TCP 三次握手分析 - 第 1 步

利用协议分析软件的输出，例如 Wireshark 的输出，您可以研究 TCP 三次握手的操作：

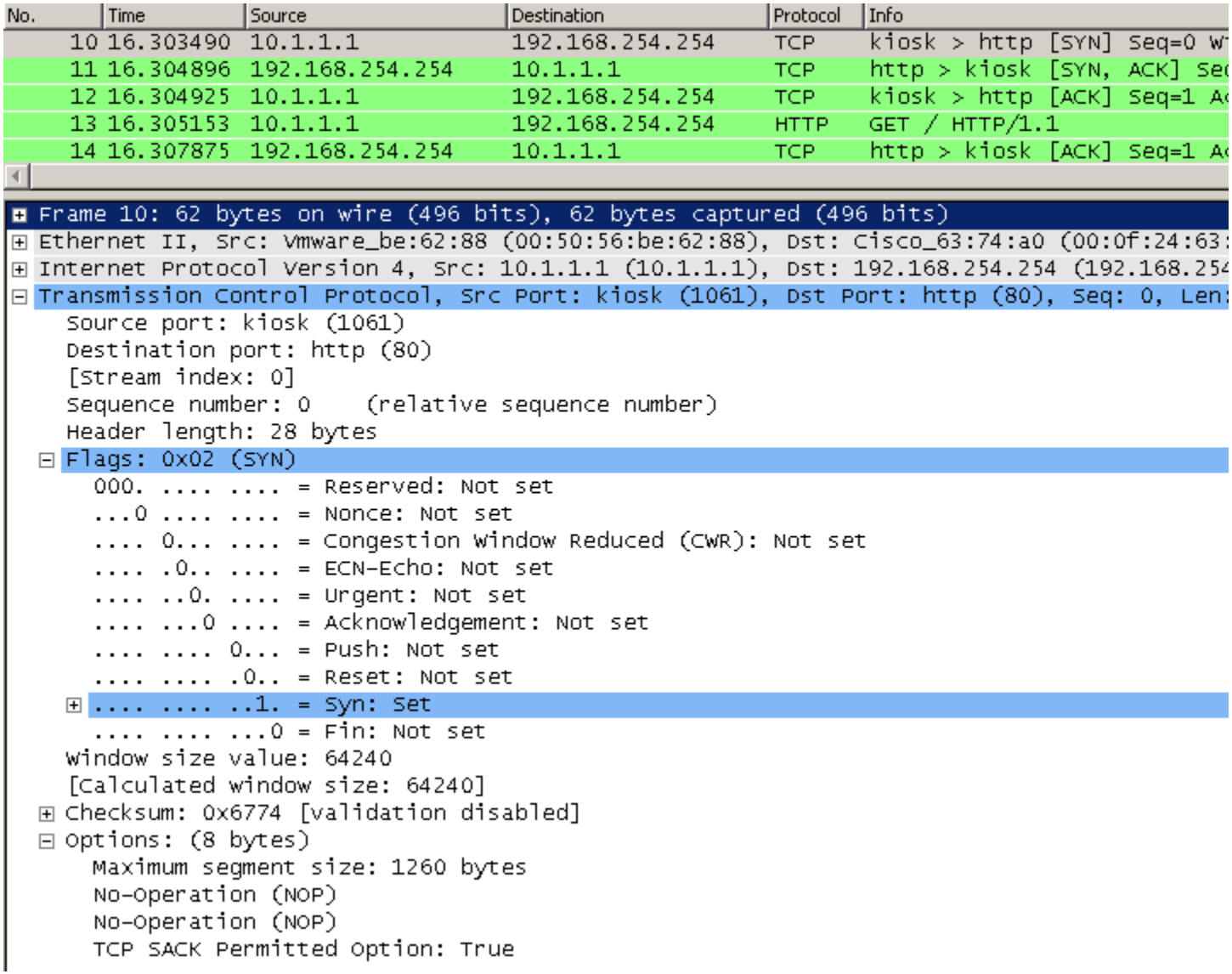
第 1 步：源客户端请求与服务器进行客户端-服务器通信会话。

TCP 客户端发送带同步序列号 (SYN) 控制标志设置的数据段，指示包含在报头中的序列号字段的初始值，用以开启三次握手。序列号的初始值称为初始序列号 (ISN)，由系统随机选取，并用于跟踪会话过程中从客户端到服务器的数据流。在会话过程中，每从客户端向服务器发送一个字节的数据，数据段报头中包含的 ISN 值就要加 1。

