2021

SMARTS-GENESIS

ПРОГРАММНАЯ СИСТЕМА ИНТЕГРИРОВАННОГО УПРАВЛЕНИЯ (СИУ)

ОПИСАНИЕ ПРОГРАММЫ

ПСУ

АО «СМАРТС»

genesis@smarts.ru
+7 (846) 212-99-70
г. Самара, ул. Дачная, д.2, корп.2

СМАРТС

Оглавление

Оглавление 1

1. Общие сведения 2

1.1. Обозначение и наименование программы 2

1.2. Программное обеспечение, необходимое для функционирования программы 2

1.3. Языки программирования, на которых написана программа 2

2. Функциональное назначение 3

2.1. Классы решаемых задач 3

2.2. Назначение программы 3

2.3. Сведения о функциональных ограничениях на применение 3

3. Описание логической структуры 4

3.1. Используемые методы 5

3.2. Структура программы с описанием функций составных частей и связи между ними 5

3.2.1. Программный модуль обеспечения защиты виртуальных линий передачи данных (МЗВС) 5

3.2.2. Программный модуль rest_forward.py для ПрКт ПКС 6

3.3. Связи программы с другими программами 7

4. Используемые технические средства 8

5. Вызов и загрузка 8

6. Входные данные 8

7. Выходные данные 11

1. Общие сведения

1.1. Обозначение и наименование программы

Программа «Программные средства управления каналами связи (передачи данных) между компонентами МГРЦ (ПСУ)» имеет следующие атрибуты:

- Наименование исполняемого файла - rest_forward.py;
- Размер исполняемого файла - 22237 байт;
- Версия файла - 5.1;
- Версия продукта - 5.1;
- Внутреннее имя - rest_forward;
- Исходное имя файла - rest_forward.py;
- Название продукта - rest_forward;
- Описание версии файла - Module for Controller Release.

1.2. Программное обеспечение, необходимое для функционирования программы

Системные программные средства, используемые программой ПСУ:

- ОС NauLinux 7.4;
- SDN контроллер RYU 4.15;
- SDN коммутатор Open vSwitch 2.8.2 с поддержкой OpenFlow 1.3;
- OpenSSL 1.0.2k.

1.3. Языки программирования, на которых написана программа

Исходным языком данной разработки является Python. Среда разработки – текстовый редактор vi, интерпретатор и среда выполнения – Python 2.7.

2. Функциональное назначение

2.1. Классы решаемых задач

ПСУ предназначен для обеспечения защиты виртуальных линий передачи данных (МЗВС) и создания виртуальных каналов обмена данными между компонентами МГРЦ в соответствии с требованиями по качеству обслуживания (ТКО).

В соответствии с ТКО ПСУ формирует таблицы маршрутизации сетевых пакетов и по протоколу OpenFlow версии 1.3 посылает команды управления аппаратным сетевым коммутаторам ПКС для создания виртуальных каналов передачи данных между ЦОД МГРЦ.

Создание вариантов и изменение топологии виртуальных каналов достигается за счет программного модуля ПрКт ПКС, позволяющего переключать каналы в соответствии с ТКО в динамическом режиме.

МЗВС предоставляет интерфейс и кодирует данные, поступающие на данный интерфейс с целью их дальнейшей пересылки и декодирования, организуя соединение уровня L2 сетевой модели OSI.

2.2. Назначение программы

Основное назначение программы «ПСУ» обеспечение защиты виртуальных линий передачи данных (МЗВС) и создания виртуальных каналов обмена данными между компонентами МГРЦ.

2.3. Сведения о функциональных ограничениях на применение

Программа «ПСУ» не имеет ограничений на применение.

3. Описание логической структуры

Функциональная схема ПСУ представлен на рисунке 3.1. На схеме отображены логические элементы модуля и варианты взаимодействия между ними.

