

# Содержание

<b>Parameters description</b> .....	3
RTT .....	3
Retransmits .....	4



# Parameters description

## RTT

Round-trip time, RTT is the time taken to send the signal, plus the time it takes to confirm that the signal was received. This delay time, therefore, consists of the signal transmission time between two points within the same flow.

The term **flow** in DPI refers to all network activity within the source/destination socket (source IP:port / destinationIP:port).

Since the entire flow between the client and server goes through DPI, RTT calculation is performed by DPI for two directions:

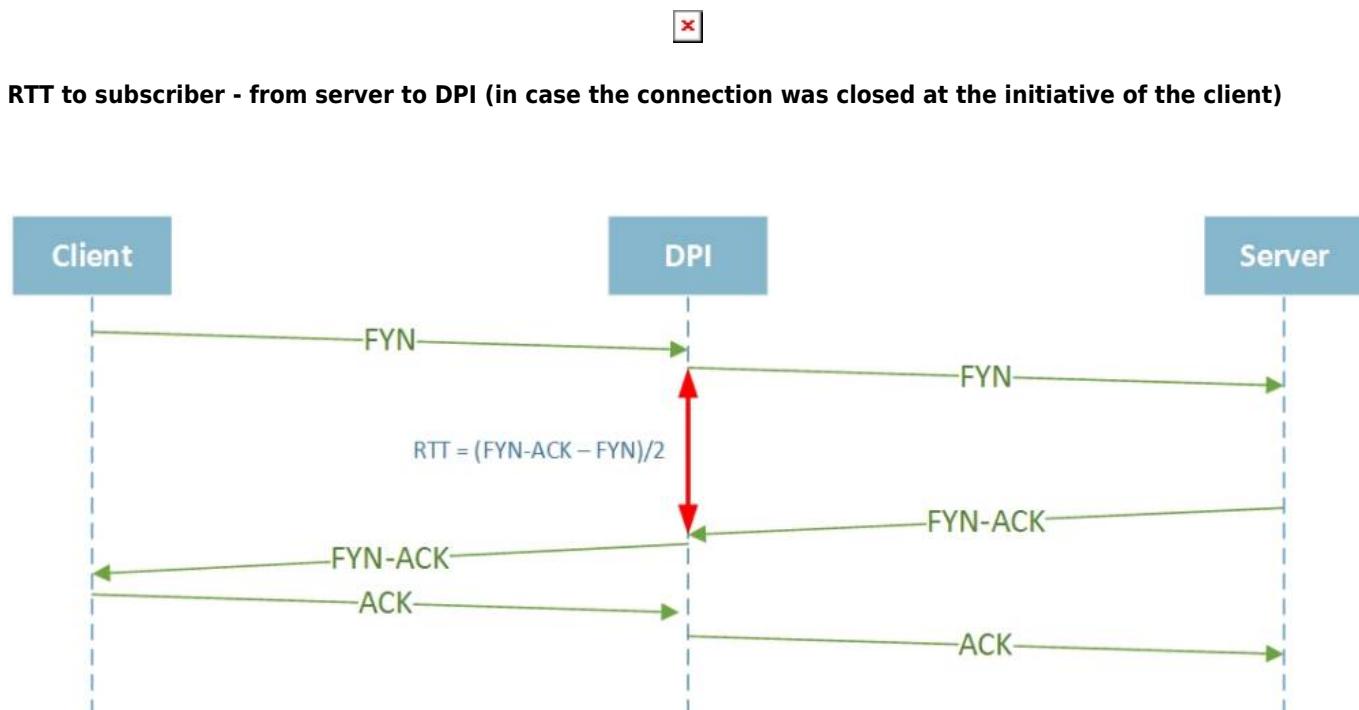
1. From subscriber to DPI (GUI has the following notation: **from subscriber**)
2. From server to DPI (GUI has the following notation: **from server**)

Registration of each new flow is performed by DPI on SYN/ACK received response basis instead of on SYN message sent from the initiator of the TCP connection, therefore, the RTT calculation is based on the difference between the transmission and reception of the following messages:

The client can be a server or the server can be a client depending on initiator of the TCP connection (TCP SYN). So the logic of RTT calculation also changes and calculation is done the other way around.

!!! It is important to know that RTT is only considered for session-oriented (TCP) connections. RTT calculation is not performed in case of UDP.

### RTT from subscriber to DPI



## RTT to subscriber - from server to DPI (in case the connection was closed at the initiative of the server)



### TCP protocol specificity and RTT calculation

There are many different situations affecting the RTT calculation for a particular flow due to some TCP protocol features.

#### RTT from subscriber to DPI equals null for some flows



It corresponds the situation when DPI did not receive ACK sent from client to the DPI in response to received SYN/ACK. The situation can be caused by several reasons, for example, if the client physically disconnected the wire or sent RST. In all the situations mentioned above, the DPI will set "0" value in RTT from the client to the DPI for the given flow.

#### RTT for some flows take very large values (tens of seconds)



For example, this situation may occur in the case of TCP HALF CLOSED CONNECTION, i.e. when one of the connection participants terminates the data transmission, but still continues to receive the data from the remote side. In this case, the transmitting side can send FYN/ACK only after the data transfer is completed, so it will cause the significant increasing of RTT value.

## Retransmits

1. Total retransmit percentage
2. The percentage of retransmissions in case of subscriber outgoing traffic
3. The percentage of retransmissions in case of subscriber incoming traffic



Retransmission types:

- 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 канала 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 (transmission 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 снiffeра Wireshark и означает, что отправитель повторно отправляет пакеты, на которые получатель уже отправил подтверждение.