СОДЕРЖАНИЕ

Перечень сокращений и условных сокращений

API	–	(от англ. Application Programming Interfaces) интерфейс программирования приложений, интерфейс прикладного программирования;
DNS	–	(от англ. Domain Name System) система доменных имён;
DoS	–	(от англ. Denial of Service) отказ в обслуживании;
ICMP		протокол межсетевых управляющих сообщений
ISO	–	(от англ. International Organization for Standardization) Международная организация по стандартизации
JRE	–	(от англ. Java Runtime Environment) среда выполнения для Java;
RMON		стандартная контрольная спецификация удаленного мониторинга, которая позволяет различным сетевым мониторам и консольным системам обмениваться данными мониторинга сети;
SNMP		протокол сетевого управления, который предоставляет данные управления в виде переменных в управляемых системах, организованных в информационной базе управления (MIB), которые описывают состояние системы и ее конфигурацию;
WinAPI	–	(от англ. Windows API) интерфейс программирования приложений в операционных системах семейства Microsoft Windows;
ИС	–	информационные системы;
МДЦ	–	Международный детский центр;
ОС	–	операционные системы;
ПО	–	программное обеспечение;
ФГБОУ	–	Федеральное государственное бюджетное образовательное учреждение;

ВВЕДЕНИЕ

Стремительное развитие локальных вычислительных сетей (ЛВС) сопровождается увеличением не только количества подключаемых персональных компьютеров, но и серверных станций и специальных сетевых устройств, обеспечивающих их поддержку и функционирование. К специальным сетевым устройствам относятся маршрутизаторы, коммутаторы, медиаконвертеры, межсетевые экраны и т. д. Администраторам ЛВС все сложнее осуществлять мониторинг работоспособности узлов ЛВС и ежедневно анализировать регистрационную информацию, относящуюся к сети в целом.

Проблема, исследуемая в выпускной квалификационной работе. Бесперебойное функционирование локальной вычислительной сети невозможно без программного обеспечения, проводящего мониторинг и диагностику работоспособности сети. Имеющееся программное обеспечение либо является слишком многофункциональным, сложным и трудоемким в обслуживании, либо дорогим и требующим дополнительного сопровождения разработчика, либо не решает ряда необходимых задач анализа. Разработка информационной системы мониторинга и анализа ЛВС даст возможность отслеживать проблемы предоставления ресурсов локальной вычислительной сети и предоставлять качественные интернет-услуги пользователям лагеря «Артек».

Цель выпускной квалификационной работы – анализ технологий мониторинга локальной вычислительной сети и на основе проведенного исследования разработка информационной системы мониторинга локальной вычислительной сети лагеря «Артек»

Задачи выпускной квалификационной работы:

1. Анализ технологий мониторинга ЛВС.
2. Исследование средств проектирования информационной системы.

3. Разработка информационной системы мониторинга локальной вычислительной сети лагеря «Артек».

Объект и предмет исследования выпускной квалификационной работы

Объект исследования выпускной квалификационной работы – локальная вычислительная сеть.

Предмет исследования выпускной квалификационной работы – технологии и средства проектирования и разработки информационной системы мониторинга локальной вычислительной сети лагеря «Артек».

Теоретическая значимость выпускной квалификационной работы состоит в описании технологии разработки информационной системы мониторинга и анализа локальной вычислительной сети лагеря «Артек».

Практическая значимость выпускной квалификационной работы состоит в практическом применении предложенной технологии и внедрении информационной системы мониторинга ЛВС в лагере «Артек»

Новизна магистерского исследования составила следующие положения:

1) Предложен мониторинг ЛВС в реальном времени с целью обнаружения сбоев сетевого оборудования за счет обработки результатов эхо-запросов по протоколу ICMP.

2) Получило дальнейшее развитие управление устройствами сети с целью ликвидации неисправностей за счет удаленного доступа по протоколу SSH.

3) Разработано графическое представление состояния узлов сети в определенные промежутки времени для отслеживания периодичности сбоев за счет организации хранения результатов мониторинга в БД SQLite.

Исследование, проведенное в выпускной квалификационной работе, прошло апробацию на конференции «Дистанционные образовательные технологии», а также была опубликована статья:

Информационная система мониторинга компьютерной сети международного детского центра «Артек» / В.Н. Таран, Д.С. Шкабатур //

Дистанционные образовательные технологии: материалы III Всероссийской научно-практической интернет-конференции (г. Ялта, 17-22 сентября 2018года) / отв. ред. В.Н. Таран. – Симферополь: ИТ «АРИАЛ», 2018. – С. 180-184.

РАЗДЕЛ 1. ИССЛЕДОВАНИЕ ТЕХНОЛОГИЙ МОНИТОРИНГА ЛОКАЛЬНОЙ ВЫЧИСЛИТЕЛЬНОЙ СЕТИ

1.1. Технологии мониторинга компьютерных сетей

В последние годы информационные технологии претерпевают значительные и постоянные изменения. По некоторым оценкам, за последние пять лет объем сетевого трафика локальных сетей вырос в десять раз. Таким образом, локальные сети должны обеспечивать все большую пропускную способность и необходимый уровень качества обслуживания. Однако какие бы ресурсы ни имела сеть, они все-таки конечны, поэтому для сети необходима возможность управления трафиком.

А для того чтобы управление было максимально эффективным, требуется возможность контроля над пакетами, передающимися между устройствами вашей сети. Также у администратора существует великое множество обязательных для исполнения ежедневных операций. Это, например, проверка правильности функционирования электронной почты, просмотр регистрационных файлов на предмет выявления ранних признаков неисправностей, контроль за подключением локальных сетей и за наличием системных ресурсов. И здесь на помощь могут прийти средства, применяемые для мониторинга и анализа вычислительных сетей.

Чтобы не запутаться в многообразии методик, средств и продуктов, созданных для мониторинга, начнем с краткого описания нескольких крупных классов этих продуктов.

Системы управления сетью (Network Management Systems) – это централизованные программные системы, которые собирают данные о состоянии узлов и коммуникационных устройств сети, а также о трафике, циркулирующем в сети. Эти системы не только осуществляют мониторинг и анализ сети, но и выполняют в автоматическом или полуавтоматическом режиме действия по управлению сетью – включение и отключение портов

устройств, изменение параметров адресных таблиц мостов, коммутаторов и маршрутизаторов и т.п.

Средства управления системой. Средства управления системой часто выполняют функции, аналогичные функциям систем управления, но по отношению к другим объектам. В первом случае объектом управления является программное и аппаратное обеспечение компьютеров сети, а во втором – коммуникационное оборудование. Вместе с тем некоторые функции этих двух видов систем управления могут дублироваться, например, средства управления системой могут выполнять простейший анализ сетевого трафика.

Встроенные системы диагностики и управления. Эти системы выполняются в виде программно-аппаратных модулей, устанавливаемых в коммуникационное оборудование, а также в виде программных модулей, встроенных в операционные системы. Они выполняют функции диагностики и управления только одним устройством, и в этом их основное отличие от централизованных систем управления.

Анализаторы протоколов. Представляют собой программные или аппаратно-программные системы, которые ограничиваются, в отличие от систем управления, лишь функциями мониторинга и анализа трафика в сетях. Хороший анализатор протоколов может захватывать и декодировать пакеты большого количества применяемых в сетях протоколов – обычно несколько десятков. Анализаторы протоколов позволяют установить некоторые логические условия для захвата отдельных пакетов и выполняют полное декодирование захваченных пакетов, то есть показывают в удобной для специалиста форме вложенность пакетов протоколов разных уровней друг в друга с расшифровкой содержания отдельных полей каждого пакета.

Экспертные системы. Системы этого вида аккумулируют человеческие знания о выявлении причин аномальной работы сетей и возможных способах приведения сети в работоспособное состояние. Экспертные системы часто реализуются в виде отдельных подсистем различных средств мониторинга и

анализа сетей: систем управления сетями, анализаторов протоколов, сетевых анализаторов. Простейшим вариантом экспертной системы является контекстно-зависимая help-система. Более сложные экспертные системы представляют собой так называемые базы знаний, обладающие элементами искусственного интеллекта.

Многофункциональные устройства анализа и диагностики. В последние годы в связи с повсеместным распространением локальных сетей возникла необходимость разработки недорогих портативных приборов, совмещающих функции нескольких устройств: анализаторов протоколов, кабельных сканеров и даже некоторых возможностей ПО для сетевого управления.

Кроме визуального наблюдения за состоянием сетевых устройств, программное обеспечение дает возможность организации проверки хостов и служб (в том числе и локальных ресурсов или интернет-серверов) с помощью разнообразных сетевых протоколов настройки и использования удобного способа оповещения системных администраторов о положительных или отрицательных результатах тестирования. Способы могут быть разными: появление сообщения на экране компьютера ИТ специалиста, специальный звук, отправка электронного письма или смс на телефон. Приложение для мониторинга компьютеров в локальной сети предприятия в ряде случаев может перезапустить какую-либо удаленную службу либо выполнить заранее написанный для него скрипт (тогда некоторые сбои будут устраняться в автоматическом режиме).

Если в программе реализована такая функция, то все подключенные к сети устройства будут наглядно отображаться на ее схеме. Уже по одному виду их иконок специалисту станет понятно, какие из них работают нормально, а какие функционируют неверно. Такая возможность облегчает диагностику групповых сбоев. Полученные результаты тестирования заносятся в единую базу данных; по мере накопления статистической

информации можно будет строить графики, чтобы изучать изменение отклика устройств и отслеживать другие проверяемые параметры.

Современные программы мониторинга компьютеров в локальной сети позволяют создать своего рода пульт управления сетевой инфраструктурой предприятия, при помощи которого отвечающий за сеть сотрудник сможет как наблюдать за ее важными элементами и проверять параметры оборудования, так и вести эффективное управление удаленными хостами. С помощью контекстного меню хостов можно просматривать различные данные об удаленных узлах в сети: проверять информацию по SNMP с коммутаторов, получить доступ к реестрам удаленных компьютеров, просматривать запущенные процессы и журналы событий, производить перезапуск служб и выполнять другие действия. Некоторые программы ведут не только мониторинг ресурсов компьютера, но и помогают проводить учет устройств и программных приложений на сетевых ПК. Благодаря им системный администратор имеет возможность получать практически любые сведения об аппаратном и программном обеспечении на компьютерах в корпоративной сети. Сбор данных проходит удаленно, это позволяет не мешать работать сотрудникам предприятия, и экономит рабочее время системных администраторов.

Программы наблюдения за ПК в локальной сети позволяют вести точный учет аппаратного обеспечения. ИТ специалисты смогут оперативно узнать о пропаже и неисправности какого-либо компонента либо о его замене. При обнаружении изменений они заносятся в журнал, и о них оповещается администратор сети. Если нужно отслеживать определенные параметры на рабочих станциях пользователей с конкретной частотой и получать оповещения при их изменении, возможно будет настроить сбор данных по расписанию. В этом случае мониторинг ресурсов компьютера будет вестись в автоматическом режиме.

Многие программы не только формируют отчеты по компонентам компьютеров, но и следят за их работоспособностью – контролируется функционирование жестких дисков и их температура. Когда какой-либо диск перегревается или приложение выдает прогноз о том, что он может выйти из строя, сисадмин увидит отчет, содержащий критические замечания по работе ПК с предупреждением.

При необходимости софт мониторинга ПО в локальных сетях предприятия позволяет отслеживать и изменения в установленных программах. В случае, когда права пользователей строго не ограничиваются, кто-либо из сотрудников предприятия может установить на своем компьютере нежелательное или нелицензионное приложение. При мониторинге, когда происходит инсталляция или удаление программного обеспечения, любое изменение фиксируется и заносится в журнал. Это означает, что системный администратор будет всегда знать, какие именно программы и куда были инсталлированы или откуда удалены.

Часто при мониторинге ресурсов компьютера происходит отслеживание серийных номеров и лицензий программ, подсчет количества установок ПО, контроль правильного использования серийных номеров. Все эти меры реально помогают избежать проблем при проверке корпоративного программного обеспечения на предмет лицензионной чистоты. С целью повышения уровня безопасности и отказоустойчивости компьютеров ряд специальных программ мониторинга имеют функции просмотра проведенных обновлений ПО и системы и составления отчета по работе антивирусного софта и актуальности его баз.

Данные, собираемые программой с сетевых компьютеров и отображаемые на экране ПК системного администратора, могут быть занесены в отчет. Далее их можно распечатать или экспортировать в определенную базу. Кроме автоматически собираемой информации, многие приложения для мониторинга ресурсов компьютера разрешают вводить

вручную серийные номера оборудования, номера офисов их пользователей и их контактные данные.

Таким образом, на сегодняшний день существует большое количество различных технологий мониторинга компьютерных систем, использующих различные сетевые протоколы.

1.2.Протоколы мониторинга компьютерных сетей

1.2.1. Протокол межсетевых управляющих сообщений ICMP

Протокол межсетевых управляющих сообщений ICMP преимущественно используется для уведомления об ошибках передачи данных. Он используется сетевыми устройствами, включая маршрутизаторы, для отправки сообщений об ошибках и оперативной информации, указывающей, например, что запрошенная услуга недоступна или что хост, или маршрутизатор не могут быть достигнуты. ICMP отличается от транспортных протоколов, таких как TCP и UDP, тем, что он обычно не используется для обмена данными между системами, и не используется обычным сетевым приложениям конечного пользователя (за исключением некоторых диагностических инструментов, таких как ping и traceroute).

Сообщения ICMP обычно используются для целей диагностики или контроля и генерируются в ответ на ошибки в IP-операциях (как указано в RFC 1122). ICMP-ошибки направляются на исходный IP-адрес исходного пакета.

Каждое устройство (например, промежуточный маршрутизатор), перенаправляющее IP-датаграмму, сначала уменьшает поле времени для жизни (TTL) в заголовке IP на единицу. Если итоговый TTL равен 0, пакет отбрасывается и время ICMP, повышенное в транзитном сообщении, отправляется на адрес источника дейтаграммы.

Многие широко используемые сетевые утилиты основаны на сообщениях ICMP. Команда traceroute может быть реализована путем

передачи IP-дейтаграмм с помощью специально заданных полей заголовка IP TTL и поиска времени ICMP, превышенного в пути, и недостижимых сообщений, сгенерированных в ответ. Связанная утилита ping реализована с использованием ICMP-эхо-запроса и сообщений эхо-ответа.

ICMP использует базовую поддержку IP, как если бы это был протокол более высокого уровня, однако ICMP на самом деле является неотъемлемой частью IP. Хотя ICMP-сообщения содержатся в стандартных IP-пакетах, сообщения ICMP обычно обрабатываются как особый случай, отличающийся от обычной обработки IP-адресов. Во многих случаях необходимо проверить содержимое ICMP-сообщения и доставить соответствующее сообщение об ошибке в приложение, ответственное за передачу IP-пакета, запрашивающего отправку ICMP-сообщения.

ICMP – это протокол сетевого уровня. Номер порта TCP или UDP не связан с ICMP-пакетами, так как эти числа связаны с транспортным уровнем выше.

Таким образом протокол ICMP представляет собой наиболее простой сетевой протокол, применяющийся для базового мониторинга сети и выявления проблем.

1.2.2. Протокол сетевого управления SNMP

Этот протокол интернет-стандарта для сбора и организации информации об управляемых устройствах в IP-сетях и для изменения этой информации для изменения поведения устройства. Устройства, которые обычно поддерживают SNMP, включают в себя

- кабельные модемы,
- маршрутизаторы,
- коммутаторы,
- серверы,
- рабочие станции,

– принтеры и так далее.

SNMP широко используется в управлении сетью и для сетевого мониторинга. SNMP предоставляет данные управления в виде переменных в управляемых системах, организованных в информационной базе управления (MIB), которые описывают состояние системы и ее конфигурацию. Затем эти переменные могут быть дистанционно запрошены.

Разработаны и развернуты три значимые версии SNMP.

SNMPv1 является исходной версией протокола.

Более свежие версии, SNMPv2c и SNMPv3, улучшают производительность, гибкость и безопасность.

SNMP является компонентом пакета интернет-протокола, как определено целевой группой Internet Engineering Task Force (IETF). Он состоит из набора стандартов для управления сетью, включая протокол уровня приложения, схему базы данных и набор объектов данных.

При типичном использовании SNMP один или несколько компьютеров, называемых менеджерами, выполняют задачу мониторинга управления группы хостов или устройств в компьютерной сети. Каждая управляемая система выполняет программный компонент, называемый агентом, который передает информацию через SNMP менеджеру.

Сеть, управляемая SNMP, состоит из трех ключевых компонентов:

- Управляемые устройства
- Агент - программное обеспечение, которое работает на управляемых устройствах
- Станция управления сетью (NMS) - программное обеспечение, которое работает у менеджера

Управляемое устройство является сетевым узлом, который реализует SNMP-интерфейс, который позволяет однонаправленный (только для чтения) или двунаправленный (чтение и запись) доступ к информации, специфичной для узла. Управляемые устройства обмениваются узловыми данными с NMS.

Иногда называемые сетевые элементы, управляемые устройства могут быть устройствами любого типа, включая маршрутизаторы, серверы доступа, коммутаторы, кабельные модемы, мосты, концентраторы, IP-телефоны, IP-камеры, компьютерные хосты и принтеры.

Агент - это программный модуль сетевого управления, который находится на управляемом устройстве. Агент имеет локальное знание управляющей информации и переводит эту информацию в форму, определенную SNMP.

На станции управления сетью выполняются приложения, которые контролируют и управляют управляемыми устройствами. NMS обеспечивают основную часть ресурсов обработки и памяти, необходимых для управления сетью. Одна или несколько NMS могут существовать в любой управляемой сети.

Агент SNMP предоставляет данные управления управляемыми системами как переменные. Протокол также разрешает активные задачи управления, такие как изменения конфигурации, посредством удаленной модификации этих переменных. Переменные, доступные через SNMP, организованы в иерархии. Сам SNMP не определяет, какие переменные должна предлагать управляемая система. Скорее, SNMP использует расширяемый дизайн, который позволяет приложениям определять свои собственные иерархии. Эти иерархии описываются как информационная база управления (MIB). MIB описывают структуру данных управления подсистемой устройства; они используют иерархическое пространство имен, содержащее идентификаторы объектов (OID).

SNMP работает на прикладном уровне пакета интернет-протокола. Все сообщения SNMP передаются через протокол пользовательских дейтаграмм (UDP). Агент SNMP принимает запросы на порт 16 UDP. Менеджер может отправлять запросы от любого доступного порта источника к порту 161 в агенте. Ответ агента отправляется обратно в исходный порт диспетчера.

Менеджер получает уведомления (Ловушки и InformRequests) на порту 162. Агент может генерировать уведомления из любого доступного порта. При использовании с защитой транспортного уровня или защитой транспортного уровня датаграммы запросы принимаются на порт 10161, а уведомления отправляются на порт 10162 [3].

SNMPv1 определяет пять основных блоков данных протокола (PDU). В SNMPv2 добавлены два других блока PDU, GetBulkRequest и InformRequest, а в SNMPv3 добавлен PDU отчета. Все блоки протоколов SNMP построены следующим образом:

IP-заголовок UDP-версия заголовка сообщества PDU-тип request-id error-status error-index привязки переменных

Семь типов PDU SNMP, идентифицированные полем типа PDU, следующие:

– GetRequest- запрос менеджера-агента для получения значения переменной или списка переменных. Желаемые переменные указываются в привязках переменных (поле значения не используется). Получение указанных переменных значений должно выполняться агентом. Возвращается ответ с текущими значениями.

– SetRequest - запрос менеджера-агента для изменения значения переменной или списка переменных. Переменные привязки указываются в теле запроса. Изменения во всех указанных переменных должны выполняться агентом. Возвращается ответ с (текущими) новыми значениями для переменных.

– GetNextRequest - запрос менеджера-агента для обнаружения доступных переменных и их значений. Возвращает ответ с привязкой переменной для лексикографически следующей переменной в MIB. Весь MIB агента можно пройти путем итеративного применения GetNextRequest, начиная с OID 0. Строки таблицы можно прочитать, указав OID столбца в привязках переменных запроса.

– GetBulkRequest - запрос менеджера-агента для нескольких итераций GetNextRequest. Оптимизированная версия GetNextRequest. Возвращает ответ с несколькими привязками переменных, идущими от привязки переменных или привязок в запросе. Для контроля поведения ответчика используются конкретные не повторительные PDU и max-repetitions. GetBulkRequest был представлен в SNMPv2.

– Response - возвращает привязки переменных и подтверждение от агента к менеджеру для GetRequest, SetRequest, GetNextRequest, GetBulkRequest и InformRequest. Отчеты об ошибках предоставляются по положению об ошибках и по индексам ошибок. Хотя он использовался как ответ на получение и набор, этот PDU назывался GetResponse в SNMPv1.

– Trap - асинхронное уведомление от агента к менеджеру. В то время как в другой SNMP-связи менеджер активно запрашивает информацию у агента, это PDU, которые отправляются от агента менеджеру без явного запроса.

– InformRequest - подтвержденное асинхронное уведомление. Этот PDU был введен в SNMPv2 и первоначально был определен как менеджер для обмена менеджером.

