоящем документе используются нижеперечисленные сокращения.

Система – система многофункциональной связи «ИСЕТЬ».

АРМ – автоматизированное рабочее место.

АСОДУ – автоматизированная система оперативно-диспетчерского управления.

БД – база данных.

ЗИП – запасные части, инструменты, принадлежности и материалы.

ИП – источники питания (ШИП).

КД – конструкторская документация.

КИ – конвертер интерфейса.

КИПиА – контрольно-измерительные приборы и автоматика.

ПО – программное обеспечение.

ПС – паспорт.

УБП – устройство бесперебойного питания.

РЭ – руководство по эксплуатации.

РП – руководство пользователя.

СПИН – система передачи информации.

ТО – техническое обслуживание.

ШИП, ШИП-И1 – шахтный источник питания.

ЦЭВМ – цифровая электронная вычислительная машина (компьютер).

ЯСУ – Ящик монтажный ЯСУ-XX.Y.ZZ.

ШТД – шлюз-точка доступа (ШТД-РТД в режиме шлюза).

РТД – ретранслятор-точка доступа (ШТД-РТД в режиме ретранслятора).

ИКН – интерком носимый.

ПУС – подземный узел связи СПИН 0020-КН04.11.1.

НУС – наземный узел связи СПИН 0020-КН04.21.3.

БА – блок аккумуляторный.

РД – рабочая документация на внедрение СМС «ИСЕТЬ».

АСН – анализатор сети носимый.

МИВ – метка интеллектуальная встраиваемая.

СТП – светильник точного позиционирования.

1 Назначение и область применения

СМС «ИСЕТЬ» — автоматизированная система многофункциональной связи и позиционирования предназначена для:

1. обеспечения беспроводной голосовой связью работников, находящихся в горных выработках, между собой, с диспетчером на поверхности;
2. определения положения работников, транспорта, подвижного оборудования, контрольно-измерительных приборов и средств автоматики в подземных выработках шахт и рудников;
3. передачи данных между электронными системами работников, транспорта, подвижного оборудования, средств КИПиА и диспетчерской шахты и формирования резервного канала аварийного оповещения.

Комплекс СМС Исеть разработан на базе технологии Voice & location (далее по тексту — Voiloc).

Технология Voiloc обеспечивает инфраструктуру радиопокрытия, локальное позиционирование, передачу голосовых данных и телеметрии.

2 Состав и описание комплекса

2.1 Возможности применяемой технологии

Технология «Voiloc» реализует следующие базовые возможности, используемые в ПАК «СМС Исеть»:

1. Многоканальная голосовая связь между находящимися в зоне радиопокрытия ИКН между собой, с пультами диспетчеров или ПО «Голосовой клиент».
2. Устойчивость функционала голосовой связи к отказу сервера. При отключении сервера подсистема голосовой связи «СМС Исеть» запоминает последнюю рабочую конфигурацию и продолжает функционировать без возможности изменения настроек вплоть до отключения электропитания базовых станций, обеспечивающих радиопокрытие.
3. Запись разговоров и их хранение на сервере.
4. Определение местоположения ИКН и меток МИВ внутри зоны радиопокрытия, предоставление информации об их расположении средствами API, хранение истории местоположений на сервере.
5. Регистрация событий с различных датчиков с устройств, расположенных в области радиопокрытия, хранение истории событий на сервере.
6. Средства диагностики инфраструктуры радиопокрытия.
7. Предоставление API для интеграции с внешними системами.

2.2 Аппаратное обеспечение

Программное обеспечение Voiloc устанавливается на Сервер, на котором устанавливается ядро системы Voiloc, и автоматизированные рабочие места (АРМ), на которых устанавливаются специальные утилиты для доступа к различным функциям Комплекса.

2.2.1 Сервер

Сервер — ЭВМ, предназначенная для выполнения на ней сервисного программного обеспечения СМС «Исеть», управляющего всей инфраструктурой радиопокрытия (аппаратной частью Комплекса). В качестве сервера могут использоваться сертифицированные компьютеры промышленного или общего исполнения, совместимые с архитектурой IBM PC, основные требования к которым приведены в таблице 1. Питание сервера осуществляется от устройств бесперебойного питания.

Таблица 1 - Основные требования к серверу

| Параметр | Значение |
|---|--|
| Процессор, не хуже | Intel Xeon 4*Core, 2ГГц |
| Объём оперативной памяти, ГБ, не менее | 16 ¹ |
| Объём дисковой подсистемы, ГБ, не менее | 1000 ² |
| Количество сетевых адаптеров Ethernet, не менее | 1 |
| Совместимость с GNU/Linux, версия | Rosa Fresh Server 12.4; Astra Linux CE 2.12; Debian 9.0 и выше |
| Совместимость с системой виртуализации, версия | VMware ESXi 5.0 и выше; Proxmox VE 5.4 и выше |

2.2.2 Автоматизированные рабочие места

АРМ – ЭВМ, предназначенная для выполнения клиентских программ Комплекса, таких как: АРМ Диспетчера, АРМ Ламповщика, АРМ Инженера и других. В качестве АРМ могут использоваться сертифицированные компьютеры промышленного или общего исполнения, совместимые с архитектурой IBM PC, основные требования к которым приведены в таблице 2. Питание АРМ осуществляется от устройств бесперебойного питания.

Таблица 2 – Основные требования к АРМ

| Параметр | Значение |
|---|---|
| Процессор, не хуже | Intel Core i5, 2ГГц |
| Объём оперативной памяти, ГБ, не менее | 4 |
| Объём дисковой подсистемы, ГБ, не менее | 250 |
| Количество сетевых адаптеров Ethernet, не менее | 1 |
| Совместимость с GNU/Linux, версия | Debian 11.0 и выше; Rosa Fresh LXQt 12.4 |
| Совместимость с Windows версия | 7 и выше |
| Разрешение монитора | не менее 2560x1440 ³ |

¹Рассчитывается на этапе проектирования и зависит от количества устройств в составе СМС «Исеть»

²Рассчитывается на этапе проектирования и зависит от требований к хранению данных в СМС «Исеть», для систем позиционирования рекомендуется использовать SSD, каждая 1000 сотрудников дают до 100 Гб данных на диске в сутки

³Для АРМ Ламповщика допускается меньшее разрешение, для ПО Оператор шахты рекомендуется 2 использовать монитора

2.3 Программное обеспечение

В Комплекс СМС «Исеть» входят следующие виды программного обеспечения:

- серверное ПО, устанавливаемое на сервер;
- клиентское ПО, устанавливаемое на рабочие станции;
- стороннее ПО;
- микропрограммное обеспечение, работающее на устройствах, входящих в аппаратную часть Комплекса.

Настоящее Руководство рассматривает только серверное и клиентское ПО, аппаратная часть комплекса рассматривается в документе ИГТ.091000.000.00.000РЭ СМС Исеть Руководство по эксплуатации. Стороннее ПО описано в части взаимодействия с Комплексом, полную инструкцию на стороннее ПО необходимо получать у его производителя.

Структура программного обеспечения и диаграмма потоков данных Комплекса СМС «Исеть» приведена на рисунке 1.

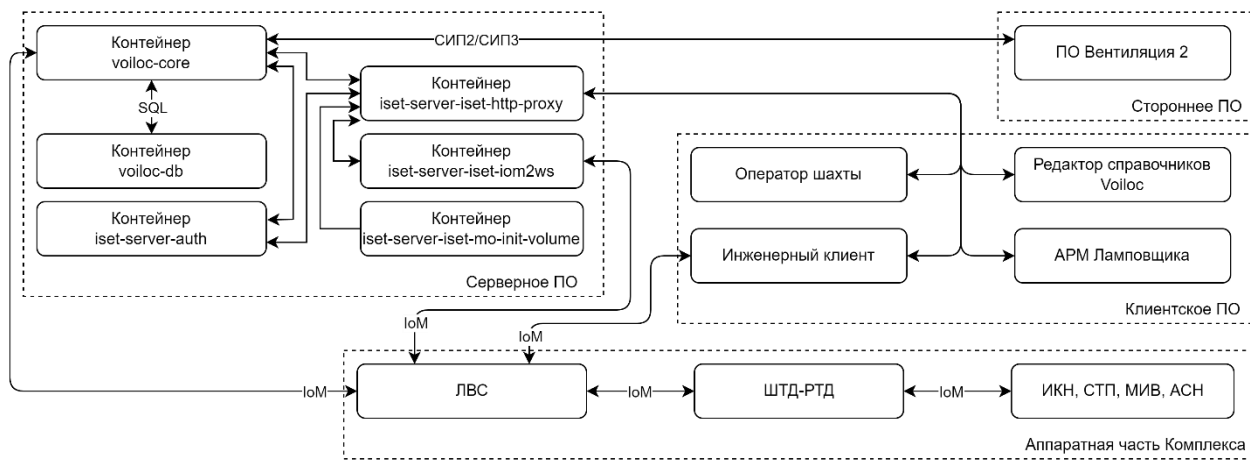


Рисунок 1 Структура ПО и диаграмма потоков данных

2.3.1 Серверное ПО

Серверное ПО — специализированное сервисное ПО, принимающее и обрабатывающее запросы от клиентского ПО с целью предоставления услуг по управлению комплексом «СМС Исеть».

Серверное ПО поставляется в виде набора контейнеров Docker:

1. voiloc-core — совокупность программных компонент, реализующих управление инфраструктурой радиопокрытия СМС «ИСЕТЬ», основной модуль системы.
2. voiloc-db — СУБД, предназначенная для хранения данных СМС «Исеть».
3. iset-server-iset-iom2ws – контейнер, который позволяет организовать аудиосвязь в клиентском ПО Оператор шахты.
4. iset-server-auth – система авторизации.
5. iset-server-iset-mo-init-volume – контейнер, содержащий web-версию ПО Оператор шахты.
6. iset-server-iset-http-proxy – единая точка входа для http-трафика клиентского ПО.

Структура подсистемы управления контейнера voiloc-core приведена на рисунке 2.

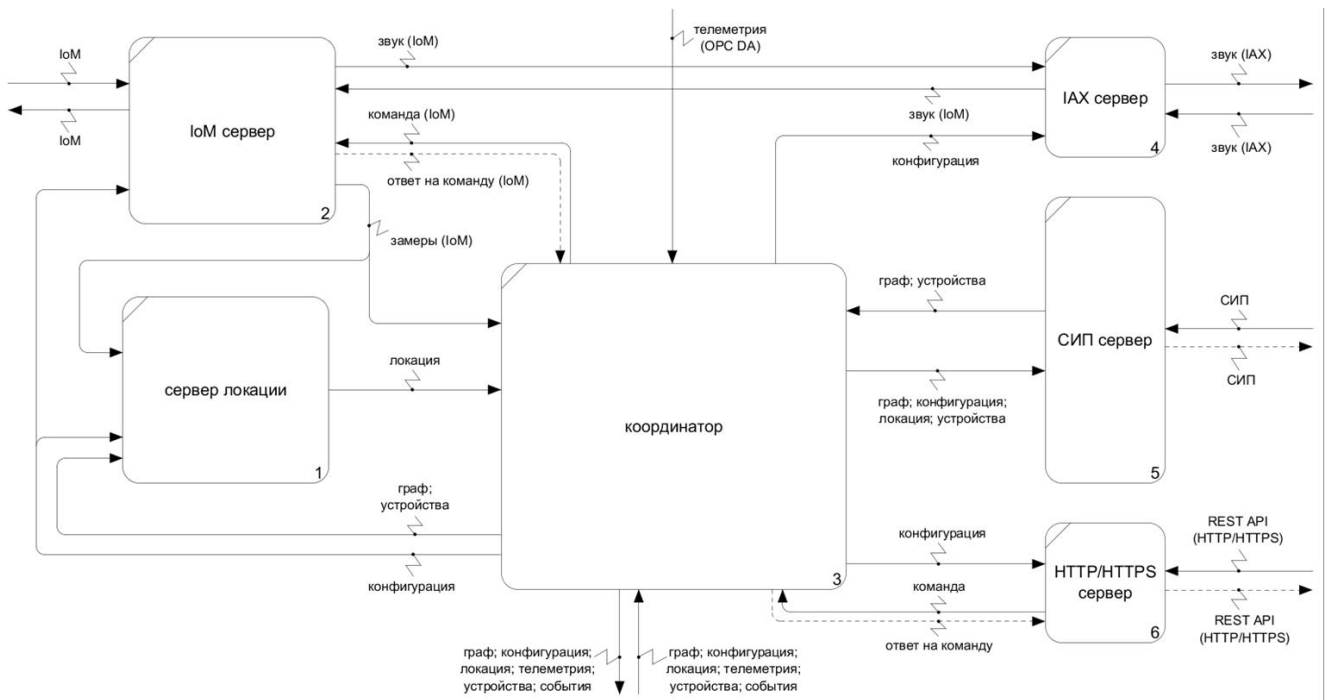


Рисунок 2 Структура серверного ПО, подсистема управления контейнера voiloc-core

Функциональное назначение компонент серверного ПО:

1. IoM-сервер — компонент, отвечающий за реализацию протокола взаимодействия аппаратного обеспечения в составе инфраструктуры радиопокрытия СМС «ИСЕТЬ» на стороне сервера, отвечает за обработку данных, связанных с локальным позиционированием и телеметрией.
2. Voip-сервер — компонент, отвечающий за реализацию протокола приёма голосовых данных СМС «ИСЕТЬ».
3. Сервер локации — компонент, реализующий алгоритмы расчёта локации по графу на основе данных от IoM-сервера и СИП-сервера.
4. СИП-сервер — компонент, отвечающий за реализацию протокола обмена данными с ПО «Вентиляция 2» (параметры конфигурации местоположения базовых станций, представление локации устройств).
5. HTTP/HTTPS-сервер — компонент, отвечающий за реализацию REST API и GraphQL, используемых для подключения клиентского ПО.
6. Координатор — компонент, отвечающий за взаимодействие всех компонент и регламентирующий целевое использование аппаратных ресурсов сервера компонентами серверного ПО.

Подсистема хранения в контейнере voiloc-db и построена на базе СУБД PostgreSQL 9.6, доступ к базе данных осуществляется через сервер Voiloc, средствами, предусмотренными REST API и GraphQL, прямой доступ к данным запрещён.

Работа серверного ПО обеспечивается средой Java Runtime Environment 8, установленной внутри контейнера voiloc-core.

2.3.2 Клиентское ПО

Клиентское ПО — кроссплатформенное программное обеспечение, взаимодействующее с сервером и устройствами по сети, устанавливаемое на рабочие места.

Работа клиентского ПО, описанного в данном руководстве, обеспечивается средой Java Runtime Environment 8.

Руководство к ПО Оператор шахты и Редактор справочников Voiloc находится в документе ИГТ.091000.401.00.000РП АРМ Оператор шахты.

Взаимодействие клиентского ПО с устройствами и сервером СМС Исеть осуществляется следующими путями (указаны настройки по умолчанию, конкретные порты и адреса задаются в конфигурационном файле сервера):

- 1) REST API, HTTP протокол, реализованный на сервере, доступен по адресу:
http://<IP сервера>:8090/voiloc;
- 2) ЮМ, собственный протокол СМС Исеть, реализованный поверх мультикаст группы 239.239.101.101 через порт 6050;
- 3) прямой доступ к ШТД для обновления встроенного ПО реализован по IP адресу ШТД через порт 1234.

3 Клиентское программное обеспечение

Клиентское ПО поставляется в виде архивов, не требующих инсталляции, содержащих исполняемый и конфигурационные файлы.

Установка заключается в получении и установке среды исполнения Java, извлечении файлов из архива в целевой каталог, внесения необходимых изменений в конфигурационные файлы. У пользователя, использующего клиентское ПО, должны быть права на запись в соответствующий каталог.

Для запуска используется исполняемый файл с расширением exe в Windows (или jar в Linux).

3.1 Среда исполнения Java

Все клиенты в составе «СМС Исеть», рассмотренные в данном руководстве (кроме ПО «Вентиляция 2»), являются кроссплатформенными и могут быть запущены на платформах Windows и Linux.

Кроссплатформенность обеспечивает среда Java Runtime Environment (JRE) 8, которую выпускает несколько производителей:

- 1) Oracle Java, для коммерческого использования требуется платная подписка, <https://www.java.com/ru/download/manual.jsp>;
- 2) Amazon Corretto, бесплатно для коммерческого использования <https://docs.aws.amazon.com/corretto/latest/corretto-8-ug/downloads-list.html>, для Windows рекомендуется устанавливать версию JDK;
- 3) для Debian в репозитории доступен пакет `openjdk-8-jre`, бесплатный для коммерческого использования.
- 4) в ОС Rosa Fresh 12 требуется установить пакеты `java-1.8.0-openjdk` и `java-1.8.0-openjdk-openjfx`

Инсталляция среды JRE производится в соответствии с рекомендациями производителя.