如图 1 所示，协议分析器的输出结果中显示了 SYN 控制标志和相应的序列号。

SYN 控制标志已设置，且相应的序列号为 0。尽管图中的协议分析器显示序列号和确认号的相应值，但真正的值是 32 位的二进制数。图中显示了以十六进制显示的四个字节。

图 1 - TCP 三次握手 (SYN)



TCP 三次握手分析 - 第 2 步

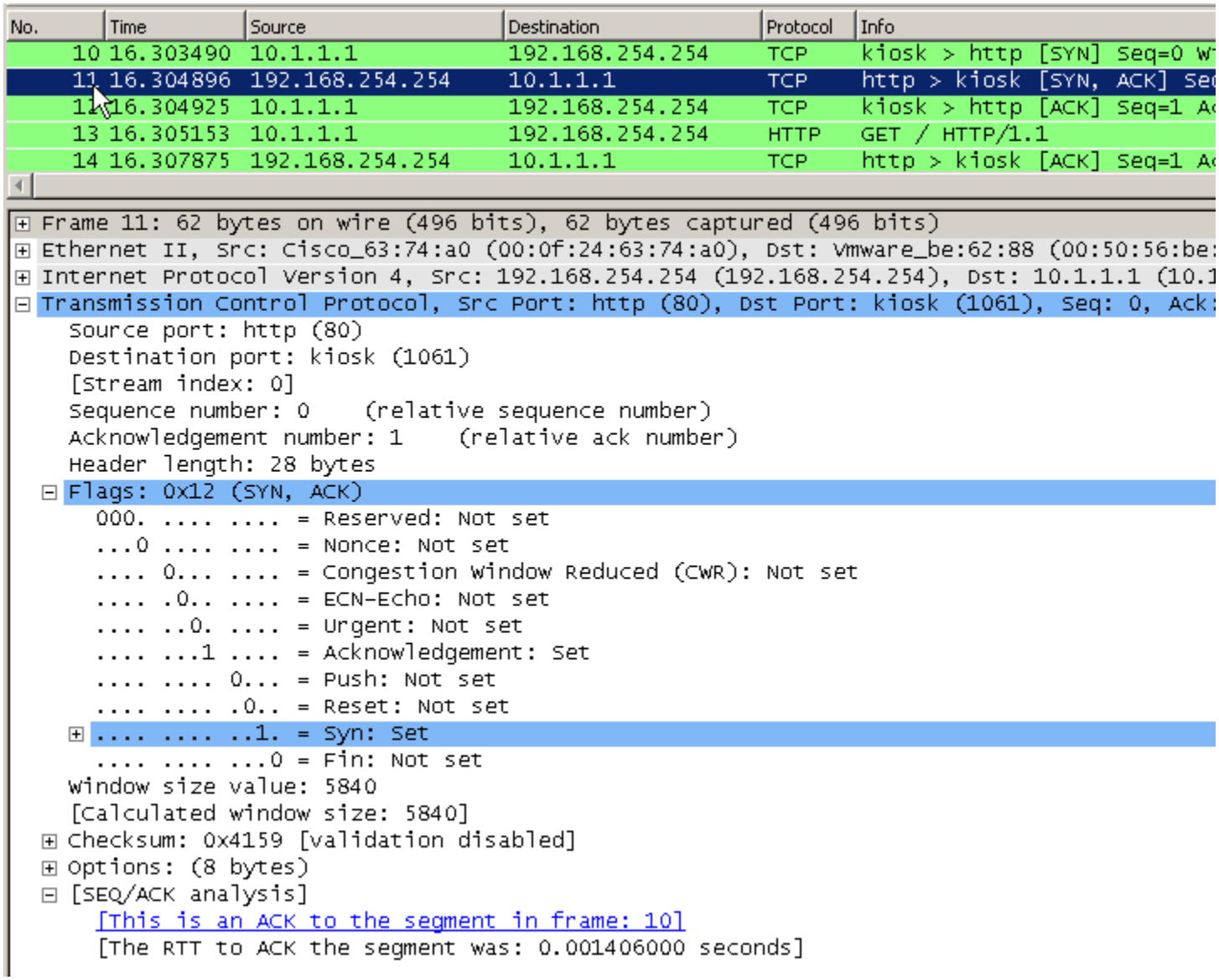
第 2 步：服务器确认客户端-服务器通信会话，并请求服务器-客户端通信会话。

TCP 服务器必须确认从客户端处收到 SYN 数据段，从而建立从客户端到服务器的会话。为了达到此目的，服务器应向客户端发送带确认 (ACK) 标志设置的数据段，表明确认号有效。客户端将这种带确认标志设置的数据段理解为确认信息，即服务器已收到从 TCP 客户端发出的 SYN 信息。

