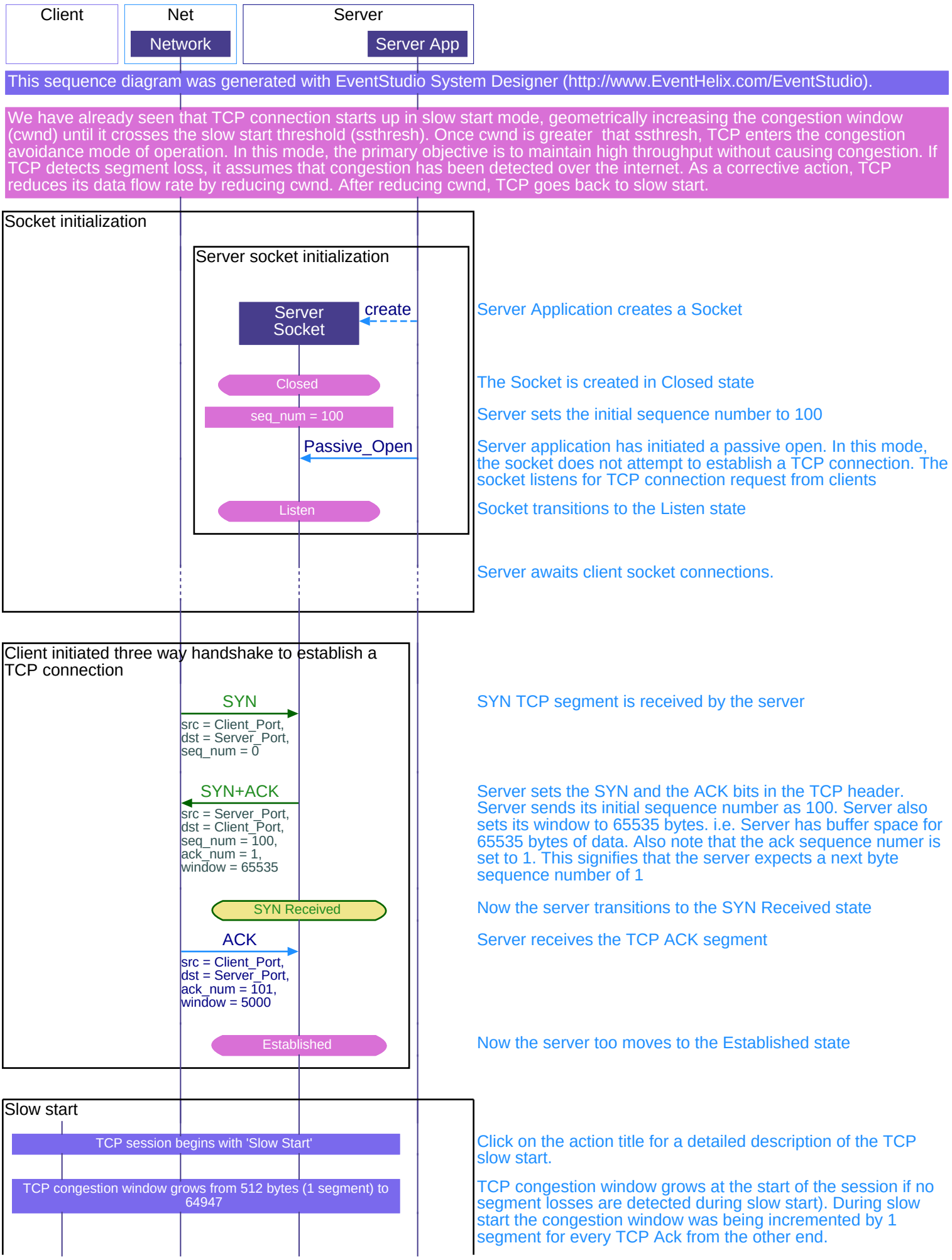
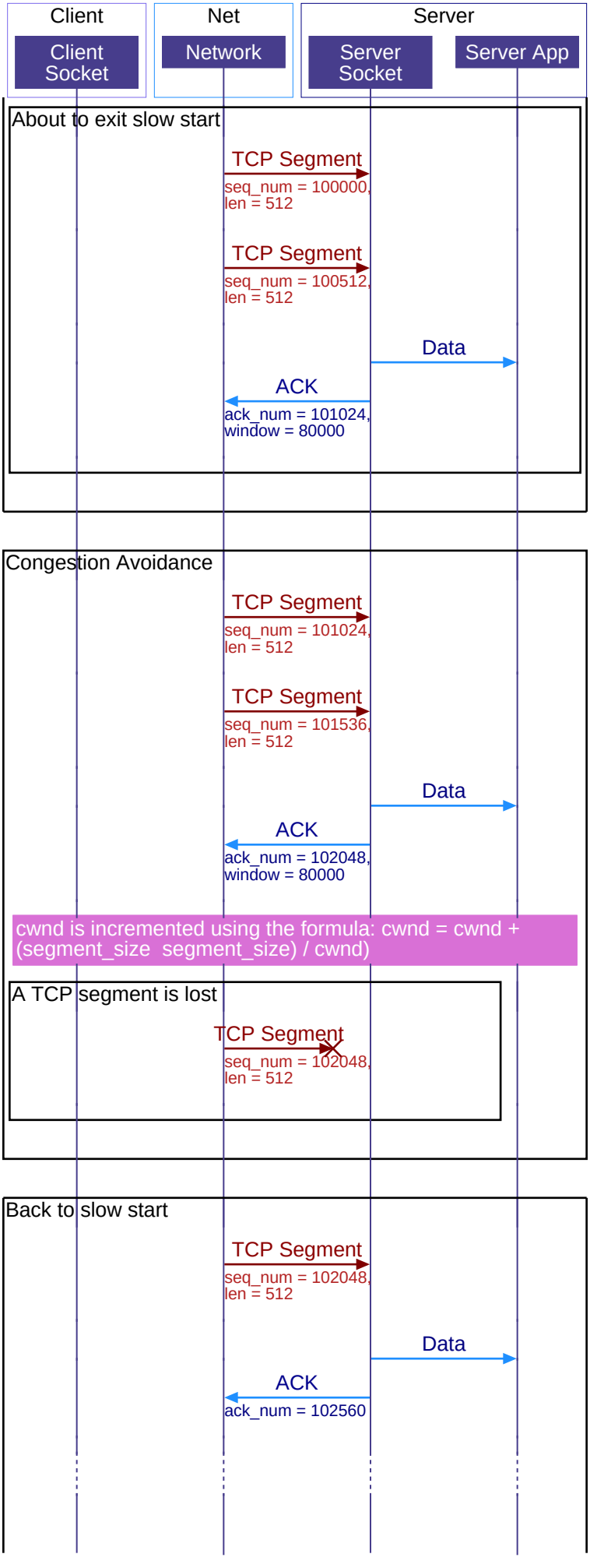