3.2 Инженерный клиент

Инженерный клиент предназначен для инженерного персонала, ответственного за пусконаладку и сопровождение Комплекса, и реализует инструменты управления и мониторинга «СМС Исеть».

Основной функционал программы:

- 1) настройка параметров конфигурации радиопокрытия;
- 2) настройка параметров аппаратного обеспечения;
- 3) мониторинг состояния радиопокрытия в режиме реального времени;
- 4) мониторинг параметров аппаратного обеспечения;
- 5) предоставление инструментов визуализации результатов аналитической обработки данных серверным ПО в виде отчётов и графиков;
- 6) сопоставление точек доступа с моделью пространства, полученной СИП-сервером;
- 7) предоставление средств обновления микропрограмм аппаратного обеспечения подземной части комплекса.

3.2.1 Внешний вид основного окна программы

Внешний вид основного окна программы приведён на рисунке 3.

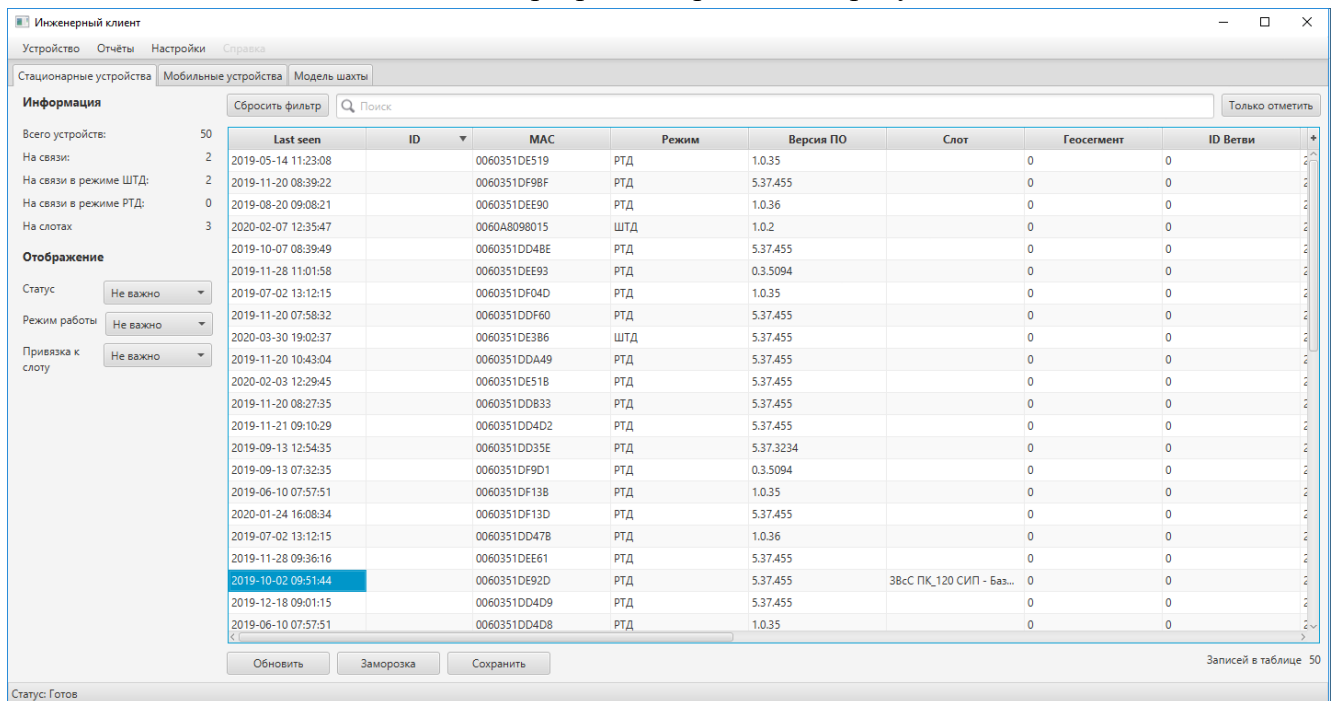


Рисунок 3 Инженерный клиент

Программа отображает радиоустройства СМС «ИСЕТЬ» в виде таблицы, окно программы поделено на вкладки «Стационарные устройства» и «Мобильные устройства». Стационарные устройства – это точки доступа (ШТД-РТД) СМС «ИСЕТЬ», мобильные устройства – это интеркомы носимые (ИКН), метки (СТП, МВИ) и анализаторы (АСН) СМС «ИСЕТЬ». Вкладка «Модель шахты» предоставляет справочную информацию о загруженной модели из ПО «Вентиляция 2».

Инженерный клиент позволяет производить сортировку и фильтрацию выводимых устройств, для удобства их администрирования. Также можно добавлять и удалять из таблицы

столбцы. Отсортированную и отфильтрованную таблицу можно экспортировать в CSV формат.

Главное окно программы позволяет добавлять и удалять отображаемые столбцы для стационарных и мобильных устройств, в зависимости от типа устройства доступны следующие столбцы:

Таблица 3 – Параметры устройств, отображаемые в главном окне Инженерного клиента

| Название | Отображаемая информация |
|----------------|---|
| Last seen | Время последнего выхода на связь |
| ID | Идентификатор устройства, задаваемый пользователем в свободной форме |
| MAC | MAC адрес устройства |
| Режим | Режим работы устройства (для стационарных устройств) |
| Версия ПО | Версия встроенного ПО |
| Слот | Место установки, заданное через ПО «Вентиляция 2», к которому привязано стационарное устройство |
| Геосегмент | Геосегмент, присвоенный стационарному устройству |
| ID Ветви | Идентификатор ветви, на которой установлено стационарное устройство |
| Мощность | Мощность передатчика стационарного устройства |
| IAA | Период отправки в микросекундах Alive пакетов устройством |
| Статус | Статус устройства (наличие связи) |
| Доступность, % | Доступность устройства (% полученных Alive пакетов за последние 10 минут) |
| Uptime | Время непрерывной работы устройства |
| DC-DC | Напряжение на плате DC-DC преобразователя ШТД-РТД |
| MCU | Напряжение на процессоре ШТД-РТД |
| IP | IP адрес стационарного устройства |
| АКБ | Наличие аккумулятора у стационарного устройства |
| Носитель | Носитель, к которому привязано мобильное устройство |
| Уровень заряда | Уровень заряда аккумулятора у мобильного устройства |

3.2.2 Настройка устройств

Для доступа к настройкам устройства необходимо дважды кликнуть на устройстве в таблице основного окна, либо выделить строку в таблице и зайти в меню «Устройство/Настроить параметры». Пример окна настройки устройства приведен на рисунке 4.

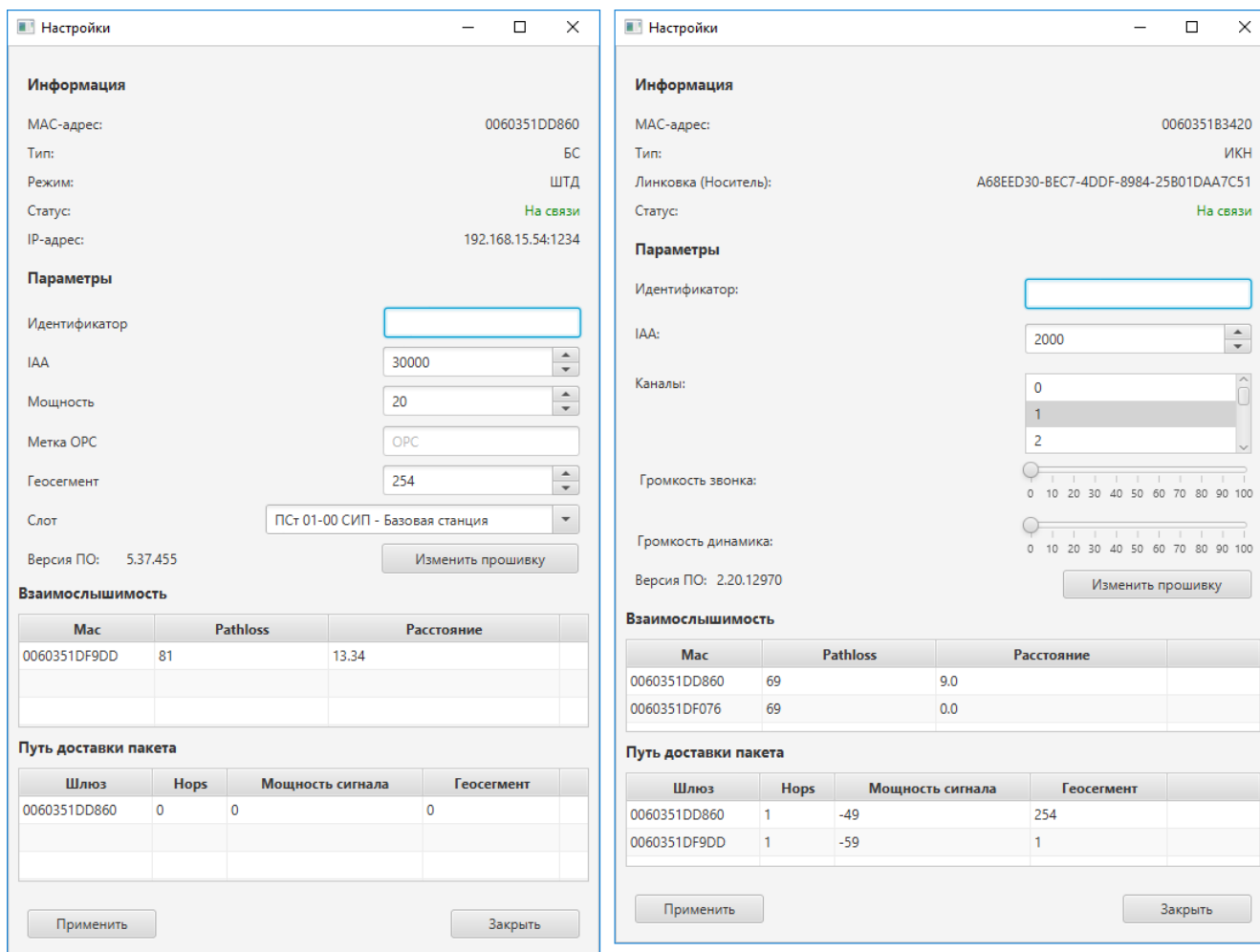


Рисунок 4 Окно настройки устройства (слева – стационарное устройство, справа – носимое)

В окне настройки можно изменить следующие параметры устройства:

Таблица 4 – Параметры, отображаемые в окне настройки устройства

| Название | Отображаемая информация |
|---------------|---|
| Идентификатор | Задание идентификатора устройства в свободной форме, в качестве идентификатора может служить любая уникальная строка, состоящая из латинских и русских букв и цифр, например, инвентарный номер |
| IAA | Установка период отправки в микросекундах Alive пакетов устройством |
| Мощность | Установка мощность передатчика для стационарных устройств |
| Метка OPC | Задание OPC тэга для управления электропитанием стационарных устройств, при использовании ШИП-И1, подключенных к СМС Исеть |
| Геосегмент | Задание номера геосегмента для стационарных устройств |
| Слот | Привязка стационарного устройства к месту установки, загруженному вместе со схемой из ПО «Вентиляция 2» |
| Версия ПО | Версия встроенного ПО, при нажатии на кнопку «Изменить прошивку» откроется диалог обновления встроенного ПО |

| | |
|--------------------|--|
| Каналы | Установка списка разрешенных каналов для ИКН, для выбора более одного канала необходимо зажать клавишу «Ctrl» и отметить левой кнопкой мыши разрешенные каналы |
| Громкость звонка | Задание громкости звонка для индивидуальных вызовов на ИКН |
| Громкость динамика | Задание громкости разговорного динамика на ИКН |

Примечание: параметры «IAA» и «Мощность» используются только при проведении пусконаладочных работ и испытаний.

Также окно настройки содержит таблицу «Взаимослышимость», которая дает возможность посмотреть уровень связи выбранного устройства с соседними ШТД/РТД, и таблицу «Путь доставки пакета», в которой можно увидеть шлюз, через который поступают пакеты от выбранного устройства на сервер и количество ретрансляций по пути от устройства к шлюзу. Данную информацию можно использовать для анализа работы радиосети и поиска мест с большими потерями в эфире.

3.2.3 Обновление встроенного ПО

При нажатии на кнопку «Обновление прошивки» в окне настроек устройства откроется окно обновления встроенного ПО устройства (Рисунок 5).

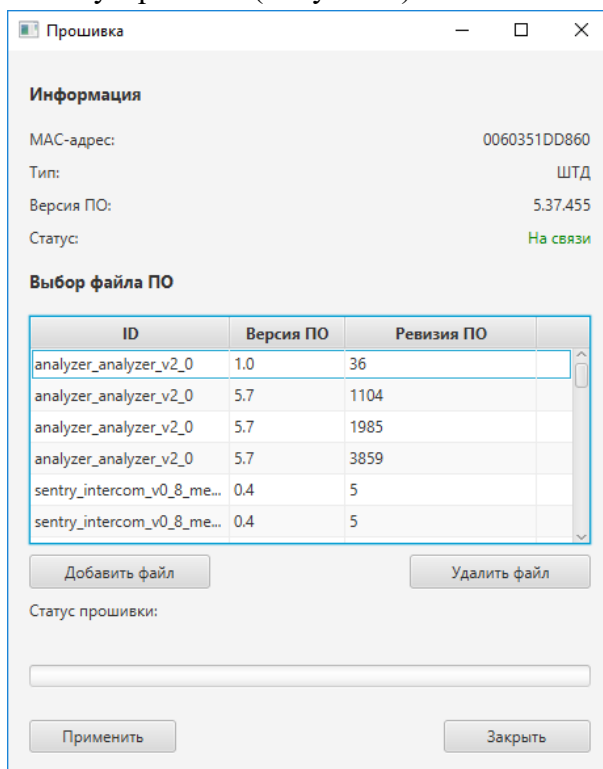


Рисунок 5 Окно обновления встроенного ПО

Для нормальной работы данной функции необходимо чтобы компьютер, с которого производится обновление, был в одной подсети с устройствами СМС Исеть. Обновлять можно только те устройства, которые в данный момент находятся на связи. Для запуска процедуры обновления необходимо выбрать левой кнопкой мыши в таблице нужную версию ПО и нажать кнопку «Применить», после чего в нижней части экрана появится индикатор процесса обновления. Длительность процесса обновления ПО зависит от уровня связи с устройством (потери сигнала на каждой ретрансляции в цепочке от сервера к устройству).

Для добавления в таблицу других версия встроенного ПО необходимо нажать кнопку «Добавить файл» и в стандартном диалоге операционной системы выбрать нужный файл с ПО. Для удаления из таблицы необходимо выбрать в таблице версию ПО и нажать на кнопку «Удалить файл».

В случае неудачного завершения процесса обновления ПО можно повторить попытку через 1-2 минуты.

Для того, чтобы записать в память Анализатора встроенное ПО для других устройств СМС Исеть необходимо в Анализаторе зайти в менеджер прошивок и из Инженерного клиента «обновить» встроенное ПО Анализатора программным обеспечением от нужного типа устройства.

3.2.4 Модель шахты

Программа позволяет посмотреть выгруженную из ПО «Вентиляция 2» модель шахты на вкладке основного окна «Модель шахты» (Рисунок 6).

| ID | Имя ветви | Длина |
|--------|------------------------|--------|
| 42402 | Конь. штрек 2В | 210.98 |
| 42393 | Вент. штрек 1В | 220.8 |
| 42401 | Конь. штрек 2В | 145.05 |
| 42392 | Вент. штрек 1В | 125.07 |
| 41045 | Вентиляционный штр... | 259.34 |
| 42345 | Конь. штрек 2В | 224.99 |
| 42336 | Вент. штрек 1В | 239.99 |
| 48598 | Дренажный штрек 10Л | 22.88 |
| 41769 | Дренажный квершлаг | 35.25 |
| 49744 | Вентиляционный штр... | 254.75 |
| 303860 | Сбойка №1 | 24.96 |
| 46315 | Конвейерный штрек 1... | 24.72 |
| 303877 | Маг. пут. штрек | 170.22 |
| 303532 | Конвейерный штрек 6Л | 120.0 |
| 42146 | Вент. штрек. 2В | 161.24 |
| 303743 | Вент. штрек 6Л | 170.0 |
| 42810 | Вент. штрек 3В | 171.68 |
| 42035 | Вент. штрек 8В | 26.44 |
| 304333 | Путевой ствол | 308.65 |
| 303438 | Конь. штрек 7Л | 2.1 |
| 304330 | Конвейерный ствол | 57.79 |
| 304140 | Южный фланговый ук... | 10.02 |

Рисунок 6 Модель шахты

На данной вкладке можно получить следующую информацию:

Таблица 5 – Информация, отображаемая на вкладке "Модель шахты"

| Название | Отображаемая информация |
|-----------|--|
| ID | Внутренний идентификатор выработки, присвоенный ей в ПО «Вентиляция 2» |
| Имя ветви | Название выработки |
| Длина | Длина участка выработки (ветвь) в метрах |

Механизм работы данной вкладки аналогичен механизму работы вкладок «Стационарные устройства» и «Мобильные устройства».