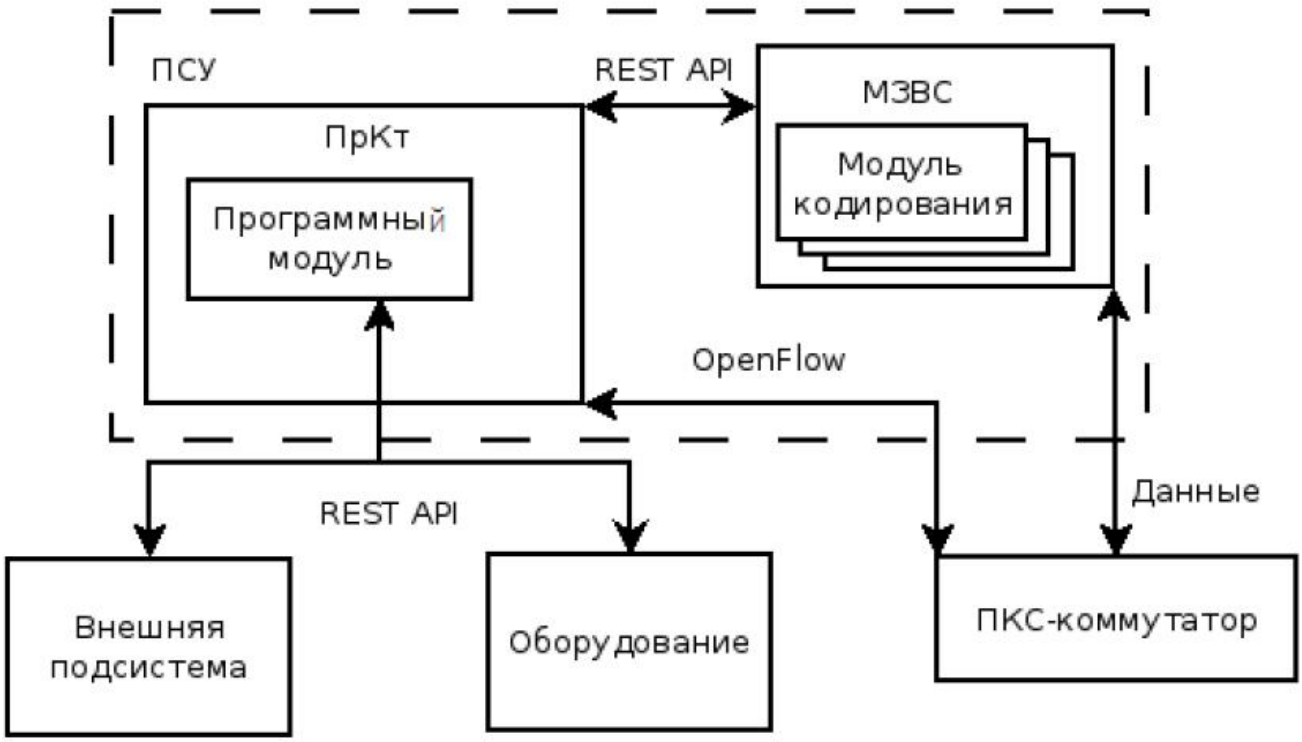

Рисунок 1. Функциональная схема ПСУ

Модуль rest_forward.py позволяет ПСУ осуществлять перенаправление потоков данных между клиентом и различными серверами без внесения изменения в конфигурацию клиента. Перенаправление осуществляется на уровне L2 модели OSI с возможностью указания VLAN.

Запросы между ПСУ и КСУВ осуществляются по протоколу HTTP/HTTPS по спецификации RESTful API. Согласно запросу, REST API модуль может производить следующие действия: восстановление прежних связей, связь между подсетями с учетом VLAN, переключение между каналами в соответствии с требованиями по качеству обслуживания (ТКО), с учетом данных о состоянии сетевых коммутаторов ПК и квантовых и не квантовых каналов. Переключение осуществляется между каналами с кодированием и без кодирования. ПрКт ПКС формирует запрос к МЗВС для кодирования канала. Сформированные команды управления исполняются ПрКт ПКС и отображается результат выполнения этих команд.

3.1. Используемые методы

В реализованном алгоритме выполняется преобразование поступающих данных формата JSON в правила сетевого коммутатора протокола OpenFlow. При этом используются методы поиска информации о состоянии системы и имеющихся сущностях с целью выявления принадлежности конфигурируемых в запросах объектов запущенному инстансу ПСУ и классификации приходящих запросов по признакам типа связываемых услуг и принадлежности его к тому или иному порту коммутатора.

3.2. Структура программы с описанием функций составных частей и связи между ними

В состав ПСУ входят следующие подсистемы:

- Программный модуль обеспечения защиты виртуальных линий передачи данных (МЗВС);
- Программный контроллер – ПрКт программно-конфигурируемой сети (ПКС).

МЗВС состоит из модуля хранения ключей с обменом данными по протоколу TCP/IP и модуля создания интерфейсов для кодирования и декодирования данных, поступающих в виртуальный канал передачи.