确认号字段的值等于 ISN 加 1。此时创建从客户端到服务器的会话。ACK 标志在会话期间保持设置。回想一下，客户端和服务器之间的会话实际上是由两个单向的会话组成的：一个是从客户端到服务器的会话，另一个则正好相反。在三次握手过程的第二步中，服务器必须发起到客户端的响应。为开启会话，服务器应采用与客户端同样的方法使用 SYN 标志。该操作设置报头中的 SYN 控制标志，从而建立从服务器到客户端的会话。SYN 标志表明序列号字段的初始值已包含在报头中，且该值用于跟踪会话过程中从服务器返回客户端的数据流。

如图 2 所示，协议分析器的输出结果中显示了 ACK 和 SYN 控制标志的设置，以及相应的序列号和确认号。

图 2 - TCP 三次握手 （SYN、ACK）



TCP 三次握手分析 - 第 3 步

第 3 步：源客户端确认服务器-客户端通信会话。

最后，TCP 客户端发送包含 ACK 信息的数据段，以示对服务器发送的 TCP SYN 信息的响应。在该数据段中，不包括用户数据。确认号字段的值比从服务器接收的 ISN 值大 1。一旦在客户端和服务器之间建立了双向会话，该通信过程中交换的所有数据段都将包含 ACK 标志设置。

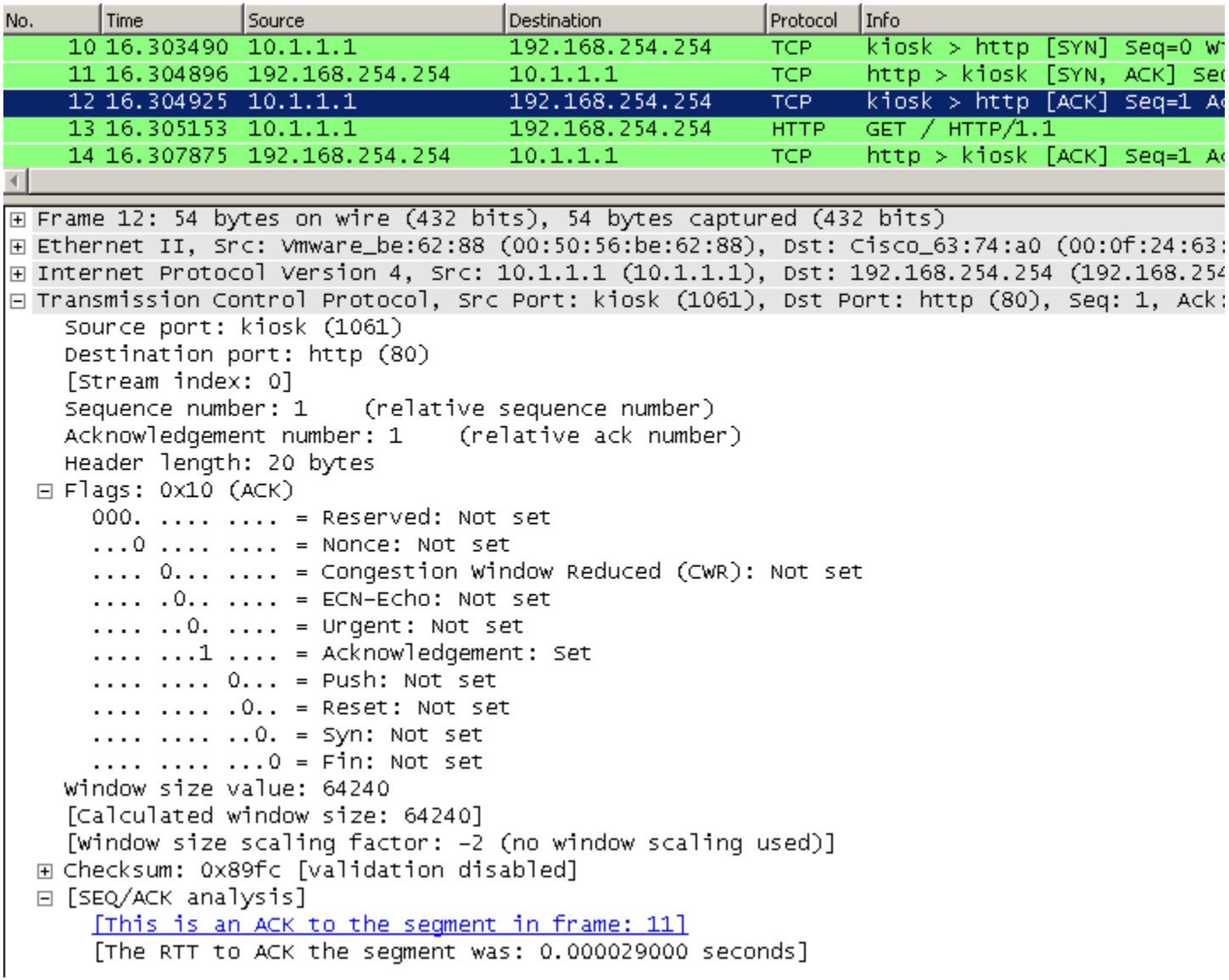
如图 3 所示，协议分析器的输出结果中显示了 ACK 控制标志，以及相应的序列号和确认号。

