

Мантуров Д.О.
Описание

Репозиторий проекта:
<https://github.com/manturovDan/GossipTTL>

14. Разработать программу GossiperWithTTL. Программа рассылает и принимает 2 типа сообщений: о том, что хочет получать слухи от других таких же программ (посылается на специальный MAC, конфигурируемый для группы) при старте программы, принимая подобные сообщения, заносит MAC источник в список сплетниц; периодически рассылает сплетни (с TTL = M) N случайным хостам из списка сплетниц, при получении подобного сообщения, если TTL > 0 пересылает N случайным хостам из списка, уменьшая TTL на 1, кроме источника источника сообщения.

Программа написана на java с использованием библиотеки pcap4j (<https://github.com/kaitoy/pcap4j>).

Код находится в пакете src/app

Программа имеет 3 потока: главный поток, GUI, Listener.

Программа была протестирована на linux. Необходимо запускать с правами суперпользователя. Так как GUI разработан на JavaFX для запуска необходимо указать путь к библиотеке. Запуск скомпилированной программы производится следующим образом: `sudo java -jar --module-path /home/kali/Desktop/javafx-sdk-11.0.2/lib --add-modules javafx.controls,javafx.fxml GossipTTL.jar`

Сразу после запуска в консоли выбирается нужный нам сетевой интерфейс, затем записывается N для каждого хоста, далее через запятую вводятся все MAC адреса, куда необходимо отправить информацию о том, что новый хост в системе.

```
NIF[0]: wlp5s0
      : link layer address: 34:e1:2d:64:54:88
      : address: /192.168.0.104
      : address: /fe80:0:0:0:c5dd:e5cc:abc4:f34e
NIF[1]: lo
      : link layer address: 00:00:00:00:00:00
      : address: /127.0.0.1
      : address: /0:0:0:0:0:0:0:1
NIF[2]: any
      : description: Pseudo-device that captures on all
interfaces
NIF[3]: enp3s0
```

```
      : link layer address: 8c:16:45:d9:3f:a3
NIF[4]: virbr0
      : link layer address: 52:54:00:78:e5:97
      : address: /192.168.122.1
NIF[5]: docker0
      : link layer address: 02:42:55:e9:fa:83
      : address: /172.17.0.1
NIF[6]: bluetooth-monitor
      : description: Bluetooth Linux Monitor
NIF[7]: nflog
      : description: Linux netfilter log (NFLOG)
interface
NIF[8]: nfqueue
      : description: Linux netfilter queue (NFQUEUE)
interface
NIF[9]: bluetooth0
      : description: Bluetooth adapter number 0
NIF[10]: virbr0-nic
      : link layer address: 52:54:00:78:e5:97
```

Select a device number to capture packets, or enter 'q'
to quit > 0

My Mac address :[34:e1:2d:64:54:88]

Insert count of resend destinations (N):

2

Insert dst MACs (throw comma):

ff:ff:ff:ff:ff:ff

reg: ff:ff:ff:ff:ff:ff

register to: ff:ff:ff:ff:ff:ff from: 34:e1:2d:64:54:88

[Ethernet Header (14 bytes)]