Обобщив вышесказанное, можно сделать вывод, что, несмотря на ряд достоинств, протокол SNMP представляет собой достаточно сложный для реализации протокол, поддерживающийся достаточно ограниченным числом устройств.

1.2.3. Дистанционный мониторинг сети (RMON)

RMON был разработана IETF для поддержки мониторинга и анализа протоколов локальных сетей. Первоначальная версия (иногда называемая RMON1) была посвящена информации уровня OSI 1 и уровня 2 в сетях Ethernet и Token Ring. Он был расширен до RMON2, который добавляет поддержку мониторинга сетевого и прикладного уровня и SMON, добавляет

поддержку коммутируемых сетей. Это стандартная отраслевая спецификация, которая обеспечивает большую часть функциональных возможностей, предлагаемых проприетарными анализаторами сети. Агент RMON встроен во многие высококлассные коммутаторы и маршрутизаторы.

Удаленный мониторинг (RMON) - стандартная контрольная спецификация, которая позволяет различным сетевым мониторам и консольным системам обмениваться данными мониторинга сети. RMON предоставляет сетевым администраторам больше свободы в выборе консолей мониторинга сети с функциями, отвечающими их конкретным сетевым потребностям. Реализация RMON обычно работает в модели клиент / сервер. Устройства мониторинга (обычно называемые «зонды» в этом контексте) содержат программные агенты RMON, которые собирают информацию и анализируют пакеты. Эти зонды действуют как серверы, а приложения Network Management, которые общаются с ними, действуют как клиенты. Хотя конфигурация и сбор данных агента использует SNMP, RMON предназначен для работы иначе, чем другие системы на базе SNMP:

У зондов больше ответственности за сбор и обработку данных, что снижает трафик SNMP и нагрузку на обработку клиентов.

Информация передается только в приложение управления, если требуется, вместо непрерывного опроса и мониторинга. Короче говоря, RMON предназначен для мониторинга на основе потока, тогда как SNMP часто используется для управления на основе устройств. RMON похож на другие технологии мониторинга потока, такие как NetFlow и SFlow, поскольку собранные данные в основном касаются шаблонов трафика, а не состояния отдельных устройств. Одним из недостатков этой системы является то, что удаленные устройства несут большую часть бремени управления и требуют больше ресурсов для этого.

RMON состоит из десяти групп:

- статистика: статистика LAN в реальном времени, например, использование, столкновение, ошибки CRC;
- история: история выбранной статистики;
- тревога: определения сообщений SNMP SNMP, которые должны быть отправлены, когда статистика превышает определенные пороговые значения;
- хосты: конкретная локальная статистика LAN, например, байтов, отправленных / полученных, отправленных / полученных кадров;
- ведущие вершины: запись N наиболее активных подключений за определенный период времени;
- матрица: принятая схема трафика между системами;
- фильтр: определяет интересующие образцы пакетов данных, например MAC-адрес или порт TCP;
- захват: сбор и пересылка пакетов, соответствующих фильтру;
- событие: отправьте предупреждения (SNMP-ловушки) для группы «Тревога»;
- Token Ring: расширения, характерные для Token Ring.

Таким образом, протокол RMON представляет собой расширение SNMP, предназначенный для сбора и анализа информации о характере данных, передаваемых по сети. Однако сбор информации осуществляется с помощью специальных аппаратных агентов, что приводит к необходимости закупки дополнительного оборудования и внесения изменений в уже существующие ЛВС, а протокол RMON представляет собой расширение SNMP, предназначенный для сбора и анализа информации о характере данных передаваемых по сети. Однако сбор информации осуществляется с помощью специальных аппаратных агентов, что приводит к необходимости закупки дополнительного оборудования и внесения изменений в уже существующие ЛВС.

1.3.Анализ существующих программных средств, реализующих мониторинг компьютерных систем

1.3.1. Zabbix – инструмент мониторинга

Zabbix – это программный инструмент для мониторинга открытых источников для различных ИТ-компонентов, включая сети, серверы, виртуальные машины (VM) и облачные сервисы. Zabbix предоставляет доступ к таким показателям мониторинга, как использование сети, загрузка процессора и потребление дискового пространства. Программное обеспечение контролирует работу в Linux, Hewlett Packard Unix (HP-UX), Mac OS X, Solaris и других операционных системах (ОС); однако мониторинг Windows возможен только с помощью специальных программ агентов.

Zabbix может быть развернут для контроля на основе как агентов так и без них. Агенты установлены на ИТ-компонентах для проверки производительности и сбора данных. Затем агент возвращается к централизованному серверу управления Zabbix. Эта информация включена в отчеты или представлена визуально в графическом пользовательском интерфейсе Zabbix (GUI). Если есть какие-либо проблемы в отношении того, что отслеживается, Zabbix отправит уведомление или оповещение пользователю. Мониторинг осуществляется такой же образом, используя существующие ресурсы в системе или устройстве для эмуляции агента.

Веб-интерфейс Zabbix позволяет пользователям просматривать свою ИТ-среду с помощью настраиваемых панелей мониторинга на основе виджетов, графиков, сетевых карт, слайд-шоу и отчетов. Например, пользователь может настроить отчет для отображения показателей, связанных как с соглашениями об уровне обслуживания (SLA), так и с ключевыми показателями эффективности (KPI) при загрузке ЦП.

Zabbix работает в трех режимах обнаружения.

– Обнаружение сети: периодически сканирует ИТ-среду и записывает тип устройства, IP-адрес, статус, время простоя и время простоя.

– Обнаружение низкого уровня: автоматически создает элементы, триггеры и графики на основе обнаруженного устройства. Низкоуровневое обнаружение может создавать показатели из идентификаторов объектов Simple Network Management Protocol (SNMP), служб Windows, запросов на структурированный запрос запросов (SQL), связанных с базами данных (ODBC), сетевых интерфейсов и т.д.

– Автоматическое обнаружение: автоматически начинает мониторинг любого обнаруженного устройства с помощью агента Zabbix.

Zabbix может отправлять уведомления по электронной почте на основе предопределенных событий в ИТ-среде пользователя. Еще один способ для пользователей Zabbix оставаться в курсе своей ИТ-среды – это мобильные приложения от таких поставщиков, как M7 Monitoring или их собственное создание.

Zabbix предлагает несколько вариантов мониторинга вне агентов. Простая проверка может проверить доступность и отзывчивость стандартной службы, такой как уведомления или HTTP.

Расширения управления Java (JMX), веб-мониторинг и другие методы также являются альтернативой использованию агентов. В Zabbix JMX можно использовать для мониторинга приложений на Java. Веб-мониторинг используется для проверки доступности веб-сайтов и поддерживает HTTP и HTTPS. Zabbix собирает данные, касающиеся средней скорости загрузки сценария, ошибок и сообщений об ошибках, времени отклика и т.д.

Интерфейс программирования Zabbix представляет собой веб-интерфейс для создания новых приложений, автоматизации задач и интеграции с сторонним программным обеспечением, таким как go-zabbix, Zabbix: Tiny или Zabbix отправитель. JavaScript Object Notation (JSON -

формат) используется для базировать API в качестве интерфейсного веб-интерфейс.

API Zabbix состоит из множества методов, которые сгруппированы в отдельные API-интерфейсы, каждый из которых выполняет определенную службу. Например, метод создания нового хоста - host.create; метод входа в систему в качестве администратора - user.login.

Используя API, пользователи могут создавать приложения для работы и отображения информации Zabbix.

Также Zabbix имеет встроенную поддержку шаблонов. Шаблоны - это настраиваемые надстройки, расширяющие функциональность Zabbix. Некоторые шаблоны сделаны Zabbix и поставляются в комплекте с готовым к использованию программным обеспечением, а другие – пользователями Zabbix. Шаблоны позволяют пользователям Zabbix отслеживать сетевые устройства от таких поставщиков, как Cisco, Dell, HP и Juniper. Другие шаблоны могут использоваться для мониторинга серверов IBM, HP и Super Micro. Шаблоны для служб на основе приложений включают Microsoft Exchange и Exchange Server, Zenoss, PowerDNS, Authoritative Server Stats и другие. Шаблоны могут быть созданы для мониторинга ОС и гипервизоров.

Таким образом, ПО Zabbix, представляет собой свободную систему мониторинга разнообразных сервисов компьютерной сети и сетевого оборудования. Несмотря на большое количество достоинств данная система обладает и рядом недостатков таких как: сложность в установке и внедрении, требовательность к ресурсам, невозможность обеспечения отказоустойчивости.

1.3.2. Nagios – мониторинг локальных сетей на базе UNIX

Nagios был разработан для Linux, но теперь способен отслеживать операционные системы на базе UNIX. Он состоит из трех компонентов:

демона, веб-интерфейса и плагинов. Nagios состоит из четырех основных типов объектов, которые перечислены ниже.

- Команды: они используются для взаимодействия с плагинами и для управления обработчиками событий, уведомлениями и проверками.
- Контактные и контактные группы: они определяют лиц, с которыми необходимо связаться в случае события.
- Группы хостов и хостов: они используются для указания служб и хостов в определенной сети.
- Уведомления. Они определяют, какой контент должен быть отправлен контактам и контактным группам, когда обнаружены.

Мониторинг хоста и службы определяют, какие плагины вызывать для получения статуса хоста или службы. Определения в плагинах проверяются на этом конкретном хосте / службе. Если видно, что значения выходят за порог, Nagios уведомляет контакты / контактные группы. Nagios возвращает любой из следующих четырех кодов статуса для любого события:

1. OK
2. Предупреждение
3. Критический
4. Неизвестно

Nagios можно настроить для проведения пассивных или активных проверок. В активных проверках хост, который так часто запускает устройства или службы Nagios для получения информации о статусе, тогда как пассивные проверки инициируются и выполняются внешними приложениями / процессами. Затем результаты отправляются на хост мониторинга (где работает Nagios) для обработки. Основное различие между активными и пассивными проверками заключается в том, что активные проверки инициируются и выполняются Nagios, в то время как пассивные проверки выполняются внешними приложениями. Пассивные проверки могут использоваться для мониторинга асинхронных служб и мониторинга служб,

расположенных за брандмауэром. Активные проверки могут использоваться для проверки по требованию и регулярных проверок интервалов. Результаты могут контролироваться через веб-интерфейс, основанный на CGI.

Главные особенности системы:

- Планирование обновления инфраструктуры до того, как устаревшие системы вызывают сбои
- Отвечать на вопросы при первом признаке проблемы
- Автоматическое исправление проблем при обнаружении
- Согласовать ответы технической команды
- Обеспечьте соблюдение SLA вашей организации
- Обеспечить минимальные последствия для ИТ-инфраструктур на нижней строке вашей организации
- Мониторинг всей инфраструктуры и бизнес-процессов

Обобщив вышесказанное Nagios представляет собой программу с открытым исходным кодом, предназначенную для мониторинга компьютерных сетей и систем. К недостаткам данной системы можно отнести отсутствие встроенных средств визуализации, сложность масштабирования и отсутствия возможности вносить оперативные изменения в конфигурацию системы.

1.3.3. Spiceworks Network Monitor – мониторинг серверов и сетевых устройств

Spiceworks Network Monitor - бесплатный инструмент, предназначенный для мониторинга и статистики в реальном времени для серверов и сетевых устройств, поддерживающих SNMP. Хотя он бесплатный, он не является программным продуктом с открытым исходным кодом, так же в правом верхнем углу присутствует реклама. Сетевой монитор Spiceworks можно использовать вместе с инструментами технической поддержки Spiceworks и инструментами управления ресурсами, но далее он рассматривается как отдельный программный продукт.

Сетевой монитор работает с любой версией Windows с Windows Server 2008 R2, и мы установили его на сервере Windows 2012 R2, на котором не было никаких других служб. Установка и настройка и настройка не вызывает никаких затруднений: просто загрузите программное обеспечение с сайта Spiceworks и запустите программу установки. После завершения работы ярлык на рабочем столе приведет к веб-интерфейсу, где можно завершить процесс начальной настройки.

Перед тем, как войти в систему мониторинга, вам понадобится учетная запись Spiceworks - нужно будет перейти на spiceworks.com, чтобы установить ее. Для этого потребуется имя, адрес электронной почты и пароль. После входа в систему нам был представлен экран панели управления по умолчанию. Ниже горизонтального меню и предупреждающих полосок в верхней части экрана есть поля для сетевых и сетевых наблюдателей. Ниже расположены пробелы для добавления 3 устройств для более тщательного мониторинга, в которых будет отображаться более подробная информация о каждом из этих устройств.

При первом входе в систему пользовательская панель автоматически открывается через левую треть страницы, раздражающе скрывая элементы панели инструментов под ней. Основная справочная информация, которую он отображает, не очень полезна, но после ее закрытия она остается закрытой для будущих логинов, если она не будет повторно открыта.

Помимо основной информационной панели, для устройств есть выделенные страницы, где можно добавлять и просматривать детали машин, которые вы хотите отслеживать, и параметры, в которых можно настраивать оповещения и добавлять пользователей. Существует также опция меню для справки, но вместо того, чтобы сразу предоставить вам соответствующую документацию, она приведет вас на сайт сообщества Spiceworks, где нужно настроить имя форума, прежде чем сможете получить доступ к любой полезной информации. В процессе создания учетной записи настраивается профиль для своей компании, а также один для себя, и хотя можно пропустить

большую часть этого, это раздражает, если вам просто нужна немедленная помощь с программным обеспечением.

Еще более раздражающе, как только вы зарегистрировали профиль, вы попадаете на главную страницу сообщества Spiceworks вместо справочных страниц сетевого монитора. К счастью, после регистрации в будущем попытки получить доступ к справке через веб-интерфейс Network Monitor вы попадете прямо на страницу поддержки сообщества, где вы можете искать ранее заданные вопросы или перейти на форумы поддержки.

Программное обеспечение отслеживает 26 различных параметров для каждого устройства, разделенных на 5 категорий: хост, процессор, память, диск и сеть. Они могут отслеживать конкретные условия, такие как постоянная высокая загрузка процессора, спайки в использовании памяти, низкое дисковое пространство и узкие места в сети. Однако, в отличие от более полнофункциональных решений мониторинга (как коммерческих, так и открытых), у него нет возможности отслеживать конкретные процессы или каким-либо образом создавать пользовательские предупреждения для определенных условий ошибки.

На странице настроек есть только две вкладки, перечисленные в левой части экрана, «Мониторы по умолчанию» и «Учетные записи пользователей». Первый позволяет настроить пороговые значения по умолчанию для контролируемых параметров и выбрать, какие будут генерировать оповещения по электронной почте. Вторая вкладка позволяет добавлять или удалять пользователей для системы мониторинга. К сожалению, пользователи не могут быть добавлены напрямую. Вместо этого вы вводите свое имя и адрес электронной почты, и система отправляет им приглашение по электронной почте со ссылкой. Если они войдут в систему, для них автоматически будет создана учетная запись spiceworks.com.

Страница устройств начинается с одного только перечисленного устройства: машины, на которой работает программа мониторинга. Нажатие

кнопки добавления устройства открывает новую панель с вкладками для добавления компьютеров под управлением Windows или Linux, а также сетевых устройств, таких как маршрутизаторы и брандмауэры. Чтобы добавить устройство, все, что вам нужно, это его IP-адрес или имя хоста и логин для этой системы с достаточными привилегиями. При добавлении сервера Windows мы использовали учетную запись с правами локального администратора на этом сервере. Для серверов Linux мы использовали стандартные учетные записи пользователей, которым был предоставлен полный доступ к sudo.

После добавления устройства пороговые значения и параметры электронной почты для каждого предупреждения могут быть оставлены по умолчанию или настроены для этого устройства. Мы моделировали различные системные ошибки на наших тестовых серверах. Большинство из них сообщалось точно, но, когда мы использовали программный инструмент для поддержания центрального процессора нашего сервера Windows на 100%, Spiceworks Network Monitor показал правильность загрузки процессора на своем графике, но его список процессов показал только 50% загрузки процессора для а не почти 100% нагрузки, отображаемой диспетчером задач на самом сервере.

Оповещения отображаются в веб-интерфейсе сетевого монитора и отправляются по электронной почте, если вы включили оповещения по электронной почте для данного параметра. Мы обнаружили, что письмо первому пользователю прибыло незамедлительно, но письмо второму пользователю, которого мы добавили, всегда было на 15 минут позже первого. Нелья указать, какие оповещения будут отправляться пользователям, либо: всем пользователям отправляются все предупреждения. Это нормально, если вы контролируете только несколько серверов, но в более крупной компании, где разные сотрудники могут нести ответственность за разные группы машин, могут возникнуть проблемы с ограниченными возможностями настройки для

оповещений. Пока предупреждение не будет очищено, электронные письма с напоминанием отправляются каждые 30 минут. Окончательное письмо отправляется после устранения проблемы.

Стоит отметить, что сообщения электронной почты отправляются через системы spiceworks.com, вместо того, чтобы использовать собственный внутренний почтовый сервер, поэтому, если интернет-соединение не работает, вы не получите предупреждающие сообщения. Сетевой монитор аналогично полагается на Spiceworks для аутентификации пользователей, поэтому вы не можете установить средство мониторинга на сервере в защищенном разделе сети с ограниченным доступом в Интернет.

Хотя мы обнаружили, что Spiceworks Network Monitor был быстр и прост в установке и настройке, отсутствие детализации конфигурации и невозможность определения пользовательских предупреждений были разочаровывающими. В сочетании с его зависимостью от внешних систем spiceworks.com для аутентификации пользователей и электронной почты и ограниченными возможностями мониторинга эти факторы делают его непригодным для настроек мониторинга корпоративного уровня или даже для среднего бизнеса. Это странно, так как Network Monitor может, по данным Spiceworks, отслеживать до 1000 устройств без ущерба для замедления.

ПО Spiceworks Network Monitor обладает удобным интерфейсом, предназначенным для быстрой настройки и поэтому хорошо подходит для организации системы мониторинга в небольших ЛВС. Администраторы с более жесткими требованиями к мониторингу, которые хотят иметь возможность гибко настроить систему под свои требования, должны выбрать более настраиваемый инструмент сетевого мониторинга. ПО Spiceworks Network Monitor обладает удобным интерфейсом, предназначенным для быстрой настройки и поэтому хорошо подходит для организации системы мониторинга в небольших ЛВС. Администраторы с более жесткими требованиями к мониторингу, которые хотят иметь возможность гибко

настроить систему под свои требования, должны выбрать более настраиваемый инструмент сетевого мониторинга.

1.3.4. Eltex.EMS – система управления сетевым оборудованием

Eltex.EMS - это централизованная система управления сетевым оборудованием производства Eltex Enterprise Ltd.

Система Eltex.EMS представляет собой архитектуру на основе клиент-сервер. Один сервер доступа представлен веб-интерфейсом, который позволяет осуществлять независимое одновременное управление различными сетевыми элементами.

Подсистема управления автоматизацией (Northbound Interface) позволяет взаимодействовать с EMS с превосходным поставщиком OSS / BSS. В частности, он позволяет стыковать с биллинговой системой оператора с использованием открытых стандартизованных протоколов, что позволяет автоматизировать рутинные операции, такие как массовое отключение абонентских портов в случае неоплаченных услуг и последующее повторное подключение в случае оплаты, а также изменение конфигурации устройств.

Возможность установки пакетов для используемого типа оборудования позволяет оптимально загружать серверные ресурсы оператора, отслеживать данные о состоянии сети в реальном времени и эффективное использование человеческих ресурсов. Таким образом, вы можете получить наиболее эффективный выход из системы.

Система Eltex.EMS может быть представлена в виде стандартных Linux-дистрибутивов двух основных популярных форматов rpm и deb, а также в виде готового ISO-образа, который может быть быстро установлен как на реальном хосте, так и на супервизоре виртуальных машин. Это позволяет быстро развернуть систему мониторинга в кратчайшие сроки.

Таким образом Eltex.EMS это централизованная система управления сетевым оборудованием производства ООО «Предприятие «ЭЛТЕКС». К

недостаткам можно отнести ресурсоемкость, достаточно высокую стоимость и закрытость системы.

Выводы по первому разделу

На сегодняшний день существует большое количество различных технологий мониторинга компьютерных систем, использующих различные сетевые протоколы. Протокол ICMP представляет собой наиболее простой сетевой протокол, применяющийся для базового мониторинга сети и выявления проблем.

ПО Zabbix, представляет собой свободную систему мониторинга разнообразных сервисов компьютерной сети и сетевого оборудования. Несмотря на большое количество достоинств данная система обладает и рядом недостатков таких как: сложность в установке и внедрении, требовательность к ресурсам, невозможность обеспечения отказоустойчивости. Spiceworks Network Monitor - бесплатный инструмент, предназначенный для мониторинга и статистики в реальном времени для серверов и сетевых устройств, поддерживающих SNMP. Хотя он бесплатный, он не является программным продуктом с открытым исходным кодом, так же в правом верхнем углу присутствует реклама. Сетевой монитор Spiceworks можно использовать вместе с инструментами технической поддержки Spiceworks и инструментами управления ресурсами, но я рассматриваю его как отдельный программный продукт. Eltex.EMS это централизованная система управления сетевым оборудованием производства ООО «Предприятие «ЭЛТЕКС». К недостаткам можно отнести ресурсоемкость, достаточно высокую стоимость и закрытость системы.

