# TCP Connection Establishment and Termination

# Connection Establishment and Termination

## reference

[reference 1](1-doc/Understanding%20TCP%20Sequence%20and%20Acknowledgment%20Numbers%20-%20PacketLife.net.htm) [tcp\_example.cap](1-doc/TCP_example.cap)

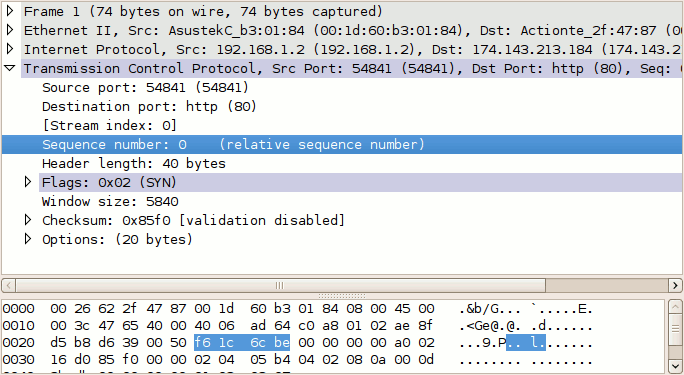
## Sequence and Acknowledgment Numbers

The client on either side of a TCP session maintains a 32-bit sequence number it uses to keep track of how much data it has sent. This sequence number is included on each transmitted packet, and acknowledged by the opposite host as an acknowledgement number to inform the sending host that the transmitted data was received successfully.

When a host initiates a TCP session, its initial sequence number is effectively random; it may be any value between 0 and 4,294,967,295, inclusive. *However, protocol analyzers like Wireshark will typically display*relative*sequence and acknowledgement numbers in place of the actual values.* These numbers are relative to the initial sequence number of that stream. This is handy, as it is much easier to keep track of relatively small, predictable numbers rather than the actual numbers sent on the wire.

### wireshark中显示的seq是相对值，真实值可以在ASCII decode中看到。

For example, the initial relative sequence number shown in packet #1 is 0 (naturally), while the ASCII decode in the third pane shows that the actual sequence number is 0xf61c6cbe, or 4129057982 decimal.

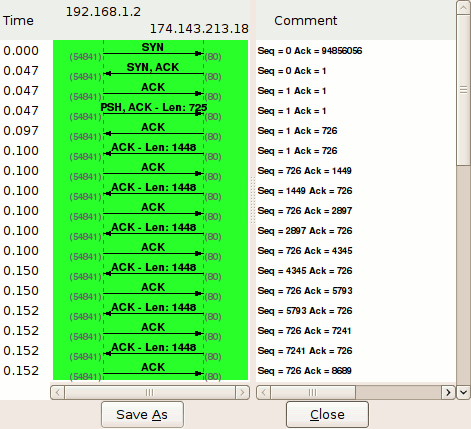


#### wireshark显示真实的sequence number。

Edit🡪Preference🡪protocols🡪TCP🡪uncheck Relative sequence numbers and window scaling

### 直观的查看TCP flow

To better understand how sequence and acknowledgement numbers are used throughout the duration of a TCP session, we can utilize Wireshark's built-in flow graphing ability. Navigate to **Statistics > Flow Graph...**, select **TCP flow** and click **OK**. Wireshark automatically builds a graphical summary of the TCP flow.



### Sequence number和Acknowledgement number指明流的位置