Destination address: ff:ff:ff:ff:ff:ff

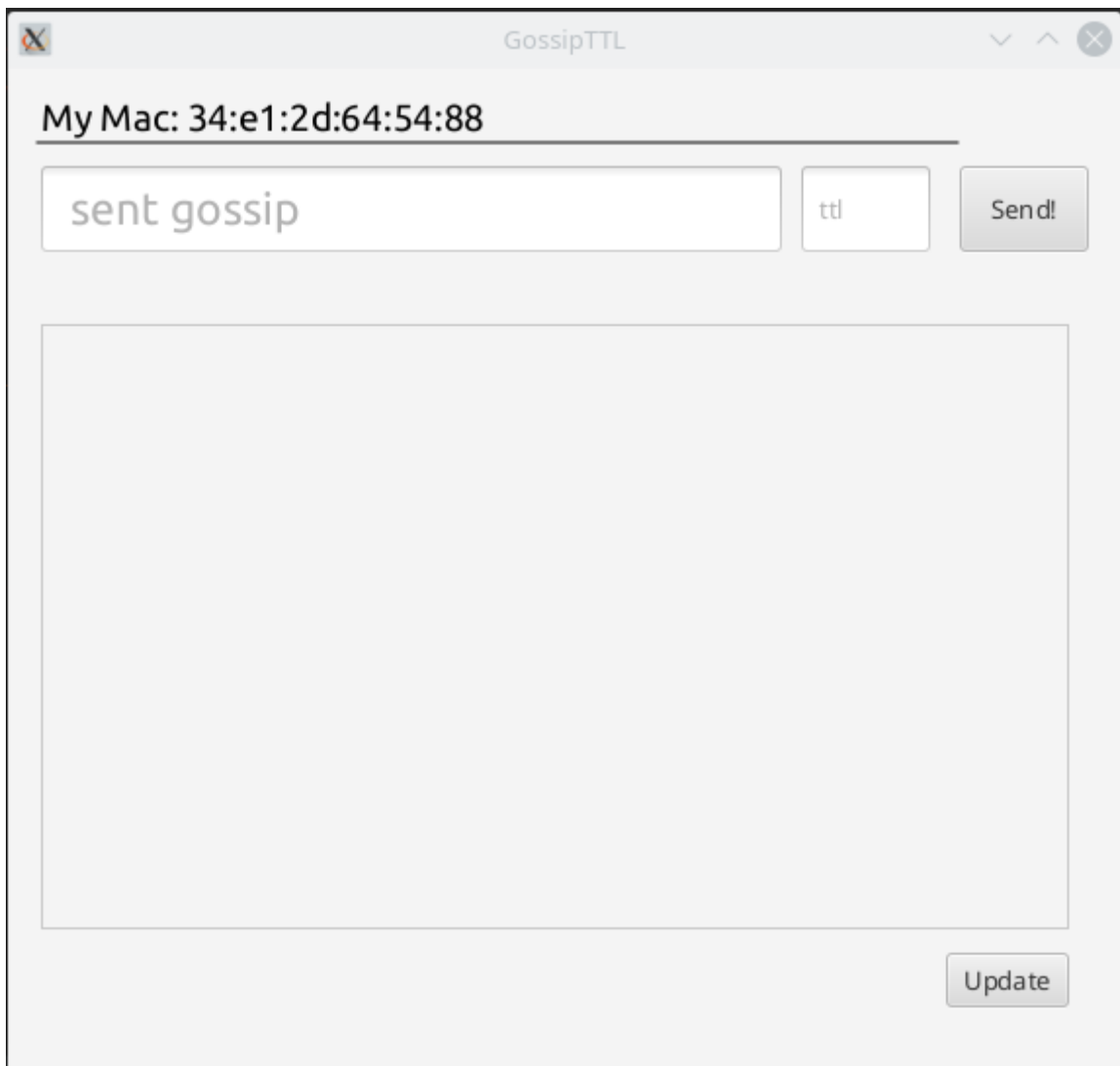
Source address: 34:e1:2d:64:54:88

Type: 0x9001 (unknown)

[Ethernet Pad (1 bytes)]

Hex stream: 00

После этого откроется окно пользовательского интерфейса.



В верхние поля ввода вводится сплетня и её время жизни.

В нижней области выводятся Mac-адреса, в которые мы отправляем (обновляются при нажатии UPDATE).

В консоль пишется всё, что происходит: получение и отправка сообщений.