РАЗДЕЛ 2. ИССЛЕДОВАНИЕ МЕТОДОВ И СРЕДСТВ ПОСТРОЕНИЯ СОВРЕМЕННЫХ КОМПЬЮТЕРНЫХ СЕТЕЙ

2.1. Особенности построения современных компьютерных сетей

На сегодняшний день считается общепринятой трехуровневая иерархическая модель построения компьютерных сетей.

Иерархическая модель сети – трехуровневая модель организации сети компании, впервые предложенная инженерами Cisco Systems. Подразделяет сеть компании на три уровня иерархии: ядро сети, уровень распределения, уровень доступа

Каждый слой имеет свои собственные цели и функции. Это помогает сделать сеть масштабируемой, стабильной и детерминированной. Следуя этой модели, администраторы могут поддерживать сеть как самым естественным образом. Как показано на рисунке 2.1, иерархическая сетевая модель использует три уровня. Это уровни Core, Distribution и Access.

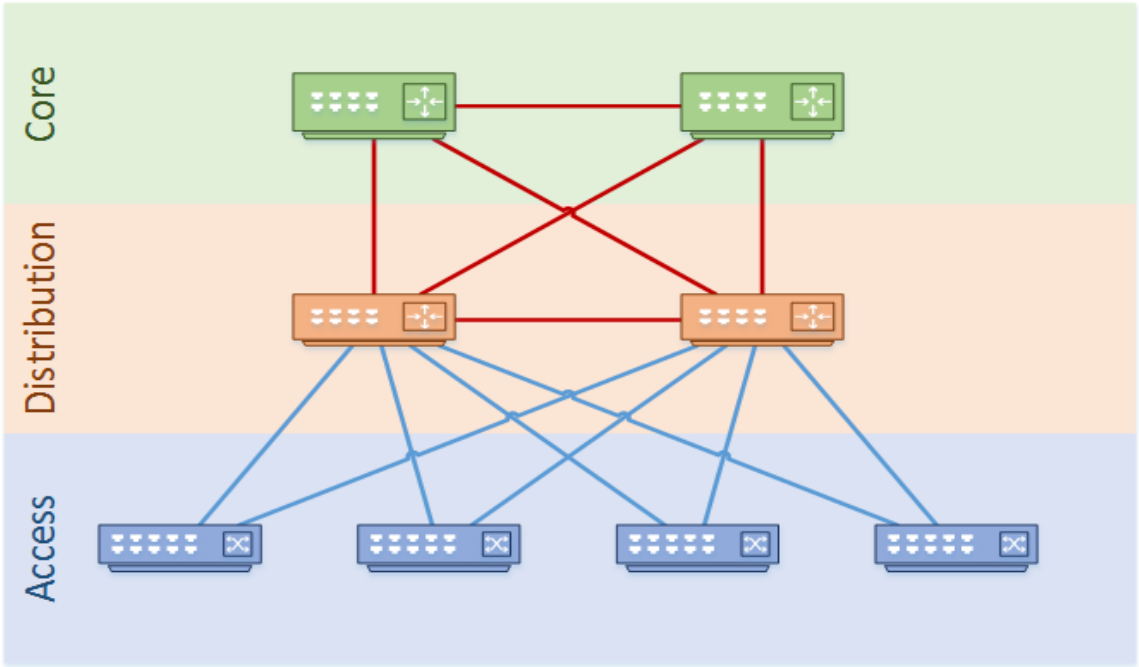

Рис. 2.1. Иерархическая трехуровневая модель сети

Требования и функции каждого уровня различны. Чтобы решить эту проблему, подход к дизайну каждого уровня должен быть индивидуальным.

Уровень доступа предназначен для удовлетворения потребностей подключения к конечным устройствам. Уровень доступа - это край сети, в которой подключены хост-устройства. Это включает в себя рабочие станции и принтеры. Устройства, которые расширяют сеть, например телефоны и точки доступа, также присоединяются сюда. Это слой, на котором администратор проводит большую часть своего времени. Это очень функциональный слой, так как он должен поддерживать множество разных конечных точек, как показано на рисунке 2.2.

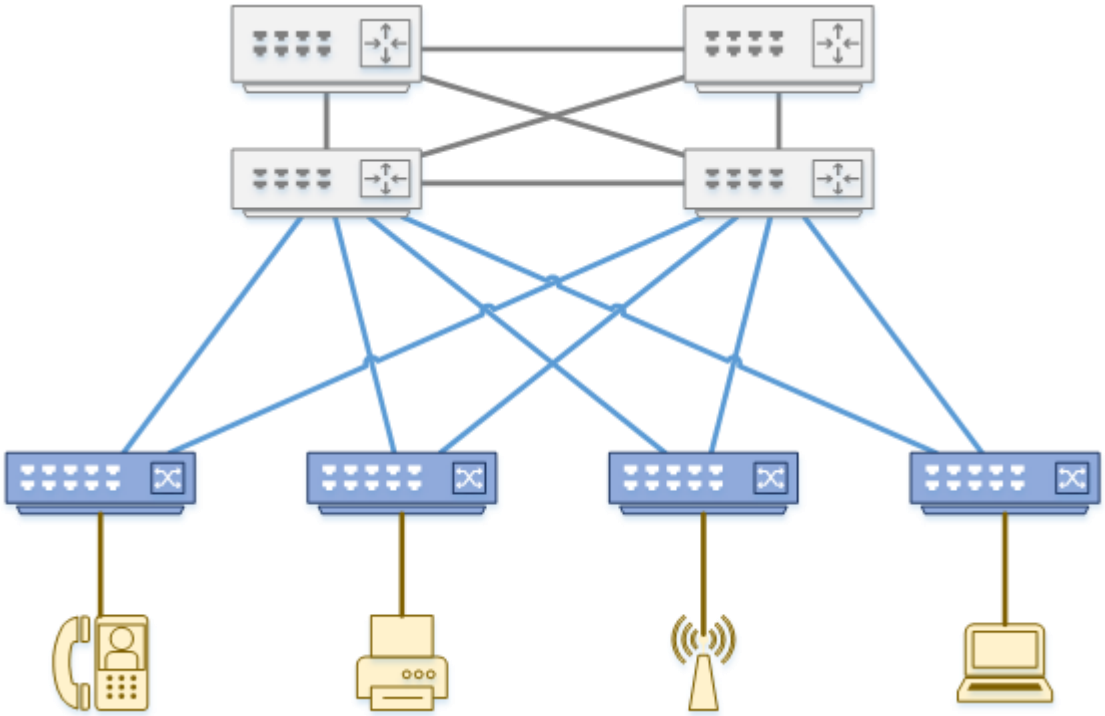

Рис. 2.2. Подключение коммутаторов уровня доступа с резервированием

Этот уровень обычно включает в себя такие сервисы как

- Открытие и настройка - CDP и LLDP
- Безопасность и сетевая идентификация - 802.1x, безопасность портов, отслеживание DHCP, DAI, Source Guard, Identity Based Network Services и Web-auth
- Признание приложения - маркировка QoS, контроль, очередность, NBAR

– Управление сетью - протоколы маршрутизации, связующее дерево, DTP, LACP, UDLD, FlexLink
– Физическая инфраструктура - PoE

Уровень распределения - многоцелевой слой. В частности, ему необходимо агрегировать трафик уровня доступа и перенаправить его в остальную сеть. В сети, вероятно, много коммутаторов уровня доступа. Каждый из этих переключателей имеет восходящие линии связи с коммутаторами уровня распространения. Многие конечные устройства через коммутаторы уровня доступа агрегируются на уровне распространения.

Использование коммутаторов распределения для совокупного трафика логически создает «блоки распределения». Рассмотрим университетский городок с четырьмя зданиями. Каждое здание имеет два распределительных коммутатора и восемь переключателей уровня доступа. Каждое из этих зданий является распределительным блоком. Трафик может маршрутизировать между блоками распределения через основной уровень, как показано на рисунке 2.3.

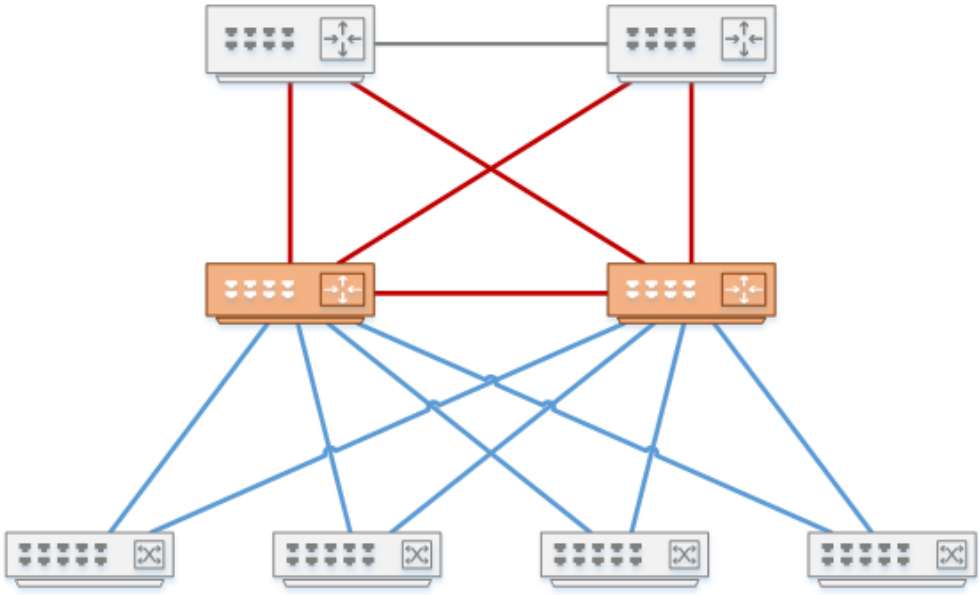

Рис. 2.3 Резервирование коммутаторов уровня распределения

Уровень распределения обеспечивает точку демаркации между блоком и остальной частью сети (рисунок 2.4). Это делает его хорошим местом для

применения сетевых политик. Это также хорошая граница безопасности между уровнем доступа и остальной частью сети.

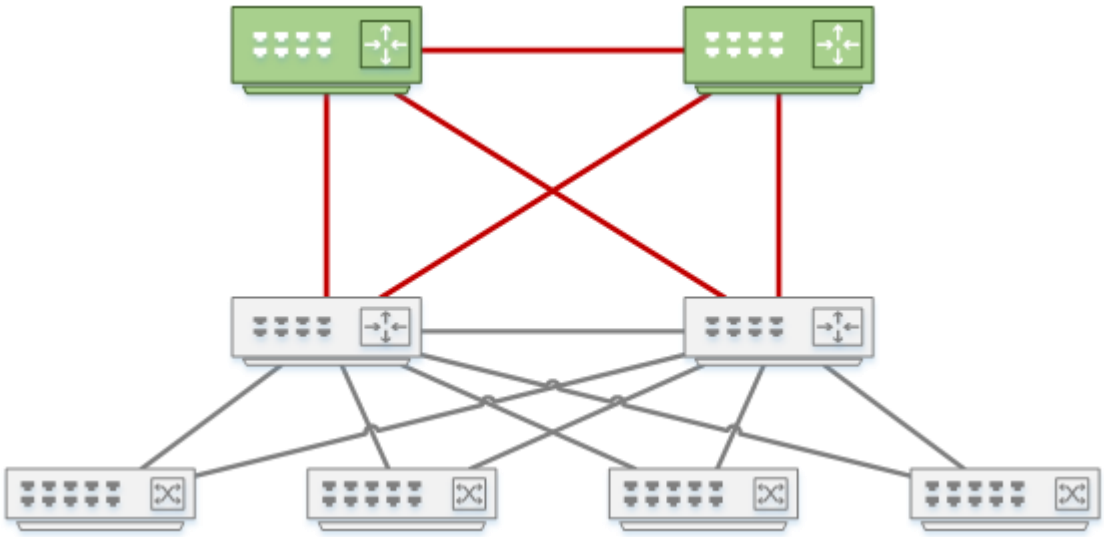

Рис. 2.4. Маршрутизация трафика на коммутаторах уровня ядра

Уровень ядра – основной слой сети. Он соединяет все блоки распределения. Основной слой имеет одну цель и не требует много функций. Нет никаких политик безопасности, никакого QoS и никаких конечных точек. Ключевыми принципами проектирования основного слоя является то, что он должен быть быстрым, он должен быть всегда доступен, и он должен быть надежным. Крайне важно, что нет единственной точки отказа. Если произошел сбой, восстановление должно быть как можно быстрее. Ядро работает с чистыми каналами уровня 3 (маршрутизируемыми портами) и настраиваемыми протоколами маршрутизации.

Таким образом общепринятой считается трехуровневая иерархическая модель ЛВС. Достоинства подобной структуры в том, что трафик пользователей с множества коммутаторов уровня доступа агрегируется на родительском узле распределения, маршрутизируется или коммутируется по необходимости на вышестоящее ядро, на соседний узел распределения или непосредственно между самими пользователями с разных узлов доступа. А каждое ядро маршрутизирует или коммутирует трафик между несколькими

узлами распределения, которые непосредственно включены в него, или между соседними ядрами.

2.2. Описание архитектуры построения компьютерной сети ФГБОУ "МДЦ "Артек"

ЛВС ФГБОУ «МДЦ «Артек» представляет собой сложную распределенную сеть выстроенною в соответствии с общепринятыми нормами и правилами построения сетей. Логическая топология сети представлена на рисунке 2.5

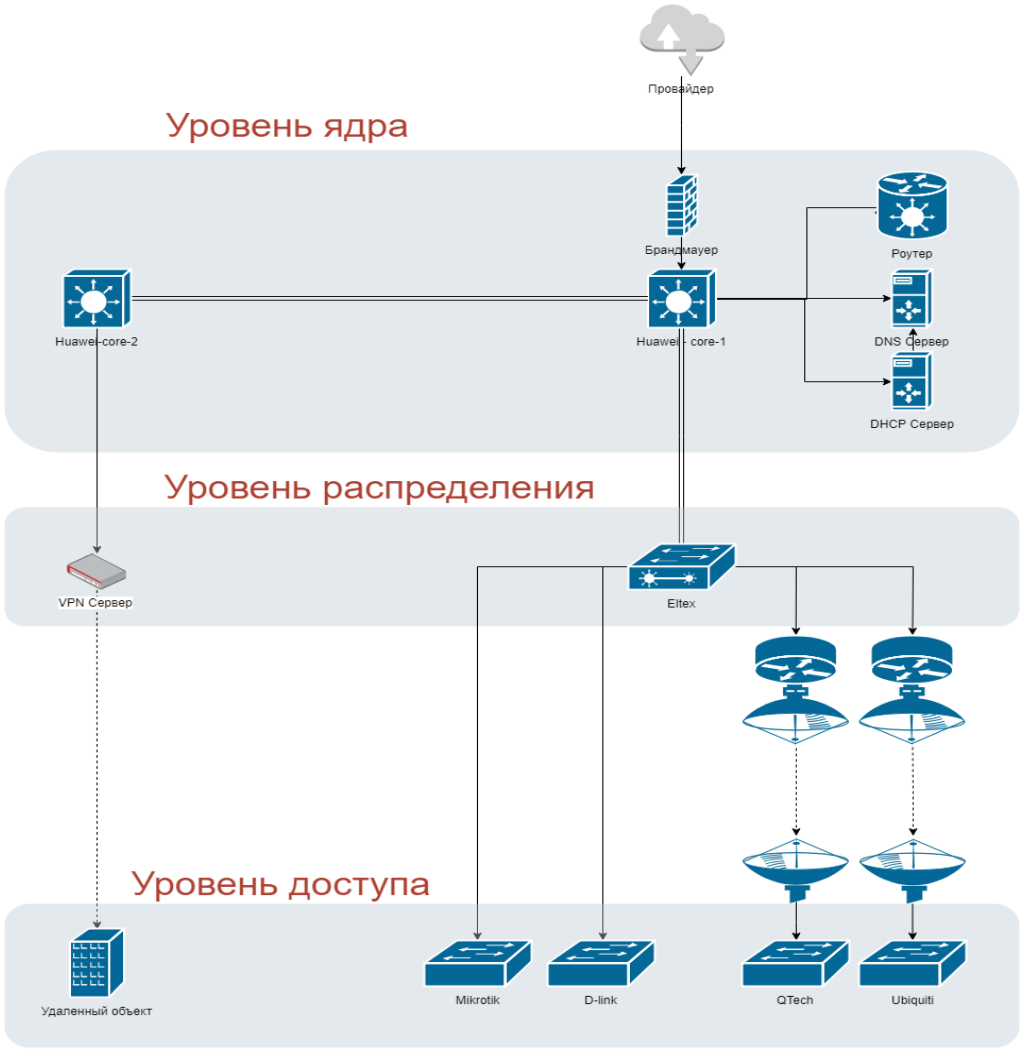

Рис. 2.5. Логическая топология сети ФГБОУ «МДЦ «Артек»

ЛВС ФГБОУ «МДЦ «Артек» построен согласно общепринятой трехуровневой иерархической модели сети. Оборудование уровня ядра и распределения обладает двукратным резервированием и может быть оперативно заменено в случае выхода из строя любой единицы оборудования. Так же все активное оборудование ЛВС подключено к источникам бесперебойного питания, обеспечивающего стабилизацию подачи электроэнергии на оборудование и бесперебойную работу в случае временных проблем с городской электрической сетью.

ЛВС ФГБОУ «МДЦ «Артек» представляет собой сложную распределенную сеть выстроенною в соответствии с общепринятыми нормами и правилами построения сетей.

2.3. Описание и особенности работы активного сетевого оборудования ФГБОУ "МДЦ "Артек"

Рассмотрим подробнее активное сетевое оборудование ФГБОУ «МДЦ «Артек»:

- высокоскоростные коммутаторы;
- маршрутизаторы уровня доступа (более 18 моделей разных производителей);
- устройства радио-релейной связи (Air Fiber);
- камеры видеонаблюдения ;
- IP-телефоны.

2.3.1. Оборудование уровня ядра

За оборудование уровня ядра сети отвечает коммутатор высокоскоростного доступа и многосервисного агрегирования Huawei S5720-56C-PWR-EI-AC1 (рис. 2.6), который базируется на современной аппаратуре

и программном обеспечении универсальной платформы маршрутизации (VRP).

Рис. 2.6. Huawei S5720-56C-PWR-EI-AC1

S5720 поддерживает технологию Super Virtual Fabric (SVF), которая

- виртуализирует функции ядра,
- агрегации,
- доступа,
- беспроводные точки доступа на одном устройстве для упрощенного управления.

Виртуализация SVF также обеспечивает возможность включения и выключения коммутаторов и точек доступа; и поддерживает настройку конфигурации на основе профиля и автоматическую доставку конфигураций от основных устройств к устройствам агрегации и доступа.

Графический интерфейс управления коммутатором Huawei представлен на рисунке 2.7.

Простота эксплуатации обеспечивает развертывание с нулевым касанием, что позволяет заменять неисправные устройства без дополнительной настройки; плюс развертывание на основе USB, пакетная конфигурация и пакетное удаленное обновление - значительно сокращая затраты на O & M.

