TCP is a connection-oriented protocol that enables end-to-end communications to be established by handshaking. User data may be forwarded two-way over the connection once it is established. UDP is an easier message-based connectionless protocol than TCP, meaning that there is no dedicated end-to-end connection and information is passed from source to destination from the source in one direction without checking the receiver availability or status. TCP controls the acknowledgement, broadcast and timeout of messages. TCP attempts to send messages that get lost throughout the way, so there are no missing data in TCP and if there are ever several timeouts, the link is discontinued. When a UDP message is sent, there is no guarantee that the message reaches its destination. For UDP, there is no acknowledgement, retransmission, or timeout. When two messages are sent sequentially, the first message reaches the destination first. If data segments are not in the correct order, TCP buffers can delay data until all the data can be reordered prior to delivery however when using UDP, it is impossible to predict which messages arrive. The TCP protocol has comprehensive algorithms to ensure correct data delivery. The combination of two TCP connections forces both TCP connections to operate in parallel. However TCP was not intended to work like this, and in different situations, problems are likely to occur. The retransmission problems, TCP meltdown and double retransmission problems occur because of TCP tunnelling. Problems can happen when both connections are stacked and packets are being retransmitted. In past work, it is not quite clear how serious the problems with retransmission are, in relation to TCP in tunnelling. Automatic recovery from any loss or dropped data was provided for TCP protocol suite. This protocol has to be able to retrieve from any host's failure anywhere in the network and any time during data transfer. When TCP packets are sent across the network from one endpoint to the other, the data packets are reorganized in the same sequence generated by the sender. The protocol detects the removal, reordering, duplicating and corrupting of segments of the data stream by the network. The transmitter can even broadcast damaged segments. That makes TCP a trustworthy protocol which is very reliable. The double retransmission, however, creates latency. Retransmission, message acknowledgement and timeouts are regulated by TCP. TCP delivers lost messages in several attempts along the way. There are no missing data in TCP transmission and the connection is dropped if there are ever multiple timeouts. There is no guarantee that the message will reach its destination when a UDP message is sent. There is no retransmission, timeout and acknowledgement. When two packets of data are sent in sequence, the first message reaches the destination first. When data segments arrive in the incorrect order, TCP buffers hold the data until all data is re-ordered prior to the transmission while using UDP, no prediction can be made of the order in which the messages arrive. TCP was designed to create an efficient, low-head protocol, a protocol suite that would transfer a minimum amount of 'extra' data. This additional information, called overhead, is a package for the transmission of the data and enables the transmission of data. TCP Tunnel is a technology that adds and transfers packets sent as a single TCP connection between the end hosts. By using a TCP tunnel you can improve the equity between aggregated ows and transparently transmit a number of protocols through a firewall. Many of the applications currently use a TCP tunnel, including as Secure Shell (SSH), Virtual Tunnels (VTun), and Http Tunnel (HTun). However, since more applications on end hosts generally have TCP, there are simultaneous operations and mutual interference between two TCP congestion monitoring systems, like end-to-end TCP and tunnel TCP. The use of a TCP tunnel under certain conditions can greatly impact the TCP performance from end to end. It is more explicitly known that the end-to-end TCP passage is degraded output for a period of time when using a TCP tunnel. The problem is referred to as TCP meltdown.