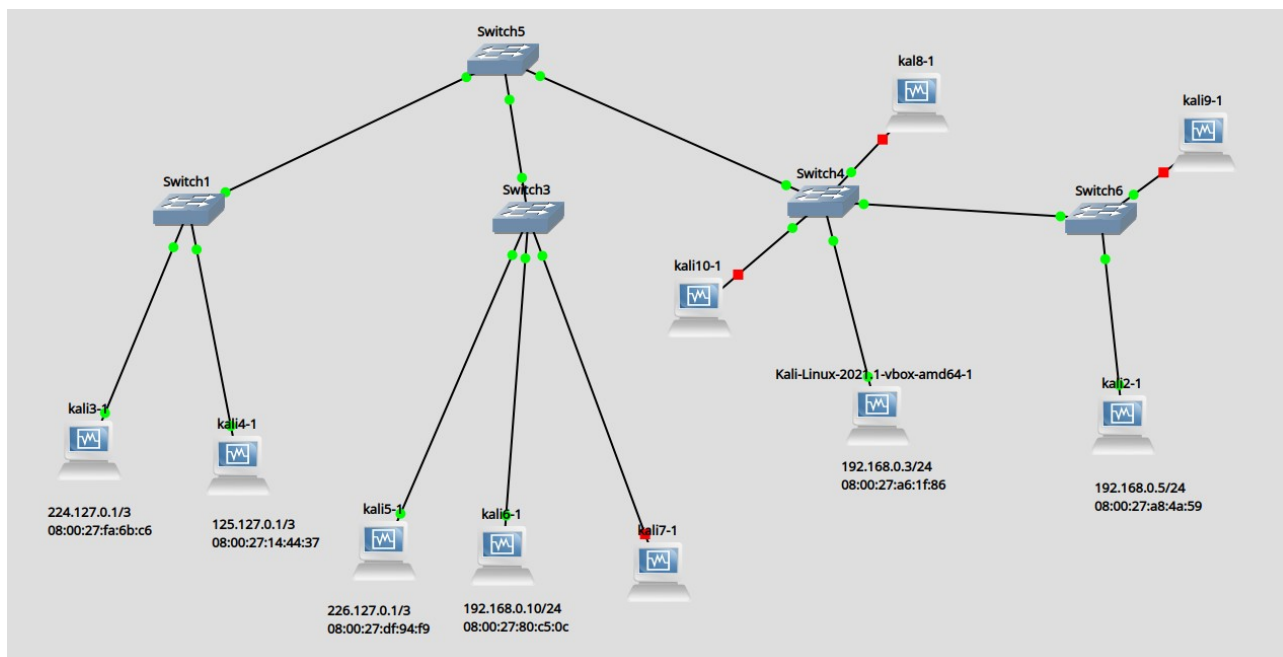
Пакет app содержит

- файл Runner, отвечающий за запуск приложения и взаимодействие компонентов.
- Подпакет frame с классами, имплементирующими кадры обоих типов
- Подпакет Gui – диалоговое окно
- Подпакет listener, читающий и парсящий входные пакеты
- Подпакет sender отвечающий за формирование и отправку кадров обоих типов

Классы SendEthernetRequest и SnifferApp являются

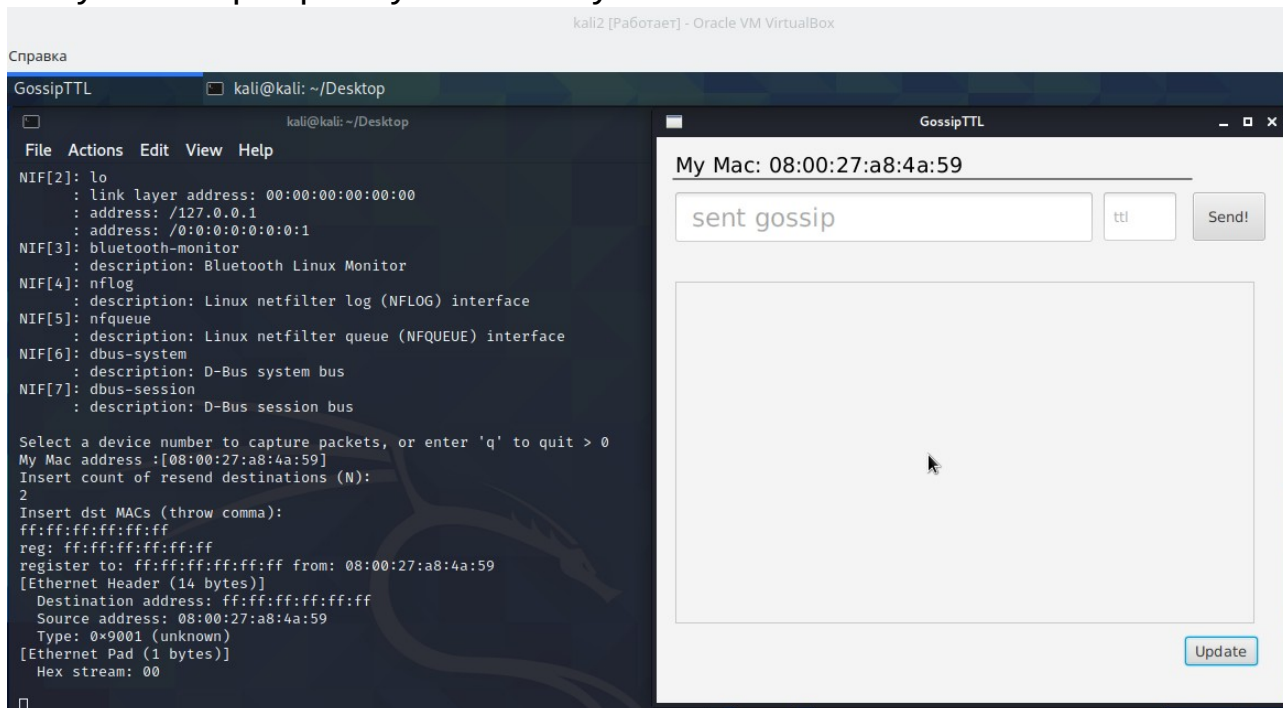
отладочными и не компилируются.