通过以下方式，可以加强数据网络的安全性：

- 拒绝建立 TCP 会话；
- 只允许建立特定服务的会话；
- 只允许已建立会话之间的通信。

以上安全策略可以应用于所有 TCP 会话，也可以仅应用于某些选定会话。

图 3 - TCP 三次握手 (ACK)



TCP 会话终止分析

若要关闭连接，数据段报头必须设置完成 (FIN) 控制标志。为终止每个单向 TCP 会话，需采用包含 FIN 数据段和 ACK 数据段的二次握手。因此，若要终止 TCP 支持的整个会话过程，需要实施四次交换，以终止两个双向会话，如图 1 所示。

注意：在本部分中，为了更容易理解，采用了客户端和服务器这两个术语进行说明。实际上，终止的过程可以在任意两台具有开放会话的主机之间展开：

第 1 步：当客户端的数据流中没有其他要发送的数据时，它将发送带 FIN 标志设置的数据段；

第 2 步：服务器发送 ACK 信息，确认收到从客户端发出的请求终止会话的 FIN 信息；

第 3 步：服务器向客户端发送 FIN 信息，终止从服务器到客户端的会话；

第 4 步：客户端发送 ACK 响应信息，确认收到从服务器发出的 FIN 信息。

当客户端没有其他要传输的数据时，它将在数据段报头中设置 FIN 标志。然后，会话中的服务器端发送包含 ACK 标志设置的一般数据段信息，通过确认号确认已经收到所有数据。当所有数据段得到确认后，会话关闭。

另一方向的会话采用相同的方式关闭。接收方在数据段的报头中设置 FIN 标志，然后发送到发送方，表明没有其他需要发送的数据。返回的确认信息确定已接收所有数据，随即该方向的会话关闭。

请参阅图 4 和图 5，查看数据段报头中的 FIN 和 ACK 控制标志，从而关闭 HTTP 会话。

也可以通过三次握手方式关闭连接。当客户端没有其他要传输的数据时，它将向服务器发送 FIN 信息。如果服务器也没有其他要传输的数据，它将发送同时包含 FIN 和 ACK 标志设置的响应信息，将两步并作一步。最后，客户端返回 ACK 信息。

图 4 - TCP 会话终止 (FIN)

