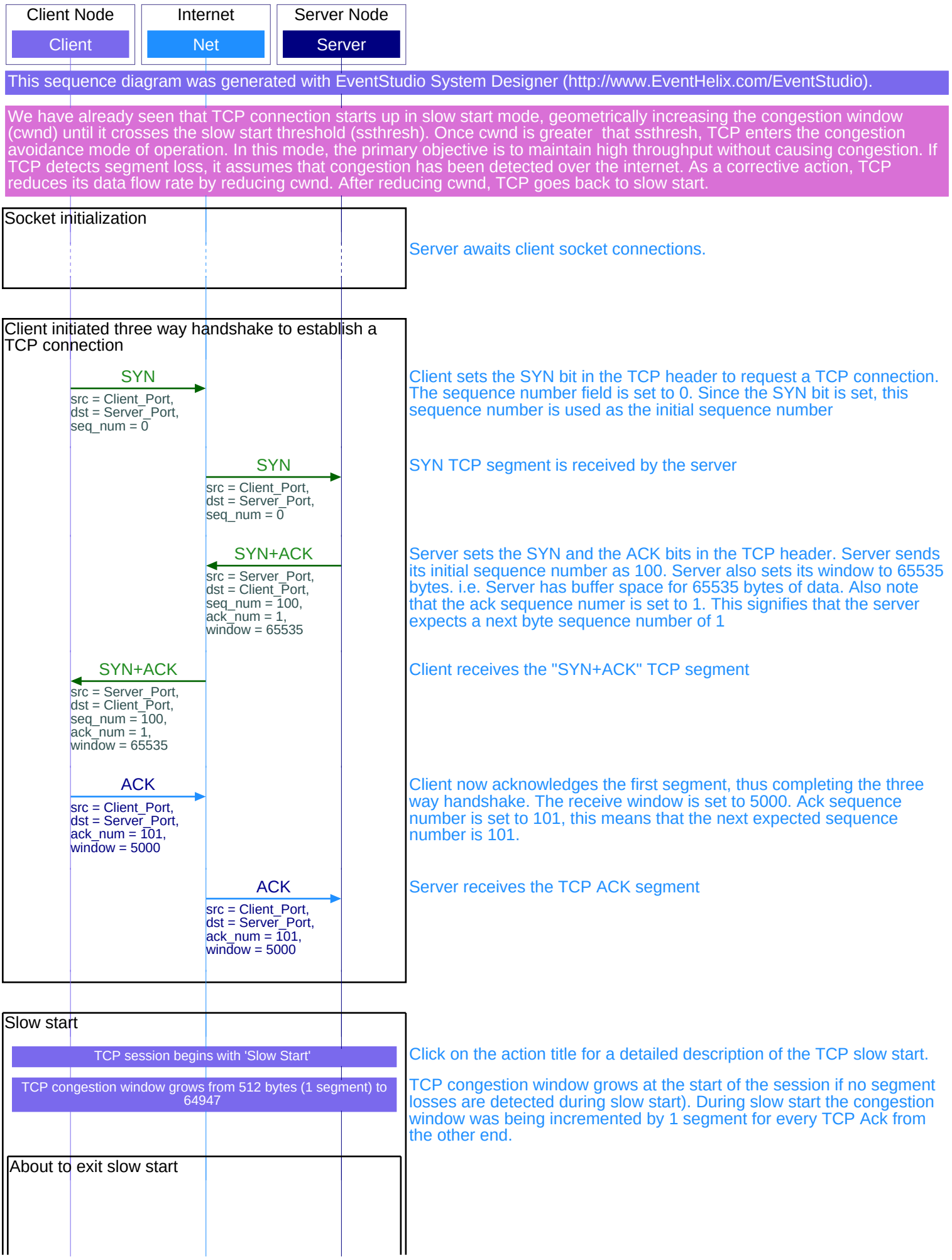
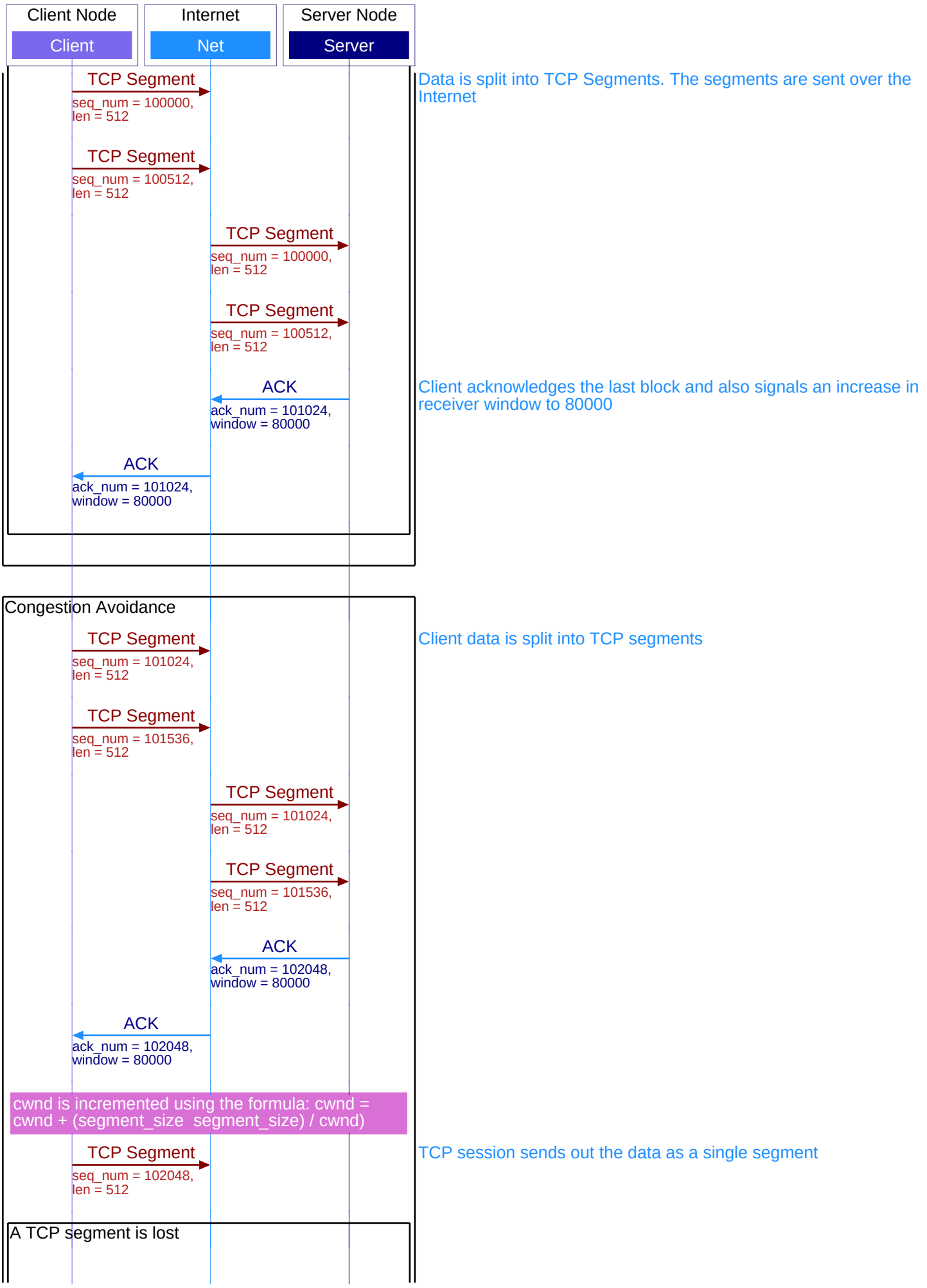
