

Генератор пакетов RTP.

Генератор должен формировать пакеты RTP, имитирующие запросы нескольких клиентов RTP на наш сервер.

Количество клиентов должно устанавливаться от 1 до 255. Для каждого клиента должен быть свой IP, MAC, ClockIdentity. Задается начальное значение параметра, а для других клиентов эти параметры формируются прибавлением 1.

Задается до 4 IP которые исключаются из списка вывода

Интервал между запросами должен устанавливаться от 10мкс (как получится, но не менее 40мс) до 60 сек. Интервал между запросами должен быть минимальным: при скорости 100 Мбит/с минимальный интервал между запросами составляет 10мкс, что соответствует максимальной загрузке сети

Количество запросов в секунду должно быть не менее 100.

Порядок вывода клиентов может быть или последовательным (IP увеличивается на 1 или случайным в заданном диапазоне IP)

destination IP - задает константой адрес сервера

destination Mac - задает константой адрес сервера

TimeStamp - текущее время задается константой или берется из системного времени.

RTP пакет (Delay_Req Message) передаётся по протоколу UDP через 319 порт, таким образом задача сводится к формированию UDP пакета с произвольными данными.

Состав требуемого пакета в Таблице 1. Формат пакета IEEE 802.3 UDP/IP.

Таблица 1

| Протокол | Смещение, байт | Размер поля, байт | Название | Действия с полем |
|---------------------------|----------------|-------------------|--|-------------------------------------|
| Ethernet II or IEEE 802.3 | 0 | 8 | Preamble/SFD 0x55,0x55,...0x55,0xD5 | Не меняется. |
| | 8 | 6 | Destination MAC | Не меняется. |
| | 14 | 6 | Source MAC | Меняется, для каждого клиента свой. |
| | 20 | 2 | Ethertype or length | Не меняется. |
| IP | 22 | 1 | Version, Internet Header Length | Не меняется. |
| | 23 | 1 | Differentiated Services Code Point, Explicit Congestion Notification (DSCP, ECN) | Не меняется. |
| | 24 | 2 | Total Length | Не меняется. |
| | 26 | 2 | Identification | Не меняется. |
| | 28 | 2 | Flags, Fragments Offset | Не меняется. |
| | 30 | 1 | Time to Live (TTL) | Не меняется. |
| | 31 | 1 | Protocol | Не меняется, должен быть UDP, 17. |
| | 32 | 2 | Header Checksum | Не меняется. |

| | | | | |
|--|----|---|---------------------|-------------------------------------|
| | 34 | 4 | Source address | Меняется, для каждого клиента свой. |
| | 38 | 4 | Destination address | Не меняется, адрес нашего сервера. |

| | | | | | |
|-----|----|----|-------------------------|--------------------|---|
| UDP | 42 | 2 | Source Port Number | | Не меняется., должен быть 319 |
| | 44 | 2 | Destination Port Number | | Не меняется., должен быть 319 |
| | 46 | 2 | Length | | Не меняется., должно быть 52 |
| UDP | 48 | 2 | Checksum | | Не меняется |
| PTP | 50 | 1 | 4 бита | Transport Specific | Не меняется, должно равняться 0. |
| | | | 4 бита | Message Type | Не меняется, должно равняться 1. |
| | 51 | 1 | Version PTP | | Не меняется, должно равняться 2. |
| | 52 | 2 | Message length | | Не меняется, должно равняться 44. |
| | 54 | 2 | Subdomain number | | Не меняется, должно равняться 0. |
| | 56 | 2 | Flags | | Не меняется, должно равняться 0. |
| | 58 | 12 | Correction Field | | Не меняется, должно равняться 0. |
| | 70 | 8 | ClockIdentity | | Меняется, у каждого клиента свой номер. |
| | 78 | 2 | SourcePortID | | Не меняется, должно равняться 1 или любое другое число, может быть различное у каждого клиента. |
| | 80 | 2 | SequenceID | | Меняется, инкрементируется с каждым запросом, для каждого клиента отдельно. |
| | 82 | 1 | Control | | Не меняется, должно равняться 1. |
| | 83 | 1 | LogMessagePeriod | | Не меняется, должно равняться 127. |
| | 84 | 6 | TimeStamp (seconds) | | Можно не менять или подставить текущее время. |
| | 90 | 4 | TimeStamp (nanoseconds) | | Можно не менять или подставить текущее время. |
| | 94 | 4 | FCS | | Не меняется. |

В таблице зеленым цветом указаны данные, которые формирует операционная

система или сетевая карта и т. п. Синим цветом указаны поля, которые должны настраиваться и на которые стоит обратить внимание. Красным цветом указаны PTP данные, которые формируются и настраиваются генератором пакетов, некоторые из этих данных постоянны.

Описание некоторых полей пакета.

- 1) Поле Destination MAC должно соответствовать адресу сервера, к примеру: 01:00:5e:00:01:81 (настройки спросить у программистов).
- 2) Поле Source MAC для каждого клиента имеет свое значение, может быть выбрано случайно, но не должно соответствовать широковещательным и мультикастовым адресам.
- 3) Поле Source address – IP адрес клиента, у каждого клиента свой адрес, который должен принадлежать диапазону локальной сети, в которой находится сервер, а это значит, что количество участников сети превышает 255 (255 клиентов и 1 сервер) и маска подсети должна быть установлена на значение 255.255.0.0. (касается настройки сервера и вообще организации локальной сети).
- 4) Поле Destination address – IP адрес сервера, «мультикастовый» адрес. К примеру: 224.0.1.129 (спросить у программистов).
- 5) Поле Source Port Number – порт отправителя UDP пакета, для PTP сообщения этот порт должен равняться 319.
- 6) Поле Destination Port Number – порт получателя UDP пакета, для PTP сообщения этот порт должен равняться 319.
- 7) Поле Length – количество байт, передаваемых по протоколу UDP, включая данные, порт отправителя, порт получателя, контрольную сумму UDP и само поле Length. Для PTP сообщения количество байт равно 52, из них 8 байт заголовок UDP, а 44 байт данные PTP. Для самоконтроля.
- 8) Поле ClockIdentity – идентификационный номер клиента. Номера клиентов могут быть выбраны случайно. Количество клиентов должно быть не меньше 255.
- 9) Поле SourcePortID — порт клиента. Не должен быть нулевым. Может быть одинаковым у разных клиентов.
- 10) Поле SequenceId – поле номера запроса. Для каждого клиента ведется свой счетчик запросов. Значение этого поля увеличивается на 1 при каждом запросе. Должно обязательно меняться от запроса к запросу.
- 11) Поле TimeStamp – метка времени клиента. Содержит время, когда клиент был отправлен. В это поле можно подставить текущее время.

Значения остальных полей постоянны и указаны в таблице.

Дополнительно

Заложить возможность случайной очередности передачи запросов.

Программа может быть выполнена на языке Python (Константин уже это делал).

Пример пакета PTP в Wireshark:

\\SERVER2KB\archive\project\ptp\PTP_packet_gen\PTP_request.pcap

Последующие модификации

1. Прием и анализ полей ответных пакетов PTP Delay_Resp Message
- 2/ Поддержка VLAN IEEE 802.1Q