Санкт-Петербургский государственный университет

Кафедра системного программирования

Улучшение сетевых возможностей ОСРВ Embox

Тагильцев Семен Вадимович

10.06.2024

Научный руководитель: к.ф.-м.н. Д.В. Луцив, доцент кафедры системного программирования

Рецензент: К.К. Смирнов, старший преподаватель кафедры ИАС

Консультант: А.В. Бондарев, генеральный директор ООО «Ембокс»

Санкт-Петербург
2024

Введение

- Сетевые технологии являются основой современной жизни
- Постоянно растущие объемы передаваемой информации влекут за собой появление сетевых технологий, в частности, SDN[1]
- Embox[2] – отечественная ОС, долгое время разрабатывалась выпускниками СП
  - Кроссплатформенная (arm, x86, risc-v, e2k)
  - Обладает высоким уровнем безопасности, может использоваться в критической информационной инфраструктуре (КИИ)
  - Используется на сетевых устройств

[1] SDN https://en.wikipedia.org/wiki/Software-defined_networking
[2] Embox http://emboxing.ru/

Постановка задачи

Цель: Реализовать базовую поддержку программно-определяемых сетей (Software-defined networking, SDN) в ОС Embox

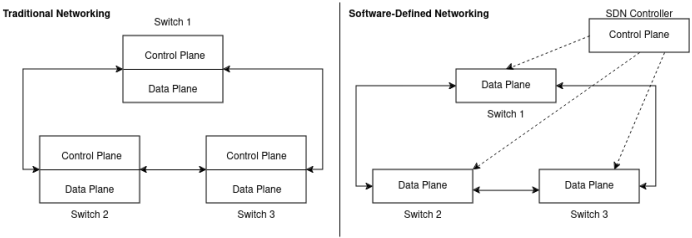
Задачи:

1. Выполнить обзор текущих сетевых возможностей ОС Embox, современных подходов к построению сетевой инфраструктуры и выделить основную функциональность для SDN
2. Спроектировать решение, обеспечивающее базовую поддержку SDN в ОС Embox
3. Улучшить текущие сетевые возможности ОС Embox, реализовав спроектированное решение
4. Выполнить тестирование реализованного решения

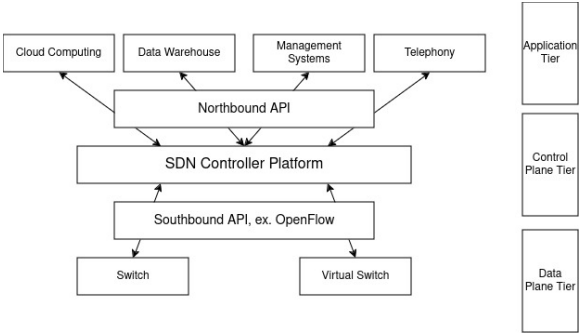
Традиционная сеть

- Ограничения в гибкости и масштабируемости
- Сложность управления традиционными сетями
- Проблема безопасности

Выход из ситуации – Software-Defined Networking (SDN)



Software-Defined Networking (SDN)



С точки зрения ОС Embox интересны управляющий уровень (control plane) и передающий уровень (data plane)

P4

P4 (Programming Protocol-Independent Packet Processors)[3] – это язык программирования, созданный для описания протоколов обработки пакетов на сетевых устройствах, таких как коммутаторы и маршрутизаторы

Основные компоненты:

- p4c – модульный компилятор для языка программирования P4
- PI (Protocol Independent) – содержит реализацию абстрактного интерфейса для взаимодействия между P4-программируемыми сетевыми устройствами и SDN-контроллерам
- behavioral-model (bmv2) – включает в себя несколько реализаций программных коммутаторов

[3] P4: https://p4.org/

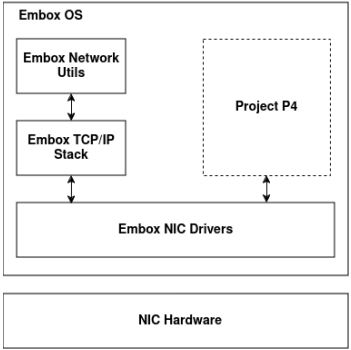
Сетевые возможности Embox

- Архитектура сетевой подсистемы ОС Embox похожа на аналогичную в ОС Linux
- Драйверы для различных сетевых контроллеров
- Протоколы сетевого стека
- Применялась для построения традиционных сетей

Архитектура

Решением является портирование проекта P4 в Embox

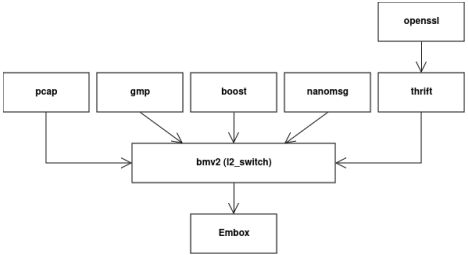
- Появится представление работоспособного коммутатора на основе ОС Embox
- Даст возможность создавать масштабируемые, безопасные, управляемые сети, используя Embox