ПрКт ПКС состоит из базовой платформы контроллера и программного модуля создания виртуальных каналов.

3.2.1. Программный модуль обеспечения защиты виртуальных линий передачи данных (МЗВС)

Программный модуль keyworker.c осуществляет прием ключей от генератора случайных битовых последовательностей, вычисление их хэш-суммы, добавление ключей в базу данных, доступ к ключам по запросу через RESTful API по хэшу, либо по запросу на выдачу нового ключа, имеет внутренние счетчики доступа к ключам с целью выдачи актуальных ключей.

Обмен данными между ПрКт ПКС и модулем keyworker.c осуществляется по протоколу HTTP/HTTPS по спецификации RESTful API: по методу POST / выполняется загрузка нового ключа.

Модуль выводит отладочную информацию в стандартный вывод и имеет ряд конфигурационных параметров.

Программный модуль ctapudp.c осуществляет создание интерфейсов для кодирования и декодирования данных, поступающих в виртуальный канал передачи. Данный модуль выполняет кодирование приходящих Ethernet-фреймов в режиме AES-256-CBC с использованием программных средств библиотеки OpenSSL и их последующую упаковку в UDP-пакеты с пересылкой на модуль декодирования, реализуя при этом L2-туннель. При этом информация о ключах поступает из модуля keyworker.c, для модуля, выполняющего кодирование производится извлечение нового ключа из модуля keyworker.c по запросу POST / с payload “last”. Модуль, выполняющий декодирование, вычисляет хэш ключа и обращается к keyworker.c для получения ключа по хэшу с запросом POST / с payload “keyХЭШ”.

Также, модуль ctapudp.c осуществляет сжатие данных с использованием программных средств библиотеки zlib, и предполагает возможность задействования 2-х каналов передачи данных для отправки UDP-пакетов. Модули МЗВС с подобными преобразованиями пакетов могут быть использованы для передачи данных путем добавления каналов в модуль ПрКт.

Модуль выводит отладочную информацию в стандартный вывод и имеет ряд конфигурационных параметров.

3.2.2. Программный модуль rest_forward.py для ПрКт ПКС

Программный модуль rest_forward.py осуществляет связь между подсетями с учетом VLAN, переключение между каналами в соответствии с требованиями по качеству обслуживания (ТКО).

Команды и данные между ПСУ и КСУВ осуществляются по протоколу HTTP/HTTPS по спецификации RESTful API и посылаются к ПрКт ПКС для осуществления следующих функций:

- По методу GET /activate восстанавливается текущее состояние каналов;

- По методу GET /channel/{channel}/{sla} происходит переключение канала {channel} на ТКО {sla};
- По методу DELETE /channel/{channel} удаляется канал {channel};
- По методу POST /channel загружаются данные о канале в формате JSON;
- По методу POST /ovs загружаются конфигурации о сетевом коммутаторе ПКС в формате JSON;
- По методу POST /host загружаются конфигурация хоста ПСУ.

ПрКт ПКС по методу POST /qkey/{ovs}/{sla} формирует запрос к МЗВС при этом используется информация о сетевом коммутаторе ПКС {ovs} и канале с заданным ТКО {sla}.

ПрКт ПКС по методу GET /qchannel/{ovs}/{sla} происходит переключение всех каналов заданного коммутатора {ovs} на ТКО {sla}.

Сформированные команды передаются от ПрКт к сетевым коммутаторам ПКС по протоколу OpenFlow версии 1.3. При этом в качестве команд выступают в том числе таблицы маршрутизации сетевых пакетов.

Создание вариантов топологии виртуальных каналов подразумевает наличие совокупности правил, что обеспечивается массивом структур, передаваемых по методу POST /channel при этом топология. Динамическое изменение топологий возможно по запросу POST /channel путем подгрузки изменений в структуру топологии. При этом возможно использование двух независимых параллельных каналов путем задания варианта топологии виртуальных каналов с включением в них МЗВС, сконфигурированных на использование двух каналов передачи данных для отправки UDP-пакетов (в которых в свою очередь пересылаются Ethernet-фреймы).

Политики использования виртуальных каналов передачи задаются в вариантах обеспечения требуемого качестве обслуживания посредством перенаправления трафика, либо, путём выключения прохождения трафика по каналу.

