Table of Contents

2 Описание параметров 3
RTT 3
Ретрансмиты 4

2 Описание параметров

RTT

Время приема-передачи (англ. round-trip time, RTT) — это время, затраченное на отправку сигнала, плюс время, которое требуется для подтверждения, что сигнал был получен. Это время задержки, следовательно, состоит из времени передачи сигнала между двумя точками в пределах одного flow.
За flow в DPI принимается вся сетевая активность в рамках source/destination socket (source IP:port /destination IP:port).
Так как весь flow между клиентом и сервером проходит через DPI, подсчет RTT на DPI производится для двух направлений:

1. От клиента до DPI (обозначение в GUI **от абонента**)
2. От сервера до DPI (обозначение в GUI **к абоненту**)

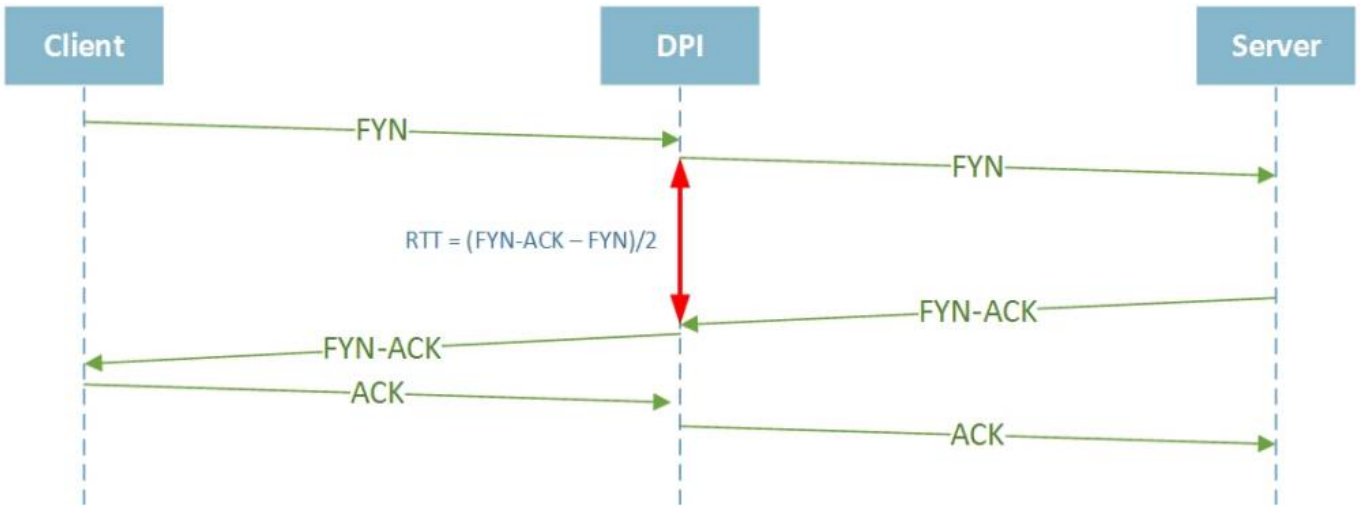
Регистрация каждого нового flow на DPI производится не по сообщению SYN от инициатора TCP соединения, а при получении ответа SYN/ACK, поэтому подсчет RTT производится исходя из разницы передачи и приема последующих сообщений:

Клиент может являться сервером, а сервер - клиентом, в зависимости от того, кто инициализирует TCP соединение (TCP SYN). Соответственно, логика подсчета RTT тогда тоже меняется, и подсчет ведется наоборот.

!!! Важно понимать, что RTT считается только для session-oriented (TCP) соединений. Для UDP подсчет RTT не производится.

RTT от абонента до DPI

RTT к абоненту - от сервера до DPI (в случае если завершение было инициировано со стороны клиента)



RTT к абоненту - от сервера до DPI (в случае если завершение было инициировано со стороны сервера)

Особенности протокола TCP и расчет RTT

В виду особенностей протокола TCP, возможно множество различных ситуаций, влияющих на подсчет RTT для конкретного flow.

RTT от клиента до DPI для некоторых flow равен нулю

В случае, если DPI не получил ACK от клиента на отправленный SYN/ACK. Подобная ситуация может случиться по нескольким причинам, например, клиент разорвал соединение физически, либо прислал RST. Во всех подобных ситуация, DPI проставит значение “0” в RTT от клиента до DPI для данного flow.

RTT для некоторых flow принимают очень большие значения (десятки секунд)

Например, такая ситуация может возникнуть в случае TCP HALF CLOSED CONNECTION (наполовину закрытый TCP), когда один из участников соединения прекращает передачу данных, однако все еще может получать данные от удаленной стороны. В таком случае, передающая сторона может послать FYN/ACK только после завершения передачи данных, в следствии чего, значение RTT значительно возрастет.

Ретрансмиты

1. Общий процент ретрансмитов
2. Процент ретрансмитов, когда трафик от абонета
3. Процент ретрансмитов, когда трафик к абонету

Виды перезапросов:

- TCP Retransmission – классический тип повторной передачи пакетов. Отправитель пакета, не получив подтверждения получения от адресата по истечении таймера retransmission timer, отправляет пакет повторно автоматически, предполагая, что он потерян по пути следования. Значение таймера подстраивается гибко и зависит от кругового времени передачи по сети для конкретного канал связи. Как он рассчитывается можно узнать в RFC6298 Computing TCP's Retransmission Timer.
- TCP Fast Retransmission – отправитель отправляет повторно данные немедленно после предположения, что отправленные пакеты потеряны, не дожидаясь истечения времени по таймеру (ransmission timer). Обычно триггером для этого является получение нескольких подряд (обычно три) дублированных подтверждений получения с одним и

тем же порядковым номером.Например, отправитель передал пакет с порядковым номером 1 и получил подтверждение – порядковый номер плюс 1, т.е. 2. Отправитель понимает, что от него ждут следующий пакет с номером два. Предположим, что следующие два пакета потерялись и получатель получает данные с порядковым номером 4. Получатель повторно отправляет подтверждение с номером 2. Получив пакет с номером 5, отправитель все равно отправляет подтверждение с номером 2. Отправитель видит три дублированных подтверждения, предполагает, что пакеты 2, 3 были потеряны и шлет их заново, не дожидаясь таймера.

- Spurious Retransmission – этот тип повторной передачи появился в версии 1.12 сниффера Wireshark и означает, что отправитель повторно отправляет пакеты, на которые получатель уже отправил подтверждение.