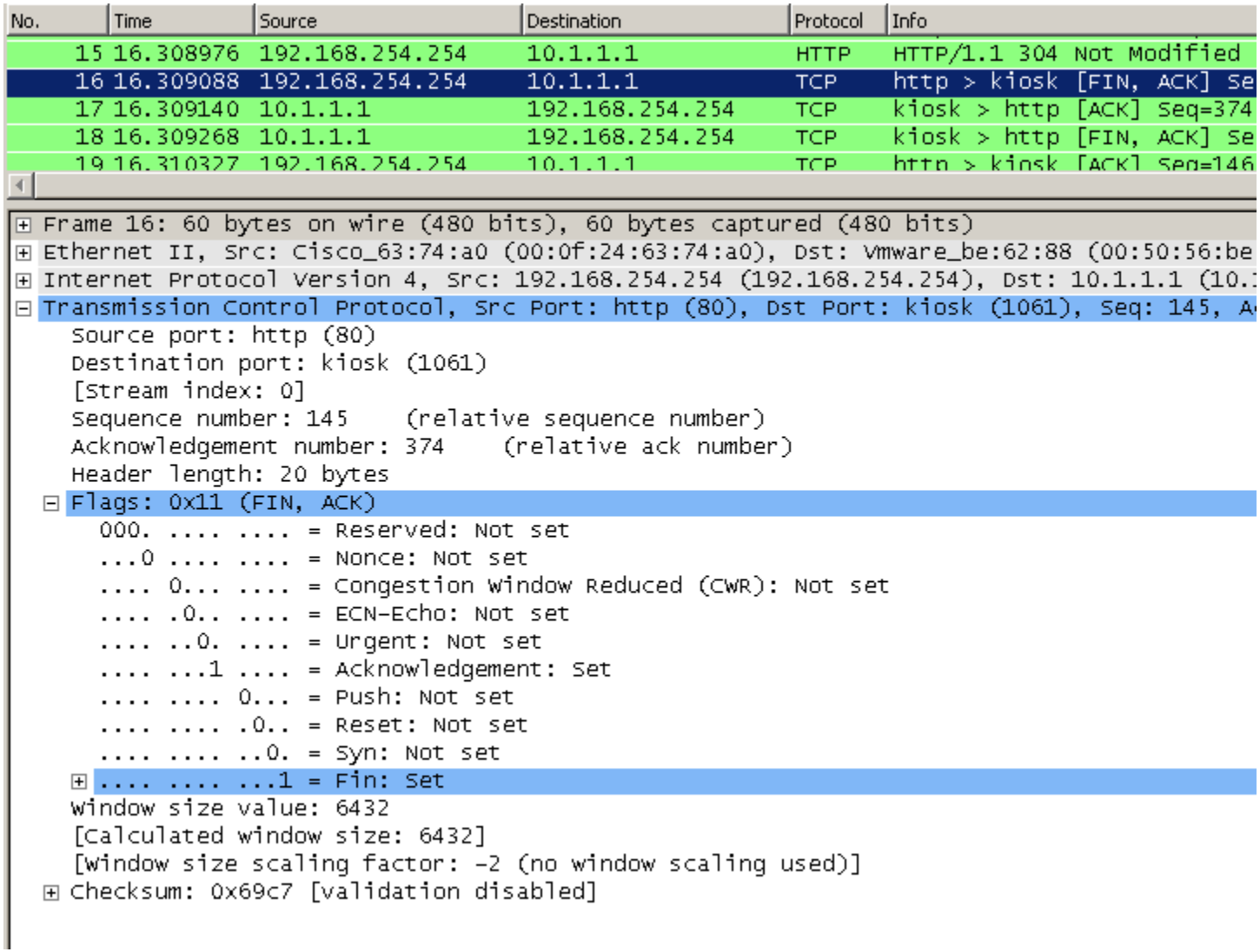
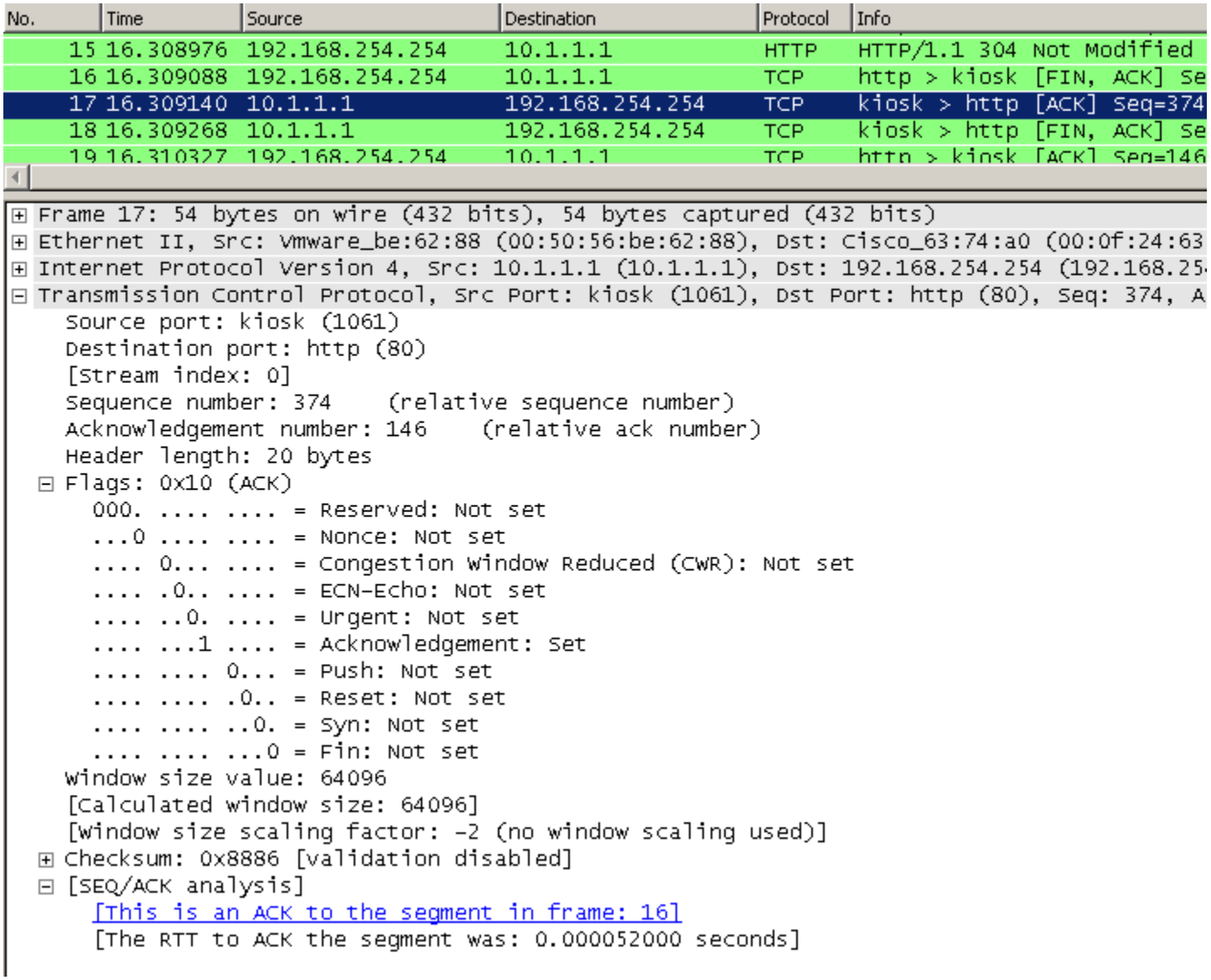