При двойном клике по таблице появится окно с таблицей загруженных из ПО «Вентиляция 2» слотов (Рисунок 7):

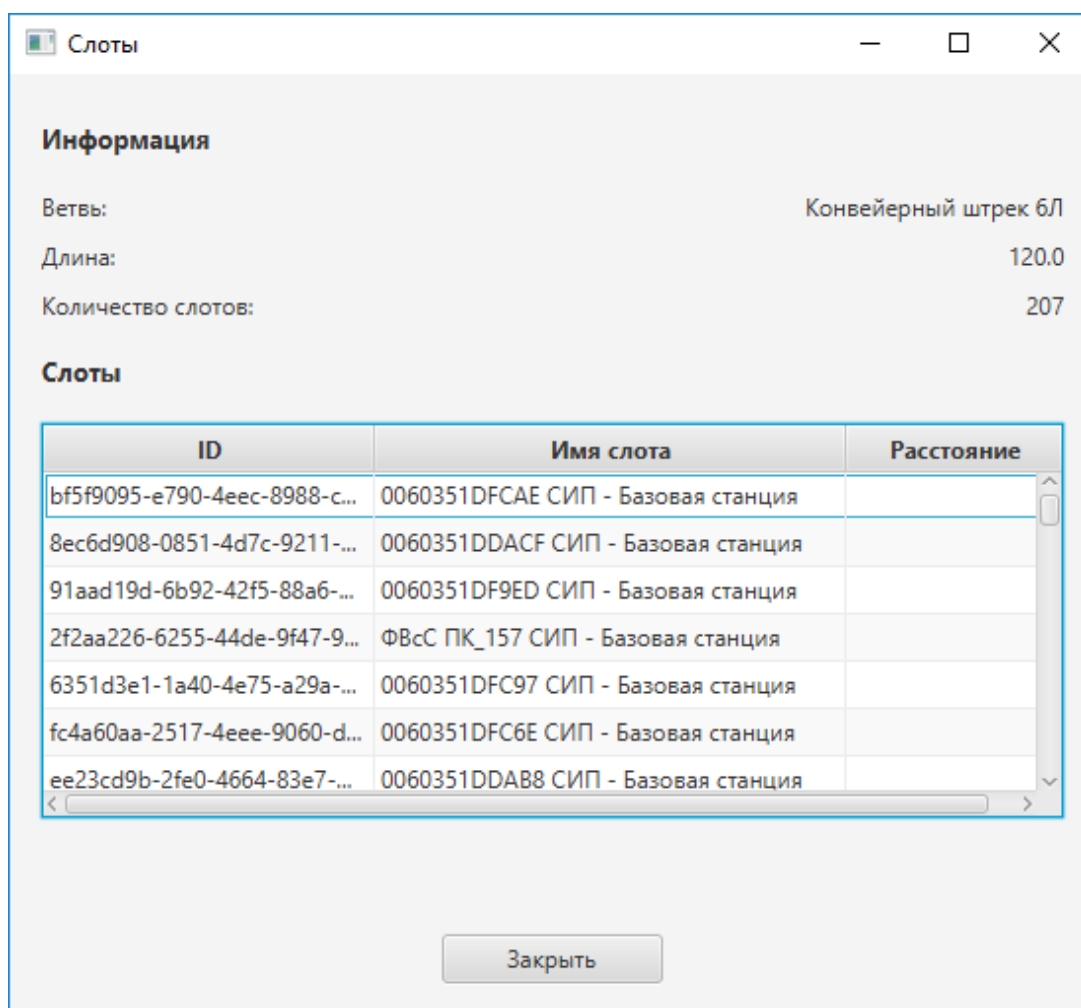


Рисунок 7 Информация о слотах

3.2.5 Отчеты

Программа позволяет создавать различные отчеты о работе системы. Доступны как табличные отчеты, так и графические. Для создания отчета необходимо зайти в меню «Отчеты» и выбрать тип отчета (табличный или графический). После выбора типа отчета необходимо выбрать устройство по которому будет создан отчет, временной промежуток, за который необходимо получить информацию и параметр, изменение значения которого будет представлено в отчете. Для отчетов доступны следующие параметры:

- давление;
- температура;
- потери пакетов;
- внешнее питание;
- внутреннее питание.

Внешний вид графического отчета представлен на Рисунке 8:

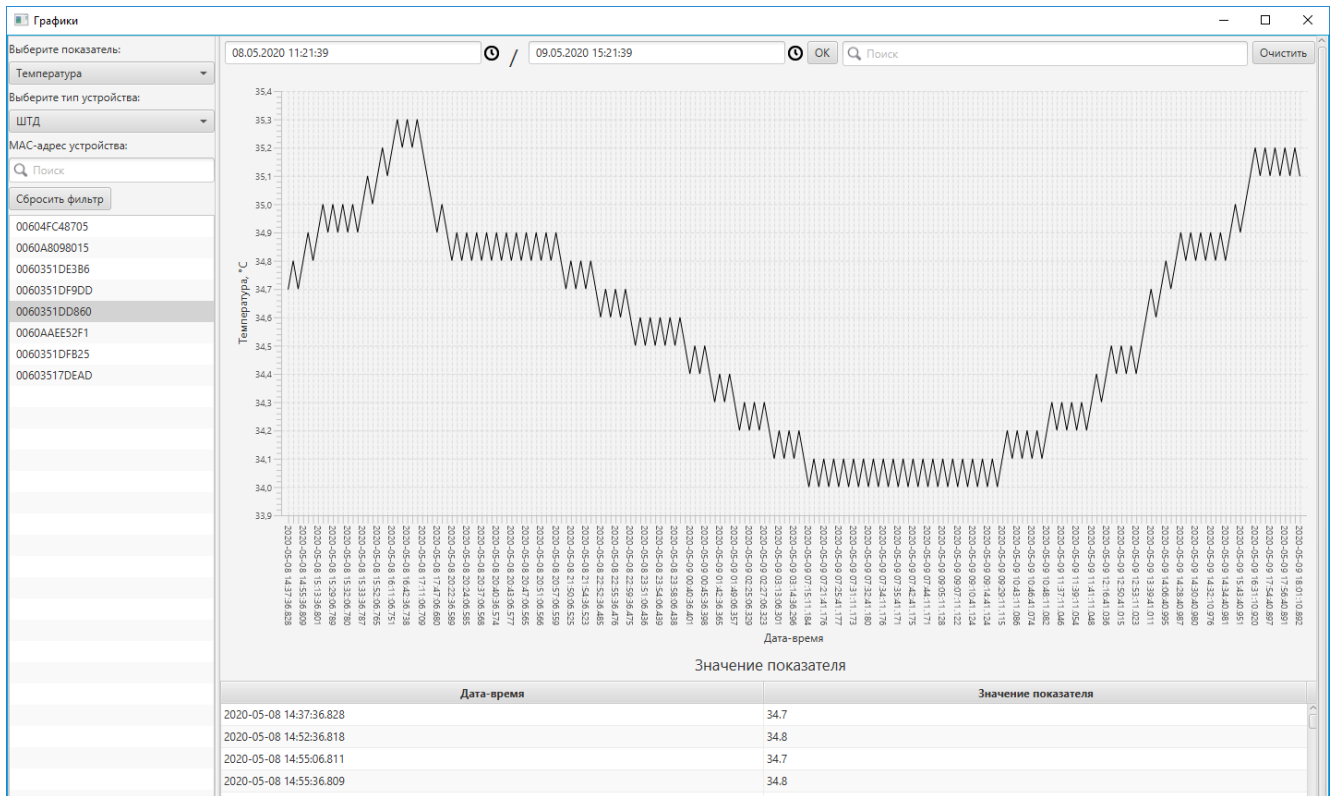


Рисунок 8 Графический отчет

3.2.6 Настройка

Часть настроек программы вынесена в меню «Настройки»:

Таблица 6 - Настройки программы через меню

| Название | Описание |
|------------------------------|--|
| Использовать локальное время | Если данная строка отмечена галочкой, то все время в программе отображается в том часовом поясе, который установлен на компьютере, если строка не отмечена, то время отображается в часовом поясе GMT0 |
| Использовать короткие MAC | Если данная строка отмечена, то MAC адреса устройств отображаются в сокращенном виде (последние 4 цифры) |

В каталоге с программой должен находиться файл `.auth` в котором должен быть записан токен авторизации для приложения с правами `VOILOC_ADMIN` и `VOILOC_READ`.

Основная настройка программы производится через конфигурационный файл `config.properties`, расположенный в папке с программой. В конфигурационном файле доступны следующие параметры:

Таблица 7 – Настройки, доступные в конфигурационном файле

| Название | Доступные значения | Описание |
|----------------------|--------------------|---|
| add_settings | true/false | Разрешение на производство настройки радиоустройств |
| add_location | true/false | Добавление или удаление вкладки с моделью шахты |
| iom_server_host | IP адрес | Адрес сервера Voiloc |
| auth_url | URL | Адрес сервиса авторизации, дополнительно необходимо в файл <i>.auth</i> записать токен авторизации |
| firmware_load_folder | Путь к папке | Папка, в которой находится встроенное ПО для радиоустройств |
| csv_save_folder | Путь к папке | Папка, в которую сохраняются файлы CSV |
| update_rate | Целое число | Частота обновления информации в программе в миллисекундах, по умолчанию 3000 мс |
| local_time | true/false | Использование локального времени |
| short_macs | true/false | Отображение коротких или полных MAC адресов радиоустройств |
| self_address | IP адрес | IP адрес интерфейса для приемки и отправки IoM сообщений (необходимо указывать при наличии нескольких сетевых интерфейсов), по умолчанию определяется автоматически |
| self_port | Число | UDP порт для отправки IoM сообщений, по умолчанию 6050 |
| mcast_address | IP адрес | Адрес multicast группы для отправки IoM сообщений, по умолчанию 239.239.101.101 |
| statDev. | – | Набор параметров, которые формируются автоматически, такие как: размеры окна, выбранные столбцы и пр. |

Пример конфигурационного файла приведен в Листинге 1:

```
#Mon Jan 18 14:24:32 YEKT 2021
add_settings=true
statDev.visibility.STATDEVEDGEID=true
local_time=true
mobDev.width.MOBDEVCARRIER=302.0
firmware_load_folder=C:\\Users\\Remote\\Desktop
statDev.order=0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,16,15
statDev.width.STATDEVIP=175.0
statDev.width.STATDEVBATTERY=110.0
short_macs=false
add_location=true
iom_server_host=192.168.15.5
auth_url=http://192.168.15.5:8090/auth/refresh/
```

Листинг 1 Пример конфигурационного файла *config.properties*

Таблица 8 – Поля таблицы светильников

| Название | Описание |
|-------------------|--|
| № светильника | Номер светильника |
| Дата выдачи | Дата выдачи светильника в ламповой |
| Время выдачи | Время выдачи светильника в ламповой |
| Табельный номер | Табельный номер сотрудника, которому выдали светильник |
| ФИО | Фамилия имя и отчество сотрудника, которому выдали светильника |
| Должность | Должность сотрудника, которому выдали светильника |
| Инвентарный номер | Инвентарный номер светильника |

В верхней части экрана кнопки-индикаторы, которые отражают количество светильников, которые находятся в разных состояниях. При нажатии на кнопку-индикатор в таблице подсвечивается первый светильник из соответствующий группы. Строки таблицы, соответствующие разным состояниям, подсвечены теми же цветами, что и кнопки им соответствующие.

Таблица 9 – Кнопки-индикаторы

| Кнопка | Описание |
|-------------|---|
| Темно-серая | Выключенные, не выданные светильники (состояние «Неактивен») |
| Серая | Включенные, не выданные светильники (состояние «Активен») |
| Голубая | Выданные светильники (состояние «Выдан») |
| Синяя | Выданные светильники, не прошедшие самотестирование |
| Красная | Выданные светильники, самотестирование которых завершилась неудачей |
| Зеленая | Выданные светильники, успешно прошедшие самотестирование |
| Желтая | Выданные светильники, не сданные после смены |

В нижней части экрана находятся кнопки-фильтры, скрывающие или показывающие светильники, которые находятся в состояниях, описанных в Таблице 9.

Рядом с кнопками-фильтрами находится кнопка «группировка», которая переключает режим формирования таблицы светильников. При включенном режиме группировки (нажатой кнопке) строки в таблице группируются по состоянию светильников. При деактивированной кнопке таблица работает в режиме очереди: строки отсортированы по времени изменения состояния (в верху таблицы находятся светильники, чье состояние изменилось недавно).

В нижней части справа находятся кнопки для ручного управления выдачей фонарей:

Таблица 10 – Кнопки-индикаторы

| Кнопка | Описание |
|---------|-----------------------------------|
| Выдача | Принудительная выдача светильника |
| Замена | Временная замена светильника |
| Приемка | Прием светильника |

3.3.2 Подготовка к использованию программы

Для использования программы необходимо настроить конфигурационный файл сервера Voilos, а именно:

- установить минимальную продолжительность смены (по умолчанию 15 минут);
- установить номер геосегмента для ламповой (по умолчанию 254);

- установить номер геосегмента для самотестирования (по умолчанию 254).

В помещении ламповой установить ШТД с выбранным номером геосегмента.

Для каждого сотрудника необходимо предварительно привязать светильник к человеку в Редакторе справочников.

3.3.3 Выдача светильников

Выдача светильников с помощью Считывателя меток МИВ-00

Порядок действий для автоматизированной выдачи светильника с использованием Считывателя:

- 1) для выдачи светильника необходимо снять с зарядного стола, после чего светильник переходит в состояние «Активен»;
- 2) поднести метку МИВ-00 к Считывателю, при удачной попытке считывания считыватель воспроизведет однократный звуковой и световой сигнал (подробнее о сигналах считывателя можно прочитать в разделе 3.3.7, посвященном программе NFC Reader), светильник перейдет в состояние «Выдан»;
- 3) светильник передается сотруднику, который, находясь в Ламповой, нажатием кнопки на светильники должен запустить процедуру самотестирования, в ходе которой светильник проверит связь с сервером и подаст сигнал лампой;
- 4) после сигнала лампы необходимо повторно нажать кнопку на светильнике, что завершит процедуру самотестирования, в случае удачного завершения светильник перейдет в состояние «Выдан с успешной проверкой», в случае неудачного завершения процедуры самотестирования светильник перейдет в состояние «Выдан с неуспешной проверкой».

Выдача светильников без использования Считывателя меток МИВ-00

Порядок действий для автоматизированной выдачи светильника без использования Считывателя:

- 1) для выдачи светильника необходимо снять с зарядного стола, после чего светильник переходит в состояние «Активен»;
- 2) светильник передается сотруднику, который, находясь в Ламповой, нажатием кнопки на светильники должен запустить процедуру самотестирования, в ходе которой светильник проверит связь с сервером и подаст сигнал лампой;
- 3) после сигнала лампы необходимо повторно нажать кнопку на светильнике, что завершит процедуру самотестирования;
- 4) при входе в радиозону с другим геосегментом светильник перейдет в состояние «Выдан».

Выдача светильников в ручном режиме

Порядок действий при ручной выдаче светильника:

- 1) для выдачи светильника необходимо снять с зарядного стола, после чего светильник переходит в состояние «Активен»;
- 2) в АРМ «Ламповщика» выбрать снятый светильник и нажать кнопку «Выдача», светильник перейдет в состояние «Выдан»;

- 3) светильник передается сотруднику, который, находясь в Ламповой, нажатием кнопки на светильники должен запустить процедуру самотестирования, в ходе которой светильник проверит связь с сервером и подаст сигнал лампой;
- 4) после сигнала лампы необходимо повторно нажать кнопку на светильнике, что завершит процедуру самотестирования, в случае удачного завершения светильник перейдет в состояние «Выдан с успешной проверкой», в случае неудачного завершения процедуры самотестирования светильник перейдет в состояние «Выдан с неуспешной проверкой».

