



OTUS C++ Basic

**Меня хорошо видно
&& слышно?**



Защита проекта

Тема: ISIS database replicator



Чурсин Александр

План защиты



Цели проекта

Что планировалось

Используемые
технологии

Что получилось

Схемы/архитектура

Выводы

Цели проекта



1. Создание приложения для решения рабочих задач по воспроизведению проблем заказчиков путем симуляции лабораторной/продуктовой среды заказчика в части протокола динамической маршрутизации ISIS (ISO/IEC 10589:2002).
2. Изучение работы существующих Open Source решений, реализующих протоколы динамической маршрутизации (FRR, Xorp)



Что планировалось

1. Идея возникла почти параллельно с проектом BNG blaster от RtBrick (стартап, развивающий Open BNG – открытая платформа для решений, обеспечивающих широкополосный доступ) <https://rtbrick.github.io/bngblaster/quickstart.html#isis>.
2. Изначально по C++ было минимум знаний по курсу в университете. Проект BNG blaster реализован на C и использует “негибкие” форматы данных(mrt rfc6396, не довелось видеть на практике, использовался для BGP, развития не получил, судя по документации) .
Поэтому было принято решение делать свое приложение и на C++ и использовать json как источник данных.
3. Коммерческие продукты тестирования сетей IXIA и Spirent не поддерживают функционал импорта ISIS данных. Поддержка генерации топологий присутствует, но на практике означает, что для симуляции окружения в среде заказчика необходимо повторить его топологию вручную.

Используемые технологии



1. Для организации сетевого взаимодействия в проекте активно используется библиотека Boost::Asio https://www.boost.org/doc/libs/1_75_0/doc/html/boost_asio.html
2. Центральная структура управления программой реализована с использованием одно заголовочной библиотеки TinyFSM <https://github.com/digint/tinyfsm>
3. STL
4. JSON парсинг реализован с использованием библиотеки Нильса Лохманна <https://github.com/nlohmann/json>



Что получилось



1. Удалось реализовать полностью архитектуру с отладкой обмена основными структурами данных.
2. По времени проект занял примерно месяц.
3. Во время реализации проекта удалось разобрать существующие продукты по крайней мере в части необходимой для собственной реализации.
4. На данном этапе проект завершен на этапе Proof of Concept, необходимый функционал работает, но для финального применения необходимо реализовать поддержку всех TLV протокола ISIS, это потребует значительного времени, большая часть из которого – написание кода структур, то есть более рутинная часть нежели выработка решений по архитектуре.

Что получилось



```
RP/0/RP0/CPU0:dut#show isis database
```

```
Sun Nov 20 13:20:42.466 UTC
```

```
IS-IS 1 (Level-1) Link State Database
```

LSPID	LSP Seq Num	LSP Checksum	LSP Holdtime/Rcvd	ATT/P/OL
dut.00-00	* 0x00000001	0xc1bb	65533/*	0/0/0

```
Total Level-1 LSP count: 1    Local Level-1 LSP count: 1
```

Только свой PDU

```
IS-IS 1 (Level-2) Link State Database
```

LSPID	LSP Seq Num	LSP Checksum	LSP Holdtime/Rcvd	ATT/P/OL
dut.00-00	* 0x00000002	0xf65d	65533/*	0/0/0

```
Total Level-2 LSP count: 1    Local Level-2 LSP count: 1
```

```
RP/0/RP0/CPU0:dut#show isis adjacency
```

```
Sun Nov 20 13:20:51.679 UTC
```

```
IS-IS 1 Level-1 adjacencies:
```

System Id	Interface	SNPA	State	Hold	Changed	NSF	IPv4	IPv6
			BFD	BFD				

```
IS-IS 1 Level-2 adjacencies:
```

System Id	Interface	SNPA	State	Hold	Changed	NSF	IPv4	IPv6
			BFD	BFD				

Нет соседей

Test topology running ISIS level 2

Cisco XRv 9000

Juniper vMX



JSON

Cisco XRv DUT



Gi0/0/0/1

eth0



Что получилось



```
root@salt:/var/tmp# ./repl eth0 prepared.json
```

```
Found LSPs: 2
```

```
LSP-ID 0001.0001.0001.00-00
```

```
hostname: vmx1
```

```
LSP-ID 0002.0002.0002.00-00
```

```
hostname: xrv9000
```

```
ISIS Adj is : Down
```

```
ISIS Adj is : Init
```

```
ISIS Adj is : Up
```

```
flooding LSPs >> DUT
```

Установлено соседство,
отправлены данные

Test topology running ISIS level 2

Cisco XRv 9000

Juniper vMX



show isis database extensive
show isis hostname

JSON

Cisco XRv DUT



Gi0/0/0/1

eth0



ISIS P2P Adjacency

LSPs



Что получилось



```
RP/0/RP0/CPU0:dut#show isis adjacency
Sun Nov 20 13:22:58.591 UTC
```

IS-IS 1 Level-1 adjacencies:

System Id	Interface	SNMPA	State	Hold	Changed	NSF	IPv4	IPv6
			BFD	BFD				

IS-IS 1 Level-2 adjacencies:

System Id	Interface	SNMPA	State	Hold	Changed	NSF	IPv4	IPv6
			BFD	BFD				
0001.0000.0001	Gi0/0/0/1	*PtoP*	Up	29	00:00:20	No	None	None

Total adjacency count: 1

```
RP/0/RP0/CPU0:dut#show isis database
Sun Nov 20 13:23:14.457 UTC
```