3.3. Связи программы с другими программами

ПрКт ПКС представляет собой модульную систему на базе свободно-распространяемого ПКС контроллера Ryu. Специализированный модуль обеспечивает требуемую другим подсистемам функциональность и доступен через

REST API. Запросы между ПСУ и КСУВ осуществляются по протоколу HTTP/HTTPS по спецификации RESTful API. Согласно запросу, REST API модуль может производить следующие действия: восстановление прежних связей, связь между подсетями с учетом VLAN, переключение между каналами в соответствии с требованиями по качеству обслуживания (ТКО). Переключение осуществляется между каналами с кодированием и без кодирования, либо с комбинацией различных типов кодирования.

4. Используемые технические средства

В состав используемых технических средств входит:

- Виртуальная машина 1: не менее 1 vCPU, 2GB RAM, 20 GB HDD;
- Виртуальная машина 2: не менее 1 vCPU, 2GB RAM, 20 GB HDD;
- Устройства ввода информации;
- Наличие доступа к удаленному узлу.

5. Вызов и загрузка

Загрузка и запуск программы «ПСУ» осуществляется путем запуска сервисов (при старте системы они стартуют автоматически):

sudo systemctl start psu psu-agent

6. Входные данные

В качестве входных данных ПСУ используются данные в формате JSON, поступающие по REST API на ПрКт ПКС и содержащие информацию о сетевом окружении объединяемых проектов пользователей: физические интерфейсы коммутаторов, VLAN сегменты, подсети. Также входными данными является конфигурация виртуального канала между объединяемыми проектами.

Формат входной информации приведен в таблицах 6.1 - 6.3.

Таблица 6.1. Добавление хоста ПСУ (Выполняется при развертывании)

Формат запроса	Описание формата
{	
"1": {	Номер ЦОД
"address": "http://172.17.20.41:1001"	Адрес для доступа к ПСУ с других ЦОД
}	
}	

Таблица 6.2. Добавление коммутатора с поддержкой OpenFlow 1.3 (Выполняется при развертывании)

Формат запроса	Описание формата
{	
"3": {	Идентификатор коммутатора в БД Идентификатор коммутатора OpenFlowВходные порты
"dpid": "000016ccbbe0d642",	
"intport": {	
"1": {	Идентификатор порта коммутатораТип услуг, подключенных к порту
"type": "ceph"	
}, "2": {	
"type": "vm"	Идентификатор порта коммутатораТип услуг, подключенных к порту
}	
},	
"outport": {	
"3": {	
"sla": "1",	Выходные порты
"desc": "normal"	Идентификатор порта коммутатора Вид ТКО, обеспечиваемых каналом МЗВС за данныпортом
},	
"4": {	
"sla": "2",	
"desc": "quantum",	Идентификатор порта коммутатора Вид ТКО, обеспечиваемых каналом МЗВС за данныпортом
"mzvs": "1"	
}	Идентификатор МЗВС, на который должны поступатьключи
},	
"mzvs": {	
"1": ["http://127.0.0.1:77	Адреса хранилищ ключей МЗВС
", "http://127.0.0.1:55554"]	
}	
}	

}	

Таблица 6.3. Добавление виртуальных каналов (выполняется КСУВ)

Формат запроса	Описание формата

{	
"1": {	Идентификатор канала в БД
"src": {	Информация о первом
"ovs": 1,	модуле канала
"type":	Идентификатор коммутатора
"vm",	в БД
"vlan": 12,	Тип услуг, которые должны быть
"subnet": "172.16.1.0/24"	доступны VLAN подсети, которая
},	должна быть доступнаПодсеть,
"dst": {	которая должна быть доступна
"ovs": 2,	
"type":	Информация о втором
"ceph",	модуле канала
"vlan": 11,	Идентификатор коммутатора
"subnet": "172.16.2.0/24"	в БД
},	Тип услуг, которые должны быть
"sla": "1",	доступны VLAN подсети, которая
"active": "1",	должна быть доступнаПодсеть,
"qos": 10	которая должна быть доступна
}	
}	ТКО устанавливаемого
	соединения необходимо ли
	сделать канал активным
	Ограничение пропускной способности МБ/с

7. Выходные данные

Выходными данными являются данные, передаваемые по протоколу HTTP по методам GET, DELETE и POST согласно спецификации REST API. В качестве формата представления данных используется формат JSON. Выходные данные содержат информацию о статусе выполнения команды вида:

{"request": "chan activate", "channel_id": 2, "mode": 0}