Данная программа была протестирована вот на этой сети:

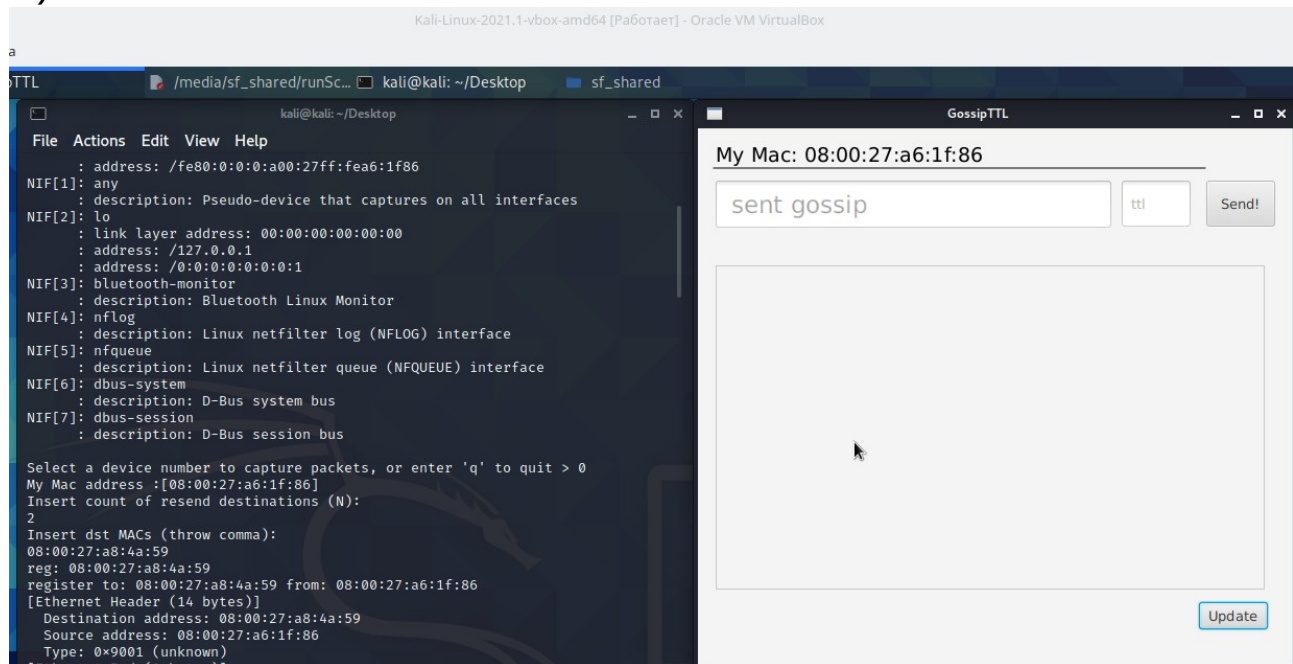


Хосты, помеченные красным неактивны, для них не хватило памяти.

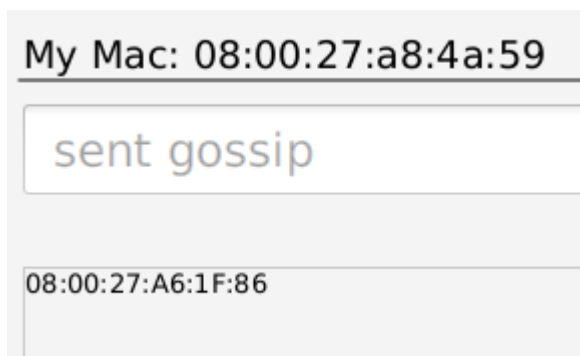
Запускаю программу на хосту kali-2 с N = 2.