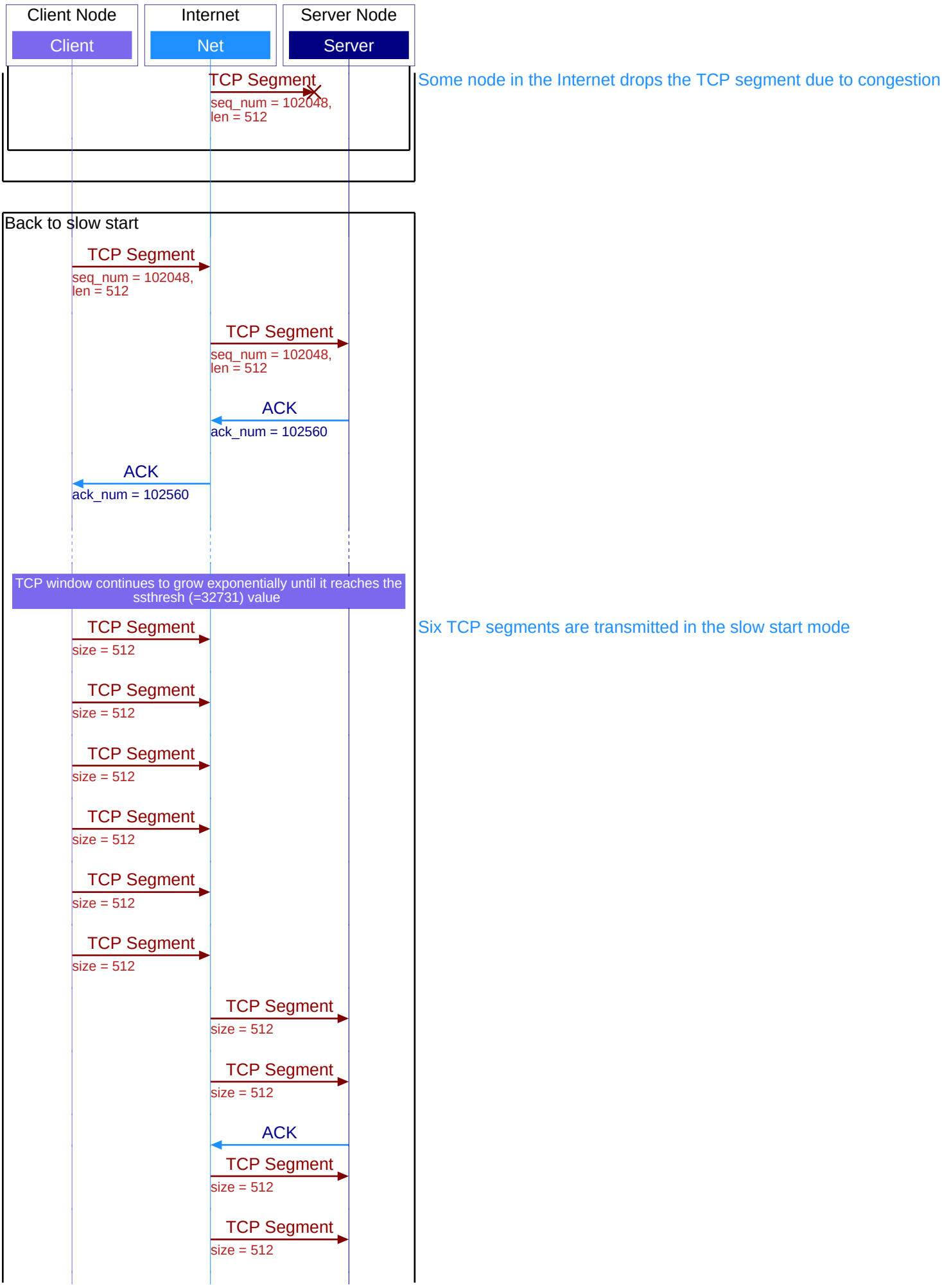
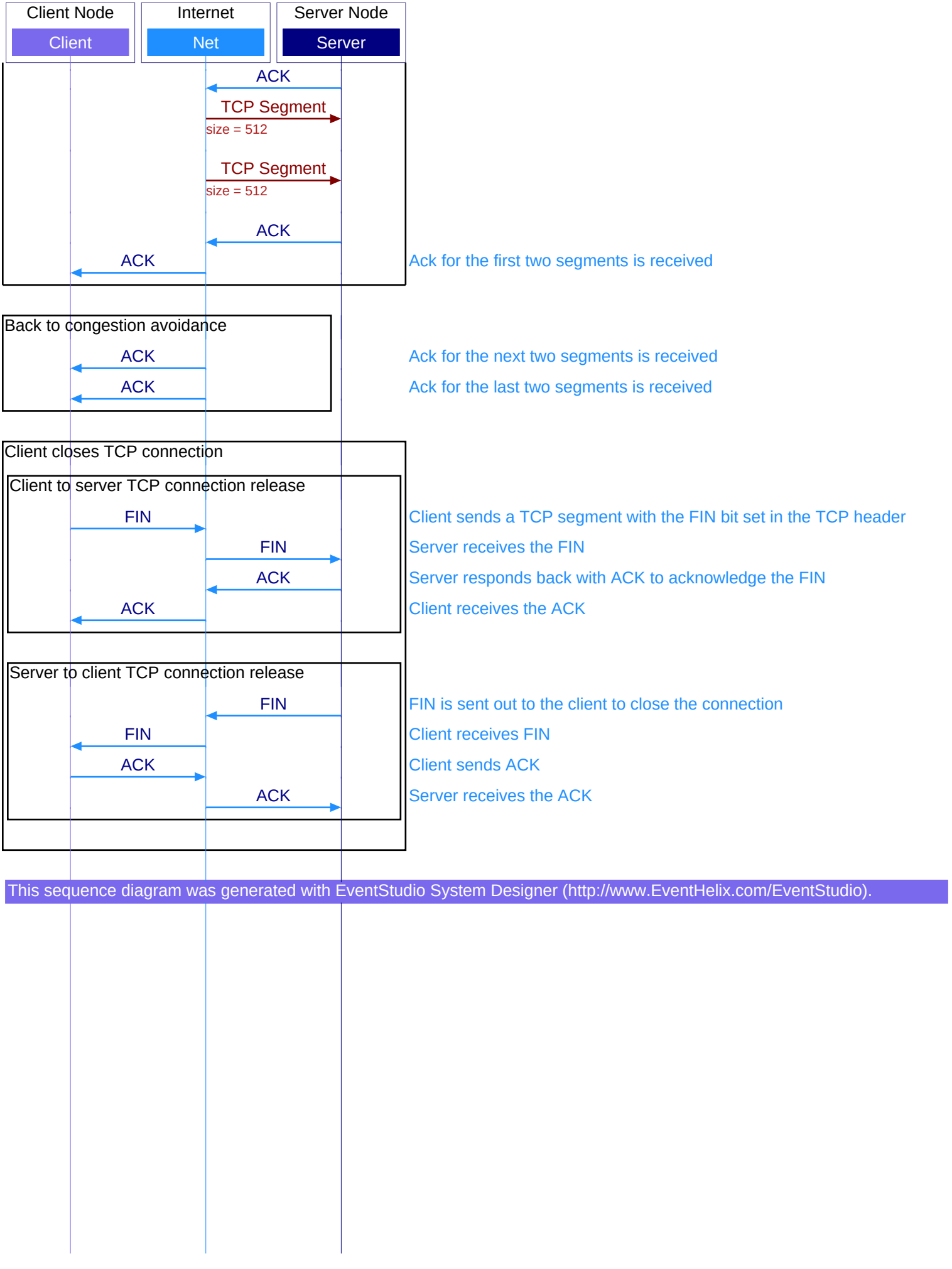

Client Node
Client
Internet
Net
Server Node
Server
This sequence diagram was generated with EventStudio System Designer (http://www.EventHelix.com/EventStudio).