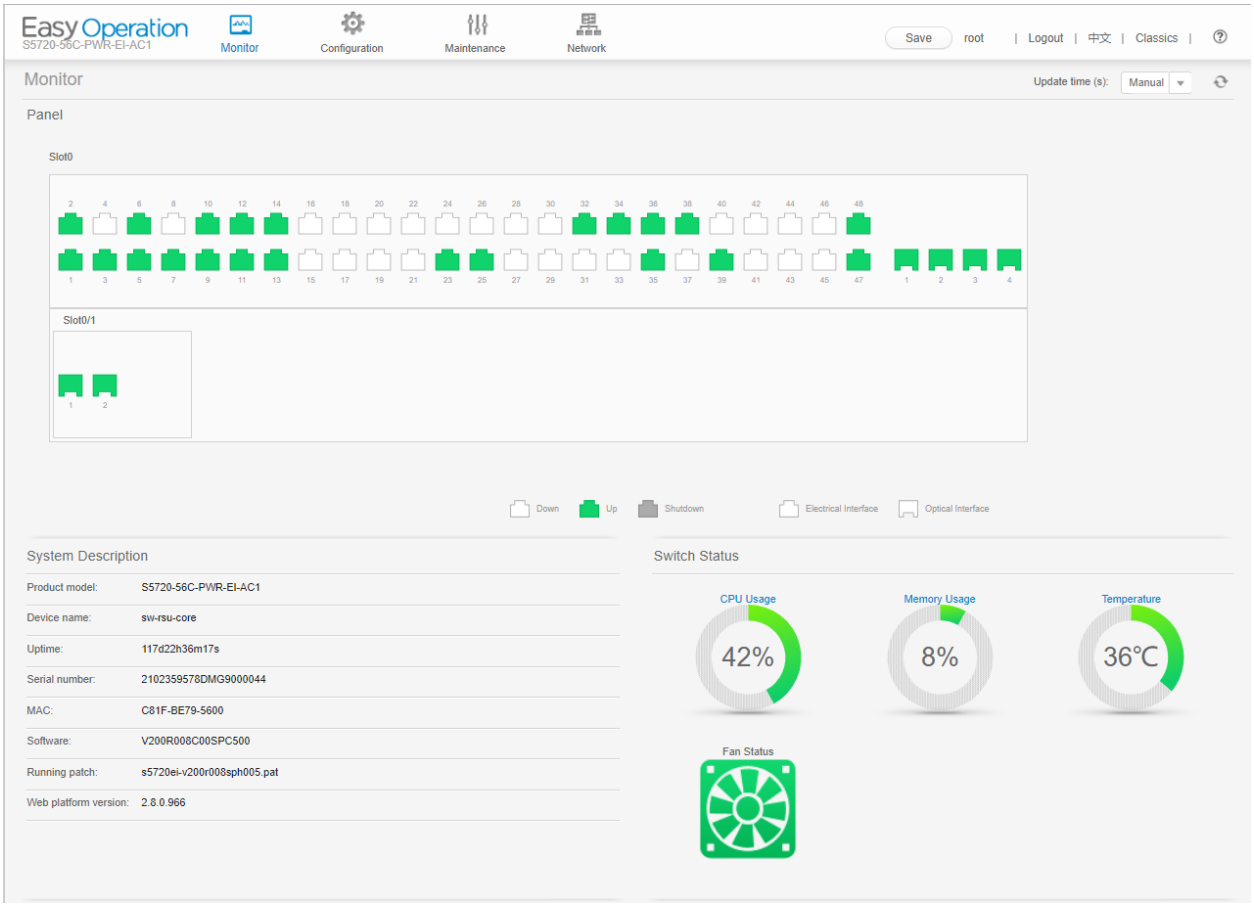

Рис. 2.7. Графический интерфейс управления коммутатором Huawei

Дополнительная гибкость обеспечивается поддержкой протокола Simple Network Management Protocol (SNMP) v1, v2c и v3; интерфейс командной строки (CLI), сетевые системы управления сетью и Secure Shell (SSH) версии 2.0. Поддержка удаленного мониторинга сети (RMON), нескольких журнальных хостов, сбора статистики трафика портов и всесторонней проверки качества сети в процессах консолидации и реорганизации сети.

В дополнение к традиционному протоколу Spanning Tree Protocol (STP), протоколу Rapid Spanning Tree Protocol (RSTP) и протоколу множественного связующего дерева (MSTP) S5720-EI поддерживает технологию Smart Ethernet Protection (SEP), разработанную Huawei, и новейшую коммутационную защиту Ethernet Ring Protection Switching (ERPS). SEP является протоколом защиты от кольцевой защиты, характерным для уровня Ethernet-канала, и обеспечивает поддержку топологий кольцевой сети, таких как открытое

кольцо, замкнутое кольцо и топологии каскадных кольцевых каналов. ERPS, определенный в ITU-T G.8032, обеспечивает переключение защиты на миллисекундах на основе традиционных функций Ethernet MAC и мостов.

S5720-EI поддерживает протокол резервирования Smart Link и Virtual Router (VRRP), который реализует резервное копирование восходящих ссылок. Один коммутатор S5720-EI может подключаться к нескольким коммутаторам агрегации через несколько каналов, что значительно улучшает доступность и надежность.

Поддержка протокола Link Layer Discovery Protocol (LLDP) позволяет подключенным устройствам обмениваться информацией о соединении и динамически предоставлять параметры, необходимые для поддержки политик VLAN, безопасности и QoS для устройств IP-телефонии.

Механизмы обнаружения множественного соединения включают в себя Ethernet OAM (IEEE 802.3ah / 802.1ag / ITU Y.1731) и двунаправленное обнаружение пересылки (BFD).

Технология iStack от Huawei объединяет несколько коммутаторов в один логический коммутатор. Кластерные коммутаторы могут комбинироваться в избыточных конфигурациях для повышения надежности сети и использоваться с агрегацией каналов между устройствами для повышения надежности соединения. Масштабируемость упрощается, поскольку количество портов, пропускную способность и общую пропускную способность можно увеличить, добавив коммутаторы в стек - без нарушения сети. Конфигурация и управление устройствами также упрощены, что упрощает O & M и снижает совокупную стоимость владения.

2.3.2. Оборудование уровня распределения

Так как локальная вычислительная сеть значительно распределена географически, большая часть удаленных объектов подключена посредством волоконно-оптических линий связи (ВОЛС). Коммутация ВОЛС значительно

отличается по используемым технологиям, стандартам и интерфейсам подключения от типичных коммутаторов, работающих посредством стандарта Ethernet. Поэтому в качестве коммутатора уровня распределения в данной ЛВС используется специализированный коммутатор Eltex MES2124F 28-port Fiber 1G Managed Switch (рис. 2.8).

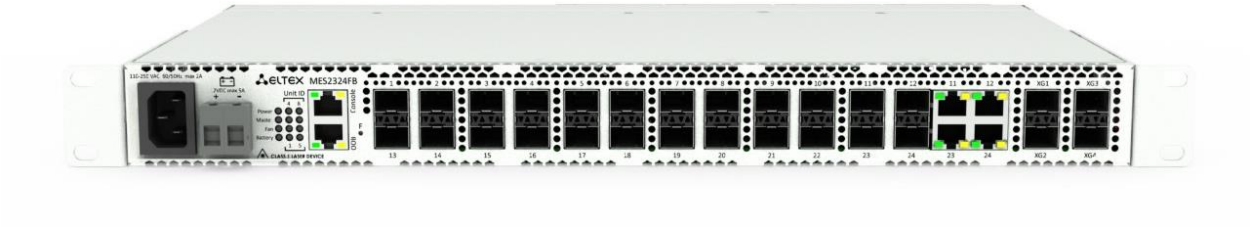

Рис. 2.8. Коммутатор Eltex MES2124F 28-port Fiber 1G Managed Switch

Функциональные возможности коммутатора обеспечивают

- физическое стекирование,
- поддержку виртуальных локальных сетей,
- многоадресных групп рассылки,
- расширенные функции безопасности.

Коммутатор имеет возможность подключения аккумуляторной батареи для обеспечения гарантированного питания в случае пропадания первичной сети 220В. Коммутатор оснащен блоком питания, который позволяет заряжать АКБ при наличии питания 220В. Система резервного питания позволяет следить за состоянием первичной сети и извещать о переходе с одного типа питания на другой.

В коммутаторах MES используется технология эффективной защиты от скачков напряжения питания (до 6 кВ), вызванного грозовыми разрядами.

Также данный коммутатор обладает рядом дополнительных функций:

- расширенные функции L2;
- поддержка стекирования;
- поддержка Multicast (IGMP Snooping, MVR);

– расширенные функции безопасности (L2-L4 ACL, IP Source Guard, Dynamic ARP Inspection и др.)

Веб-интерфейс панели администрирования одинаков для всех коммутаторов производства компании Eltex и представлен на рисунке 2.7.

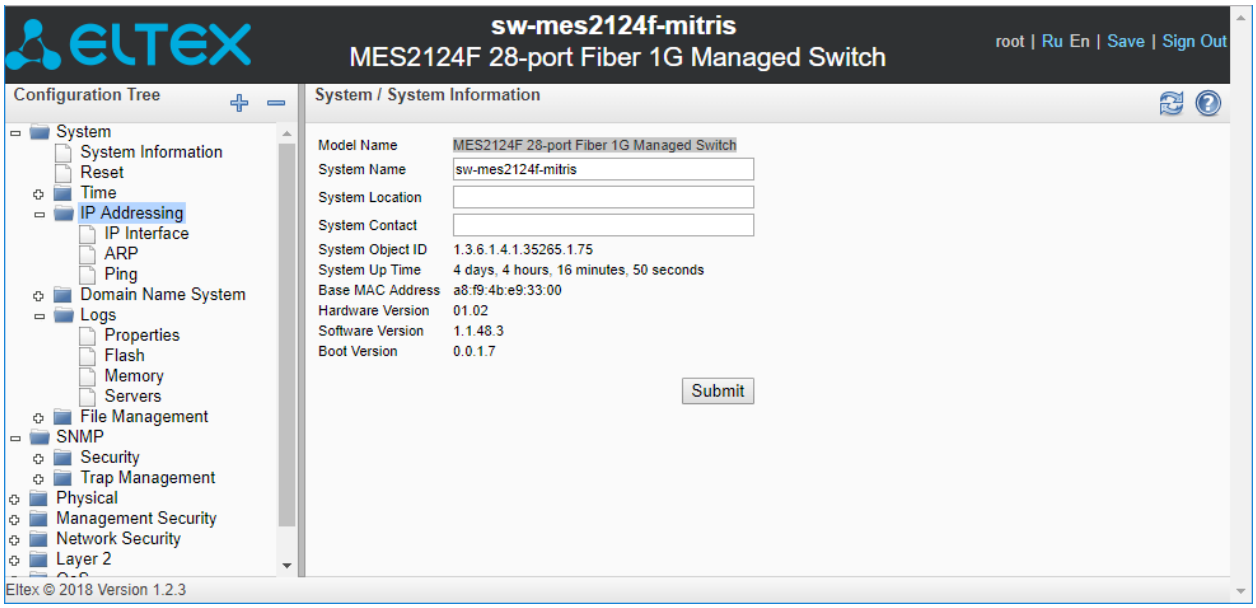

Рис. 2.7. Графический интерфейс управления коммутатором Eltex

2.3.3. Оборудование уровня доступа

Уровень доступа необходим для функционирования рабочих станций и серверов в сети. Коммутаторы уровня доступа отличаются небольшим количеством портов (4 – 24), более низкой стоимостью по отношению к коммутаторам уровня ядра, и упрощенной настройкой.

В виду того, что оборудование уровня доступа взаимодействует непосредственно с оборудованием конечных пользователей сети в организациях и предприятиях сравнимых по масштабу с ФГБОУ Артек используется большое количество различных моделей коммутаторов, поступивших на баланс в разное время и обладающих различной функциональностью и интерфейсом администрирования. В связи с этим комплексный мониторинг коммутаторов уровня доступа представляет собой одну из наиболее сложных задач.

Конкретно в локальной вычислительной сети ФГБОУ МДЦ Артек используются коммутаторы следующих производителей: HP, Q-tech, Mikrotik, Ubiquiti, Netgear, D-link. Поддерживаемые данными коммутаторами протоколы удаленного мониторинга представлены в таблице 2.1.

Таблица 2.1

Поддержка протоколов удаленного мониторинга коммутаторами разных производителей

№	Производитель	Поддержка протокола				
		ICMP	Telnet	SSH	SNMPv1	SNMPv2
1	Hewlett-Packard	Да	Да	Да	Да	Да
2	Q-tech	Да	Да	Нет	Нет	Да
3	Mikrotik (SwOS)	Да	Да	Да	Да	Нет
4	Ubiquiti	Да	Нет	Да	Нет	Нет
5	Netgear	Да	Да	Да	Да	Да
6	D-link	Да	Да	Нет	Нет	Нет

В сети ФГБОУ МДЦ «Артек» за уровень доступа отвечают следующие модели коммутаторов:

1. HPE OfficeConnect 1910 24 Switch (рис. 2.8). Обеспечивает безопасный, простой в использовании графический интерфейс для конфигурирования (рис. 2.9).

Рис. 2.8. HPE OfficeConnect 1910 24 Switch

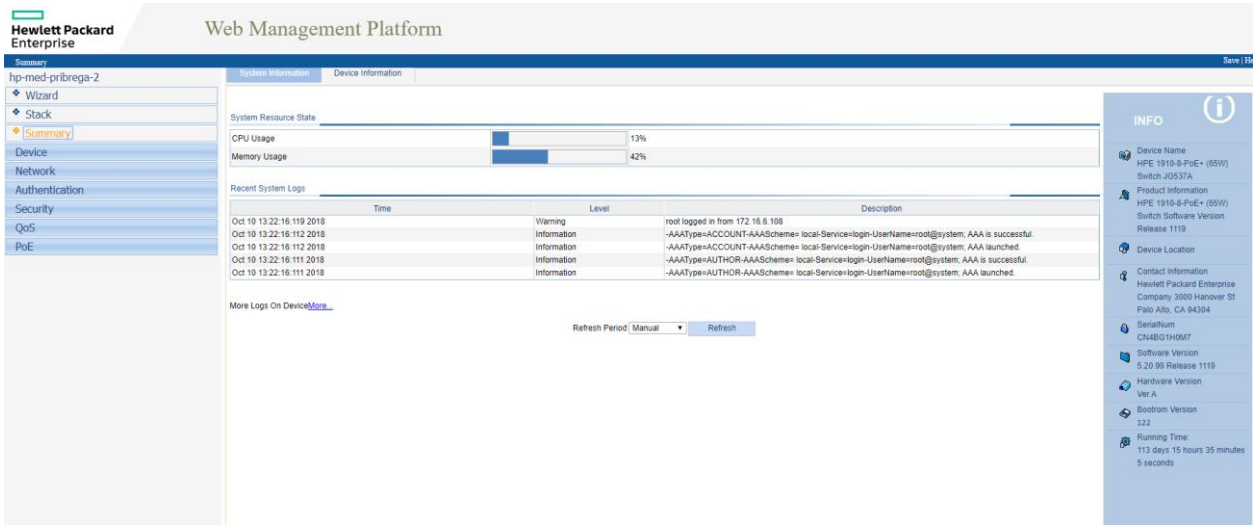

Рис. 2.9. Графический интерфейс управления коммутатором HP

SNMPv1, v2c и v3 облегчает управление коммутатором, поскольку устройство может быть обнаружено и контролироваться с помощью станции управления SNMP.

Единое управление IP позволяет управлять четырьмя устройствами HPE OfficeConnect 1910 с использованием единого веб-интерфейса; упрощает настройку нескольких устройств в одной сети.

2. DLink DES-1210-28P Fast Ethernet Switch (рис. 2.10). Данный коммутатор имеет несколько вариантов управления, обеспечивает быстрое развертывание, расширение инфраструктуры, а также плавное обновление. Графический интерфейс управления коммутатором D-link представлен на рисунке 2.11.

Рис. 2.10. DLink DES-1210-28P Fast Ethernet Switch

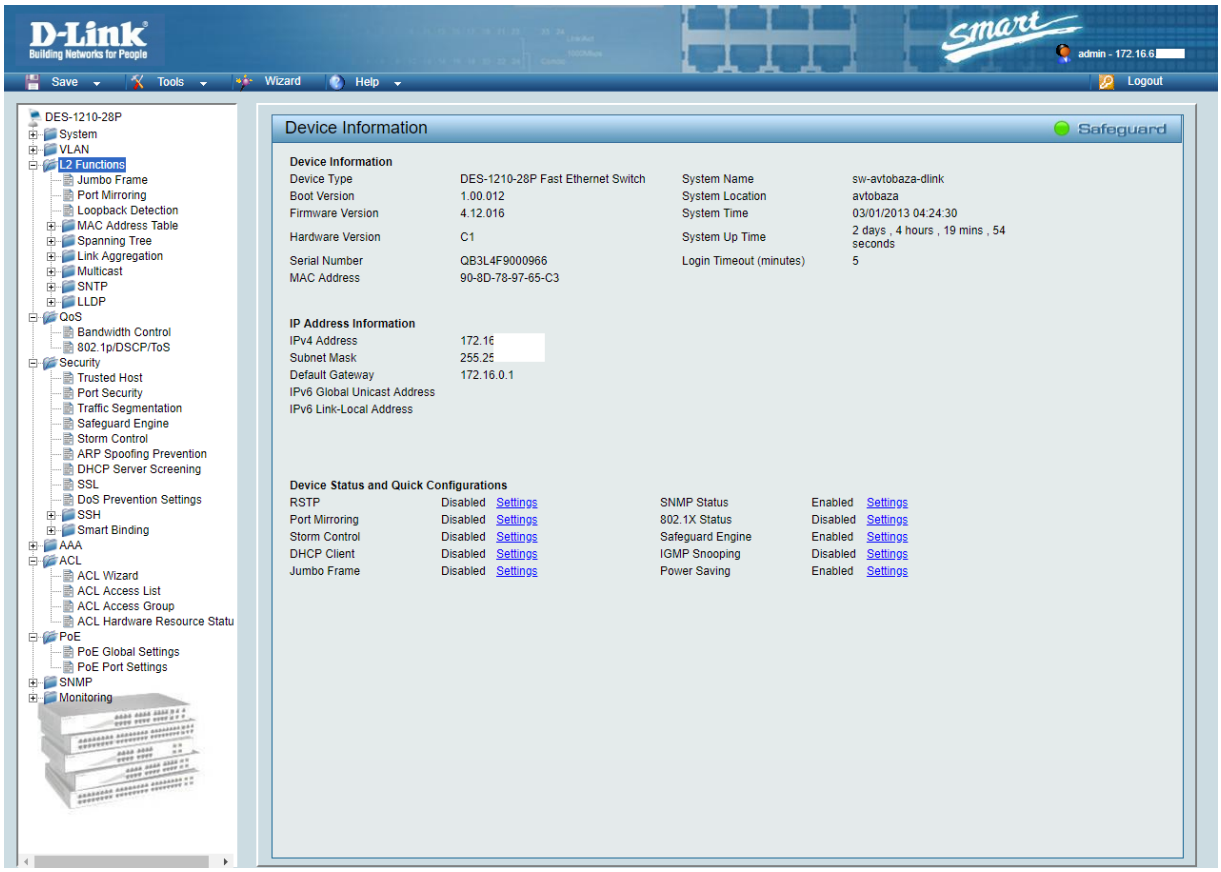

Рис. 2.11. Графический интерфейс управления коммутатором D-link

Имеется поддержка PoE, что упрощает развертывание на его основе IP-камер, VoIP-телефонов, точек беспроводного доступа и других устройств поддерживающий Power-over-Ethernet. Общий запас мощности 193 Вт. С помощью веб-интерфейса можно управлять различными функциями PoE, такими как удаленная перезагрузка камер и телефонов из любой точки сети, в том числе через Интернет.

Вся конфигурация может настраиваться через web-интерфейс независимо от установленной операционной системы клиента. При начальной настройке происходит поиск всех коммутаторов в сети, позволяя нам быстро назначать IP-адреса и маски подсети. Поддерживает одновременное обновление прошивки нескольких коммутаторов, что позволяет сэкономить много времени.

3. Ubiquiti TOUGHSwitch PoE PRO 8-port Gigabit Switch представляет собой коммутатор начального уровня в линейке производителя Ubiquiti характерной чертой которого является простота конфигурирования посредством графического интерфейса (рисунок 2.12).

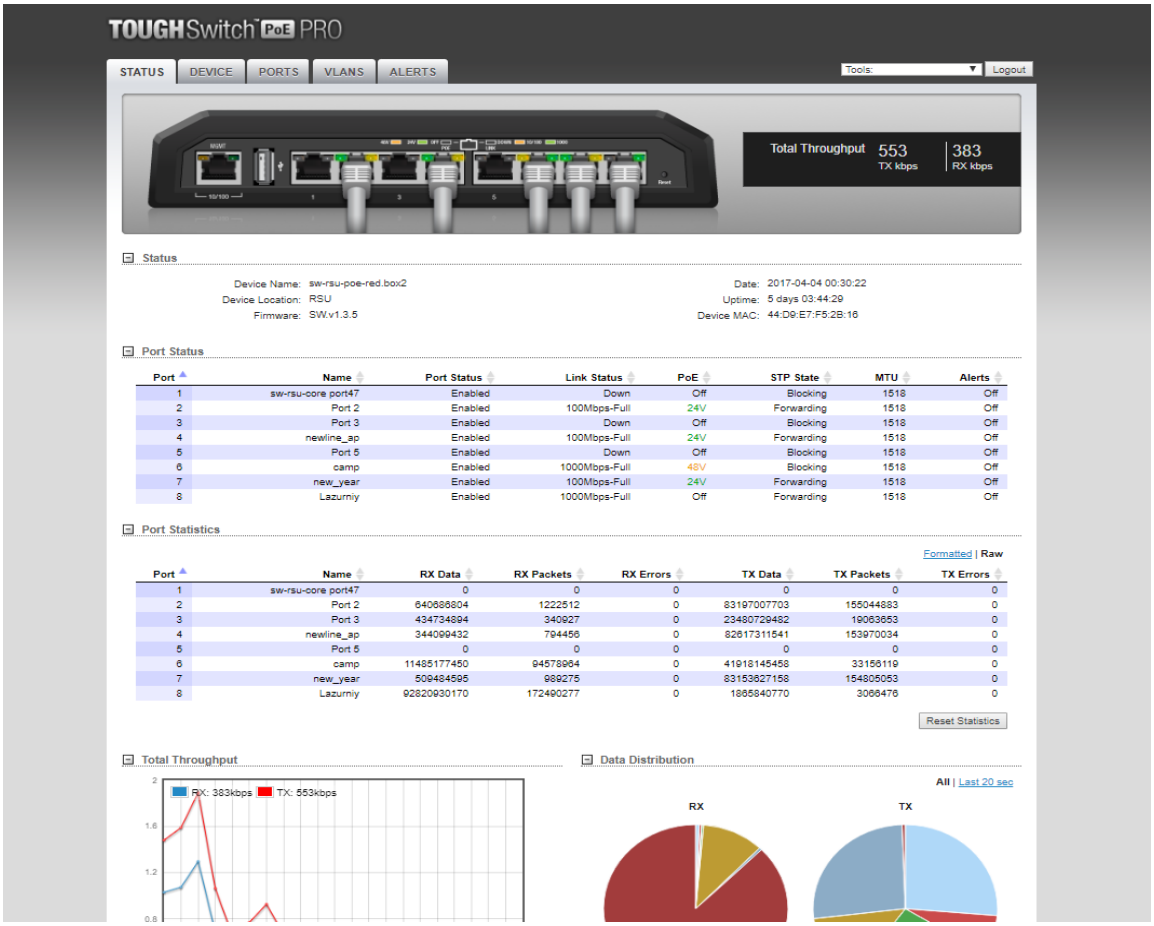

Рис. 2.12. Графический интерфейс управления коммутатором Ubiquiti

Таким образом, комплексный мониторинг коммутаторов уровня доступа представляет собой нетривиальную техническую задачу ввиду гетерогенности используемого сетевого оборудования. Единственным протоколом, гарантированно поддерживающимся коммутаторами различных производителей, является ICMP.