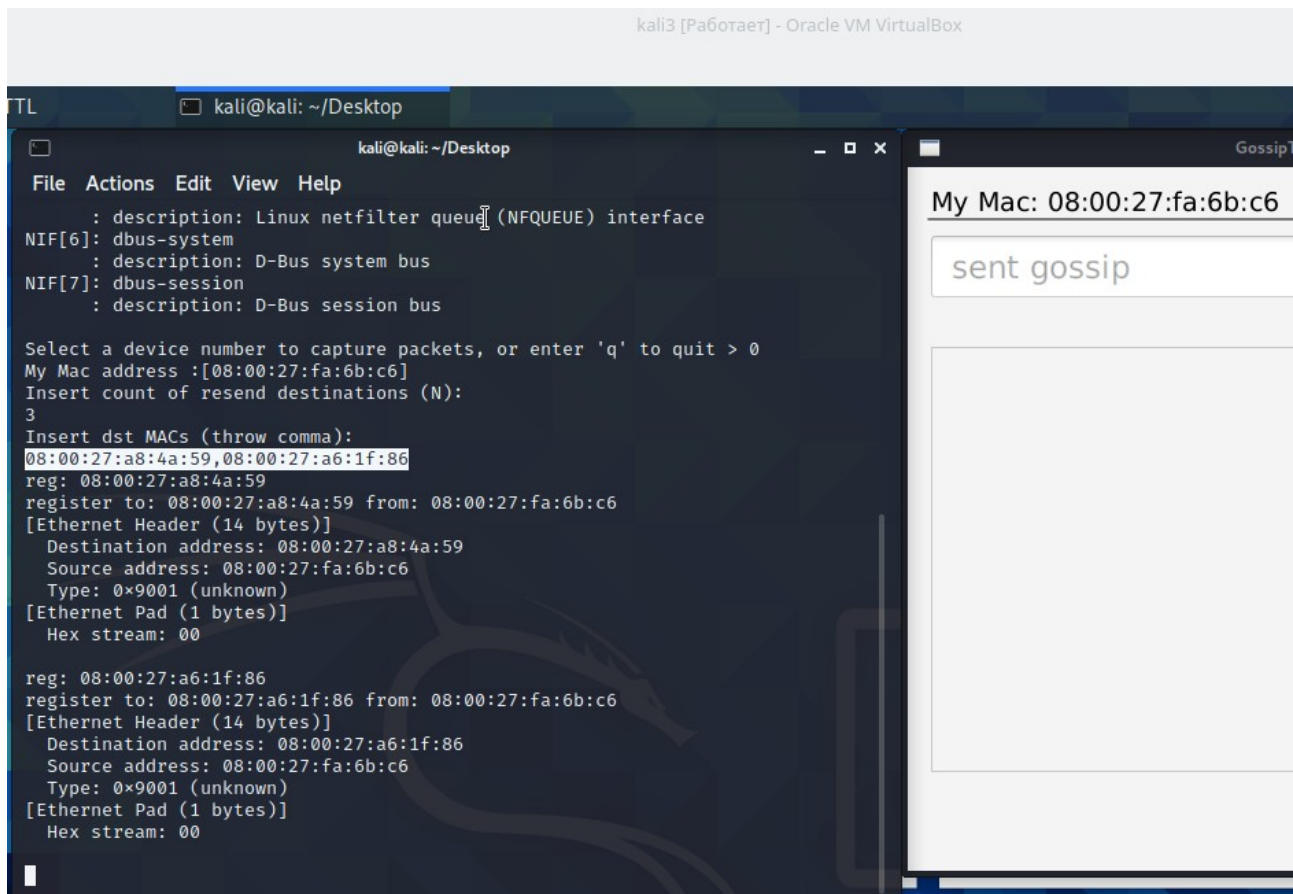
Запускаю программу на хосту №1 и пишу Mac хоста 2 (N = 2).



На 2м хосту появился мак адрес первого хоста.



На 3ем (N=3) хосту пишу маки 1го и 2го хостов.