We have already seen that TCP connection starts up in slow start mode, geometrically increasing the congestion window (cwnd) until it crosses 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 awaits client socket connections.
Client initiated three way handshake to establish a TCP connection
SYN
src = Client_Port,
dst = Server_Port,
seq_num = 0
Client sets the SYN bit in the TCP header to request a TCP connection. The sequence number field is set to 0. Since the SYN bit is set, this sequence number is used as the initial sequence number
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+ACK
src = Server_Port,
dst = Client_Port,
seq_num = 100,
ack_num = 1,
window = 65535
Client receives the "SYN+ACK" TCP segment
ACK
src = Client_Port,
dst = Server_Port,
ack_num = 101,
window = 5000
Client now acknowledges the first segment, thus completing the three way handshake. The receive window is set to 5000. Ack sequence number is set to 101, this means that the next expected sequence number is 101.
ACK
src = Client_Port,
dst = Server_Port,
ack_num = 101,
window = 5000
Server receives the TCP ACK segment
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.
About to exit slow start


Client Node
Client
Internet
Net
Server Node
Server
TCP Segment
seq_num = 100000,
len = 512
TCP Segment
seq_num = 100512,
len = 512
TCP Segment
seq_num = 100000,
len = 512
TCP Segment
seq_num = 100512,
len = 512
ACK
ack_num = 101024,
window = 80000
ACK
ack_num = 101024,
window = 80000
Data is split into TCP Segments. The segments are sent over the Internet
Client acknowledges the last block and also signals an increase in receiver window to 80000
Congestion Avoidance
TCP Segment
seq_num = 101024,
len = 512
TCP Segment
seq_num = 101536,
len = 512
TCP Segment
seq_num = 101024,
len = 512
TCP Segment
seq_num = 101536,
len = 512
ACK
ack_num = 102048,
window = 80000
ACK
ack_num = 102048,
window = 80000
cwnd is incremented using the formula: cwnd = cwnd + (segment_size segment_size) / cwnd)
TCP Segment
seq_num = 102048,
len = 512
Client data is split into TCP segments
TCP session sends out the data as a single segment
A TCP segment is lost


Client Node
Client
Internet
Net
Server Node
Server
TCP Segment
seq_num = 102048,
len = 512
Some node in the Internet drops the TCP segment due to congestion
Back to slow start
TCP Segment
seq_num = 102048,
len = 512
TCP Segment
seq_num = 102048,
len = 512
ACK
ack_num = 102560
ACK
ack_num = 102560
TCP window continues to grow exponentially until it reaches the ssthresh (=32731) value
TCP Segment
size = 512
TCP Segment
size = 512
TCP Segment
size = 512
TCP Segment
size = 512
TCP Segment
size = 512
TCP Segment
size = 512
Six TCP segments are transmitted in the slow start mode
TCP Segment
size = 512
TCP Segment
size = 512
ACK
TCP Segment
size = 512
TCP Segment
size = 512


Client Node
Client
Internet
Net
Server Node
Server
ACK
TCP Segment
size = 512
TCP Segment
size = 512
ACK
ACK
Ack for the first two segments is received
Back to congestion avoidance
ACK
ACK
Ack for the next two segments is received
Ack for the last two segments is received
Client closes TCP connection
Client to server TCP connection release
FIN
FIN
ACK
ACK
Client sends a TCP segment with the FIN bit set in the TCP header
Server receives the FIN
Server responds back with ACK to acknowledge the FIN
Client receives the ACK
Server to client TCP connection release
FIN
FIN
ACK
ACK
FIN is sent out to the client to close the connection
Client receives FIN
Client sends ACK
Server receives the ACK
This sequence diagram was generated with EventStudio System Designer (http://www.EventHelix.com/EventStudio).