3.3.4 Прием светильника

Светильник, находящийся в состоянии «Выдан», можно принять одним из следующих способов:

- 1) поднести метку МИВ-00 к Считывателю, при удачной попытке считывания считыватель воспроизведет однократный звуковой и световой сигнал, светильник перейдет в состояние «Активен»;
- 2) поставить светильник на зарядку, после чего он перейдет в состояние «Неактивен»;
- 3) в АРМ «Ламповщика» выбрать светильник и нажать на кнопку «приём», после чего светильник перейдут в состояние «Активен».

3.3.5 Временная замена светильника

В случае необходимости временной замены светильника необходимо соблюдение следующих условий:

- 1) заменяемый светильник должен быть в состоянии «Активен» или «Неактивен» (светильники в состоянии «Выдан» необходимо сначала принять);
- 2) подменный светильник должен быть в исправном состоянии и быть привязан к носителю в Редакторе связей и справочников.

Для выдачи подменного светильника необходимо произвести следующие действия:

- 1) в АРМ «Ламповщика» выбрать в таблице заменяемый светильник и нажать на кнопку «замена»;
- 2) откроется окно замены светильника (Рисунок 10) с таблицей активных не выданных светильников, выбрать в этой таблице подменный светильник и нажать кнопку «замена»;
- 3) подменный светильник временно (на одну смену) будет привязан к сотруднику, чей светильник был заменен;
- 4) выдать подменный светильник.

Содержание ярлыка Листинге 2:

```
[Desktop Entry]
Version=1.0
Type=Application
Name=LAMP
Comment=Voiloc Lamp Client
Exec=java -jar -ea voiloc-lamp-client.jar
Path=/home/user/vlamper
Terminal=false
StartupNotify=false
Icon=/home/user/vlamper/vlamper.png
Name[ru_RU.UTF-8]=АРМ Ламповщика
```

Листинг 2 Пример конфигурационного файла ярлыка

Файл ярлыка нужно сделать доверенным для исполнения, нажав правой кнопкой на файл и выбрав соответствующий пункт. Также необходимо создать ярлык в автозапуске, для этого необходимо зайти в меню в Пуск/Параметры/Настройка LXQt/Настройка сеанса/Автозапуск и нажать кнопку «Добавить» в секции «Автозапуск только для LXQt», задать имя и ввести команду для запуска (подставив актуальный путь до файла):

```
java -jar -ea /home/user/vlamper/voiloc-lamp-client.jar
```

Файл с ярлыком автозапуска находится в `/home/user/.config/autostat/`, в случае, если автозапуск не работает, необходимо также сделать ярлык автозапуска доверенным.

Настройка программы осуществляется через конфигурационный файл `vlamper.cfg`, расположенный в папке с установленной программой. В конфигурационном файле доступны следующие параметры:

Таблица 11 – Настройки, доступные в конфигурационном файле

| Название | Доступные значения | Описание |
|--|---------------------------|--|
| voiloc_url | URL | Адрес для доступа к REST API сервера Voiloc |
| voiloc.update_interval-devices | Целое положительное число | Интервал обновления данных об устройствах, сек |
| voiloc.update_interval-virtual_devices | Целое положительное число | Интервал обновления данных о виртуальных устройствах, сек |
| voiloc.update_interval-carriers | Целое положительное число | Интервал обновления данных о носителях, сек |
| fullscreen | true/false | Флаг полноэкранного режима |
| voiloc.auth.url | URL | Адрес сервиса авторизации, по умолчанию <code>:8090/auth/refresh/</code> |
| voiloc.auth.token | Токен | Параметр содержит либо строку с токеном авторизации, либо ссылку на файл с токеном, для работы программы необходимо создать пользователя с правами APP_VOILOC_LAMP и выпустить ему токен |

Пример конфигурационного файла представлен в Листинге 3:

```
# путь к VoiLOC REST API
voiloc.url = http://192.168.15.5:8090/voiloc/

# интервал обновления данных об устройствах, сек
voiloc.update_interval-devices = 1
voiloc.update_interval-virtual_devices = 1

# интервал обновления данных о носителях, сек
voiloc.update_interval-carriers = 15

# флаг полноэкранного режима по-умолчанию
fullscreen = true

# путь к auth сервису (выдача access-токенов)
voiloc.auth.url = http://192.168.15.5:8090/auth/refresh/

# refresh-токен, указывается или сам токен или ссылка на файл с токеном
voiloc.auth.token = "c113h02fw0002p62yqu8w299g.pj0SNm0xTG66/sIShnRnfbr/BW+sbhCsjwIkM/CLcsyB/KwDKJEzAXB+3Akpge3Y7w4SCRI9Cl0xxAI93i3YfA=="
#voiloc.auth.token = ./auth
```

Листинг 3 Пример конфигурационного файла vlamper.cfg

3.3.7 NFC Reader

NFC Reader – это программа без пользовательского интерфейса, предназначенная для получения данных со Считывателя меток МИВ-00 и передачи их на сервер СМС Исеть. Как правило, считыватель меток подключается к компьютеру в Ламповой, на котором установлен АРМ Ламповщик.

Установка NFC Reader

NFC Reader поставляется в виде zip архива с docker-контейнером и драйвером для Считывателя меток МИВ-00. Компьютер, на который производится установка, должен работать под ОС Linux, в примере рассмотрим установку на OS Rosa Fresh 12.4. На время установки компьютер должен иметь подключение к интернету во время установки ПО и к нему должен быть подключен считыватель по USB.

Перед установкой необходимо загрузить docker и настроить его:

```
sudo dnf install docker docker-compose
sudo systemctl enable docker
sudo usermod -aG docker <имя_пользователя>
```

Для установки драйвера необходимо распаковать архив в любую папку, перейти в нее и запустить установку драйвера от имени суперпользователя:

```
sudo ./install_driver_linux.sh
```

После установки драйвера нужно перезапустить компьютер и проверить, что в каталоге */dev/* появился файл *microm**, вместо * может быть любое число – это номер считывателя, его необходимо запомнить для последующей настройки программы.

Далее необходимо в каталоге с программой отредактировать файл *docker-compose.override.yml*, пример конфигурационного файла представлен в Листинге 4:

```
## Iset reader user configuration

services:
  iset-reader:
    environment:
      # [REQUIRED] Iset server URL
      - ISET_READER_SERVER_URL=http://192.168.15.5:8090

      # [REQUIRED] Reader usb numbers microm0, microm1 will correspond to "0,1". If
not set, determine automatically
      - ISET_READER_USB_NUMS=[1]

      # [REQUIRED for existing reader] GUID on an already existing reader. If provided,
the reader will not be registered on the server
      # - ISET_READER_GUID="d575ca63-6cb5-44fc-82ce-a2c9fc840583"

      # [OPTIONAL] Reader name (will be registered on ISET server)
      - ISET_READER_NAME=lamp-station-1

      # [OPTIONAL] Authentication refresh token (with APP_VOILOC_READER role). Required
if server authorization enabled
      # -
      ISET_READER_AUTH_REFRESH_TOKEN="clkqw0a4n0008p1395u501e14.etxwBxX2RSQhVnyqkuDMqRaakvx
SKhwE2Z50eN08rM476p1YJsh1V/N5WPaBNu7ZI7OMJs0wH371fB87bv1CA=="

  devices:
    # - /dev/microm0:/dev/microm0
    - /dev/microm1:/dev/microm1
```

```
# - /dev/microm2:/dev/microm2
# - /dev/microm3:/dev/microm3
# - /dev/microm4:/dev/microm4
# - /dev/microm5:/dev/microm5
```

Листинг 4 Пример конфигурационного файла *docker-compose.override.yml*

Таблица 12 – Настройки, доступные в конфигурационном файле

| Название | Доступные значения | Описание |
|--------------------------------|--------------------|---|
| ISET_READER_SERVER_URL | URL | Адрес для доступа к API сервера Voiloc |
| ISET_READER_USB_NUMS | Массив чисел | Номера считывателей, которые подключены к компьютеру |
| ISET_READER_GUID | GUID | ID считывателя, устанавливается только если производится обновление ПО, либо производится замена компьютера |
| ISET_READER_NAME | Строка | Имя считывателя, отображаемое в системе |
| ISET_READER_AUTH_REFRESH_TOKEN | Токен | Токен авторизации на сервере СМС Исеть, для работы считывателя необходимо создать пользователя с правами APP_VOILOC_READER и выпустить ему токен |
| devices: | Устройства | В данной секции необходимо раскомментировать только те устройства считывателей, которые подключены к компьютеру, в противном случае программа считывателя не запустится |

После внесения настроек необходимо запустить установку контейнера с программой считывателя командой:

```
./deploy.sh
```

Посмотреть логи работы программы можно командой

```
docker logs -f iset-reader
```

После завершения работы скрипта необходимо перезагрузить компьютер. После перезагрузки в списке устройств (каталог /dev) должно появиться устройство microm. Если устройство не найдено, то необходимо повторить установку драйверов устройства (от имени суперпользователя, находясь в папке, с распакованным архивом), после чего снова перезагрузить компьютер:

```
# ./dist/install_driver_nix.sh
```

Управление службой NFC Reader

Посмотреть логи работы программы можно командой:

```
docker logs -f iset-reader
```

Узнать состояние службы:

```
docker-compose ps
```

Остановить службу:

```
docker-compose down iset-reader
```

Запустить службу:

```
docker-compose up -d iset-reader
```

Работа с программой

После установки служба NFC Reader автоматически загружается при включении компьютера и после загрузки ОС можно сразу пользоваться Считывателем меток МИВ-00. Для считывания метки необходимо поднести метку МИВ-00 к зоне считывания Считывателя. В случае успешной регистрации метки Считыватель произведет однократный звуковой и световой сигналы.

Если метка считалась, но отправка информации на сервер закончилась неудачей (сервер прислал ошибку, либо отсутствует связь с сервером), то Считыватель произведет трехкратный звуковой сигнал. В этом случае необходимо проверить настройки программы, связь с сервером и его работоспособность.

Если считывание метки не удалось (считать информацию не получилось частично или полностью), то считыватель не воспроизведет звуковых и световых сигналов. В этом случае необходимо проверить питание метки работоспособность фонаря.

Более подробная информация, касающаяся нюансов технических нюансов по работе программы содержится в архиве с установочными файлами программы.

3.4 ПО Вентиляция 2

ПО «Вентиляция 2» — это специализированный программный комплекс, не являющийся частью СМС Исеть, созданный компанией Шахтэксперт, для решения задач проветривания горных выработок и противоаварийной устойчивости горных предприятий - угольных шахт, рудников, тоннелей.

Помимо основного назначения ПО «Вентиляция 2» предоставляет широкий набор инструментов для создания компьютерной модели шахты и обеспечивает возможность двустороннего обмена данными с сервером СМС Исеть по протоколу СИП версии 3 и версии 2. Таким образом, основным назначением ПО «Вентиляция 2» в рамках совместного использования с комплексом СМС Исеть является:

- 1) предоставление актуальной модели выработок серверу СМС Исеть;
- 2) предоставление инструментов указания местоположения базовых станций относительно узлов модели выработок;
- 3) предоставление информации о местоположении базовых станций (слотов) серверу СМС Исеть;
- 4) визуальное отображение местоположения ИКН и СТП на модели выработок в реальном времени на основании данных, получаемых с сервера СМС Исеть.

Для случаев, когда функционал локального позиционирования не требуется, использование ПО «Вентиляция 2» является избыточным.

Подробную информацию о ПО «Вентиляция 2» можно получить на сайте разработчика <https://minesoft.ru/cooling/>.

4 Серверное программное обеспечение

Перед установкой серверного программного обеспечения рекомендуется настроить как минимум наземную часть сетевой инфраструктуры, конфигурация которой должна обеспечивать:

- 1) изолированный сегмент сети ПАК «СМС Исеть» (не рекомендуется использование активного сетевого оборудования других автоматизированных систем);
- 2) прохождение трафика multicast (для программного и аппаратного обеспечения используется групповой адрес — 239.239.101.101).

Также на время установки серверного ПО серверу необходим доступ в интернет для установки внешних пакетов ПО.

4.1 Установка и настройка ОС

4.1.1 Установка операционной системы

Серверное ПО «СМС Исеть» предназначено для работы в среде Linux, рекомендуется использовать дистрибутив Rosa Fresh Server версии 12.4 и выше. Загрузить установочный диск можно на странице <https://rosa.ru/rosa-linux-download-links/>.

Если установка осуществляется без использования системы виртуализации через флеш-накопитель, то необходимо записывать его в «DD-режиме». Документацию по операционной системе и по установке можно получить на странице <https://rosa.ru/docs/>.

Если установка осуществляется через систему виртуализации, то можно выделить отдельные виртуальные диски для:

- 1) операционной системы (достаточно 100 Гб);
- 2) образов докер-контейнеров (достаточно 50-100 Гб);
- 3) логов системы (достаточно 50-100 Гб);
- 4) записей переговоров (достаточно 50-100 Гб);
- 5) кэшированных данных (не менее 100 Гб, зависит от количества устройств);
- 6) базы данных (зависит от количества устройств, самый объемный диск).

Выделение дисков позволит частично сохранить работоспособность сервера при переполнении одного из них. Часть дисков можно объединить в один.

Внимание: при установке и при базовой настройке операционной системы необходимо иметь подключение к сети Интернет для установки дополнительных программных пакетов.

4.1.2 Базовая настройка операционной системы

В данном разделе часть команд будет начинаться со с команды `sudo`, которая заставляет операционную систему выполнить операцию от имени администратора, в этом случае перед выполнением команды необходимо будет ввести пароль пользователя.

4.1.2.1 Статический IP адрес

Определение активных сетевых интерфейсов:

```
ip link show
```

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT
group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: ens19: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
mode DEFAULT group default qlen 1000
    link/ether 08:00:27:9d:a7:6a brd ff:ff:ff:ff:ff:ff
```

В данном случае имя физического сетевого интерфейса *ens19*.

Для установки статического IP адреса необходимо воспользоваться утилитой *nmcli*, введя команду:

```
nmcli connection modify ens19 connection.autoconnect yes ipv4.method manual
ipv4.dns 10.0.0.1 ipv4.addresses 10.0.0.5/24 ipv4.gateway 10.0.0.1
```

В команде используются следующие параметры:

Таблица 13 – Параметры, доступные в *nmcli*

| Параметр | Описание |
|------------------------|--|
| connection modify | Параметр указывает на изменяемый интерфейс (в примере <i>ens19</i>) |
| connection.autoconnect | Включать интерфейс по умолчанию (yes/no) |
| ipv4.method | Метод задания IP адреса (manual – вручную, auto – dhcp) |
| ipv4.dns | Адрес DNS сервера |
| ipv4.addresses | IP адрес и маска подсети |

После ввода команды нужно перезагрузить операционную систему командой:

```
sudo reboot
```

После задания статического IP адреса становится возможно подключаться к серверу по протоколам *ssh* и *sftp* с помощью программ:

- 1) *Putty* ssh-клиент для Windows <https://www.putty.org/>;
- 2) *FileZilla* sftp-клиент для Windows и Linux <https://filezilla.ru/>

4.1.2.2 Монтирование дисков

Для примера рассмотрим монтирование диска для хранения контейнеров и образов *docker*. Другие диски монтируются аналогичным образом, но владельцы и права директорий, в которые они монтируются будут отличаться. *Следует монтировать диски до того, как будут установлены дополнительные пакеты и установлено ПО СМС Исеть.*

Перед монтированием дисков необходимо посмотреть список накопителей, присутствующих в системе командой:

```
lsblk
```

```
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINTS
sda 8:0 0 100G 0 disk
├─sda1 8:1 0 8,6G 0 part [SWAP]
├─sda2 8:2 0 30G 0 part /
└─sda3 8:3 0 61,4G 0 part /home
sdb 8:16 0 100G 0 disk
```

В данном примере *sda* и *sdb* – диски, а *sda1*, *sda2* и *sda3* – это логические тома. Перед тем как монтировать диск, необходимо создать на нем логический том утилитой *cfdisk*:

```
sudo cfdisk /dev/sdb
```

В утилите необходимо нажать кнопку [**Новый**] и ввести размер создаваемого тома, затем нажать кнопку [**Запись**] и для подтверждения ввести *yes*, после записи изменений на диск нужно выйти, нажав кнопку [**Выход**].

После создания логического тома его необходимо отформатировать командой:

```
sudo mkfs -t ext4 /dev/sdb1
```

```
mke2fs 1.46.6 (1-Feb-2023)
/dev/sdd1 contains a ext4 file system
    created on Fri Mar 15 14:56:42 2024
Proceed anyway? (y,N)
```

```
y
```

```
Discarding device blocks: done
Creating filesystem with 26214139 4k blocks and 6553600 inodes
Filesystem UUID: 1926c113-4546-4293-b4a5-b6f14fb60300
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000, 7962624, 11239424, 20480000, 23887872
```

```
Allocating group tables: done
Writing inode tables: done
Creating journal (131072 blocks): done
Writing superblocks and filesystem accounting information: done
```

UUID созданного тома необходимо запомнить, либо посмотреть его командой:

```
sudo blkid
```

```
/dev/sdb1:    UUID="1926c113-4546-4293-b4a5-b6f14fb60300"    BLOCK_SIZE="4096"    TYPE="ext4"
PARTUUID="c7b35e34-8f15-8742-a108-a229e3042168"
/dev/sda2:    UUID="8f8f2d92-23d9-402a-8b90-593616bb4768"    BLOCK_SIZE="4096"    TYPE="ext4"
PARTUUID="a35d1716-02"
/dev/sda3:    UUID="31acb01f-77a4-46e1-8db5-6502819e0e86"    BLOCK_SIZE="4096"    TYPE="ext4"
PARTUUID="a35d1716-03"
/dev/sda1:    UUID="b79b6ed1-cab1-47a7-ab9d-8f1ed35c1fae" TYPE="swap" PARTUUID="a35d1716-01"
```

Далее необходимо создать точку монтирования командой:

```
sudo mkdir /var/lib/docker
```

Для постоянного монтирования диска необходимо записать изменения в файл *fstab*:

```
sudo nano /etc/fstab
```

В файл необходимо добавить строку:

```
UUID=1926c113-4546-4293-b4a5-b6f14fb60300 /var/lib/docker    ext4 defaults 0 0
```

*Внимание: файл *fstab* должен заканчиваться пустой строкой, иначе система не сможет монтировать диски.*

После внесения изменений в файл *fstab* необходимо перезагрузить сервер:

```
sudo reboot
```

После перезагрузки необходимо проверить правильность монтирования диска командой (в колонке MOUNTPOINTS напротив тома должна стоять нужная директория):

```
lsblk
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINTS
sda 8:0 0 100G 0 disk
├─sda1 8:1 0 8,6G 0 part [SWAP]
├─sda2 8:2 0 30G 0 part /
├─sda3 8:3 0 61,4G 0 part /home
sdb 8:48 0 100G 0 disk
└─sdb1 8:49 0 100G 0 part /var/lib/docker
```

После этого необходимо сменить владельца, группу и права чтения и записи для директории командами:

```
sudo chown root:root /var/lib/docker
sudo chmod 710 /var/lib/docker
```

На этом монтирование диска закончено.

4.1.2.3 Обновление и установка дополнительных пакетов

Перед установкой дополнительных пакетов необходимо обновить операционную систему до актуального состояния командой:

```
sudo dnf update
```

Для работы с базой данных необходимо установить пакет *postgresql*:

```
sudo dnf install postgresql
```

Для работы с *postgresql* необходимо создать пользователя командой:

```
sudo adduser -M -N postgres
```

После обновления необходимо установить *docker*:

```
sudo dnf install docker docker-compose
```

Далее необходимо добавить *docker* в автозагрузку:

```
sudo systemctl enable docker
```

Для возможности запускать контейнеры от имени пользователя необходимо добавить его в группу *docker*:

```
sudo usermod -aG docker <имя_пользователя>
```

Также для работы сервера потребуются дополнительные пакеты:

```
sudo dnf install cronie rsync sshpass
```

После установки необходимо перезагрузить систему:

```
sudo reboot
```

4.1.3 Сервер DHCP

Сервер DHCP необходим для предоставления сетевых настроек базовым станциям в режиме ШТД. Сервер может быть развернут на любом хосте в рамках сети СМС Исеть.

Допускается установка пакета *dhcp-server* (сервер DHCP) на сервере «СМС Исеть». В этом случае при отключении сервера может нарушиться работа подземной части Комплекса СМС Исеть.

Перед установкой необходимо убедиться, что сервер использует статический фиксированный IP адрес, ниже приведены инструкции по установке и настройке сервера DHCP.

Установка пакета `dhcp-server`:

```
sudo dnf install dhcp-server
```

В файле `/etc/sysconfig/dhcpd` необходимо добавить параметр `DHCPDARGS`, указав в нем имя сетевого интерфейса, отредактировать файл можно командой:

```
nano /etc/sysconfig/dhcpd
```

Пример части содержимого файла `/etc/sysconfig/dhcpd` (имя сетевого интерфейса в данном примере «ens19»):

```
DHCPDARGS=ens19
```

Настройка параметров сервера DHCP осуществляется через конфигурационный файл `/etc/dhcpd.conf`, в котором следует обратить внимание на параметры, указанные в таблице 14 (более подробную информацию о всех доступных настройках можно прочитать в инструкции к пакету `dhcp-server`).

Таблица 14 – Настройки конфигурационного файла сервера DHCP

| Параметр | Описание |
|---|---|
| <code>default-lease-time</code> | Время, на которое выдается IP адрес |
| <code>max-lease-time</code> | Максимальное время, на которое выдается IP адрес |
| <code>always-broadcast</code> | Всегда отправлять пакеты broadcast флагом |
| <code>authoritative</code> | Данный параметр позволяет ШТД быстрее получать настройки, если они были перемещены из другой подсети |
| <code>subnet</code> | Блок, описывающий подсеть, с которой работает сервер DHCP. В объявлении блока указывается адрес подсети и сетевая маска, далее в фигурных скобках указываются параметры и настройки для данной подсети. |
| <code>range</code> | Диапазон раздаваемых IP адресов, в данном примере раздаются адреса с 192.168.1.1 до 192.168.1.3 включительно |
| <code>option domain-name-servers</code> | Опция, задающая адрес DNS сервера |
| <code>host</code> | Блок, фиксирующий настройки для заданного устройства в сети, в объявлении необходимо указать имя устройства, параметры и настройки указываются в фигурных скобках. |
| <code>hardware ethernet</code> | Указывается MAC адрес устройства |
| <code>fixed-address</code> | Фиксируется IP адрес устройства, при обращении к серверу DHCP устройство будет всегда получать заданный IP адрес. |

Перед внесением изменений необходимо сделать резервную копию конфигурационного файла `/etc/dhcp/dhcpd.conf` командой:

```
sudo cp /etc/dhcpd.conf /etc/dhcpd.conf.bak
```

Очистить конфигурационный файл можно командой:

```
sudo cat /dev/null > /etc/dhcpd.conf
```

Редактирование конфигурационного файла:

```
sudo nano /etc/dhcpd.conf
```

Блоки `host` можно создавать для всех ШТД и давать им имена, соответствующие проектной документации, для удобства эксплуатации в дальнейшем. Пример конфигурационного файла:

```
default-lease-time 600;
max-lease-time 7200;
authoritative;
always-broadcast on;

subnet 192.168.1.0 netmask 255.255.255.0
{
    range 192.168.1.10 192.168.1.100;
    option domain-name-servers 192.168.1.1;
}

host ap1 {
    hardware ethernet 00:60:35:17:D8:B6;
    fixed-address 192.168.1.101;
}

host ap2 {
    hardware ethernet 00:60:35:17:D8:BB;
    fixed-address 192.168.1.102;
}
```

Листинг 5 Пример конфигурационного файла dhcpd.conf

Чтобы DHCP сервер запускался автоматически необходимо ввести команду:

```
sudo systemctl enable dhcpd.service
```

Команда запуска сервера DHCP:

```
sudo systemctl start dhcpd.service
```

Команда перезапуска сервера DHCP:

```
sudo systemctl restart dhcpd.service
```

Проверка статуса сервера DHCP:

```
sudo systemctl status dhcpd.service
```

При правильной работе сервиса в выводе команды в числе прочих должна присутствовать строчка `active (running)`.

Проверка выданных IP адресов:

```
cat /var/lib/dhcpd/dhcpd.leases
```

4.2 Установка серверного ПО

4.2.1 Установка сервера Voiloc

Перед установкой сервера необходимо создать следующие каталоги командой:

```
sudo mkdir /opt/voiloc /opt/voiloc/db_data /opt/voiloc/etc /opt/voiloc/logs
/opt/voiloc/calls /opt/voiloc/graphql_cache /opt/voiloc/iset_auth_data
```

Назначение каталогов описано в таблице 15:

Таблица 15 – Назначение каталогов сервера

| Каталог | Назначение |
|----------------------------|----------------------------|
| /opt/voiloc/db_data | База данных |
| /opt/voiloc/etc | Настройки сервера |
| /opt/voiloc/logs | Логи сервера |
| /opt/voiloc/calls | Записи голосовых вызовов |
| /opt/voiloc/graphql_cache | Кэшированные данные |
| /opt/voiloc/iset_auth_data | Данные сервиса авторизации |

После создания каталогов (и примонтирования к ним дисков, если в этом была необходимость) необходимо разрешить доступ к ним:

```
sudo chmod 777 /opt/voiloc /opt/voiloc/db_data /opt/voiloc/etc /opt/voiloc/logs
/opt/voiloc/calls /opt/voiloc/graphql_cache /opt/voiloc/iset_auth_data
```

Серверное ПО СМС Исеть поставляется в виде стэка docker-контейнеров в архиве вида *ISET-server-stack_master-35e7ee72.tar*. Данный архив необходимо скопировать на сервер и распаковать командой:

```
tar -xvf ISET-server-stack_master-35e7ee72.tar
```

Далее нужно перейти в каталог с распакованным стэком:

```
cd ISET-server-stack_master-35e7ee72/
```

Далее необходимо настроить параметры работы docker-контейнеров в файле *.env*:

```
nano .env
# ---- Common settings ----
MAIN_HTTP_PORT=8090
MAIN_HTTPS_PORT=4443

# ---- VoiLoc Core settings ----
VOILOC_DB_PORT=5432          # Database service port

### !NOTE!
### If you want to mount one of the volumes below, uncomment BOTH lines in the group
(*_MOUNT_TYPE and *_PATH).
### Change only the *_PATH value, leave the *_MOUNT_TYPE always set to 'bind'.

# -- VoiLoc database storage --
VOILOC_DB_DATA_MOUNT_TYPE=bind
VOILOC_DB_DATA_PATH=/opt/voiloc/db_data

# -- VoiLoc config storage --
VOILOC_CORE_ETC_MOUNT_TYPE=bind
VOILOC_CORE_ETC_PATH=/opt/voiloc/etc
```



```
# -- VoiLoc logs storage --
VOILOC_CORE_LOGS_MOUNT_TYPE=bind
VOILOC_CORE_LOGS_PATH=/opt/voiloc/logs

# -- [Optional] VoiLoc call records storage --
VOILOC_CORE_CALLS_MOUNT_TYPE=bind
VOILOC_CORE_CALLS_PATH=/opt/voiloc/calls

# -- [Optional] VoiLoc API cache storage --
VOILOC_CORE_GRAPHQL_CACHE_MOUNT_TYPE=bind
VOILOC_CORE_GRAPHQL_CACHE_PATH=/opt/voiloc/graphql_cache

# ---- ISET Auth settings ----

# -- [Optional] ISET Auth data --
ISET_AUTH_DATA_MOUNT_TYPE=bind
ISET_AUTH_DATA_PATH=/opt/voiloc/iset_auth_data
```

Таблица 16 – Параметры в файле .env

| Каталог | Назначение |
|--------------------------------|--|
| MAIN_HTTP_PORT | Порт для нешифрованного http трафика |
| MAIN_HTTPS_PORT | Порт для зашифрованного http трафика |
| VOILOC_DB_PORT | Порт доступа к базе данных |
| VOILOC_DB_DATA_PATH | Каталог, к которому подключается база данных из docker-контейнера |
| VOILOC_CORE_ETC_PATH | Каталог, к которому подключается docker-контейнер для хранения настроек |
| VOILOC_CORE_LOGS_PATH | Каталог, к которому подключается docker-контейнер для хранения логов сервера |
| VOILOC_CORE_CALLS_PATH | Каталог, к которому подключается docker-контейнер для хранения голосовых вызовов |
| VOILOC_CORE_GRAPHQL_CACHE_PATH | Каталог, к которому подключается docker-контейнер для хранения кэшированных данных |
| ISET_AUTH_DATA_PATH | Каталог, к которому подключается docker-контейнер с сервисом авторизации |

Если какие-либо каталоги docker-контейнеров не следует выводить на сервер, то следует закомментировать соответствующие им строки, поставив в начале строки символ «#». Например, для каталога с записями голосовых вызовов это будет выглядеть так:

```
#VOILOC_CORE_CALLS_MOUNT_TYPE=bind
#VOILOC_CORE_CALLS_PATH=/opt/voiloc/calls # Optional
```

Если вы не станете подключать какой-либо каталог из docker-контейнеров, то информация будет храниться внутри контейнера в каталоге /var/lib/docker, необходимо убедиться, что в нем достаточно дискового пространства для хранения соответствующих данных.

Для установки или обновления серверного ПО СМС Исеть необходимо установить стек docker-контейнеров командой:

```
./deploy.sh
```

После ввода команды будут загружены (или обновлены) и запущены docker-контейнеры, в случае успешного выполнения будет выведено сообщение:

```
DEPLOY SUCCESS
```

Для удобного администрирования docker-контейнеров можно установить дополнительный контейнер *portainer*, который позволит иметь доступ к управлению docker через WEB интерфейс:

```
docker volume create portainer_data
docker run -d -p 8000:8000 -p 9000:9000 --name portainer --restart=always -v
/var/run/docker.sock:/var/run/docker.sock -v portainer_data:/data
portainer/portainer-ce:latest
```

После установки необходимо перезагрузить сервер и через браузер зайти по адресу <http://<IP-адрес сервера>:9000>, где нужно будет создать пароль администратора. Далее по этому же адресу будет доступ к управлению контейнерами.

4.2.2 Настройка сервера Voiloc

Настройка компонент сервера осуществляется через конфигурационный файл */opt/voiloc/etc/voiloc.cfg*. Строки, начинающиеся с символа #, являются комментариями и не влияют на настройки сервера. Пример конфигурационного файла приведён в Листинге 6:

```
#### Настройки для подключения и работы с БД
#####
# JDBC URL для доступа к БД
db.url = jdbc:postgresql://localhost:5432/voilocdb

# пользователь и пароль для доступа к БД
db.user =
db.password =

# ограничение на максимальное количество подключений к БД
db.pool_limit = 100

# ограничение на размер единой выборки из БД, шт
db.result_count_limit = 20000

# ограничение на время выполнения единой выборки из БД, сек
db.result_time_limit = 300

# телеметрия, записываемая в БД
db.sensors = {include:[ALL], exclude:[QS_CONNECTIVITY, QS_RANGING, QS_NET_STAT, QS_LOSS_IAA]}

#### настройки встроенного ИоМ-сервера
#####
# IP-адрес интерфейса, на котором ИоМ-сервер будет принимать сообщения (по-умолчанию null (autodetect))
#iom.address = ${.host}

# порт (по-умолчанию 6050)
iom.port = 6050

# использовать UNICAST для отсылки пакетов (для совместимости с RealTrac)
# iom.force_unicast = false

# геосегмент для прохождения процедуры тестирования связи (по умолчанию 254)
iom.area_for_testing = 254

# геосегмент для мобильных РТД, которые не учитываются при расчете позиционирования (по умолчанию 255)
iom.area_for_mobile_anchor = 255
```

```

# правила игнорирования ALARM-ов в определенных геосегментах
iom.alarm_ignore_rules = { MANDOWN: {zones: [254, 253]}, FREEFALL: {zones: [254, 253]}, BUTTON: {zones:
[254, 253]} }

#### настройки встроенного HTTP/HTTPS-сервера
#####

# IP-адрес интерфейса, на котором HTTP/HTTPS-сервер будет принимать сообщения (по-умолчанию null
(autodetect))
#http.address = ${.host}

# порт (по-умолчанию 8080)
http.port = 8080

# ограничение на время выполнения единоразового HTTP запроса, сек
http.request_time_limit = 360

#### настройки для подсистемы Ламповщика
#####

# режим работы модуля Ламповщика
ext.lamp.mode = AUTO

# геосегмент для выдачи фонарей в ламповой
ext.lamp.area_out = 254

# геосегмент для приема фонарей в ламповой
ext.lamp.area_in = 254

# список рабочих смен (от получения фонаря до его сдачи)
ext.lamp.work_shifts = [{"06:00", "16:00"}, {"12:00", "22:00"}, {"18:00", "04:00"}, {"00:00", "10:00"}]