Client
Net
Network
Server
Server App
This sequence diagram was generated with EventStudio System Designer (http://www.EventHelix.com/EventStudio).
We have already seen that TCP connection starts up in slow start mode, geometrically increasing the congestion window (cwnd) until it crosses the slow start threshold (ssthresh). Once cwnd is greater that ssthresh, TCP enters the congestion avoidance mode of operation. In this mode, the primary objective is to maintain high throughput without causing congestion. If TCP detects segment loss, it assumes that congestion has been detected over the internet. As a corrective action, TCP reduces its data flow rate by reducing cwnd. After reducing cwnd, TCP goes back to slow start.
Socket initialization
Server socket initialization
Server Socket
create
Server Application creates a Socket
Closed
The Socket is created in Closed state
seq_num = 100
Server sets the initial sequence number to 100
Passive_Open
Server application has initiated a passive open. In this mode, the socket does not attempt to establish a TCP connection. The socket listens for TCP connection request from clients
Listen
Socket transitions to the Listen state
Server awaits client socket connections.
Client initiated three way handshake to establish a TCP connection
SYN
src = Client_Port,
dst = Server_Port,
seq_num = 0
SYN TCP segment is received by the server
SYN+ACK
src = Server_Port,
dst = Client_Port,
seq_num = 100,
ack_num = 1,
window = 65535
Server sets the SYN and the ACK bits in the TCP header. Server sends its initial sequence number as 100. Server also sets its window to 65535 bytes. i.e. Server has buffer space for 65535 bytes of data. Also note that the ack sequence numer is set to 1. This signifies that the server expects a next byte sequence number of 1
SYN Received
Now the server transitions to the SYN Received state
ACK
src = Client_Port,
dst = Server_Port,
ack_num = 101,
window = 5000
Server receives the TCP ACK segment
Established
Now the server too moves to the Established state
Slow start
TCP session begins with 'Slow Start'
Click on the action title for a detailed description of the TCP slow start.
TCP congestion window grows from 512 bytes (1 segment) to 64947
TCP congestion window grows at the start of the session if no segment losses are detected during slow start). During slow start the congestion window was being incremented by 1 segment for every TCP Ack from the other end.