2.4. Активное оборудование каналов радиорелейной связи

Для обеспечения доступа в Интернет отдаленных точек «ФГБОУ МДЦ «Артек» используется оборудование радиорелейной связи Ubiquiti AirFiber (рис. 2.13).

Рис. 2.13. Оборудование радиорелейной связи Ubiquiti AirFiber

Чтобы обеспечить беспроводной мост, необходимо два одинаковых устройства, состоящих из двух антенн. Первая из них работает на оправку, другая на приём (рис. 2.14).

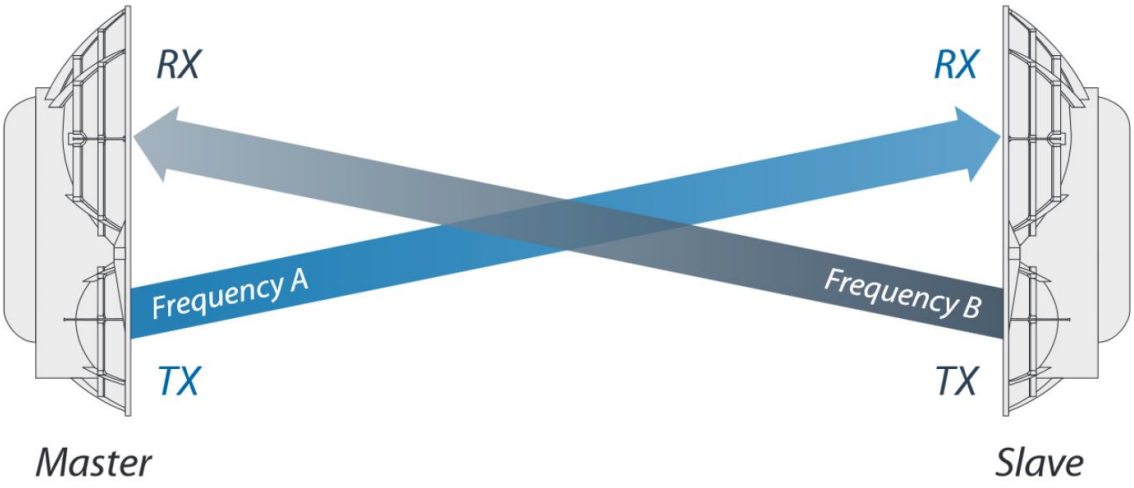

Рис. 2.14. Схема передачи данных посредством РРЛ Ubiquiti AirFiber

Такой канал связи обладает скоростью передачи данных более 1 Гб/с и может работать на расстоянии более 90 км.

Развертывание и настройка канала осуществляется с помощью интерфейса конфигурации AirOS (рис. 2.15). Помимо настройки этот

интерфейс позволяет наглядно видеть текущую скорость передачи, силу сигнала, расстояние между двумя устройствами, а также ширину канала.

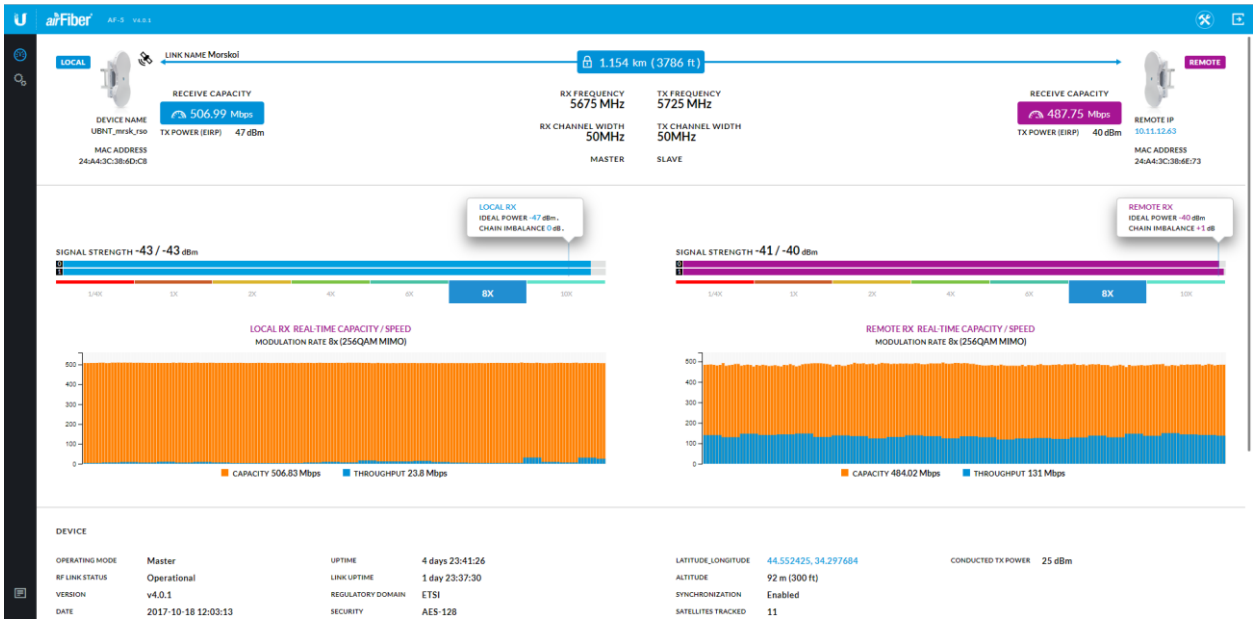

Рис. 2.15. Графический интерфейс управления РРЛ

Выводы по второму разделу

Общепринятой считается трехуровневая иерархическая модель ЛВС. Достоинства подобной структуры в том, что трафик пользователей с множества коммутаторов уровня доступа агрегируется на родительском узле распределения, маршрутизируется или коммутируется по необходимости на вышестоящее ядро, на соседний узел распределения или непосредственно между самими пользователями с разных узлов доступа. А каждое ядро маршрутизирует или коммутирует трафик между несколькими узлами распределения, которые непосредственно включены в него, или между соседними ядрами.

ЛВС ФГБОУ «МДЦ «Артек» представляет собой сложную распределенную сеть выстроенною в соответствии с общепринятыми нормами и правилами построения сетей.

Конкретно в локальной вычислительной сети ФГБОУ МДЦ Артек используются коммутаторы следующих производителей: HP, Q-tech, Mikrotik, Ubiquiti, Netgear, D-link.

Комплексный мониторинг коммутаторов уровня доступа представляет собой нетривиальную техническую задачу ввиду гетерогенности используемого сетевого оборудования. Единственным протоколом, гарантированно поддерживающимся коммутаторами различных производителей, является ICMP.

РАЗДЕЛ 3. ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ МОНИТОРИНГА ЛОКАЛЬНОЙ ВЫЧИСЛИТЕЛЬНОЙ СЕТИ ЛАГЕРЯ «АРТЕК»

3.1.Описание средств реализации программного продукта

3.1.1. Обоснование выбора платформы разработки

В качестве платформы разработки была выбрана виртуальная машина Java(JVM).

JVM – это механизм, обеспечивающий среду выполнения для управления кодом Java или приложениями. Он преобразует байт-код Java в язык машин. JVM является частью JRE (среда запуска Java). Это означает, что виртуальная машина Java

В других языках программирования компилятор создает машинный код для конкретной системы. Однако компилятор Java создает код для виртуальной машины, известной как виртуальная машина Java.

- Во-первых, код Java выполняется в байт-код. Этот байт-код интерпретируется на разных машинах
- Между хост-системой и Java-источником bytecode является языком-посредником.
- JVM отвечает за выделение памяти.

Последовательность загрузки JVM класса в память для дальнейшего выполнения представлена на рис. 3.1.

Рис. 3.1. Порядок загрузки JVM класса

Архитектура JVM содержит загрузчик классов, область памяти, механизм выполнения. Их схематическое взаимодействие представлено на рис. 3.2.

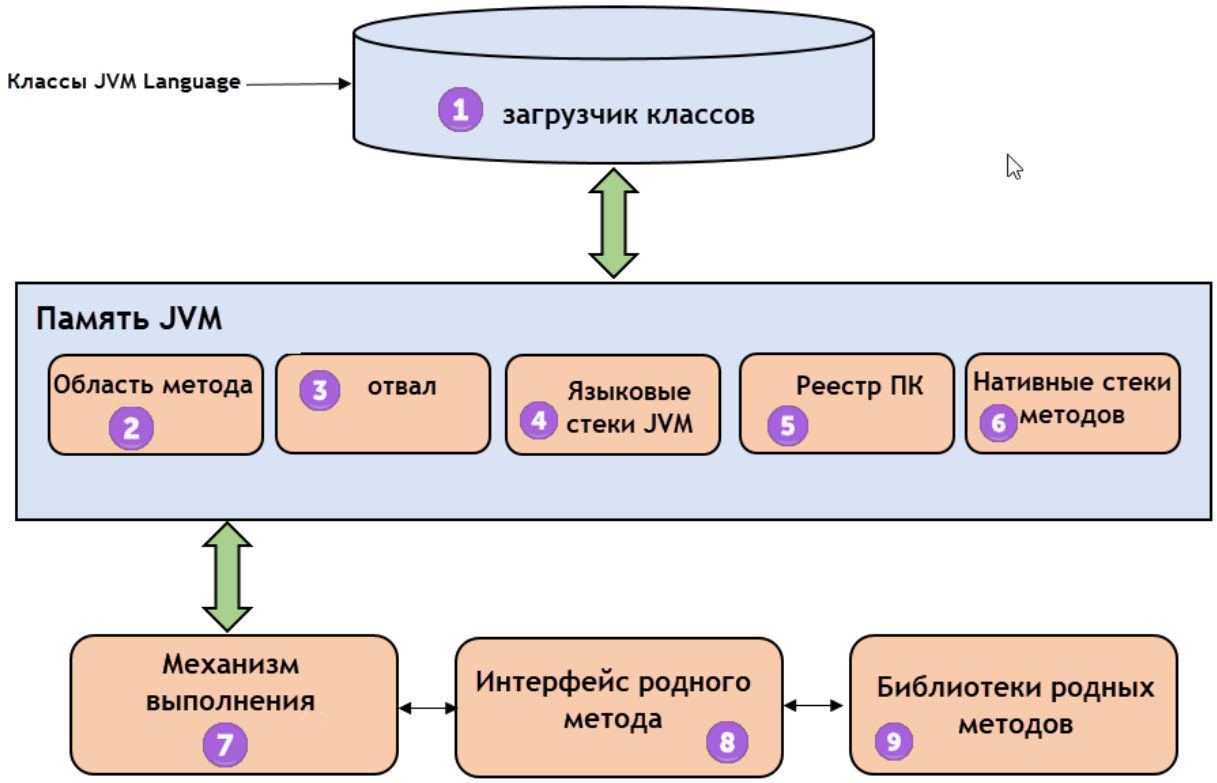

Рис. 3.2. Архитектура JVM

– Загрузчик классов - подсистема, используемая для загрузки файлов классов. Он выполняет три основные функции: Загрузка, связывание и инициализация.
– Область методов JVM хранит структуры классов, такие как метаданные, постоянный пул времени выполнения и код методов.
– Куча. Все объекты, связанные с ними переменные экземпляра и массивы хранятся в куче. Эта память является общей и разделяется между несколькими потоками.
– Стеки языков JVM хранит локальные переменные, и это частичные результаты. Каждый поток имеет свой собственный стек JVM, создаваемый одновременно с созданием потока. Новый кадр

создается всякий раз, когда вызывается метод, и он удаляется, когда процесс вызова метода завершен.

- Реристры ПК хранит адрес инструкции виртуальной машины Java, которая в настоящее время выполняется. В Java каждый поток имеет отдельный регистр ПК.
- Нативные стеки методов содержат инструкцию нативного кода, зависящую от исходной библиотеки. Он написан на другом языке вместо Java.
- Механизм исполнения - тип программного обеспечения, используемого для тестирования оборудования, программного обеспечения или полных систем. Механизм выполнения теста никогда не несет никакой информации о тестируемом продукте.
- Интерфейс на основе метода – это программная среда. Он позволяет Java-коду, запущенному в JVM, вызывать библиотеки и собственные приложения.
- Нативные библиотеки методов - это сборник Native Libraries (C, C ++), который необходим движку Execution Engine.

Благодаря такой архитектуре, JVM позволяет запускать программы, написанные на любом языке программирования имеющим возможность компиляции в byte-код. Так же это позволяет запускать один и тот же код на разных аппаратных платформах без перекомпиляции. Так же возможность использовать огромное количество готовых библиотек

3.1.2. Обоснование выбора языка программирования для реализации серверной части

В качестве языка программирования для разработки серверной части системы мониторинга был выбран язык Clojure.

Clojure – это функциональный, LISP-подобный язык программирования общего назначения, с мощной системой макросов. Главными особенностями которого являются:

- Разработка в REPL-цикле
- Функции являются объектами первого класса
- Параллельное программирование с поддержкой агентной системы, транзакционной памяти и динамических переменных
- Компиляция в Java байт код
- Тесная интеграция с JVM платформой
- Поддержка ленивых последовательностей
- Неизменяемые типы данных

Из-за тесной интеграции с JVM возможно простое использование огромного количества Java-библиотек, а написание кода в функциональном стиле уменьшает его объем и намного облегчает тестирование. Благодаря использованию неизменяемых персистентных типов данных во время выполнения параллельных операций не возникает гонок данных. Разработка в REPL-цикле а так же философия языка «код – это данные» позволяет изменять код в работающем приложении без необходимости перекомпиляции или перезапуска.

3.1.3. Обоснование выбора базы данных

SQLite – это встраиваемая библиотека, которая реализует автономный, безсерверный, транзакционный механизм управления базами данных. В отличии от большинства других баз данных SQLite не нуждается в отдельной системе управления базами данных (СУБД). Вся база данных со всеми таблицами, индексами триггерами и представлениями содержится в одном файле на диске. Формат файла базы данных является кроссплатформенным, что в свою очередь позволяет свободно переносить базу данных между различными архитектурами и системами без ее модификации.

SQLite – это компактная библиотека. Вместе со всеми файлами при включении всех функций размер базы данных не превышает 600 килобайт. Ее использование показало хорошую производительность даже в средах с небольшим количеством памяти.

Данная библиотека очень тщательно тестируется перед выпуском каждой версии и имеет репутацию очень надежной. Ее кодовая база поддерживается международной командой разработчиков.

Таким образом, использование базы данных SQLite в проекте позволило значительно снизить потребляемые системой мониторинга ресурсы в сравнении с использованием ресурсов при аналогичных других средствах разработки.

3.2. Особенности технической реализации серверной части системы мониторинга ЛВС

Перед началом проектирования информационной системы мониторинга локальной сети лагеря «Артек» необходимо определить ее функционал, т.е. функции или операции, которые будут реализованы в ИС.

Функционал информационной системы мониторинга ЛВС ФГБОУ «МДЦ «Артек»:

1) мониторинг ЛВС в реальном времени;
2) хранение результатов мониторинга в базе данных;
3) информирование администраторов ЛВС в случае сбоя важных узлов сети;
4) удаленный доступ к консоли коммутаторов по протоколу SSH
5) предоставление графического интерфейса для наглядного наблюдения за состоянием ЛВС
6) графическое представление результатов мониторинга сети за определенный период.

3.2.1. Описание файловой структуры проекта

Для управления зависимостями проекта используется инструмент автоматизации Leiningen. Следующая структура каталогов (рис.3.3) отражает идиоматический подход, который является общепринятым для проектов на языке Clojure. Благодаря унифицированной структуре сторонние программисты могут без труда разобраться в структуре чужого проекта.

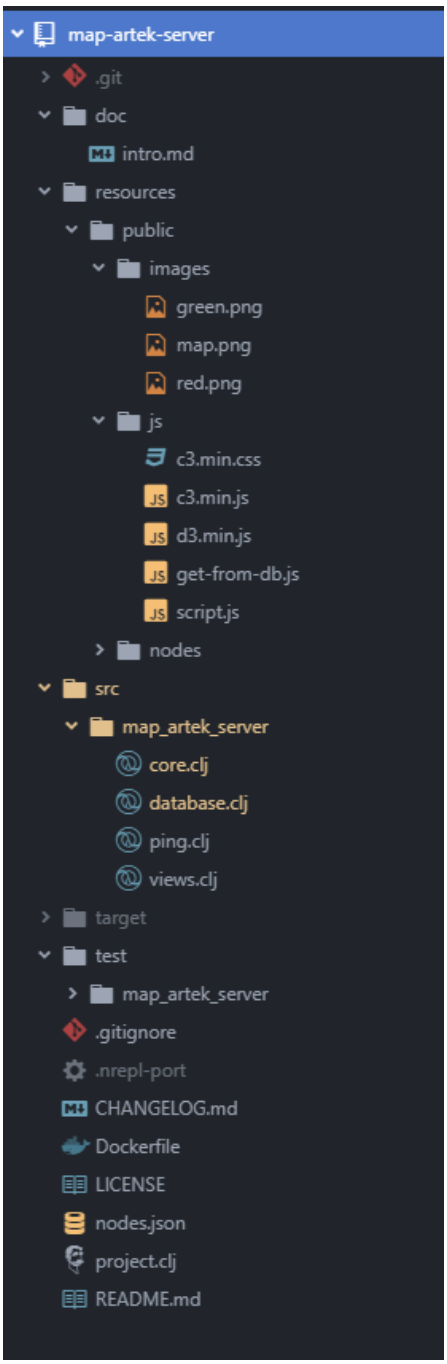

Рис. 3.3. Файловая структура проекта

В корневой папке проекта содержатся следующие файлы:

- .gitingore – список файлов, которые не будут включатся в слепок файловой системы при загрузке проекта в github репозиторий;
- Dockerfile – файл с настройками для запуска в docker контейнере;
- Nodes.json – список узлов сети, мониторинг которых производится системой;
- Project.clj – настройки проекта, здесь указывается список используемых clojure библиотек, плагинов, а также файл, содержащий точку входа в приложение;
- README.MD – содержит краткое описание проекта и инструкцию по запуску.

В папке db всего один файл database.db – база данных, в которой хранится история результатов мониторинга за последнюю неделю.

В папке src находятся файлы перечисленные ниже:

- core.clj – главная часть системы мониторинга, именно сюда подключаются все остальные файлы проекта, тут находится функция main, здесь запускается обработчик запросов и WebSocket-сервер;
- database.clj – модуль, предоставляющий интерфейс для записи и чтения из базы данных;
- ping.clj – модуль, в котором реализована функциональность отправки ICMP эхо-пакетов;
- ssh-client.clj – реализация ssh клиента для удаленного подключения к высокоскоростным коммутаторам;
- views.clj – файл описания HTML страничек с помощью структур данных языка clojure.

Хранение файлов ИС в системе контроля версий показано на рис. 3.4.