# лимит ожидания проведения процедуры проверки связи (Test_Connection) от выдачи фонаря, сек
ext.lamp.test_timeout = 180

# задержка повторной активации только что сданного фонаря через зарядку, сек
ext.lamp.reactivate_delay = 70

# задержка повторной активации только что сданного фонаря через считыватель, сек
ext.lamp.reactivate_delay_after_reader = 30

# таймаут блокировки выдачи устройства (в случае открытия диалога замены), сек
ext.lamp.lock_out_timeout = 600

# временной лимит ожидания выдачи обязательных устройств (снаряжения), отсчитывается с момента выдачи фонаря,
сек
ext.lamp.required_devices_out_timeout = 120

# временной лимит сдачи ранее выданных устройств (снаряжения), отсчитывается с момента сдачи фонаря, сек
ext.lamp.received_devices_in_timeout = 120

# список обязательных к получению устройств (снаряжения) по умолчанию
ext.lamp.required_devices = ['TAG', 'GAS_ANALYZER']

# возможные варианты устройств (снаряжения)*:]
# 'TAG' - фонарь
# 'SENTRY' - ИКН
# 'GAS_ANALYZER' - газоанализатор
# 'SELF_AID_KIT' - самоспасатель
# *каждое выдаваемое устройство (снаряжение) должно иметь RFID-метку, быть заведено в систему через Редактор
справочников
# выдача и приемка осуществляется только через считыватель

#### настройки встроенного сервера локации
#####

# тип сервера позиционирования (доступен только ToF)
#ls.impl = ToF

# максимальная скорость мобильного объекта, м/с
ls.ToF.expansion = 5.0

# радиус разброса гипотез о местонахождении объекта, при котором считается, что объект "точно" локализован,
метры
ls.ToF.max_error = 15.0

```

```

# задержка отображения позиционирования (количество измерений), чем больше это значение, тем точнее
позиционирование
ls.ToF.lag = 3

#### настройки встроенного СИП-сервера (связь с ПО Вентиляция-2)
#####

# версия СИП протокола
sip.version = 3

# IP-адрес интерфейса, на котором СИП-сервер будет принимать сообщения (по-умолчанию null (autodetect))
#sip.address = ${.host}

# порт (по-умолчанию 1555)
sip.port = 1555

# имя пользователя для подключения к СИП-серверу
# sip.user

# пароль пользователя для подключения к СИП-серверу
# sip.password

# флаг автоматического удаления координат точки доступа при её отвязке от слота
sip.delete_coords_on_unlink_slot = true

#### настройки модулей сервера
#####

# активация модулей сервера, по умолчанию активны все модули
core.modules = {include:[ALL]}

#### настройки телеметрии, обрабатываемой сервером в реальном времени
#####

# телеметрия, которую сервер собирает в текущем режиме
#core.sensors = {include:[ALL], exclude:[QS_NET_STAT]}
core.sensors = {include:[ALL]}

# верхний временной лимит между реперными значениями телеметрии, мс
core.sensors.significant_time_limit = 600000

# предельные изменения показателей телеметрии, при превышении которых показания считаются реперными
core.sensors.QS_LOSS_IAA.significant_threshold = 10.0
core.sensors.VOLTAGE_EXT.significant_threshold = 3.0
core.sensors.VOLTAGE_INT.significant_threshold = 2.0
core.sensors.TEMPERATURE.significant_threshold = 1.0
core.sensors.PRESSURE.significant_threshold = 200.0
core.sensors.HUMIDITY.significant_threshold = 5.0
core.sensors.GAS_CH4.significant_threshold = 0.2
core.sensors.GAS_O2.significant_threshold = 0.4
core.sensors.GAS_CO.significant_threshold = 8
core.sensors.GAS_CO2.significant_threshold = 2

# коэффициенты пересчета данных принятых из блока GAS-18 и единицы измерения газов
core.sensors.GAS_CH4 = {k: 0.01, units: "% об.д."}
core.sensors.GAS_O2 = {k: 0.1, units: "% об.д."}
core.sensors.GAS_CO = {k: 1, units: "ppm"}
core.sensors.GAS_CO2 = {k: 0.01, units: "% об.д."}

# предельные значения показателей телеметрии, при пересечении которых создается тревога (событие)
# задаются в формате [<предельное значение>, <количество измерений подряд с превышением>]
core.sensors.QS_LOSS_IAA.alarm_threshold = [50.0, 2]
core.sensors.VOLTAGE_EXT.alarm_threshold = [9.5, 10]
core.sensors.VOLTAGE_INT.alarm_threshold = [4.6, 10]

# показания газовой телеметрии
core.sensors.GAS.aggregation_buffer = 600000
core.sensors.GAS.failure_threshold = 40
core.sensors.GAS_CH4.alarm_threshold = {h: 0.8, H: 1, n: 3}
core.sensors.GAS_O2.alarm_threshold = {L: 19, l: 20, n: 3}
core.sensors.GAS_CO.alarm_threshold = {h: 15, H: 17, n: 3}
core.sensors.GAS_CO2.alarm_threshold = {h: 0.45, H: 0.50, n: 3}

#### настройки обработка данных позиционирования
#####

```

```

# верхний временной лимит между реперными значениями локации, мс
core.locations.significant_time_limit = 300000

# нижний пространственный лимит между реперными значениями локации, м
core.locations.significant_distance_limit = 30.0

# верхний количественный лимит фиксации перехода на ребро графа
# при отсутствии информации о нахождении на других ребрах, циклы
core.locations.enter_edge_count_limit = 5

# верхний временной лимит фиксации перехода на ребро графа
# при отсутствии информации о нахождении на других ребрах, с
core.locations.enter_edge_time_limit = 60

#### настройки автоматических аварийных оповещений
#####

# список типов устройств, для которых необходимо производить отсылку сигнала при входе в особые области
core.area_signal.device_types = ["TAG"]

# периодичность повторной отправки аварийного оповещения на устройство после его входа в особые области, с
core.area_signal.recreate_period = 300

#### настройки работы со стационарными устройствами
#####

# восстановление геосегмента стационарных устройств после перепрошивки
core.rollback_area_after_upgrade = true

# размер временного окна набора статистики для сравнения карты замеров расстояний
# между стационарными устройствами с их расстановкой на графе, час
core.validate_anchor_arrangement.period = 24

# предельно допустимая величина расхождения статистики реальных замеров
# и данных о расположении стационарных устройств на графе, м
core.validate_anchor_arrangement.error = 20.0

#### настройки работы с автоматически создаваемыми зонами
#####

# флаг автоматического создания областей под сервисные автозоны (области создаются под КАЖДУЮ точку доступа
НЕЗАВИСИМО)
core.autoareas.enable = true

# политика создания ребер для автозон
core.autoareas.coords_policy = [[0,0,0], [10,0,0], [0,0,10]]

# политика создания автозон для разных геосегментов
core.autoareas.zones_policy = {254:["LAMP", "Ламповая ${mac}"], 251:["SERVICE", "Сервисная зона"]}

#### настройки групповых аварийных оповещений
#####

# периодичность повторной отправки аварийного оповещения на устройство при отсутствии подтверждения
успешности предыдущего, с
core.emergency_notifications.resend_period = 60

# предельное количество повторений отправки аварийного оповещения на устройство при отсутствии подтверждения
успешности предыдущего
core.emergency_notifications.resend_count = 50

# предельная длительность ожидания окончания выполнения задачи групповой рассылки аварийного оповещения,
мин
core.emergency_notifications.timeout = 180

#### настройки встроенного Netty-движка
#####

# флаг использования EPOLL (по-умолчанию false)
# на текущий момент поддерживается только на платформе linux-x86_64
netty.use_epoll = false

### настройки встроенного VoIP-модуля
#####

# прокси сервер для VoIP-сервера
#voip.outbound_proxy

```

```

# IP-адрес интерфейса, на котором VoIP-сервер будет принимать сообщения (по-умолчанию null (autodetect))
#voip.address = ${.host}

# порт, на котором VoIP-сервер будет принимать сообщения
voip.port = 5060

# транспортный протокол по которому работает VoIP-сервер: TCP, UDP, TLS, STCP
#voip.transport = UDP

# начальный порт для RTP-трафика SIP-стека
voip.rtp_start_port = 8000

# количество портов для RTP-трафика SIP-стека
voip.rtp_num_ports = 500

# учётные записи внешних абонентов при использовании SIP/FXO шлюза
voip.users = {"9999":"VoilocGate"}

# номер вызываемого абонента по умолчанию
#voip.default_callee

# путь по которому хранятся файлы записей разговоров (по-умолчанию './calls/')
voip.calls_path = /opt/voiloc/calls/

### настройки встроенного OPC-клиента
#####

# IP адрес OPC-сервера
# opc.address = 192.168.8.184

# пользователь для доступа к OPC-серверу
opc.user = voiloc

# пароль для доступа к OPC-серверу
opc.password = voiloc

# идентификатор инстанса OPC-сервера
#opc.prog_id = RTS OPC server test

# идентификатор CLS_ID OPC-сервера
opc.cls_id = 8FD7B87A-E35E-47DC-A1FD-5D52233E6F2E

### настройки системы мониторинга состояния сервера
#####

# путь к внешнему скрипту обработки сообщений от системы мониторинга,
# вызывается скрипт с параметрами <ts> <tag> [параметры соответствующего события]
core.sys_monitor.cmd = /opt/voiloc/monitor.sh

# настройки мониторинга загрузки CPU [порогПроценты, размерБуфераМинуты, частотаПовтораУведомленияМинуты]
core.sys_monitor.cpu_usage = [80, 10, 60]

# настройки мониторинга загрузки ОЗУ [порогПроценты, размерБуфераМинуты, частотаПовтораУведомленияМинуты]
core.sys_monitor.mem_usage = [80, 10, 60]

# настройки мониторинга заполнения диска [[каталог, порогПроценты, размерБуфераМинуты, частотаПовтораУведомленияМинуты]]
core.sys_monitor.disk_usage = ["/opt/voiloc/calls", 80, 10, 60], ["/opt/voiloc/logs", 80, 10, 60], ["/var/lib/postgresql", 80, 10, 60]

# настройки мониторинга лога - набор масок для игнорирования событий в логе
core.sys_monitor.log_ignore = [".*No valid measurements.*", ".*Impossible to init location.*"]