Реализация

Портирован проект P4

- Выявлены зависимости у коммутатора l2_switch



- Портированы библиотеки libpcap, libgmp, thrift, openssl
- Выполнены необходимые доработки в ОС Embox
- Портирован программный коммутатор p4/l2_switch

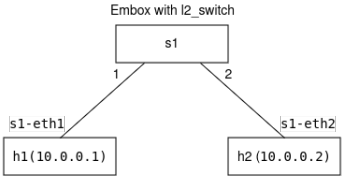
Mybuild и Makefile файлы для p4

```
package third_party.cmd

@App
@AutoCmd
@BuildDepends(embox.lib.libsupcxx)
@BuildDepends(embox.lib.libstdcxx)
@BuildDepends(embox.lib.libgcc)
@BuildDepends(third_party.boost.libs)
@BuildDepends(third_party.lib.libpcap)
@BuildDepends(third_party.lib.libgmp)
@BuildDepends(third_party.lib.openssl)
@BuildDepends(third_party.lib.thrift)
@Build(stage=2,script="$(EXTERNAL_MAKE)")
@Cmd(name = "l2_switch")
module l2_switch {
    @AddPrefix("^BUILD/extbld/^MOD_PATH/install")
    source "l2_switch.o"


    depends embox.compat.posix.util.gettimeofday
    depends embox.compat.atomic.pseudo_atomic
    depends embox.compat.libc.math
    @NoRuntime depends embox.lib.libsupcxx
    @NoRuntime depends embox.lib.libstdcxx
    @NoRuntime depends embox.lib.libgcc
    @NoRuntime depends third_party.boost.libs
    @NoRuntime depends third_party.lib.libpcap
    @NoRuntime depends third_party.lib.libgmp
    @NoRuntime depends third_party.lib.openssl
    @NoRuntime depends third_party.lib.thrift
    @NoRuntime depends embox.compat.posix.pthread_rwlock
    @NoRuntime depends embox.compat.posix.sys.mman.mmap_api
    @NoRuntime depends embox.compat.posix.util.nanosleep
    @NoRuntime depends embox.compat.libc.all
    @NoRuntime depends embox.compat.posix.pthread_key
}
```

```
PKG_VER := 1.15.0

PKG_SOURCES := https://github.com/p4lang/$(PKG_NAME)/archive/refs/tags/$(PKG_VER).tar.gz

PKG_MD5     := dc9d34e33daf40e54d9d71a69aece2dc

PKG_PATCHES := only_l2_switch.patch
PKG_PATCHES += disable_lib_check.patch
PKG_PATCHES += fix.patch

include $(EXTBLD_LIB)

$(CONFIGURE) :
    export EMBOX_GCC_LINK=full; \
    cd $(PKG_SOURCE_DIR) && ./autogen.sh && ( \
        ./configure --prefix=$(PKG_INSTALL_DIR)  \
            CC=$(EMBOX_GCC) \
            CFLAGS="$(BM_CXXFLAGS)" \
            CXX=$(EMBOX_GXX) \
            CXXFLAGS="$(BM_CXXFLAGS)" \
            LDFLAGS=$(BM_LDFLAGS) \
            --host=$(AUTOCONF_TARGET_TRIPLET) \
            --with-thrift=$(ROOT_DIR)/build/extbld/third_party/lib/thrift/install/include \
            --with-nanomsg=no \
            --enable-static \
            --disable-shared \
            --disable-undeterministic-tests \
            --with-stress-tests=no \
    )
    touch $@

$(BUILD) :
    cp $(ROOT_DIR)/third-party/p4/behavioral-model/patches/config.h $(PKG_SOURCE_DIR)/config.h
    cp $(ROOT_DIR)/third-party/p4/behavioral-model/patches/thrift_src/Makefile $(PKG_SOURCE_DIR)/thrift_src/Makefile
    cd $(PKG_SOURCE_DIR) && ( \
        $(MAKE) MAKEFLAGS='$(EMBOX_IMPORTED_MAKEFLAGS)'; \
    )
    touch $@

$(INSTALL) :
    cp $(PKG_SOURCE_DIR)/targets/l2_switch/l2_switch $(PKG_INSTALL_DIR)/l2_switch.o
    touch $@
```

Тестирование

- Создана конфигурация P4 на базе Linux
- Embox
  - Поднята виртуальная сеть с двумя хостами
  - В сеть добавлен коммутатор на базе Embox
  - Проверена работоспособность сети, успешно выполнен базовый ping между хостами



Результаты

- Произведен обзор текущих сетевых возможностей ОС Embox и современных подходов к построению сетевой инфраструктуры. Выбран подход, основанный на создании программно-конфигурируемых сетей (SDN)
- Спроектировано решение для поддержки программно-конфигурируемых сетей, основанное на проекте P4
- Спроектированное решение реализовано. Интегрированы необходимые компоненты проекта P4 вместе с зависимостями в Embox
- Выполнено тестирование решения на базе ОС Embox и проекта P4

Код проекта Embox доступен на github[4]. Имя пользователя: semtagg

[4] ОС Embox https://github.com/embox/embox