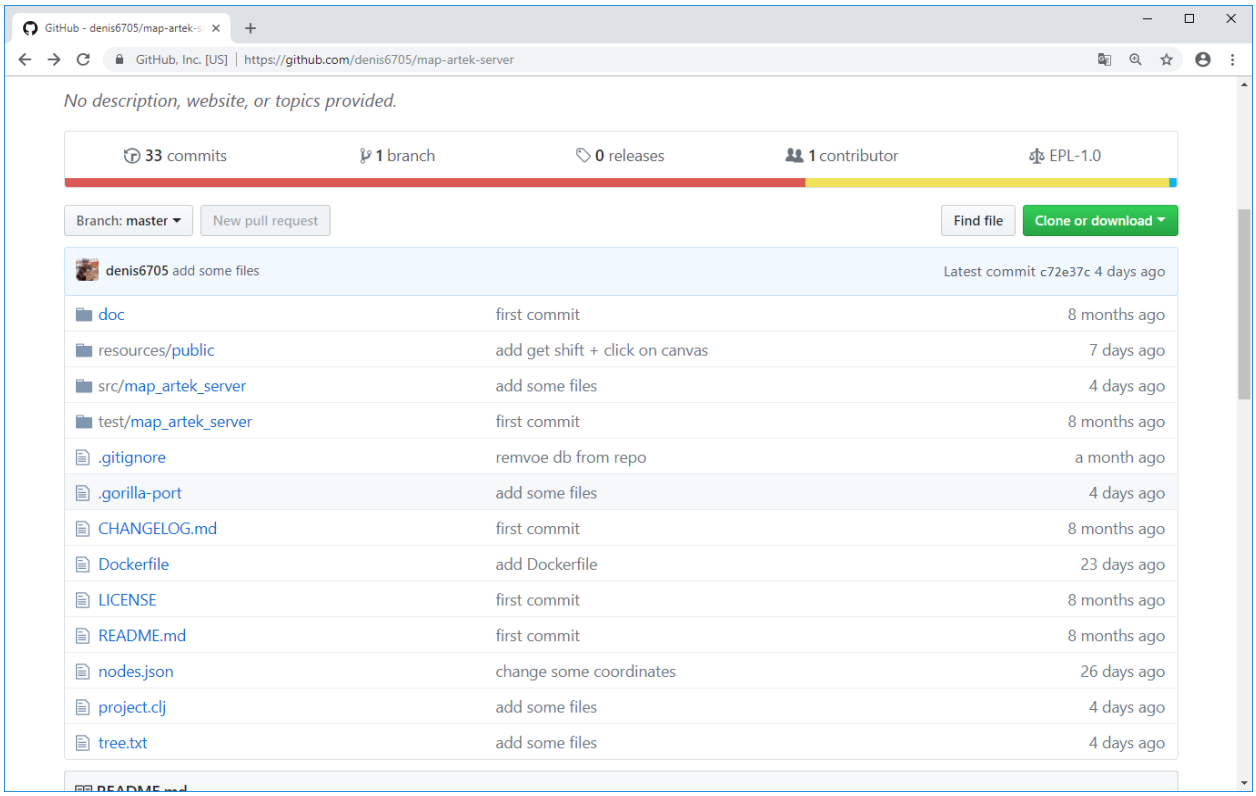

Рис. 3.4. Хранение файлов ИС в системе контроля версий

3.2.2. Общая структура серверной части

Сервер системы мониторинга ЛВС, представленный на рисунке 3.5, состоит из следующих модулей:

- обработчик запросов,
- модуль работы с базой данных (БД),
- модуль мониторинга,
- модуль оповещения,
- Secure Shell (SSH) клиента.

Серверная часть системы мониторинга реализовывает следующий функционал:

- мониторинг ЛВС в реальном времени;
- хранение результатов мониторинга в базе данных;

– информирование администраторов ЛВС в случае сбоя важных узлов сети;
– удаленный доступ к консоли коммутаторов по протоколу SSH

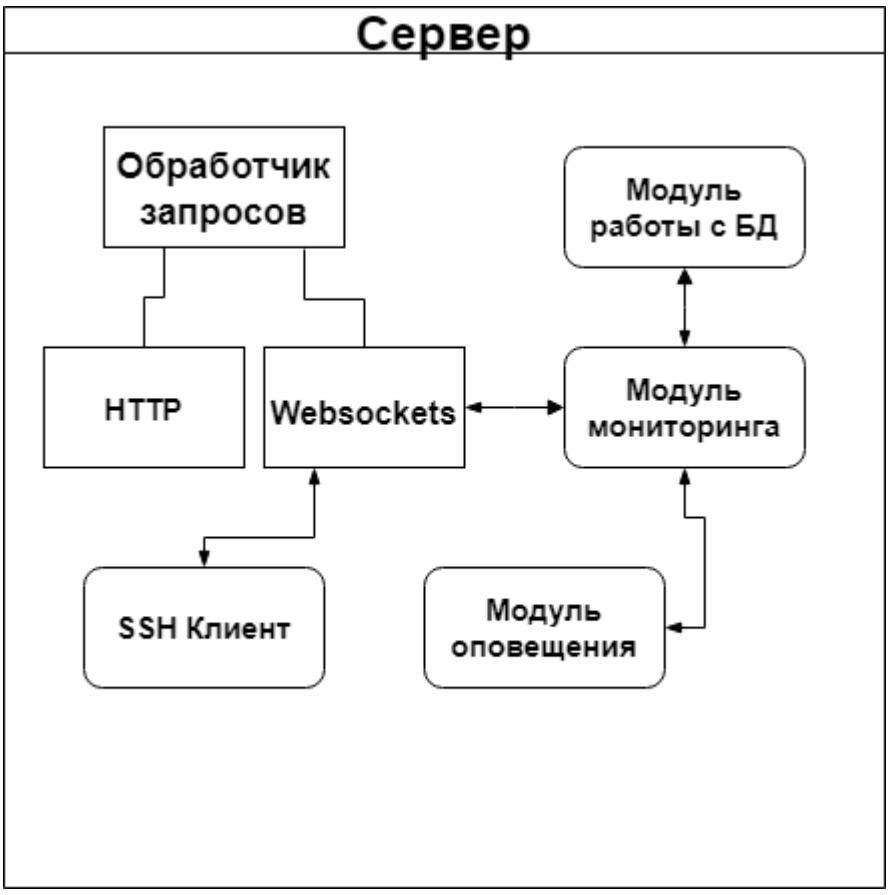

Рис. 3.5. Схема работы серверной части программы мониторинга

3.2.3. Обработчик запросов

Обработчик запросов является связующим звеном в системе мониторинга ЛВС. Он принимает все входящие запросы и в зависимости от типа протокола направляет их в другую часть системы.

В случае если запрос приходит по HyperText Transfer Protocol (HTTP) – это значит, что пользователь запрашивает одну из HyperText Markup Language (HTML) страниц. Обработчик запросов извлекает необходимые параметры и возвращает пользователю запрашиваемую HTML страницу.

Если же запрос приходит по протоколу WebSocket, никакие параметры не извлекаются, а сам запрос передается далее в модуль работы с WebSocket.

Благодаря обработчику запросов архитектура серверной части стала более модульной, так как без него в трех других модулях (HTTP, Websockets, модуль мониторинга) происходит дублирование кода.

Программный код выглядит так, как показано на рис. 3.6.

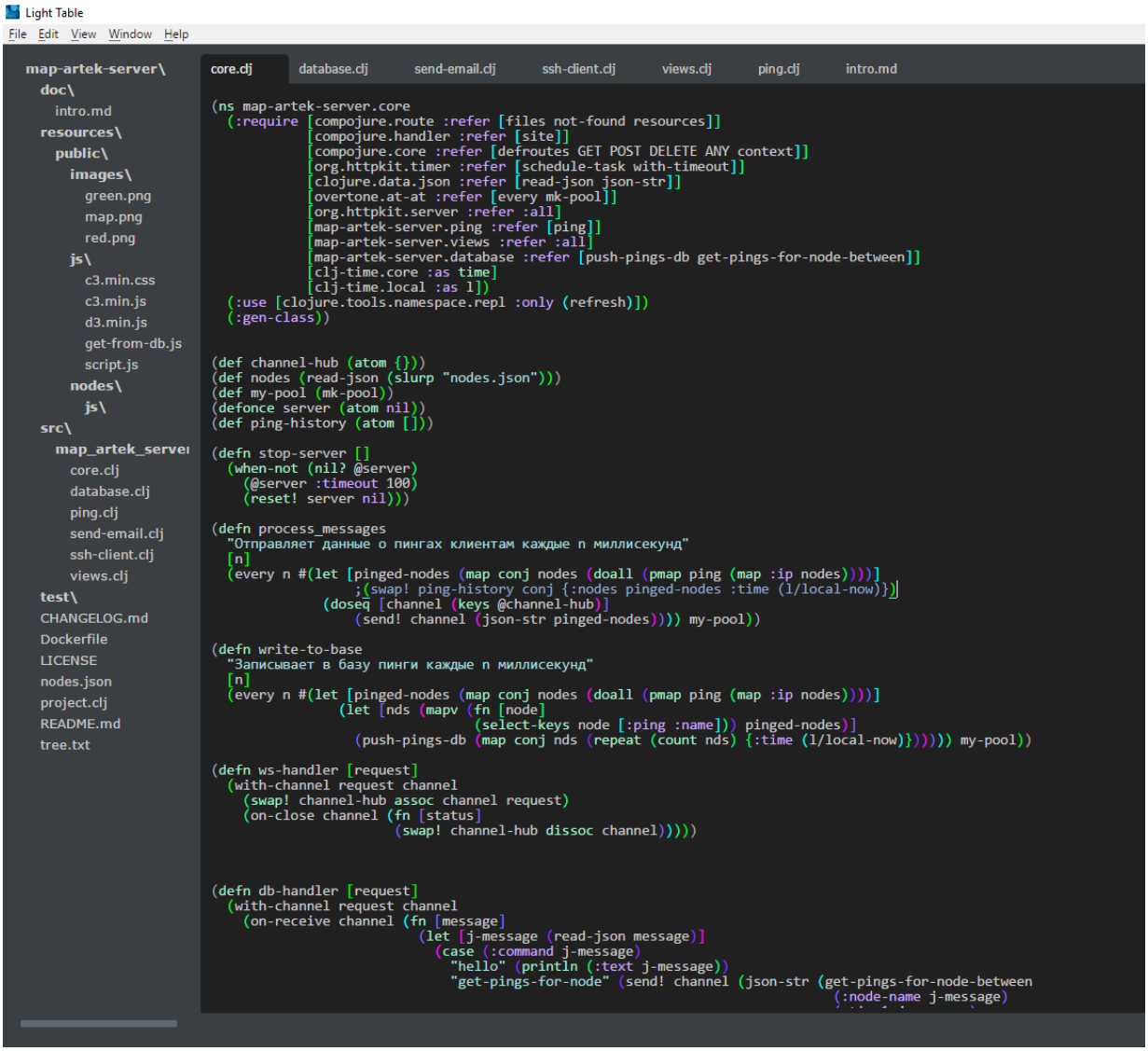

Рис. 3.6. Окно программного кода в редакторе Light Table

3.2.4. Модуль работы с WebSocket

Модуль работы с WebSocket предназначен для установления постоянной двунаправленной связи серверной части системы мониторинга с клиентской частью. Состоит из WebSoket – сервера, и менеджера сообщений.

WebSocket сервер хранит информацию о всех, подключенных в данный момент клиентах, получает от них запросы и передает их в менеджер сообщений. А также отвечает за отправку сообщений клиенту из других модулей

Менеджер сообщений разбирает и анализирует поступающие из WebSocket-сервера запросы. Первым делом он преобразует сообщения из JavaScript Object Notation (JSON) формата в хеш-таблицу. Затем в зависимости от значения ключа “command”, выполняется требуемое действие. Пример сообщения представлен на рис. 3.7.

```
{
    command :   "get-pings-from-db",
    node-name:  "Хрустальный"
    from_date:  "26.11.2018#07:00:00"
    to_date:    "26.11:2018#12:30:00"
}
```

Рис. 3.7. Пример JSON-сообщения

Структура сообщения может быть произвольного формата, единственным ограничением является присутствия ключа “command”. Менеджер сообщений по названию команды сам понимает какие еще параметры должны находится в сообщении и пытается их извлечь. В случае неудачи сообщение отбрасывается, при этом модуль продолжает нормально выполнятся.

Благодаря такой структуре, очень легко добавлять и тестировать новую функциональность без риска вывода системы из рабочего состояния.

3.2.5. Модуль мониторинга ЛВС

Модуль мониторинга ЛВС отвечает за анализ доступности объектов, подлежащих мониторингу, анализ результатов и их запись в базу данных.

Общая схема работы модуля представлена на рис. 3.8.

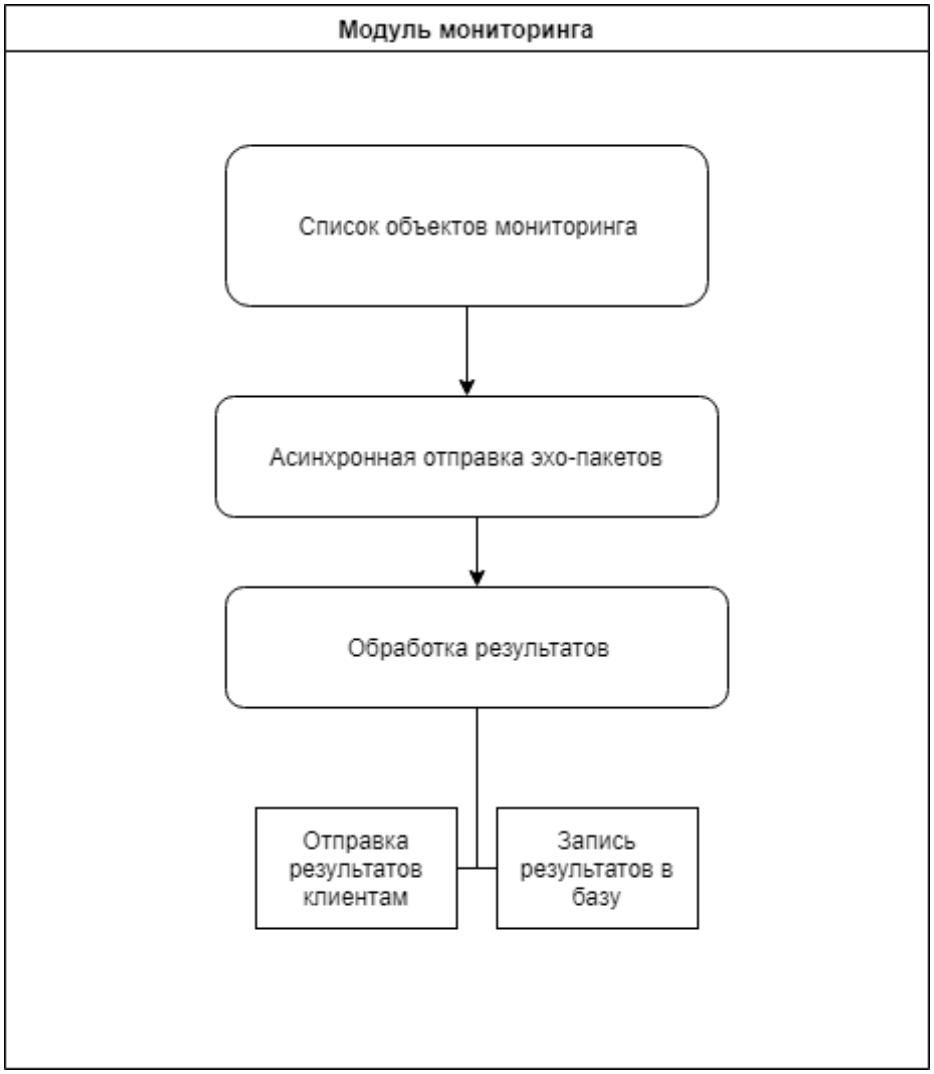

Рис. 3.8. Схема работы модуля мониторинга

Список объектов, подгружается динамически из файла в формате JSON и преобразуется в вектор хеш-таблиц. Вид одного из элементов JSON списка изображен на рисунке 3.9-3.10.

```
{
    "name":         "Янтарный",
    "ip":           "172.20.255.11",
    "importance":   "low",
    "ssh_support":  true,
    "x":            1830,
    "y":            355
}
```

Рис. 3.9. Вид JSON объекта узла сети

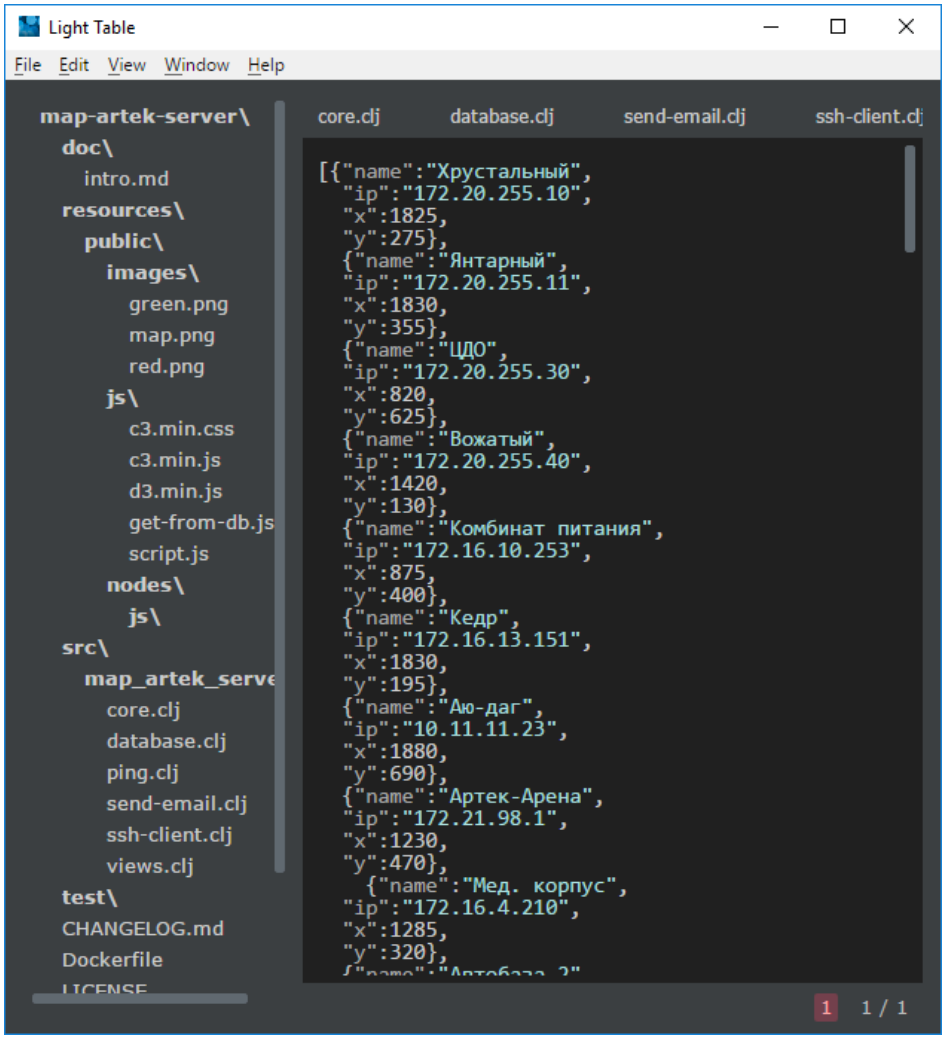

Рис. 3.10. Формат хранения и передачи данных между клиентом и сервером

Из каждой хеш-таблицы извлекается поле “:ip” и собирается в отдельный вектор, после чего этот вектор передается в функцию pmap, которая параллельно применяет функцию ping к каждому ip-адресу, в результате чего каждому узлу из списка отправляется ICMP запрос. Результатом функции

pmap является вектор хеш-таблиц вида {:status (true/false) :ping integer}. Результирующий вектор объединяется с исходным. Обработанные результаты, предварительно преобразованные в JSON, отправляются клиенту по протоколу WebSocket и записываются в базу данных.

На основании значения поля “importance” делается вывод о том к какому уровню значимости относится конкретный узел сети. В случае если его уровень важности соответствует “high” и от него нет эхо ответа в течении заданного времени, модуль мониторинга отправляет команду модулю оповещения о том, что необходимо уведомить администраторов, отвечающих за данный коммутатор.

3.2.6. SSH клиент

SSH клиент – модуль системы мониторинга, отвечающий за создание защищенного канала между серверной частью системы мониторинга и командной консолью высокоскоростных коммутаторов.

Так как не все коммутаторы имеют возможность подключения по протоколу SSH, то в файле, хранящем список подлежащих мониторингу узлов, в записи о каждом из узлов хранится свойство ssh имеющее тип Boolean. Если значение равно true, SSH клиент будет осуществлять попытку подключения, если же оно установлено в false, то подключение осуществлено не будет.

Если во время работы с коммутатором произошло отключение, выскакивает соответствующее сообщение.

При инициации подключения в веб-интерфейсе появляется сообщение «Идет подключение к высокоскоростному коммутатору «Имя корпуса …». Если подключение прошло успешно выводится сообщение «Подключение установлено». Пример подключения приведен на рисунке 3.11.

Такая интеграция SSH клиента с веб-интерфейсом системы мониторинга избавляет от необходимости использования стороннего программного обеспечения, а также позволяет подключатся к коммутаторам без ввода пароля

и имени пользователя, которые хранятся в зашифрованном файле паролей на сервере.