图 5 - TCP 会话终止 (ACK)



No.	Time	Source	Destination	Protocol	Info
15	16.308976	192.168.254.254	10.1.1.1	HTTP	HTTP/1.1 304 Not Modified
16	16.309088	192.168.254.254	10.1.1.1	TCP	http > kiosk [FIN, ACK] Se
17	16.309140	10.1.1.1	192.168.254.254	TCP	kiosk > http [ACK] Seq=374
18	16.309268	10.1.1.1	192.168.254.254	TCP	kiosk > http [FIN, ACK] Se
19	16.310327	192.168.254.254	10.1.1.1	TCP	http > kiosk [ACK] Seq=146

```
⊞ Frame 17: 54 bytes on wire (432 bits), 54 bytes captured (432 bits)
⊞ Ethernet II, Src: Vmware_be:62:88 (00:50:56:be:62:88), Dst: Cisco_63:74:a0 (00:0f:24:63
⊞ Internet Protocol Version 4, Src: 10.1.1.1 (10.1.1.1), Dst: 192.168.254.254 (192.168.25
⊟ Transmission Control Protocol, Src Port: kiosk (1061), Dst Port: http (80), Seq: 374, A
    Source port: kiosk (1061)
    Destination port: http (80)
    [Stream index: 0]
    Sequence number: 374    (relative sequence number)
    Acknowledgement number: 146    (relative ack number)
    Header length: 20 bytes
  ⊟ Flags: 0x10 (ACK)
      000. .... .... = Reserved: Not set
      ...0 .... .... = Nonce: Not set
      .... 0... .... = Congestion Window Reduced (CWR): Not set
      .... .0.. .... = ECN-Echo: Not set
      .... ..0. .... = Urgent: Not set
      .... ...1 .... = Acknowledgement: Set
      .... .... 0... = Push: Not set
      .... .... .0.. = Reset: Not set
      .... .... ..0. = Syn: Not set
      .... .... ...0 = Fin: Not set
    Window size value: 64096
    [Calculated window size: 64096]
    [Window size scaling factor: -2 (no window scaling used)]
  ⊞ Checksum: 0x8886 [validation disabled]
  ⊟ [SEQ/ACK analysis]
      [This is an ACK to the segment in frame: 16]
      [The RTT to ACK the segment was: 0.000052000 seconds]
```