IS-IS 1 (Level-1) Link State Database

LSPID	LSP Seq Num	LSP Checksum	LSP Holdtime/Rcvd	ATT/P/OL
dut.00-00	* 0x00000001	0xc1bb	65381/*	0/0/0

Total Level-1 LSP count: 1 Local Level-1 LSP count: 1
2 LSP, соответствующие тестовой топологии: xrv9000, vmx1

IS-IS 1 (Level-2) Link State Database

LSPID	LSP Seq Num	LSP Checksum	LSP Holdtime/Rcvd	ATT/P/OL
vmx1.00-00	0x00000005	0x22b7	65498/982	0/0/0
xrv9000.00-00	0x00000008	0x69c9	65498/936	0/0/0
dut.00-00	* 0x00000003	0xe7cf	65500/*	0/0/0

Total Level-2 LSP count: 3 Local Level-2 LSP count: 1

Test topology running ISIS level 2

Cisco XRv 9000

Juniper vMX



Cisco XRv DUT



Gi0/0/0/1

eth0



JSON

ISIS P2P Adjacency

LSPs



Что получилось



Router

```
RP/0/RP0/CPU0:dut(config)#int gi0/0/0/1
RP/0/RP0/CPU0:dut(config-if)#shut
RP/0/RP0/CPU0:dut(config-if)#commit
Sun Nov 20 13:26:51.424 UTC
```

Server

```
flooding LSPs >> DUT
Hold-time expired. Going Down.
ISIS Adj is : Down
```

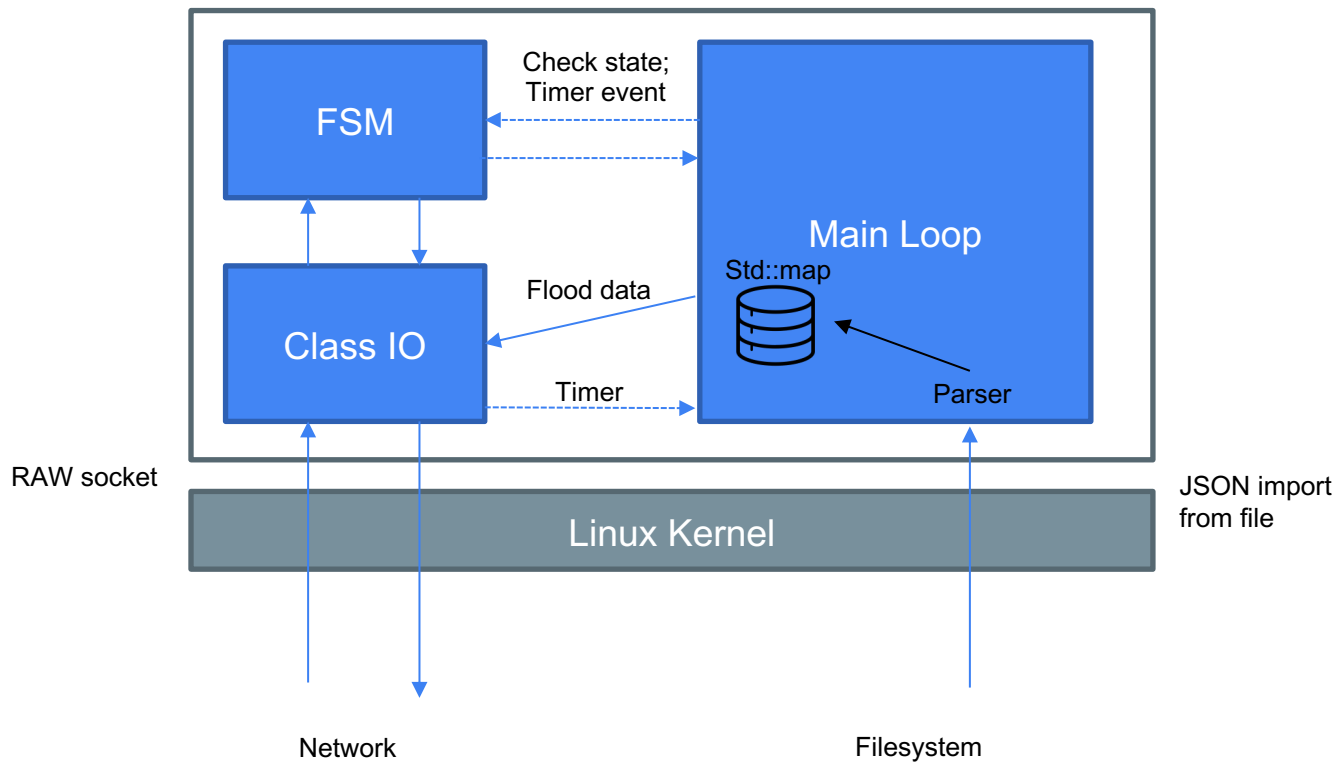
```
RP/0/RP0/CPU0:dut(config-if)#no shut
RP/0/RP0/CPU0:dut(config-if)#commit
Sun Nov 20 13:27:36.361 UTC
```

```
ISIS Adj is : Down
ISIS Adj is : Init
ISIS Adj is : Up
```

Приложение корректно отслеживает состояние тестируемого маршрутизатора



Схема архитектуры



Выводы и планы по развитию

1. Реализовать все TLV протокола ISIS.
2. При использовании istream FSM делает одно избыточное копирование пакета ввиду существующего дизайна (const & event), возможно изменение схемы парсинга без использования istream или переработка FSM, но ввиду что изначально библиотека взята как есть, потребуется время.
3. Получен опыт написания и отладки сетевого приложения на C++.

Спасибо за внимание!