Содержание

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**)

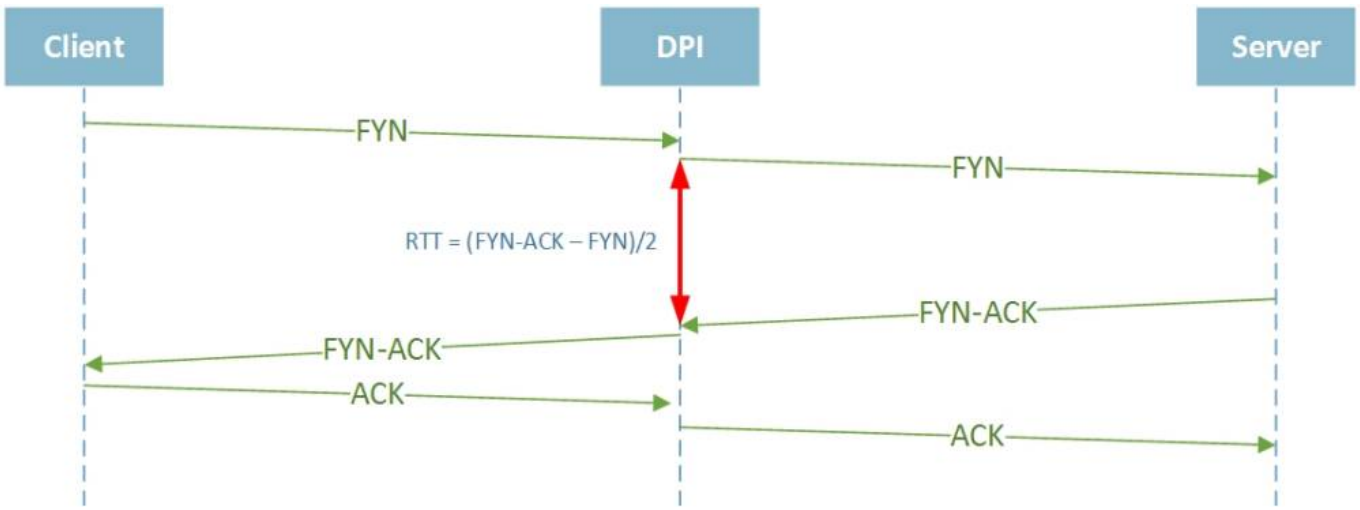
Registration of each new flow is performed by DPI on SYN/ACK received response basis instead of on SYN message sent from the initiator of the TCP connection, therefore, the RTT calculation is based on the difference between the transmission and reception of the following messages:

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 client)



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 outcoming traffic
3. The percentage of retransmissions in case of subscriber incoming traffic

Retransmission types:

- **TCP Retransmission** is a classic type of packet retransmission. The packet sender having not received acknowledgment from the addressee after the retransmission timer expires, resends the packet automatically, assuming that it is lost along the route. The timer value is flexibly adjusted and depends on the circular transmission time over the network for a particular communication channel. RFC6298 (Computing TCP's Retransmission Timer) specifies the algorithm to calculate the timer.
- **TCP Fast Retransmission** corresponds for the following case: the sender resends the data immediately after assuming that the sent packets are lost without waiting for the expiration of the transmission timer. Usually it can be triggered by receiving several consecutive (usually three) duplicate acknowledgments with the same serial number. For example, the sender transmitted a packet with sequence number 1 and received an acknowledgment equal to

sequence number plus 1, i.e. 2. The sender understands that the next packet with number two is expected. Suppose that the next two packets are lost and the recipient receives data with serial number 4. The recipient resends the acknowledgment with the number 2. Upon the receiving the packet with the number 5, the sender still sends the acknowledgment with the number 2. The sender sees three duplicate acknowledgments, assumes that the packets 2 and 3 were lost and resend them without waiting for a timer to expire.

- **Spurious Retransmission** is the type of retransmission appeared in the version 1.12 of Wireshark sniffer and means that the sender resends packets to which the recipient has already sent acknowledgment.