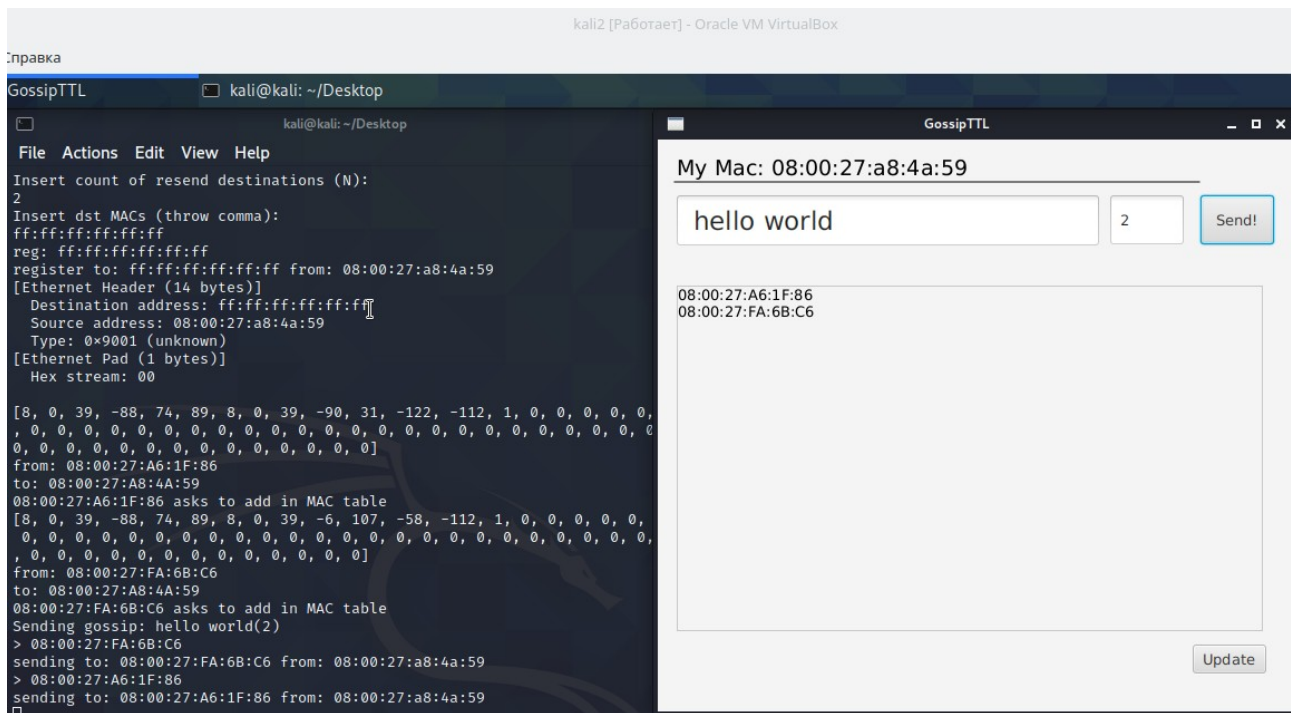


На 1м и 2м хостах обновилась Мак таблицы.

My Mac: 08:00:27:a6:1f:86	My Mac: 08:00:27:a8:4a:59
sent gossip	sent gossip
08:00:27:FA:6B:C6	08:00:27:A6:1F:86 08:00:27:FA:6B:C6

Можно отправить 1ю сплетню.

Отправил со 2го хоста



Получил на 1м и 3ем

Получение на 1м, переправка на 3ий

```
from: 08:00:27:FA:6B:C6  
to: 08:00:27:A6:1F:86  
08:00:27:FA:6B:C6 asks to add in MAC table  
[8, 0, 39, 90, 31, -122, 8, 0, 39, -88, 74, 89, -112, 1, 1, 0, 0, 0, 2, 0, 1,  
1, 104, 101, 108, 108, 111, 32, 119, 111, 114, 108, 100, 0, 0, 0, 0, 0, 0, 0,  
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]  
from: 08:00:27:A8:4A:59  
to: 08:00:27:A6:1F:86  
08:00:27:A8:4A:59 sent 'hello world' with TTL=2  
> 08:00:27:FA:6B:C6  
sending to: 08:00:27:FA:6B:C6 from: 08:00:27:a6:1f:86
```

Получение на Зем от 1го и 2го с разными TTL

```
[8, 0, 39, -6, 107, -58, 8, 0, 39, -88, 74, 119, -112, 1, 1, 0, 0, 0, 2, 0, 11  
 , 104, 101, 108, 108, 111, 32, 119, 111, 114, 108, 100, 0, 0, 0, 0, 0, 0,  
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]  
from: 08:00:27:A8:4A:59  
to: 08:00:27:FA:6B:C6  
08:00:27:A8:4A:59 sent 'hello world' with TTL=2  
[8, 0, 39, -6, 107, -58, 8, 0, 39, -90, 31, -122, -112, 1, 1, 0, 0, 0, 1, 0,  
 11, 104, 101, 108, 108, 111, 32, 119, 111, 114, 108, 100, 0, 0, 0, 0, 0, 0,  
 , 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]  
from: 08:00:27:A6:1F:86  
to: 08:00:27:FA:6B:C6  
08:00:27:A6:1F:86 sent 'hello world' with TTL=1
```

Добавил хост 4 с $N = 1$ и регистрацией на хостах 2 и 3.