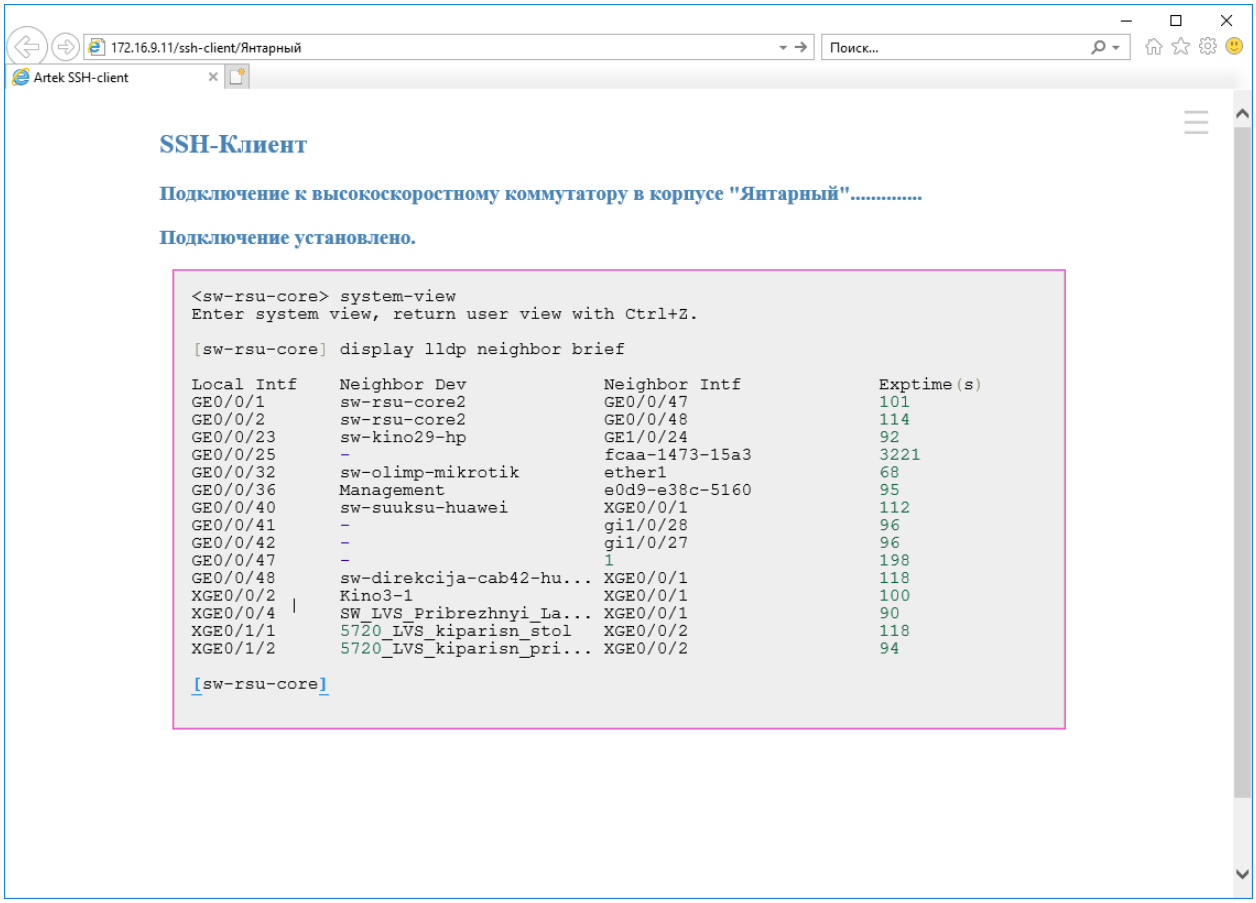

Рис. 3.11. Сеанс работы с коммутатором по протоколу SSH

3.3. Реализация клиентской части системы мониторинга компьютерной сети ФГБОУ "МДЦ "Артек"

Общая структура клиентской части изображена на рисунке и представляет из себя HTML-страничку с элементом canvas и подключенным файлом JavaScript. В JavaScript модуле реализован WebSocket-клиент, который подключается к серверу. Сервер в свою очередь начинает отправлять результаты мониторинга клиенту (рис. 3.12).

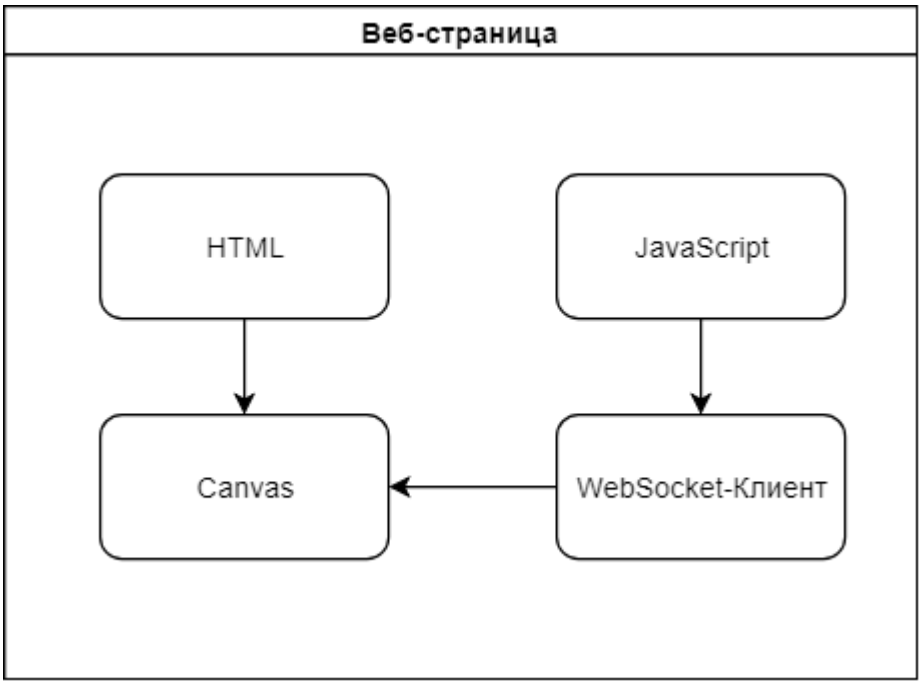

Рис. 3.12. Схема веб-страницы

Главное окно программы (рис. 3.13) представляет собой интерактивную карту предприятия (учреждения), на которой цветными индикаторами отображается текущее состоянии устройств. При щелчке правой кнопкой мыши по цветному индикатору открывается информационная станица узла сети, в которой на графике в реальном времени отображается показатель задержки.

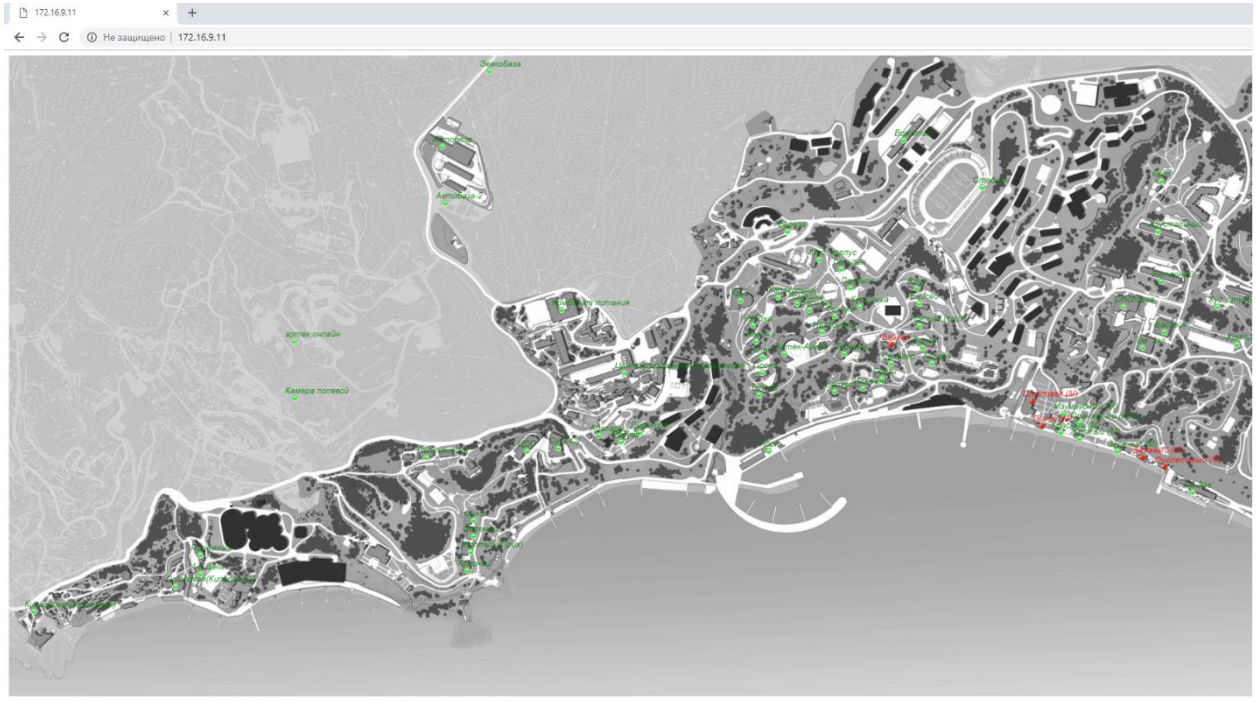

Рис. 3.13. Интерфейс программы мониторинга ЛВС

Изображение на интерактивной карте обновляется каждые несколько секунд, что позволяет видеть текущее состояние узлов сети, и быстро реагировать на происходящие сбои оборудования. Благодаря отображению информации на карте, новым сотрудникам легче понять, куда нужно ехать для устранения неполадок (в связи с большой протяженностью территории лагеря «Артек» для оперативного устранения неполадок сети в лагере имеется электромобиль, находящийся в ведении вычислительного центра).

При щелке мыши по цветному индикатору карты с нажатой клавишей Shift, открывается такое же окно, но с возможностью вывода информации из базы данных за определенный период в виде графика полученного и переданного трафика, что отображено на рисунке 3.14-3.17.

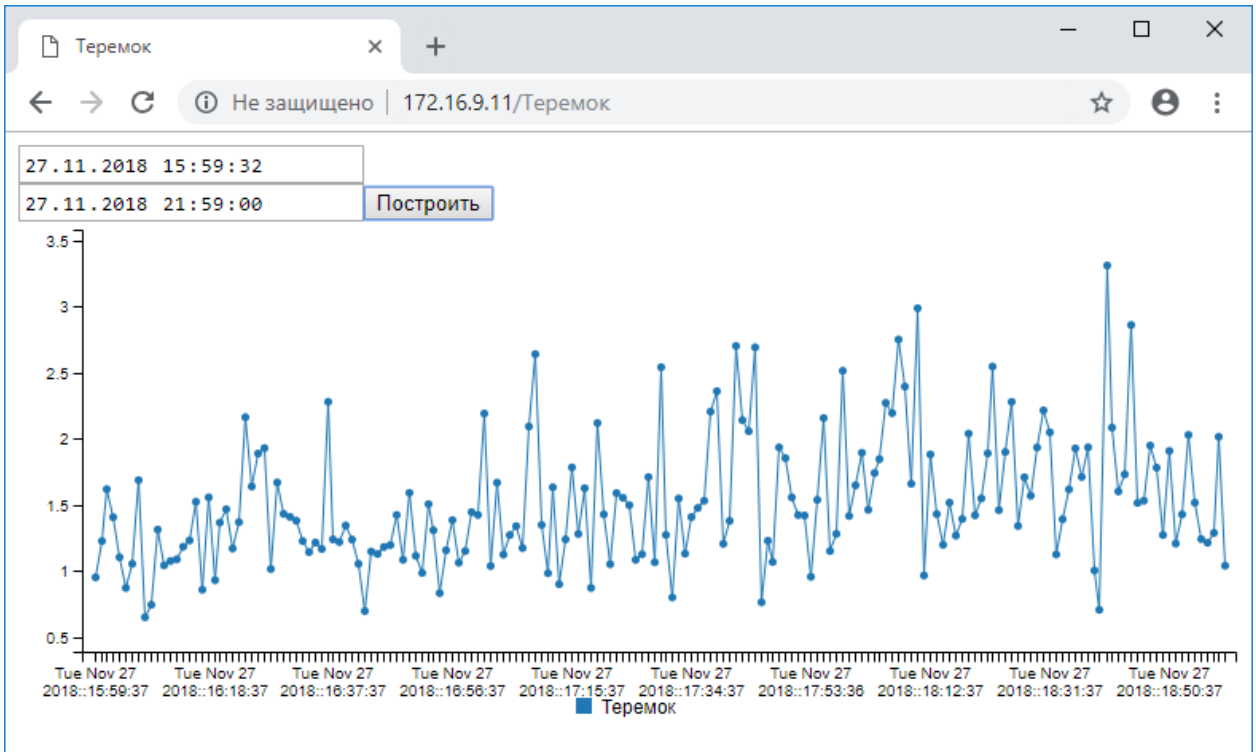

Рис. 3.14. Информация о мониторинге оборудования за указанный промежуток времени

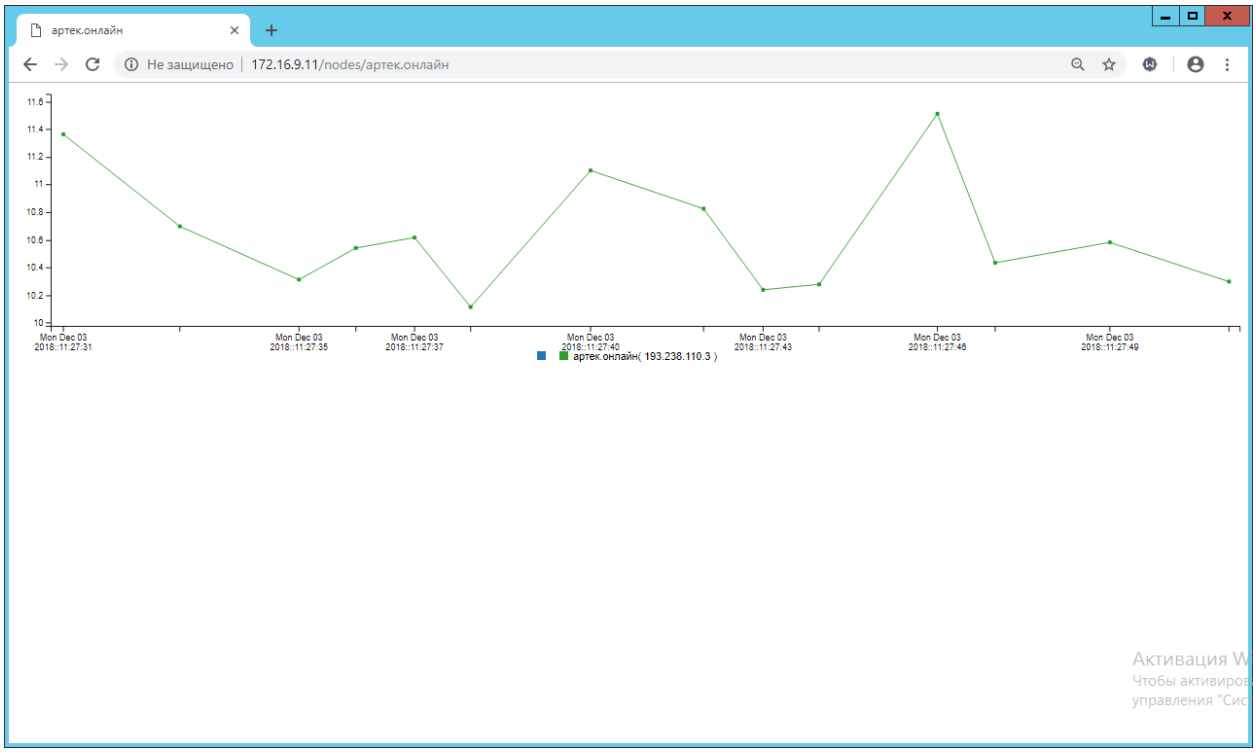

Рис. 3.15. Информация о мониторинге оборудования в режиме реального времени

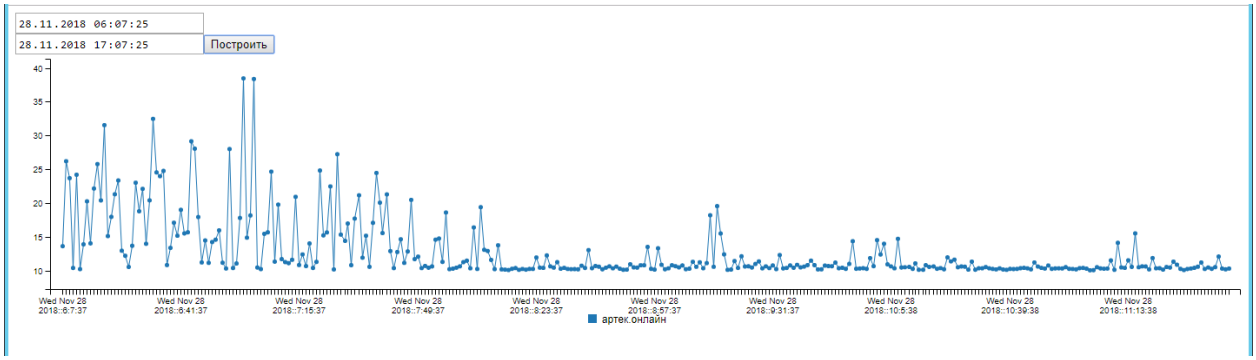

Рис. 3.16. Информация о мониторинге оборудования за указанный промежуток времени (нормальная работа оборудования – *max=400)*

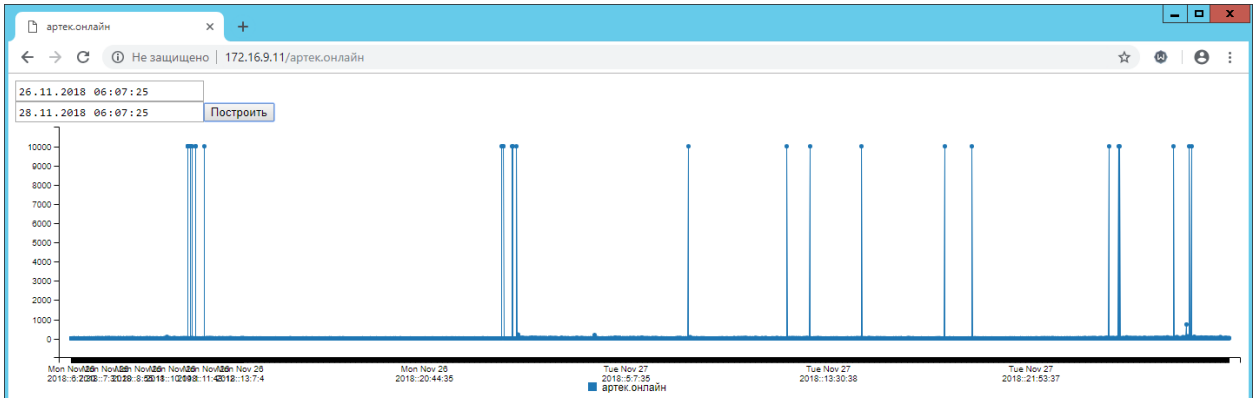

Рис. 3.17. Информация о мониторинге оборудования за указанный промежуток времени (аномальные пики до 1 000)

Использование веб-интерфейса, позволяет производить запуск информационной системы на любой операционной системе в браузере, без необходимости предварительной установки. Если раньше о проблемах сети можно было узнать по заявке в helpdesk, то сейчас вычислительный центр получает информацию о сбоях в работе оборудования и об их неисправности зачастую раньше самих пользователей, что является значительным преимуществом.

Вывод по третьему разделу

Выбранная в качестве платформы для разработки JVM позволяет запускать серверную часть системы мониторинга на любой операционной системе.

Выбор в качестве базы данных SQLite позволил сэкономить ресурсы операционной системы, так как для ее работы не нужно устанавливать систему управления базами данных.

Благодаря клиент-серверной архитектуре код проекта клиентской и серверной части разделен. Использование клиент-серверной архитектуры позволяет снизить нагрузку на сеть, так как эхо запросы отправляются с одного ip –адреса сервера.

Реализованный веб-интерфейс позволяет производить запуск системы мониторинга на любом устройстве с веб-браузером без необходимости установки.

Таким образом, разработанная система мониторинга ЛВС ФГБОУ «МДЦ «Артек» реализовывает следующий функционал:

- Мониторинг ЛВС в реальном времени;
- Хранение результатов мониторинга в базе данных;
- Информирование администраторов ЛВС в случае сбоя важных узлов сети;
- Удаленный доступ к консоли коммутаторов по протоколу SSH;
- Предоставление графического интерфейса для наглядного наблюдения за состоянием ЛВС;
- Графическое представление результат;ов мониторинга сети за определенный период.

РАЗДЕЛ 4. ОХРАНА ТРУДА И ТЕХНИКА БЕЗОПАСНОСТИ ПРИ РАБОТЕ С КОМПЬЮТЕРОМ

Требования безопасности перед началом работы

Перед началом работы следует убедиться в исправности электропроводки, выключателей, штепсельных розеток, при помощи которых оборудование включается в сеть, наличии заземления компьютера, его работоспособности.

Требования безопасности во время работы

Для снижения или предотвращения влияния опасных и вредных факторов необходимо соблюдать определенные правила и нормы.