### настройки системы кэширования архива
#####

# путь по которому хранятся кэши GraphQL-запросов
graphql.cache_path = /opt/voiloc/graphql_cache/

# интервал сбора данных по-умолчанию, сек (по умолчанию 60)
graphql.cache_build_step = 60

# глубина хранения данных по-умолчанию, час (по умолчанию 168)
graphql.cache_storage_period = 720

```

```

### настройки системы авторизации
#####

# признак активности режима авторизации через JWT-токены (по умолчанию false)
http.auth = true

# признак активности гостевого доступа при авторизации через JWT-токены (по умолчанию true)
http.auth.guests = true

# путь к публичному ключу
http.auth.public_key = /opt/voiloc/auth_public_key/access_token_sign_ES384.pub

```

Листинг 6 Пример конфигурационного файла *voiloc.cfg*

Таблица 17 – Параметры, доступные в конфигурационном файле *voiloc.cfg*

| Параметр | Значение по умолчанию | Описание |
|--|---|--|
| Настройки для подключения и работы с БД | | |
| db.url | jdbc:postgresql://127.0.0.1:5432/voilocdb | Строка доступа к базе данных, содержит IP-адрес хоста с СУБД PostgreSQL, порт СУБД и имя базы. По умолчанию используется подключение к локальной СУБД. <i>Устаревший параметр, не учитывается</i> |
| db.user | voiloc | Имя пользователя для доступа к базе данных, указанной в db.url. <i>Устаревший параметр, не учитывается</i> |
| db.password | | Пароль для доступа к базе данных, указанной в db.url. <i>Устаревший параметр, не учитывается</i> |
| db.pool_limit | 30 | Ограничение на количество подключений к БД |
| db.result_count_limit | 10 000 | Ограничение на размер единоразовой выборки из БД (в штуках), при превышении этого лимита запрос к БД отбрасывается. |

| Параметр | Значение по умолчанию | Описание |
|--|-----------------------|--|
| db.result_time_limit | 300 | Ограничение на время выполнения единоразовой выборки из БД (в секундах), при превышении этого лимита запрос к БД отбрасывается. |
| db.sensors | {include: [ALL]} | Определяет какие каналы телеметрии ⁴ будут записываться в базу данных. Задается в виде: {include: [CH1, CH2,...], exclude: [CH1, CH2,...]}, где в блоке include перечисляются каналы, которые обязательно должны обрабатываться, а в блоке exclude которые не будут обрабатываться (имеет приоритет над include). |
| Настройки встроенного IoM-сервера | | |
| iom.address | автоопределение | IP адрес подсистемы IoM. |
| iom.port | 6050 | Порт подсистемы IoM. |

⁴Доступные каналы телеметрии:

ALL – вся телеметрия;
 QS_CONNECTIVITY – включение и отключение устройств;
 QS_LOSS_IAA – уровень потерь пакетов устройств;
 QS_RANGING – измерение расстояний между устройствами;
 QS_NET_STAT – статистика по переданным пакетам;
 VOLTAGE_EXT – внешнее напряжение устройств;
 VOLTAGE_INT – внутреннее напряжение устройств;
 PRESSURE – давление;
 TEMPERATURE – температура;
 HUMIDITY – влажность;
 GAS_CH4 – концентрация метана;
 GAS_O2 – концентрация кислорода;
 GAS_CO – концентрация оксида углерода;
 GAS_CO2 – концентрация двуоксида углерода.

| Параметр | Значение по умолчанию | Описание |
|---|---|--|
| iom.force_unicast | false | Параметр для совместимости с аппаратным обеспечением, работающим на микропрограммном обеспечении Realtrac, переводит режим работы сети сервера с используемого multicast на unicast. |
| iom.area_for_testing | 254 | Геосегмент для прохождения процедуры тестирования связи |
| iom.area_for_mobile_anchor | 255 | Геосегмент для мобильных РТД, которые не учитываются при расчетах позиционирования |
| iom.alarm_ignore_rules | {MANDOWN: {zones: [254]}, FREEFALL: {zones: [254]}, BUTTON: {zones: [254]}} | Правила игнорирования тревог, приходящих с мобильных устройств, доступно только для тревог типов MANDOWN (рабочий не двигается), FREEFALL (падение), BUTTON (нажатие кнопки) |
| Настройки встроенного HTTP-сервера | | |
| http.address | автоопределение | IP адрес подсистемы HTTP/HTTPS (REST API). <i>Не рекомендуется изменять</i> |
| http.port | 8080 | Порт подсистемы HTTP/HTTPS (REST API). |
| http.request_time_limit | 360 | Ограничение на время выполнения единоразового HTTP запроса (в секундах), при превышении этого лимита HTTP запрос отбрасывается. |

| Параметр | Значение по умолчанию | Описание |
|--|-----------------------|--|
| Настройки модуля Ламповщика | | |
| ext.lamp.mode | AUTO | Режим выдачи Светильников в ламповой, возможные варианты: AUTO, READER, AREA |
| ext.lamp.area_out | 254 | Геосегмент зоны выдачи СТП |
| ext.lamp.area_in | 254 | Геосегмент зоны приемки СТП |
| ext.lamp.work_shifts | [] | Перечень рабочих смен (возможны пересечения), пример: [[“23:00”, “09:00”], [“07:00”, “17:00”], [“15:00”, “01:00”]] |
| ext.lamp.test_timeout | 300 | Лимит ожидания проведения самотестирования Светильника с момента выдачи в секундах. 0 – самотестирование не учитывается |
| ext.lamp.reactivate_delay | 30 | Ограничительная задержка повторной активации только что сданного через зарядку устройства в секундах |
| ext.lamp.reactivate_delay_after_reader | 60 | Ограничительная задержка повторной активации только что сданного через считыватель устройства в секундах |
| ext.lamp.lock_out_timeout | 600 | Таймаут блокировки выдачи устройства (при открытии диалога замены) в секундах |
| ext.lamp.required_devices_out_timeout | 120 | Временной лимит ожидания выдачи обязательных устройств (снаряжения), отсчитывается с момента выдачи фонаря, сек |

| Параметр | Значение по умолчанию | Описание |
|--|-----------------------|--|
| ext.lamp.received_devices_in_timeout | 120 | Временной лимит сдачи ранее выданных устройств (снаряжения), отсчитывается с момента сдачи фонаря, сек |
| ext.lamp.required_devices | | Список обязательных к получению устройств ⁵ (снаряжения) по умолчанию, применяется при создании нового сотрудника, пример: ['TAG','GAS_ANALYZER'] |
| Настройки сервера позиционирования | | |
| ls.ToF.expansion | 5.0 | Максимальная скорость мобильного объекта |
| ls.ToF.max_error | 15.0 | Точность позиционирования |
| ls.ToF.lag | 3 | Количество измерений, необходимое для расчета текущей позиции |
| Настройки встроенного СИП-сервера (для связи с ПО Вентиляция 2) | | |
| sip.version | 3 | Версия протокола СИП (связь с ПО «Вентиляция 2») |
| sip.address | автоопределение | IP адрес подсистемы СИП (связь с ПО «Вентиляция 2»). |
| sip.port | 1555 | Порт подсистемы СИП (связь с ПО «Вентиляция 2»). |
| sip.user | | Имя пользователя для ПО Вентиляция 2 |
| sip.password | | Пароль пользователя для ПО Вентиляция 2 |

⁵Доступные устройства:

TAG – фонарь

SENTRY – ИКН

GAS_ANALYZER – газоанализатор

SELF_AID_KIT – самоспасатель

| Параметр | Значение по умолчанию | Описание |
|---|-----------------------|---|
| sip.delete_coords _on_unlink_slot | true | Флаг автоматического удаления координат точки доступа при её отвязке от слота |
| Настройки модулей сервера | | |
| core.modules | {include: [ALL]} | Включение и отключение модулей ⁶ сервера. |
| Настройки обработки сервером телеметрии в реальном времени | | |
| core.sensors | {include: [ALL]} | Определяет какие каналы телеметрии сервер собирает и обрабатывает в режиме реального времени. Задается аналогично настройке db.sensors. |
| core.sensors. significant_time_limit | 600000 | В случае если значения телеметрии не изменяются, то показания считаются значимыми по истечению указанного времени. Параметр задается в миллисекундах. <i>В БД записываются только значимые показания.</i> |

⁶Доступные модули сервера:

ALL – все модули;

CALLS – системный модуль работы с записями разговоров (обязательный);

FW_UPGRADE – системный модуль обслуживания прошивки устройств (обязательный);

MOBILE_ANCHORS – системный модуль обслуживания мобильных РТД (обязательный);

SENSORS – системный модуль генерации телеметрии устройств и сопутствующих им событий (обязательный);

СИП – модуль интеграции с СИП-протоколом;

REST_API – модуль реализации VoiLOC REST API;

GRAPHQL_API – модуль реализации VoiLOC GraphQL API;

EXT_LAMP – модуль поддержки функционала, специфичного для АРМ "Ламповщик";

| Параметр | Значение по умолчанию | Описание |
|--|-----------------------|---|
| core.sensors.GAS_CH4. significant_threshold core.sensors.GAS_O2. significant_threshold core.sensors.GAS_CO. significant_threshold core.sensors.GAS_CO2. significant_threshold | 0.0 | Определяет на сколько должно измениться показания датчиков переносных газоанализаторов для того, чтобы эти изменения были значимыми. |
| core.sensors.QS_LOSS_IAA. significant_threshold | 10.0 | Настройка значимости для уровня потерь устройства |
| core.sensors.VOLTAGE_EXT. significant_threshold | 3.0 | Настройка значимости при измерении внешнего напряжения устройств. |
| core.sensors.VOLTAGE_INT. significant_threshold | 2.0 | Настройка значимости при измерении внутреннего напряжения устройств. |
| core.sensors.TEMPERATURE. significant_threshold | 1.0 | Настройка значимости при измерении температуры |
| core.sensors.PRESSURE. significant_threshold | 200.0 | Настройка значимости при измерении давления |
| core.sensors.HUMIDITY. significant_threshold | 5.0 | Настройка значимости при измерении влажности |
| core.sensors.GAS_CH4 core.sensors.GAS_O2 core.sensors.GAS_CO core.sensors.GAS_CO2 | | Коэффициенты пересчета для показаний газоанализатора, пример: <i>{k: 0.01, units: "% об.д."}</i> где k – коэффициент, units – единицы измерения |
| core.sensors.QS_LOSS_IAA. alarm_threshold | [50.0, 2] | Задание уставки по уровню доступности устройства, задается аналогично параметрам газа. |
| core.sensors.VOLTAGE_EXT. alarm_threshold | [10.0, 10] | Задание уставки по внешнему напряжению, задается аналогично параметрам газа. |

| Параметр | Значение по умолчанию | Описание |
|--|-----------------------|---|
| core.sensors.VOLTAGE_INT. alarm_threshold | [4.8, 10] | Задание уставки по внутреннему напряжению, задается аналогично параметрам газа. |
| core.sensors.GAS.aggregation_buffer | 600000 | Период за который усредняются показания газа на выработках, мс |
| core.sensors.GAS.failure_threshold | 10 | Предельное количество циклов опроса ожидания данных газоанализатора, после которого генерируется оповещение об отсутствии данных |
| core.sensors.GAS_CH4. alarm_threshold core.sensors.GAS_O2. alarm_threshold core.sensors.GAS_CO. alarm_threshold core.sensors.GAS_CO2. alarm_threshold | [0.0, 0] | Задание уставок по концентрации газов. При срабатывании уставки на сервере создается событие (тревога). Пример 1: <i>{h: 0.8, H: 1, n: 3}</i> где h – предаварийная высокая уставка, H – аварийная высокая уставка, n – количество циклов превышения уставки, после которого создается событие. Пример 2: <i>{L: 19, l: 20, n: 3}</i> где L – аварийная низкая уставка, l – предаварийная низкая уставка |
| Настройки постобработки результатов работы сервера позиционирования | | |
| core.locations. significant_time_limit | 600000 | В случае если человек не передвигается, то показания считаются значимыми по истечению указанного в параметре времени. Параметр задается в миллисекундах. <i>Значимые показания записываются в БД.</i> |

| Параметр | Значение по умолчанию | Описание |
|---|-----------------------|--|
| core.locations.significant_distance_limit | 100.0 | Минимальное расстояние, на которое должен переместиться человек, чтобы это перемещение считалось значимым, указывается в метрах |
| core.locations.enter_edge_count_limit | 5 | Верхний количественный лимит фиксации перехода на ребро графа при отсутствии информации о нахождении на других ребрах в циклах (для мобильных устройств цикл составляет 2 секунды) |
| core.locations.enter_edge_time_limit | 60 | Верхний временной лимит фиксации перехода на ребро графа при отсутствии информации о нахождении на других ребрах в секундах |
| Настройки автоматических аварийных оповещений | | |
| core.area_signal.device_types | [“TAG”] | Список типов устройств, для которых необходимо производить отсылку аварийного оповещения при входе в особые области |
| core.area_signal.recreate_period | 300 | Периодичность повторной отправки аварийного оповещения на устройство после его входа в особые области, сек |
| Настройки работы со стационарными устройствами | | |
| core.rollback_area_after_upgrade | true | Восстановление геосегмента стационарных устройств после обновления встроенного ПО |

| Параметр | Значение по умолчанию | Описание |
|--|-------------------------------|--|
| core.validate_anchor _arrangement.period | 24 | Размер в часах временного окна набора статистики для сравнения карты замеров расстояний между стационарными устройствами с их расстановкой на графе |
| core.validate_anchor _arrangement.error | 10.0 | Предельно допустимая величина в метрах расхождения статистики реальных замеров и данных о расположении стационарных устройств на графе. |
| Настройки работы с автоматически создаваемыми зонами (автозонами) | | |
| core.autoareas.enable | true | Флаг автоматического создания областей под сервисные автозоны (области создаются под каждую точку доступа независимо) |
| core.autoareas.coords_policy | [[0,0,0], [10,0,0], [0,0,10]] | Политика создания ребер для автозон. Формат: [startPoint, edgeShift, edgeVector] startPoint – начальная вершина первой автозоны edgeShift – сдвиг начальной вершины при переходе к следующей автозоне edgeVector – вектор между начальной и конечной вершинами ребра автозоны |

| Параметр | Значение по умолчанию | Описание |
|---|-----------------------------------|--|
| core.autoareas.zones_policy | {254:["LAMP","Ламповая \${mac}"]} | Политика создания автозон для разных геосегментов, в формате: {<Номер ГС>: [«Тип зоны», «Название зоны»],...} У ламповых следует выставлять тип «LAMP», у поверхностных зон – «SERVICE» |
| Настройки работы с групповыми аварийными оповещениями | | |
| core.emergency_notifications.resend_period | 60 | Периодичность повторной отправки аварийного оповещения на устройство при отсутствии подтверждения успешности предыдущего в секундах |
| core.emergency_notifications.resend_count | 50 | Предельное количество повторений отправки аварийного оповещения на устройство при отсутствии подтверждения успешности предыдущего |
| core.emergency_notifications.timeout | 180 | Предельная длительность ожидания окончания выполнения задачи групповой рассылки аварийного оповещения в минутах |
| Настройки Netty-движка (работа с сетью на низком уровне) | | |
| netty.use_epoll | false | Флаг использования EPOOL |
| Настройки встроенного VoIP модуля | | |
| voip.address | автоопределение | IP-адрес интерфейса, на котором VoIP-сервер будет принимать сообщения |

| Параметр | Значение по умолчанию | Описание |
|--|------------------------|---|
| voip.port | 5060 | Порт, на котором VoIP-сервер будет принимать сообщения |
| voip.rtp_start_port | 8000 | Начальный порт для RTP-трафика SIP-стека (для внешних VoIP подключений) |
| voip.rtp_num_ports | 1000 | Количество портов для RTP-трафика SIP-стека (для внешних VoIP подключений) |
| voip.users | {“9999”:”VoilocGate”} | Массив учётных записей внешних абонентов при использовании SIP/FХО шлюза. Учётные записи указываются в формате {user1:pass1, user2:pass2}. |
| voip.calls_path | /opt/voiloc/var/calls/ | Путь к записанным голосовым сообщениям. <i>Внимание: путь указан внутри контейнера, не рекомендуется изменять этот параметр</i> |
| Настройки системы мониторинга состояния сервера | | |
| core.sys_monitor.cmd | | Путь к скрипту обработки сообщений от системы мониторинга. Скрипт вызывается с параметрами командной строки <ts> <tag> <i>Путь указывается относительно файловой системы контейнера, рекомендуется использовать каталог логов, например: /opt/voiloc/monitor.sh core.sys_monitor.cpu_usage</i> |

| Параметр | Значение по умолчанию | Описание |
|---|----------------------------|---|
| core.sys_monitor.cpu_usage | | Настройки мониторинга загрузки CPU в формате: [порог загрузки %, период в минутах, частота повтора уведомления в минутах] Пример: [80, 10, 60] |
| core.sys_monitor.mem_usage | | Настройки мониторинга загрузки ОЗУ в формате: [порог загрузки в %, период в минутах, частота повтора уведомления в минутах] Пример: [80, 10, 60] |
| core.sys_monitor.disk_usage | [] | Настройки мониторинга заполнения диска, например: [["/opt/voiloc/calls", 80, 10, 60], ["/opt/voiloc/logs", 80, 10, 60]] Каталоги указываются относительно контейнера, параметры в [] аналогичны параметрам для CPU и ОЗУ |
| core.sys_monitor.log_ignore | [] | Настройки мониторинга лога - набор масок для игнорирования событий в логе, например: [".*No valid measurements.*", ".*Impossible to init location.*"] |
| Настройки системы кэширования архива | | |
| graphql.cache_path | /opt/voiloc/graphql_cache/ | Путь по которому хранятся кэши GraphQL-запросов. <i>Внимание:</i> путь указывается относительно контейнера, не рекомендуется его менять |
| graphql.cache_build_step | 60 | Интервал сбора данных по-умолчанию, сек |
| graphql.cache_storage_period | 168 | Глубина хранения данных по-умолчанию, час |

| Параметр | Значение по умолчанию | Описание |
|--------------------------------------|-----------------------|--|
| Настройки системы авторизации | | |
| http.auth | false | Признак активности режима авторизации через JWT-токены (true/false) |
| http.auth.guests | true | Признак активности гостевого доступа при авторизации через JWT-токены (true/false) |
| http.auth.public_key | | Путь к публичному ключу. <i>Внимание:</i> путь указывается относительно контейнера, не рекомендуется его менять |

4.2.3 Настройка журналов Voiloc

Файл *logback.xml* содержит настройки подсистемы логирования СМС Исеть (/opt/voiloc/var/log/voiloc.log). Файл имеет иерархическую структуру, пример файла приведён в Листинге 7:

```
<?xml version="1.0" encoding="UTF-8" ?>
<configuration>

  <appender name="STDOUT" class="ch.qos.logback.core.ConsoleAppender">
    <filter class="ch.qos.logback.classic.filter.ThresholdFilter">
      <level>WARN</level>
    </filter>
    <encoder>
      <charset>UTF-8</charset>
      <pattern>%yellow(%date{"yyyy-MM-dd'T'HH:mm:ss.SSSXXX", UTC}) %highlight(%5level) %-8marker %msg%n</pattern>
    </encoder>
  </appender>

  <!-- Основной файл лога ===== -->
  <property name="log.dir" value="${voiloc.log.dir:-.}/">
  <property name="log.file" value="${voiloc.log.file:-voiloc.log}"/>

  <appender name="FILE" class="ch.qos.logback.core.rolling.RollingFileAppender">
    <file>${log.dir}/${log.file}</file>
    <rollingPolicy class="ch.qos.logback.core.rolling.TimeBasedRollingPolicy">
      <fileNamePattern>${log.dir}/${log.file}.%d{yyyy-MM-dd}</fileNamePattern>
      <maxHistory>15</maxHistory>
      <maxFileSize>500MB</maxFileSize>
    </rollingPolicy>
    <encoder>
      <charset>UTF-8</charset>
      <pattern>%date{"yyyy-MM-dd'T'HH:mm:ss.SSSXXX", UTC} %5level %-8marker %msg%n</pattern>
    </encoder>
  </appender>

  <!-- Логирование IoM трафика от устройств -->
  <logger name="ru.lab127.voiloc.iom" level="DEBUG"/>

  <!-- Логирование HTTP запросов -->
  <logger name="ru.lab127.voiloc.http" level="WARN">
    <appender-ref ref="STDOUT"/>
    <appender-ref ref="FILE"/>
  </logger>

  <!-- Логирование GraphQL запросов -->
  <logger name="graphql" level="WARN"/>

  <!-- System monitor -->
  <logger name="ru.lab127.voiloc.core.modules.system" level="WARN"/>
```

```

<!-- Логирование запросов к БД -->
<logger name="ru.lab127.tools.common.db" level="WARN"/>

<!-- Логирование связи с ПО Вентиляция 2 -->
<logger name="ru.lab127.voiloc.sip" level="DEBUG"/>

<!-- Логирование модуля OPC -->
<logger name="org.openscada" level="WARN"/>

<logger name="io.netty" level="WARN"/>
<logger name="com.zaxxer.hikari" level="WARN"/>
<logger name="org.jinterop" level="WARN"/>
<logger name="org.apache.commons.beanutils.converters" level="WARN"/>
<logger name="ru.lab127.voiloc.core.modules" level="WARN"/>

<!-- Фильтр для отслеживания перезагрузок сервера -->
<turboFilter class="ch.qos.logback.classic.turbo.MarkerFilter">
  <Marker>INIT</Marker>
  <onMatch>ACCEPT</onMatch>
</turboFilter>
<!-- ===== -->

<!-- Логирование действий пользователей ===== -->
<property name="action.dir" value="${log.dir}/actions/">
<property name="action.file" value="actions.log"/>

<appender name="ACTIONS" class="ch.qos.logback.core.rolling.RollingFileAppender">
  <file>${action.dir}/${action.file}</file>
  <rollingPolicy class="ch.qos.logback.core.rolling.TimeBasedRollingPolicy">
    <fileNamePattern>${action.dir}/${action.file}.%d{yyyy-MM-dd}</fileNamePattern>
    <maxHistory>15</maxHistory>
    <maxFileSize>500MB</maxFileSize>
  </rollingPolicy>
  <encoder>
    <charset>UTF-8</charset>
    <pattern>%date{"yyyy-MM-dd'T'HH:mm:ss.SSSXXX", UTC} %5level %-8marker %msg%n</pattern>
  </encoder>
  <turboFilter class="ch.qos.logback.classic.turbo.MarkerFilter">
    <Marker>HTTP</Marker>
  </turboFilter>
</appender>