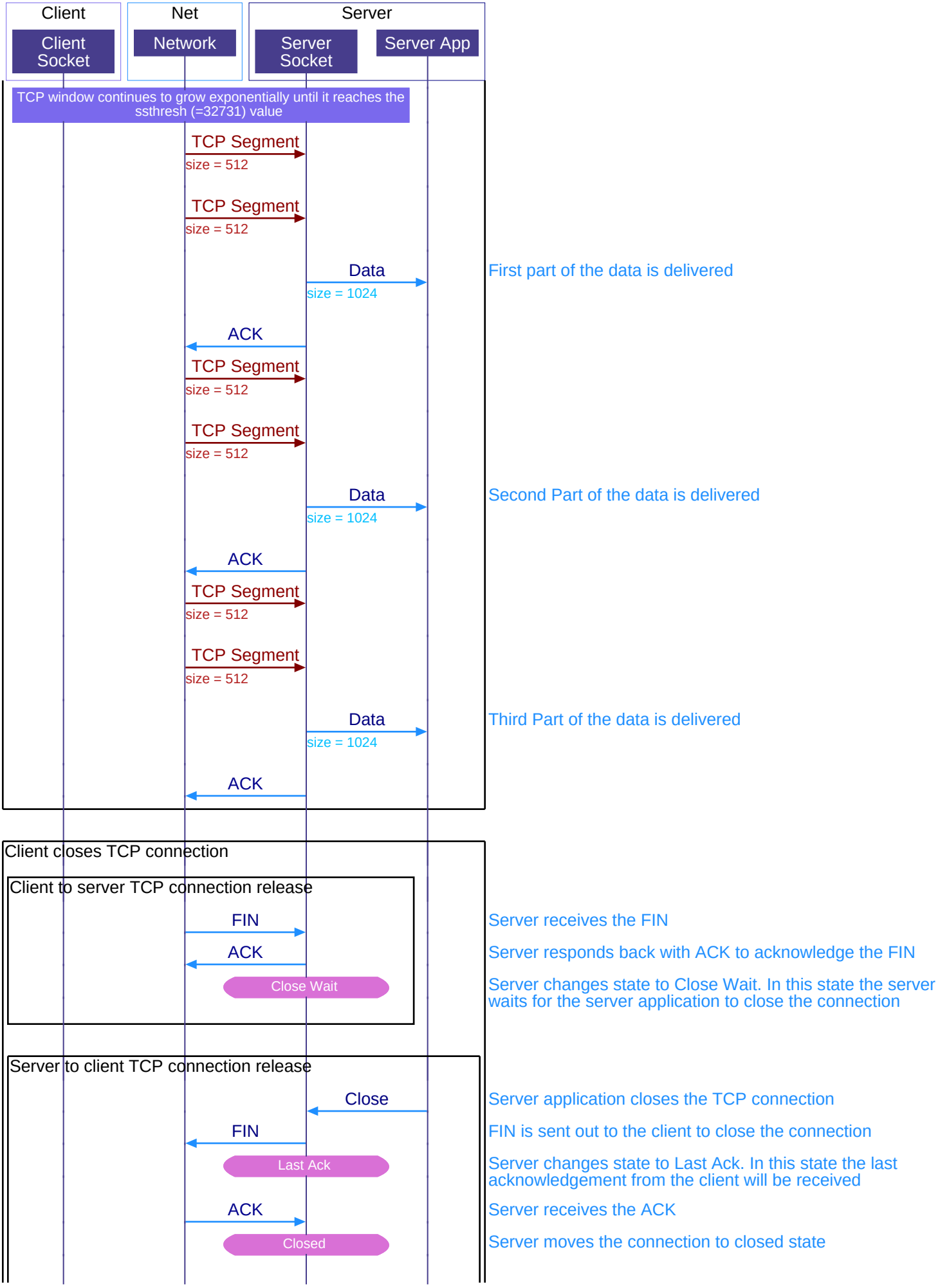
Data is forwarded to the server side application

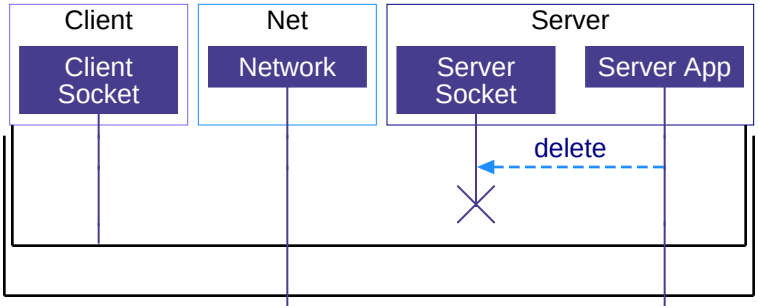
Client acknowledges the last block and also signals an increase in receiver window to 80000

Data is forwarded to the server application

Some node in the Internet drops the TCP segment due to congestion

Data is finally given to the server application


Client
Net
Server
Client Socket
Network
Server Socket
Server App
TCP window continues to grow exponentially until it reaches the ssthresh (=32731) value
TCP Segment
size = 512
TCP Segment
size = 512
Data
size = 1024
First part of the data is delivered
ACK
TCP Segment
size = 512
TCP Segment
size = 512
Data
size = 1024
Second Part of the data is delivered
ACK
TCP Segment
size = 512
TCP Segment
size = 512
Data
size = 1024
Third Part of the data is delivered
ACK
Client closes TCP connection
Client to server TCP connection release
FIN
Server receives the FIN
ACK
Server responds back with ACK to acknowledge the FIN
Close Wait
Server changes state to Close Wait. In this state the server waits for the server application to close the connection
Server to client TCP connection release
Close
Server application closes the TCP connection
FIN
FIN is sent out to the client to close the connection
Last Ack
Server changes state to Last Ack. In this state the last acknowledgement from the client will be received
ACK
Server receives the ACK
Closed
Server moves the connection to closed state



This sequence diagram was generated with EventStudio System Designer (http://www.EventHelix.com/EventStudio).