Во избежание повреждения изоляции проводов и возникновения коротких замыканий не разрешается: вешать что-либо на провода, закрашивать и белить шнуры и провода, закладывать провода и шнуры за газовые и водопроводные трубы, за батареи отопительной системы, выдергивать штепсельную вилку из розетки за шнур, усилие должно быть приложено к корпусу вилки.

Для исключения поражения электрическим током запрещается: часто включать и выключать компьютер без необходимости, прикасаться к экрану и к тыльной стороне блоков компьютера, работать на средствах вычислительной техники и периферийном оборудовании мокрыми руками, работать на средствах вычислительной техники и периферийном оборудовании, имеющих нарушения целостности корпуса, нарушения изоляции проводов, неисправную индикацию включения питания, с признаками электрического напряжения на корпусе, класть на средства вычислительной техники и периферийном оборудовании посторонние предметы.

Запрещается под напряжением очищать от пыли и загрязнения электрооборудование.

Запрещается проверять работоспособность электрооборудования в неприспособленных для эксплуатации помещениях с токопроводящими полами, сырых, не позволяющих заземлить доступные металлические части.

Недопустимо под напряжением проводить ремонт средств вычислительной техники и периферийного оборудования. Ремонт электроаппаратуры производится только специалистами-техниками с соблюдением необходимых технических требований.

Во избежание поражения электрическим током, при пользовании электроприборами нельзя касаться одновременно каких-либо трубопроводов, батарей отопления, металлических конструкций, соединенных с землей.

При пользовании электроэнергией в сырых помещениях соблюдать особую осторожность.

Требования безопасности в аварийных ситуациях

При обнаружении неисправности немедленно обесточить электрооборудование, оповестить администрацию. Продолжение работы возможно только после устранения неисправности.

При обнаружении оборвавшегося провода необходимо немедленно сообщить об этом администрации, принять меры по исключению контакта с ним людей. Прикосновение к проводу опасно для жизни.

Во всех случаях поражения человека электрическим током немедленно вызывают врача. До прибытия врача нужно, не теряя времени, приступить к оказанию первой помощи пострадавшему.

На рабочем месте запрещается иметь огнеопасные вещества

В помещениях запрещается:

- зажигать огонь;
- включать электрооборудование, если в помещении пахнет газом;
- курить;
- сушить что-либо на отопительных приборах;
- закрывать вентиляционные отверстия в электроаппаратуре.

При возникновении пожароопасной ситуации или пожара персонал должен немедленно принять необходимые меры для его ликвидации, одновременно оповестить о пожаре администрацию.

Специфика трудовой деятельности в организациях предусматривает непрерывную работу за вычислительной техникой персонала в течение всего рабочего дня.

В связи с этим, руководителю предприятия требуется в надлежащей степени организовать рабочие места в соответствии со всеми требованиями законодательства, касающимися трудовой деятельности за компьютеризированными системами.

Нормативное регулирование охраны труда при осуществлении трудовой деятельности за компьютерами осуществляется посредством следующих документов:

- Типовая инструкция ТОИ Р-45-084-01;
- СанПиН 2.2.2. / 2.4. 1340-03 (далее – СанПиН);
- ТК РФ;
- Приказ Минздравсоцразвития РФ № 302н;
- 426-ФЗ.

1. К работе на персональном компьютере (ПК) допускаются лица, прошедшие медицинское освидетельствование, вводный инструктаж, первичный инструктаж, обучение и стажировку на рабочем месте, проверку знаний требований охраны труда, имеющие группу I по электробезопасности.

2. При работе на персональном компьютере работник обязан:

1. Выполнять работу, которая определена его должностной (рабочей) инструкцией.
2. Выполнять правила внутреннего трудового распорядка.
3. Соблюдать режим труда и отдыха в зависимости от продолжительности, вида и категории трудовой деятельности.

4. Правильно применять средства индивидуальной и коллективной защиты.

5. Соблюдать требования охраны труда.

6. Немедленно извещать своего руководителя о любой ситуации, угрожающей жизни и здоровью людей, происшедшем на рабочем месте, или об ухудшении состояния своего здоровья.

7. Проходить обучение безопасным методам и приемам выполнения работ, и оказанию первой помощи пострадавшим на производстве, инструктаж по охране труда, проверку знаний требований охраны труда.

8. Уметь оказывать первую помощь пострадавшим от электрического тока и при других несчастных случаях.

9. Уметь применять первичные средства пожаротушения.

3. При эксплуатации персонального компьютера на работника могут оказывать действие следующие опасные и вредные производственные факторы:

- повышенный уровень электромагнитных излучений;
- повышенный уровень статического электричества;
- пониженная ионизация воздуха;
- статические физические перегрузки;
- перенапряжение зрительных анализаторов;
- недостаточная освещенность рабочего места.

4.Конструкция ПЭВМ должна обеспечивать возможность поворота корпуса в горизонтальной и вертикальной плоскости с фиксацией в заданном положении для обеспечения фронтального наблюдения экрана ВДТ. Дизайн ПЭВМ должен предусматривать окраску корпуса в спокойные мягкие тона с диффузным рассеиванием света. Корпус ПЭВМ, клавиатура и другие блоки и устройства ПЭВМ должны иметь матовую поверхность с коэффициентом отражения 0,4 - 0,6 и не иметь блестящих деталей, способных создавать блики.

5.Конструкция ВДТ должна предусматривать регулирование яркости и контрастности.

6.Площадь на одно рабочее место пользователей ПЭВМ с ВДТ на базе электронно-лучевой трубки (ЭЛТ) должна составлять не менее 6 $м^2$, в помещениях культурно развлекательных учреждений и с ВДТ на базе плоских дискретных экранов (жидкокристаллические, плазменные) - 4,5 $м^2$. При использовании ПВЭМ с ВДТ на базе ЭЛТ (без вспомогательных устройств - принтер, сканер и др.), отвечающих требованиям международных стандартов безопасности компьютеров, с продолжительностью работы менее 4-х часов в день допускается минимальная площадь 4,5 $м^2$ на одно рабочее место пользователя (взрослого и учащегося высшего профессионального образования).

7.Помещения, где размещаются рабочие места с ПЭВМ, должны быть оборудованы защитным заземлением (занулением) в соответствии с техническими требованиями по эксплуатации.

8.Рабочие места с компьютерами должны размещаться таким образом, чтобы расстояние от экрана одного видеомонитора до тыла другого было не менее 2м, а расстояние между боковыми поверхностями видеомониторов - не менее 1,2м.

9.Рабочие столы следует размещать таким образом, чтобы видеодисплейные терминалы были ориентированы боковой стороной к световым проемам, чтобы естественный свет падал преимущественно слева.

10.Оконные проемы в помещениях, где используются персональные компьютеры, должны быть оборудованы регулируемыми устройствами типа: жалюзи, занавесей, внешних козырьков и др.

11.Искусственное освещение в помещениях для эксплуатации ПЭВМ должно осуществляться системой общего равномерного освещения.

12.Экран видеомонитора должен находиться от глаз пользователя на расстоянии 600 - 700 мм, но не ближе 500 мм с учетом размеров алфавитно-цифровых знаков и символов.

13.Рабочая мебель для пользователей компьютерной техникой должна отвечать следующим требованиям:

– высота рабочей поверхности стола должна регулироваться в пределах 680-800мм; при отсутствии такой возможности высота рабочей поверхности стола должна составлять 725мм;

– рабочий стол должен иметь пространство для ног высотой не менее 600 мм, шириной - не менее 500 мм, глубиной на уровне колен - не менее 450 мм и на уровне вытянутых ног - не менее 650 мм;

– рабочий стул (кресло) должен быть подъемно-поворотным, регулируемым по высоте и углам наклона сиденья и спинки, а также расстоянию спинки от переднего края сиденья, при этом регулировка каждого параметра должна быть независимой, легко осуществляемой и иметь надежную фиксацию;

– клавиатуру следует располагать на поверхности стола на расстоянии 100 - 300 мм от края, обращенного к пользователю, или на специальной, регулируемой по высоте рабочей поверхности, отделенной от основной столешницы.

14. В помещениях и кабинетах, оборудованных ПЭВМ, проводится ежедневная влажная уборка и систематическое проветривание после каждого часа работы на ПЭВМ.

15.Женщины со времени установления беременности переводятся на работы, не связанные с использованием ПЭВМ, или для них ограничивается время работы с ПЭВМ (не более 3-х часов за рабочую смену).

16. В случаях травмирования или недомогания необходимо прекратить работу, известить об этом руководителя работ и обратиться в медицинское учреждение.

17. За невыполнение данной инструкции виновные привлекаются к ответственности согласно законодательства Российской Федерации.

Работнику при работе на ПК запрещается:

– прикасаться к задней панели системного блока (процессора) при включенном питании;

– переключать разъемы интерфейсных кабелей периферийных устройств при включенном питании;

– допускать попадание влаги на поверхность системного блока (процессора), монитора, рабочую поверхность клавиатуры, дисководов, принтеров и других устройств;

– производить самостоятельное вскрытие и ремонт оборудования;

– работать на компьютере при снятых кожухах;

– отключать оборудование от электросети и выдергивать электровилку, держась за шнур.

ЗАКЛЮЧЕНИЕ

На сегодняшний день существует большое количество различных технологий мониторинга компьютерных систем, использующих различные сетевые протоколы. Протокол ICMP представляет собой наиболее простой сетевой протокол, применяющийся для базового мониторинга сети и выявления проблем.

ПО Zabbix, представляет собой свободную систему мониторинга разнообразных сервисов компьютерной сети и сетевого оборудования. Несмотря на большое количество достоинств данная система обладает и рядом недостатков таких как: сложность в установке и внедрении, требовательность к ресурсам, невозможность обеспечения отказоустойчивости. Spiceworks Network Monitor - бесплатный инструмент, предназначенный для мониторинга и статистики в реальном времени для серверов и сетевых устройств, поддерживающих SNMP. Хотя он бесплатный, он не является программным продуктом с открытым исходным кодом, так же в правом верхнем углу присутствует реклама. Сетевой монитор Spiceworks можно использовать вместе с инструментами технической поддержки Spiceworks и инструментами управления ресурсами, но я рассматриваю его как отдельный программный продукт. Eltex.EMS это централизованная система управления сетевым оборудованием производства ООО «Предприятие «ЭЛТЕКС». К недостаткам можно отнести ресурсоемкость, достаточно высокую стоимость и закрытость системы.

Общепринятой считается трехуровневая иерархическая модель ЛВС. Достоинства подобной структуры в том, что трафик пользователей с множества коммутаторов уровня доступа агрегируется на родительском узле распределения, маршрутизируется или коммутируется по необходимости на вышестоящее ядро, на соседний узел распределения или непосредственно между самими пользователями с разных узлов доступа. А каждое ядро

маршрутизирует или коммутирует трафик между несколькими узлами распределения, которые непосредственно включены в него, или между соседними ядрами.

ЛВС ФГБОУ «МДЦ «Артек» представляет собой сложную распределенную сеть выстроенною в соответствии с общепринятыми нормами и правилами построения сетей. Конкретно в локальной вычислительной сети ФГБОУ МДЦ Артек используются коммутаторы следующих производителей: Huawei, Eltex, HP, Q-tech, Mikrotik, Ubiquiti, Netgear, D-link.

Комплексный мониторинг коммутаторов уровня доступа представляет собой нетривиальную техническую задачу ввиду гетерогенности используемого сетевого оборудования. Единственным протоколом, гарантированно поддерживающимся коммутаторами различных производителей, является ICMP.

Выбранная в качестве платформы для разработки JVM позволяет запускать серверную часть системы мониторинга на любой операционной системе.

Выбор в качестве базы данных SQLite позволил сэкономить ресурсы операционной системы, так как для ее работы не нужно устанавливать систему управления базами данных.

Благодаря клиент-серверной архитектуре код проекта клиентской и серверной части разделен. Использование клиент-серверной архитектуры позволяет снизить нагрузку на сеть, так как эхо запросы отправляются с одного ip –адреса сервера.

Реализованный веб-интерфейс позволяет производить запуск системы мониторинга на любом устройстве с веб-браузером без необходимости установки.

Таким образом разработанная система мониторинга ЛВС ФГБОУ «МДЦ «Артек» реализует следующий функционал:

– Мониторинг ЛВС в реальном времени;

- Хранение результатов мониторинга в базе данных;
- Информирование администраторов ЛВС в случае сбоя важных узлов сети;
- Удаленный доступ к консоли коммутаторов по протоколу SSH
- Предоставление графического интерфейса для наглядного наблюдения за состоянием ЛВС
- Графическое представление результатов мониторинга сети за определенный период;

В результате дипломного проектирования получены и практически реализованы следующие результаты исследования:

1) Предложен мониторинг ЛВС в реальном времени с целью обнаружения сбоев сетевого оборудования за счет обработки результатов эхо-запросов по протоколу ICMP.
2) Предложено управление устройствами сети с целью ликвидации неисправностей за счет удаленного доступа по протоколу SSH.
3) Разработано графическое представление состояния узлов сети в дискретные промежутки времени для отслеживания периодичности сбоев за счет организации хранения результатов мониторинга в БД SQLite.

ЛИТЕРАТУРА

1. Lonvick C. Request for Comments 3164. The BSD Syslog Protocol / Cisco Systems. San Jose, Calif., 2001.
2. Rose M. Request for Comments 3195 Reliable Delivery for syslog / Cisco Systems. San Jose, Calif., 2001.
3. Бойченко Е.В. Кальфа В. Овчинников В.В. Локальные вычислительные сети / Бойченко Е.В. Кальфа В. Овчинников В.В. - М.: Радио и связь 2000. - 500 с.
4. Бройдо В. Л. Вычислительные системы, сети и телекоммуникации: Учебник для вузов / В.Л. Бройдо. - Спб.: Питер, 2003. - 688 с.
5. Гусева А.И. Работа в локальных сетях: Учебник / А. И. Гусева. - М.: Диалог - МИФИ, 2001. - 344 с.
6. Камалян А.К., Кулев С.А., Назаренко К.Н. и др. Компьютерные сети и средства защиты информации: Учебное пособие /Камалян А.К., Кулев С.А., Назаренко К.Н. и др. - Воронеж: ВГАУ, 2003.-119с.
7. Курносов А.П. Практикум по информатике/Под ред. Курносова А.П. Воронеж: ВГАУ, 2001.- 173 с.
8. Малышев Р.А. Локальные вычислительные сети: Учебное пособие/ РГАТА. - Рыбинск, 2005. - 83 с.
9. Новиков Ю. В. Локальные сети: архитектура, алгоритмы, проектирование. / Ю. В. Новиков. - М.: ЭКОМ, 2000. - 312 с.
10. Новиков Ю. В. Основы локальных сетей / Ю. В. Новиков. - М.: ЭКОМ, 2005. - 360 с.
11. Олифер В.Г, Олифер Н.А. Сетевые операционные системы/ В.Г. Олифер, Н.А. Олифер. - СПб.: Питер, 2002. - 544 с.: ил.
12. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы /В.Г. Олифер, Н.А. Олифер. - СПб.: Питер, 2002. - 672 с.
13. Терентьев А. М. Задачи полноценного аудита корпоративных сетей // Концепции. 2003. N° 1(11). С. 94-95.

14. Терентьев А. М. Методы и средства наблюдения загрузки локальных вычислительных сетей на примере ЦЭМИ РАН: препр. / Центр. экон.-мат. ин-т Рос. акад. наук. М., 2001. № #WP/2001/110.
15. Флинт Д. Локальные сети ПК: принципы построения, реализация / Д. Флинт. - М.: Финансы и статистика, 2001. - 359 с.
16. Фридман А.Л. Основы объектно-ориентированной разработки программных систем. / А. Л. Фридман. - М.: Финансы и статистика, 2000. - 192 с.
17. Шафрин Ю.А. Основы компьютерной технологии / Ю.А. Шафрин. - М.: АБФ, 2001. - 560 с.
18. Яковлев В.А. Компьютерные сети / В.А. Яковлев. - М.: ИНФРА-М. 2001. - 244 с.

ПРИЛОЖЕНИЯ

Приложение А

Листинг модуля мониторинга (серверная часть)

```
(ns map-artek-server.core
  (:require [compojure.route :refer [files not-found resources]]
            [compojure.handler :refer [site]]
            [compojure.core  :refer  [defroutes  GET  POST  DELETE  ANY
context]]
            [org.httpkit.timer :refer [schedule-task with-timeout]]
            [clojure.data.json :refer [read-json json-str]]
            [overtone.at-at :refer [every mk-pool]]
            [org.httpkit.server :refer :all]
            [map-artek-server.ping :refer [ping]]
            [map-artek-server.views :refer :all]
            [map-artek-server.database :refer [push-pings-db get-pings-
for-node-between]]
            [clj-time.core :as time]
            [clj-time.local :as l])
  (:use [clojure.tools.namespace.repl :only (refresh)])
  (:gen-class))

(def channel-hub (atom {}))
(def nodes (read-json (slurp "nodes.json")))
(def my-pool (mk-pool))
(defonce server (atom nil))
(def ping-history (atom []))

(defn stop-server []
  (when-not (nil? @server)
    (@server :timeout 100)
    (reset! server nil)))

(defn process_messages
  "Отправляет данные о пингах клиентам каждые n миллисекунд"
  [n]
  (every n #(let [pinged-nodes (map conj nodes (doall (pmap ping (map
:ip nodes))))]
                  ;(swap! ping-history conj {:nodes pinged-nodes :time
(l/local-now)})
              (doseq [channel (keys @channel-hub)]
```

```
                (send! channel (json-str pinged-nodes)))) my-pool))

(defn write-to-base
  "Записывает в базу пинги каждые n миллисекунд"
  [n]
  (every n #(let [pinged-nodes (map conj nodes (doall (pmap ping (map
:ip nodes))))]
              (let [nds (mapv (fn [node]
                                (select-keys node [:ping :name]))
pinged-nodes)]
                (push-pings-db (map conj nds (repeat (count nds)
{:time (l/local-now)})))))) my-pool))

(defn ws-handler [request]
  (with-channel request channel
    (swap! channel-hub assoc channel request)
    (on-close channel (fn [status]
                        (swap! channel-hub dissoc channel)))))

(defn db-handler [request]
  (with-channel request channel
    (on-receive channel (fn [message]
                          (let [j-message (read-json message)]
                            (case (:command j-message)
                              "hello" (println (:text j-message))
                              "get-pings-for-node"    (send!    channel
(json-str (get-pings-for-node-between

(:node-name j-message)
(:time1 j-message)
(:time2 j-message))))
                              ))))))

(defroutes all-routes
  (GET "/db" [] #'db-handler)
  (GET "/ws" [] #'ws-handler)
  (GET "/" [] #'index)
  (GET "/nodes/:node-name" [node-name] #'node-stats)
  (GET "/:node-name" [node-name] #'node-from-db)

  (resources "/")
  (not-found "<p>Page not found.</p>"))
```

```
(defn -main
  [& args]
   (process_messages 2000)
   (write-to-base 60000)
   (reset! server (run-server (site #'all-routes) {:port 80}))
   (println "Server started on 127.0.0.1:80"))
```

Приложение Б

Листинг JavaScript модуля (клиентская часть)

```
var server_ip = document.getElementById("server-ip").className
var ws = new WebSocket("ws://" + server_ip + ":80/ws");
ctx = document.getElementById("canvas").getContext("2d");
ctx.font = "italic 10pt Arial";
map = new Image();
map.src = "images/map.png";
red = new Image();
red.src = "images/red.png";
green = new Image();
green.src = "images/green.png";
var nodes = {};

map.onload = function() {
     ctx.drawImage(map,0,0,1980,1020);
}

canvas.onclick = function(event) {
     let X1 = event.layerX - 10;
     let Y1 = event.layerY - 10;
     let X2 = event.layerX + 10;
     let Y2 = event.layerY + 10;

     console.log(event);

     nodes.forEach( (n) => {
          if( (n.x >= X1) && (n.y >= Y1) && (n.x <= X2) && (n.y <= Y2))
{
               //alert(n.name   +   "\nПинг:   "   +   (n.result   ?
n.ping.toFixed(2) + "мс":"    :\'(") + "\nIP: " + n.ip )
               if  (event.shiftKey)  window.document.location.href  =
window.location.href +  n.name;
      else  window.document.location.href  =  window.location.href  +
"nodes/" + n.name;
```

```
            }
      });
}

ws.onopen = function (event) {
    ws.onclose = function (event) {
    alert("Сервер прекратил отправлять данные");
  }
}

ws.onmessage = function (event) {
      ctx.drawImage(map,0,0,1980,1020);
      nodes = JSON.parse(event.data);
      for(node in nodes) {
            if (nodes[node].result == true){
                  ctx.drawImage(green, nodes[node].x, nodes[node].y, 10,
10);
                  ctx.fillStyle = "green";
                  ctx.fillText(nodes[node].name,        nodes[node].x-10,
nodes[node].y-2);
            } else {
                  ctx.drawImage(red,nodes[node].x,   nodes[node].y,   10,
10);
                  ctx.fillStyle = "red";
                  ctx.fillText(nodes[node].name,        nodes[node].x-10,
nodes[node].y-2);
            }
      }
}
```