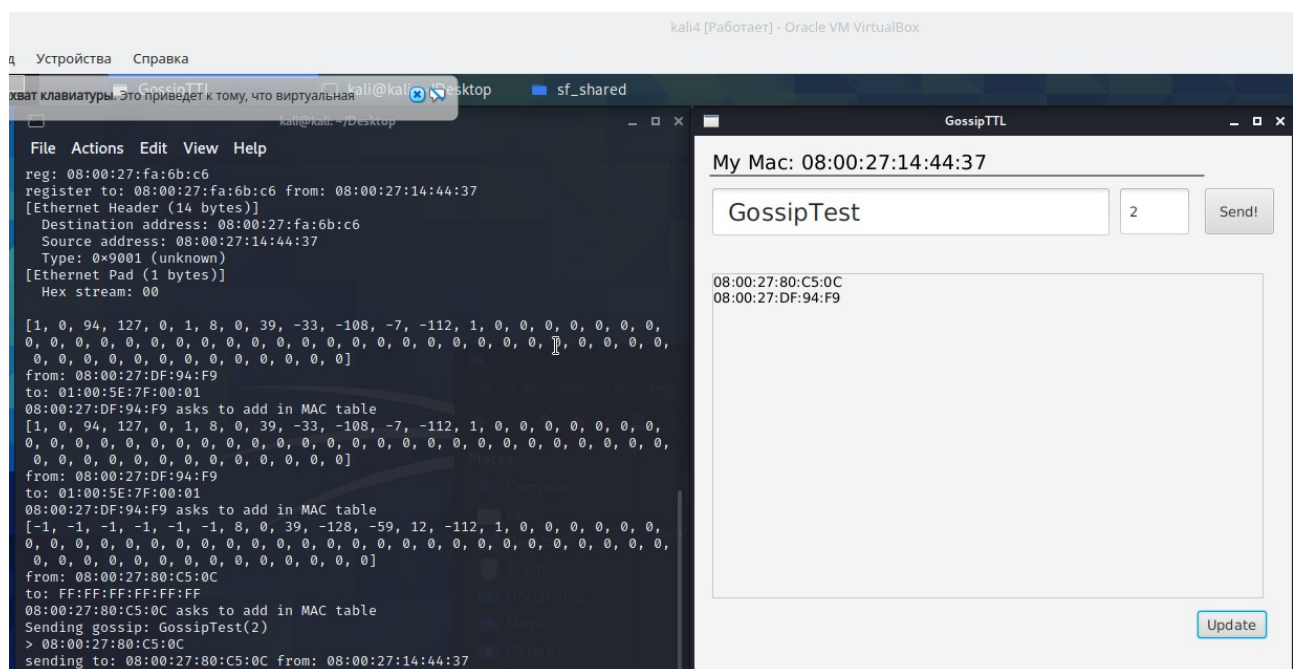
Добавил хост 5 ($N=1$) с регистрации по мультикаст адресу 01:00:5e:7f:00:01, на 2, 3, 4 хостах обновилась Mac таблицы, и на 1м тоже обновилось, т. к. используются простые коммутаторы.

1)