<logger name="ru.lab127.voiloc.http" level="INFO" additivity = "false">
  <appender-ref ref="ACTIONS"/>
</logger>

<!-- ===== -->

<root level="WARN">
  <appender-ref ref="STDOUT"/>
  <appender-ref ref="FILE"/>
</root>

</configuration>

```

Листинг 7 Пример конфигурационного файла *logback.xml*

Блоки *appender* задают выходной файл для логов, в СМС Исеть используется 3 файла: *voiloc.log*, *action.log* и *stdout*. Файл *action.log* содержит журнал действий пользователя, не рекомендуется изменять его настройки. *Stdout* – это вывод в консоль, вывод можно посмотреть в логах контейнера *voiloc-core*, предназначен для логирования значимых ошибок в работе сервера, которые можно будет посмотреть в аварийной ситуации, его настройки также не рекомендуется увеличивать, т.к. это увеличение размера логов контейнера на диске.

Секция *root level* регулирует уровень детализации журналов по умолчанию; строки *logger name* = «имя журнала» регулируют уровень детализации целевого журнала.

Параметр *level* может принимать следующие значения (указано в порядке уменьшения уровня детализации): *TRACE*, *DEBUG*, *INFO*, *WARN*, *ERROR*. *TRACE* означает вывод максимально подробной информации в журнал, *ERROR* — вывод только в случае возникновения ошибок.

Допускается изменение уровней логирования журналов компонент *IoM* и *СИИ*. При пусконаладочных работах не рекомендуется повышать уровень детализации выше *DEBUG*, чем выше уровень детализации (для крупных инсталляций) тем сильнее увеличивается размер файла журнала, что может приводить к существенному увеличению нагрузки на дисковую подсистему сервера и переполнению жёсткого диска.

4.2.4 Дополнительные настройки системы

Для нормальной работы сервера необходимо регулярно обновлять журналы БД, что ускоряет работу самой БД. Удобнее всего настроить автоматическое выполнение данной процедуры, создав файл в каталоге */etc/cron.daily/*, для создания файла необходимо выполнить команду:

```
sudo nano /etc/cron.daily/voilocdb_analyze.sh
```

В файл необходимо записать следующее:

```
#!/bin/bash
su -c "psql -h localhost -c ANALYZE -d voilocdb" postgres
```

Далее необходимо сделать файл исполняемым:

```
sudo chmod 755 /etc/cron.daily/voilocdb_analyze.sh
```

4.2.5 Резервирование сервера Voiloc

Если необходимо создать резервный сервер Voiloc, то необходимо воспользоваться механизмом репликации БД PostgreSQL, остальные данные необходимо синхронизировать с помощью *rsync*.

В одной сети не должно быть двух работающих серверов Voiloc – это может привести к неправильной работе системы позиционирования и, что особенно важно, неправильной работы системы аварийного оповещения!

Настройка основного сервера

Перед внесением изменений в настройки на основном сервере необходимо сделать резервные копии изменяемых конфигурационных файлов службы PostgreSQL (каталог БД может отличаться, в данном примере он соответствует описанной выше инструкции по установке сервера):

```
sudo cp /opt/voiloc/db_data/postgresql.conf /opt/voiloc/db_data/postgresql.conf.bak
sudo cp /opt/voiloc/db_data/pg_hba.conf /opt/voiloc/db_data/pg_hba.conf.bak
```

Далее на основном сервере необходимо внести изменения в настройки PostgreSQL, для этого открыть файл */opt/voiloc/db_data/postgresql.conf*:

```
sudo nano /opt/voiloc/db_data/postgresql.conf
```

В файле необходимо изменить (в том числе удалить из начала строк настроек знак *#*) некоторые настройки, среди которых необходимо подобрать значение параметра *wal_keep_segments* (количество файлов журнала БД, которые передаются с основного на резервный сервер), которое должно удовлетворять следующим условиям: не занимать слишком много места на основном сервере (один файл занимает 16 Мб) и быть достаточным для временного отключения резервного сервера. Отталкиваться при выборе значения следует от того, на какой объем БД увеличивается за сутки (для справки, при максимальном режиме записи данных, каждые 50 человек, постоянно находящихся в шахте, увеличивают объем БД

на 10 Гб). Остальные изменяемые в файле настройки с рекомендуемыми значениями приведены ниже:

```
listen_addresses = '*'
wal_level = hot_standby
max_wal_senders = 2
wal_keep_segments = 10000
max_replication_slots = 2
hot_standby = on
hot_standby_feedback = on
```

После этого необходимо создать специального пользователя с именем *repluser*, через которого будет производится репликация, следующей командой (где вместо *password* нужно ввести пароль для создаваемого пользователя):

```
docker exec -it voiloc-db bash
bash-5.1# sudo su -c psql postgres
postgres=# CREATE ROLE repluser WITH REPLICATION PASSWORD 'password' LOGIN;
CREATE ROLE
postgres=# \q
bash-5.1# exit
```

Далее нужно разрешить подключение к БД через сеть, для этого необходимо открыть файл */opt/voiloc/db_data/pg_hba.conf*:

```
sudo nano /etc/postgresql/9.6/main/pg_hba.conf
```

В файле нужно добавить следующие строки, где *<IP_MASTER>* – это IP адрес основного сервера, а *<IP_SLAVE>* – IP адрес резервного сервера:

| | | | | |
|------|-------------|----------|----------------|-----|
| host | all | all | <ip_MASTER>/32 | md5 |
| host | all | all | <ip_SLAVE>/32 | md5 |
| host | replication | repluser | <ip_MASTER>/32 | md5 |
| host | replication | repluser | <ip_SLAVE>/32 | md5 |

После внесения всех изменений необходимо перезапустить контейнер БД на основном сервере:

```
docker restart voiloc-db
```

Кроме того, необходимо разрешить доступ к каталогу с записями звонков:

```
sudo chmod o+rx /opt/voiloc/var/calls
```

Также необходимо установить пакет *rsync* (если он не был установлен изначально):

```
sudo dnf install rsync
```

Настройка резервного сервера

На резервном сервере необходимо установить Voiloc по настоящей инструкции с теми же настройками и дополнительным ПО, что и на основном сервере (кроме IP адреса). Оба сервера должны быть в одной локальной сети.

После установки на резервном сервере нужно дополнительно установить пакеты *rsync* и *sshpass* (если они не были установлены):

```
sudo dnf install rsync sshpass
```

Далее нужно перейти в каталог с дистрибутивом стэк сервера Voiloc (там где находится файл *docker-compose.yml*), и набрать команду для остановки стэка:

```
docker-compose down
```

Далее нужно запустить контейнер с БД в интерактивном режиме (опция `-v` указывает на каталог сервера, который будет перемонтирован к каталогу контейнера на время текущего сеанса в формате `<каталог на сервере>:<каталог в контейнере>`, в нем следует указать каталог, выделенный под БД):

```
docker run -it --rm -v /opt/voiloc/db_data:/var/lib/postgresql/data lab127/voiloc-db:9.6 /bin/bash
```

Внутри контейнера необходимо очистить каталог с данными и загрузить в него данные с основного сервера:

```
bash-5.1# rm -rf /var/lib/postgresql/data
bash-5.1# pg_basebackup -R -P --host=<IP_MASTER> --port 5432 --username=repluser
--pgdata /var/lib/postgresql/data --xlog-method=stream --write-recovery-conf
```

Система предложит ввести пароль пользователя *repluser*, после ввода пароля PostgreSQL начнет копировать данные с основного сервера. Данная процедура может занять до нескольких часов в зависимости от накопленных данных на основном сервере, поэтому рекомендуется в этот момент не запрашивать больших отчетов на основном сервере.

После окончания копирования данных с основного на резервный сервер необходимо выйти из контейнера командой:

```
bash-5.1# exit
```

На резервном сервере должен появиться файл *recovery.conf* в каталоге данных БД, пока он существует PostgreSQL будет считать сервер резервным.

Далее необходимо запустить на резервном сервере только контейнер с БД:

```
docker-compose up voiloc-db -d
```

Проверить работу репликации можно следующим способом, подключиться на резервном сервере к БД, набрав команды:

```
docker exec -it voiloc-db /bin/bash
bash-5.1# su -c "psql -d voilocdb" postgres
```

Далее вы попадете в консоль запросов к БД, определить, что данные с основного сервера копируются на резервный можно, запросив время создания последней записи в какой-нибудь таблице (при этом основной сервер должен быть в работоспособном состоянии). Пример такого запроса и ответа приведен ниже:

```
voilocdb=> SELECT MAX(_ts) FROM voiloc.devices;
           max
-----
2022-04-29 15:24:12.608841+05
(1 row)
```

Данный запрос необходимо произвести не менее двух раз с разницей в 1-2 минуты между попытками, метка времени в ответе должна увеличиться, что и будет означать работу репликации. Для выхода из консоли БД и контейнера необходимо набрать команды:

```
voilocdb=> \q
bash-5.1# exit
```

Для копирования звонков необходимо разрешить доступ к каталогам сервера:

```
sudo chmod 777 /opt/voiloc /opt/voiloc/etc /opt/voiloc/logs /opt/voiloc/calls
/opt/voiloc/graphql_cache /opt/voiloc/iset_auth_data
```


Далее необходимо проверить наличие связи с основным сервером по ssh и получить сертификат сервера, сделать это можно командой (где вместо `<user>` подставить имя пользователя на основном сервере, а вместо `<ip_MASTER>` подставить IP адрес основного сервера):

```
ssh <user>@<ip_MASTER>
```

Сервер выдаст сообщение, на которое нужно в ответ написать *yes* и нажать *Enter*:

```
The authenticity of host '192.168.0.1 (192.168.0.1)' can't be established.  
ECDSA key fingerprint is SHA256:eIAw6ndlS6quLM+2xu36zo+TXfxiRIkO6smXgtWdaLg.  
Are you sure you want to continue connecting (yes/no/[fingerprint])?
```

Далее будет предложено ввести пароль пользователя от основного сервера:

```
Warning: Permanently added '192.168.0.1' (ECDSA) to the list of known hosts.  
user@192.168.0.1's password:
```

В случае успешного подключения появится консоль основного сервера:

```
user@Voiloc:~$
```

Для выхода из нее нужно ввести команду *exit* и нажать *Enter*:

```
user@Voiloc:~$ exit
```

Далее на резервном сервере нужно создать задание планировщика для копирования данных с основного сервера, для чего ввести команду:

```
crontab -e
```

Откроется консоль редактирования файла планировщика, в конец файла нужно добавить указанные ниже строки (где `<user>` – имя пользователя на основном сервере, `<pass>` – пароль пользователя, `<IP_MASTER>` – IP адрес основного сервера), *файл обязательно должен заканчиваться пустой строкой*:

```
0 * * * * sshpass -p <pass> rsync -ctvr --delete -e ssh  
<user>@<IP_MASTER>:/opt/voiloc/calls/ /opt/voiloc/calls/  
0 * * * * sshpass -p <pass> rsync -ctvr --delete -e ssh  
<user>@<IP_MASTER>:/opt/voiloc/etc/ /opt/voiloc/etc/  
0 * * * * sshpass -p <pass> rsync -ctvr --delete -e ssh  
<user>@<IP_MASTER>:/opt/voiloc/auth_data/ /opt/voiloc/auth_data/  
0 * * * * sshpass -p <pass> rsync -ctvr --delete -e ssh  
<user>@<IP_MASTER>:/opt/voiloc/graphql_cache/ /opt/voiloc/graphql_cache/
```

После сохранения файла резервный сервер будет синхронизировать каталоги звонков, настроек, данных авторизации и кэша в начале каждого часа. Если необходимо добавить синхронизацию еще каких-либо каталогов, то нужно добавить аналогичные строки для них.

Переход на резервный сервер

В случае поломки основного сервера для перехода на резервный необходимо отключить от сети основной сервер и внести изменения в настройки резервного. Для этого нужно остановить контейнеры на резервном сервере, для этого нужно перейти в каталог со стеком Voiloc и ввести команду:

```
docker-compose down
```

Затем необходимо установить версию стека Voiloc, аналогичную той, что была установлена на основном сервере (если они отличаются), после чего снова остановить контейнеры.

Далее необходимо удалить файл *recovery.conf*:

```
docker exec -it voiloc-db bash
bash-5.1# rm -f /var/lib/postgresql/data/recovery.conf
bash-5.1# exit
```

Затем нужно перезапустить стек Voiloc:

```
docker-compose down
docker-compose up -d
```

Далее необходимо удалить из планировщика задания на копирование записей звонков, настроек, данных авторизации и кэша с основного сервера, запустив планировщик:

```
crontab -e
```

И удалив из открывшегося файла строки вида:

```
0 * * * * sshpass -p <pass> rsync -ctvr --delete -e ssh
<user>@<IP_MASTER>:/opt/voiloc/calls/ /opt/voiloc/calls/
```

После запуска сервера нужно будет поменять IP адрес сервера в настройках клиентов СМС Исеть.

После этого репликация между серверами будет разорвана. Для восстановления репликации необходимо настроить бывший резервный сервер как основной, а бывший основной настроить в качестве резервного.

Если на основном сервере используется служба DHCP сервера, то на резервном сервере его необходимо только установить, но не задавать ему настройки. Настроить службу DHCP сервера на резервном сервере можно только после перехода установления его в качестве основного. Не допускается одновременная работа двух DHCP серверов в одной локальной сети!

4.2.6 Создание долговременных архивов и чистка БД

Для создания долговременных архивов рекомендуется использовать утилиту *pg_dump*, которая создает полный слепок базы данных. Создавать такие архивы следует не чаще 1 раза в месяц с целью освободить место на высокоскоростном SSD диске сервера для текущих данных.

Создание архива

Для создания архива необходимо перейти в контейнер *voiloc-db* и выполнить команду, имя файла архива (*dump2024-04-01.sql*) может быть любым:

```
docker exec -it voiloc-db bash
bash-5-1# su postgres
/ $ pg_dump --dbname=voilocdb --schema='voiloc' --format=p --if-exists --clean -
-file '/var/lib/postgresql/data/dump2024-04-01.sql'
/ $ exit
bash-5.1# exit
```

Размер файла обычно получается в 2 раза меньше, чем размер каталога с БД. Создание архива может занять несколько часов, во время создания архива не рекомендуется запрашивать в ПО Оператор шахты отчеты и обращаться к архивным данным. Файл архива будет находиться в каталоге с данными БД.

После создания архива БД необходимо удалить старые данные (для нормального функционирования СМС Исеть рекомендуется оставлять в БД данные хотя бы за один месяц).

Удаление старых данных осуществляется следующими командами (все данные до указанной в команде будут удалены):

```
docker exec -it voiloc-db bash
bash-5-1# su postgres
/ $ psql -c "SELECT db__cleanup(voiloc.iso_to_timestamp('2024-03-01T00:00:00.000Z'))" -d voilocdb
/ $ psql -c 'ANALYZE' -d voilocdb
/ $ exit
bash-5.1# exit
```

После удаления старых данных рекомендуется перезагрузить контейнер voiloc-core.

Также необходимо сохранить записи разговоров командой (где вместо /cat/ подставить путь до каталога с архивами):

```
tar -czvf ~/calls2024-04-01.tar.gz /opt/voiloc/calls/
tar -czvf ~/actions2024-04-01.tar.gz /opt/voiloc/logs/actions/
```

Также вместе с ними следует сохранить стэк сервера *voiloc*, т.к. сохраненный архив может быть не совместим с более новым ПО в случае его обновления на сервере.

Восстановление данных из архива

Рекомендуется разворачивать архив на отдельном компьютере. На компьютере, на котором будет разворачиваться дамп с архивом, необходимо установить СМС Исеть в соответствии с настоящей инструкцией. *Компьютер, на котором разворачивается архив должен быть изолирован от сети СМС Исеть.*

После установки необходимо скопировать файл архива БД в каталог с данными БД, остановить весь стэк СМС Исеть и запустить только контейнер voiloc-db:

```
docker-compose down
docker-compose up voiloc-db -d
```

Далее нужно загрузить архив в БД:

```
docker exec -it voiloc-db bash
bash-5.1# su -c "psql -f dump2024-04-01.sql -d voilocdb" postgres
bash-5.1# exit
```

Также необходимо развернуть архив со звонками и действиями пользователя, для этого нужно скопировать его на компьютер и выполнить команды:

```
sudo tar -xzf ~/calls2024-04-01.tar.gz -C /
sudo tar -xzf ~/actions2024-04-01.tar.gz -C /
```

После разворачивания архивов нужно запустить все контейнеры:

```
docker-compose up -d
```

Архивные данные будут доступны по адресу *http://<IP адрес компьютера>:8090*