С 6го хоста ($N=2$) при запуске отсылаю бродкаст

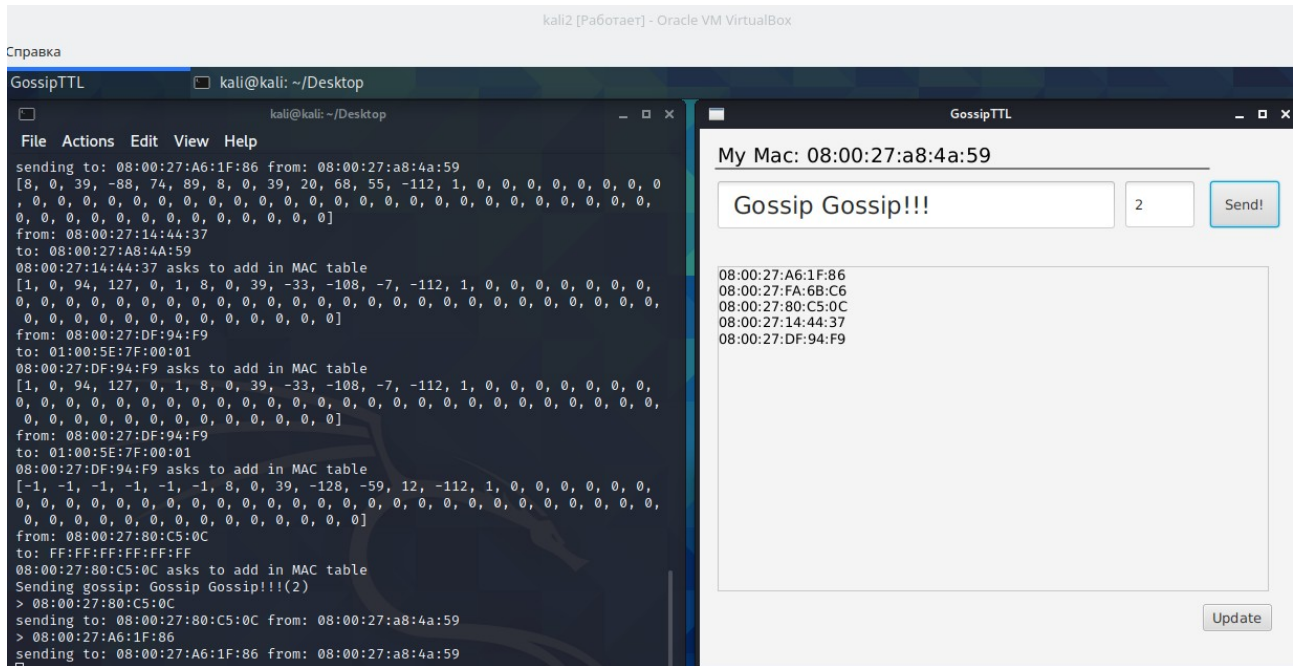
Отсылаю сплетню с 4го с $TTL = 2$;

4й хост знает 2 мак адреса, но у него $N=1$



Кадр отослан на хост 6, который никого не знает, кадр на нём и застревает.

Следующую сплетню отсылаю с хоста 2.



Отослано хостам 1 и 6.

С 1го хоста сплетня пересылается 5 и 3 хостам.

Пришло на 5й

```
[8, 0, 39, -33, -108, -7, 8, 0, 39, -90, 31, -122, -112, 1, 1, 0, 0, 0, 1, 0,
16, 71, 111, 115, 115, 105, 112, 32, 71, 111, 115, 115, 105, 112, 33, 33, 33
, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
from: 08:00:27:A6:1F:86
to: 08:00:27:DF:94:F9
08:00:27:A6:1F:86 sent 'Gossip Gossip!!!' with TTL=1
> 08:00:27:80:C5:0C
sending to: 08:00:27:80:C5:0C from: 08:00:27:df:94:f9
```

Пришло на 3й:

```
from: 08:00:27:A6:1F:86
to: 08:00:27:FA:6B:C6
08:00:27:A6:1F:86 sent 'Gossip Gossip!!!' with TTL=1
> 08:00:27:14:44:37
sending to: 08:00:27:14:44:37 from: 08:00:27:fa:6b:c6
> 08:00:27:DF:94:F9
sending to: 08:00:27:DF:94:F9 from: 08:00:27:fa:6b:c6
> 08:00:27:80:C5:0C
sending to: 08:00:27:80:C5:0C from: 08:00:27:fa:6b:c6
```

С 3го отправилось на 4й, 5й, 6й с ttl=0

```
from: 08:00:27:FA:6B:C6
to: 08:00:27:14:44:37
08:00:27:FA:6B:C6 sent 'Gossip Gossip!!!' with TTL=0
```

5:

```
from: 08:00:27:FA:6B:C6  
to: 08:00:27:DF:94:F9  
08:00:27:FA:6B:C6 sent 'Gossip Gossip!!!' with TTL=0  
█
```

6:

c 5

```
, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,  
from: 08:00:27:DF:94:F9  
to: 08:00:27:80:C5:0C  
08:00:27:DF:94:F9 sent 'Gossip Gossip!!!' with TTL=0  
5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

c 3

```
from: 08:00:27:FA:6B:C6  
to: 08:00:27:80:C5:0C  
08:00:27:FA:6B:C6 sent 'Gossip Gossip!!!' with TTL=0  
█
```