Analyse de la connexion TCP en trois étapes - Étape 1

En utilisant les informations du logiciel d'analyse de protocoles, par exemple les résultats de Wireshark, vous pouvez examiner le fonctionnement de la connexion TCP en trois étapes :

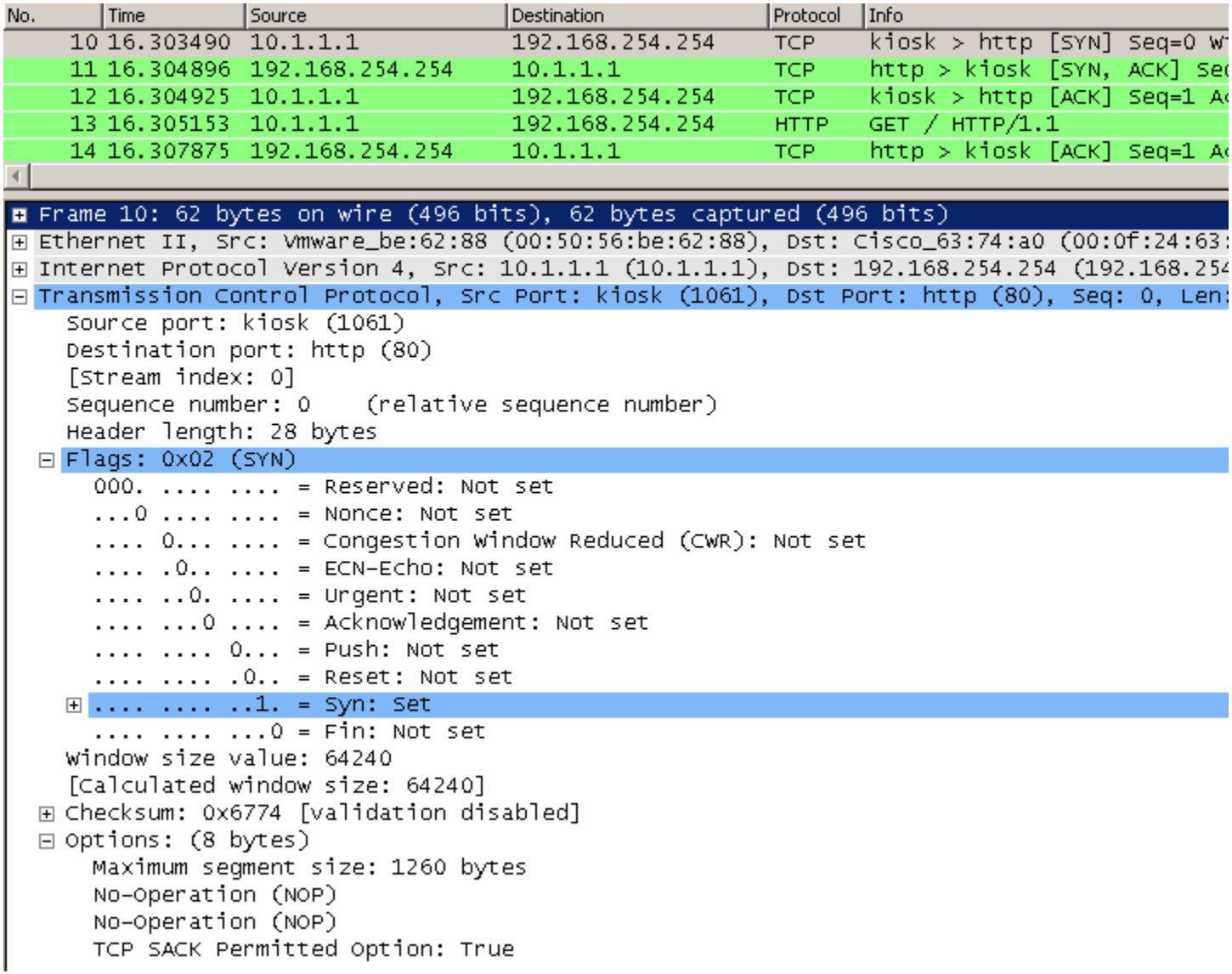
Étape 1 : Le client demande l'établissement d'une session de communication client-serveur avec le serveur.

Un client TCP initie une connexion en trois étapes en envoyant un segment contenant l'indicateur de contrôle SYN qui indique une valeur initiale dans le champ de numéro d'ordre de l'en-tête. Cette valeur initiale du numéro d'ordre, appelée ISN (Initial Sequence Number), est choisie de façon aléatoire et sert à commencer le suivi du flux de données entre le client et le serveur pour cette session. L'ISN figurant dans l'en-tête de chaque segment est incrémenté de un pour chaque octet de données envoyé par le client au serveur tandis que la conversation de données se poursuit.

Comme l'illustre la figure 1, le résultat d'un analyseur de protocole affiche l'indicateur de contrôle SYN et le numéro d'ordre relatif.

L'indicateur de contrôle SYN est défini et le numéro d'ordre relatif est égal à 0. Bien que l'analyseur de protocole dans le graphique présente les valeurs relatives des numéros d'ordre et d'accusé de réception, les valeurs réelles sont des nombres binaires de 32 bits. La figure illustre les quatre octets au format hexadécimal.

Figure 1 - Connexion TCP en trois étapes (SYN)



No.	Time	Source	Destination	Protocol	Info
10	16.303490	10.1.1.1	192.168.254.254	TCP	kiosk > http [SYN] Seq=0 W
11	16.304896	192.168.254.254	10.1.1.1	TCP	http > kiosk [SYN, ACK] Se
12	16.304925	10.1.1.1	192.168.254.254	TCP	kiosk > http [ACK] Seq=1 A
13	16.305153	10.1.1.1	192.168.254.254	HTTP	GET / HTTP/1.1
14	16.307875	192.168.254.254	10.1.1.1	TCP	http > kiosk [ACK] Seq=1 A

```
⊞ Frame 10: 62 bytes on wire (496 bits), 62 bytes captured (496 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 (192.168.254
⊟ Transmission Control Protocol, Src Port: kiosk (1061), Dst Port: http (80), Seq: 0, Len:
    Source port: kiosk (1061)
    Destination port: http (80)
    [Stream index: 0]
    Sequence number: 0    (relative sequence number)
    Header length: 28 bytes
  ⊟ Flags: 0x02 (SYN)
      000. .... .... = Reserved: Not set
      ...0 .... .... = Nonce: Not set
      .... 0... .... = Congestion Window Reduced (CWR): Not set
      .... .0.. .... = ECN-Echo: Not set
      .... ..0. .... = Urgent: Not set
      .... ...0 .... = Acknowledgement: Not set
      .... .... 0... = Push: Not set
      .... .... .0.. = Reset: Not set
    ⊞ .... .... ..1. = Syn: Set
      .... .... ...0 = Fin: Not set
    Window size value: 64240
    [Calculated window size: 64240]
  ⊞ Checksum: 0x6774 [validation disabled]
  ⊟ Options: (8 bytes)
      Maximum segment size: 1260 bytes
      No-Operation (NOP)
      No-Operation (NOP)
      TCP SACK Permitted Option: True
```

Analyse de la connexion TCP en trois étapes - Étape 2

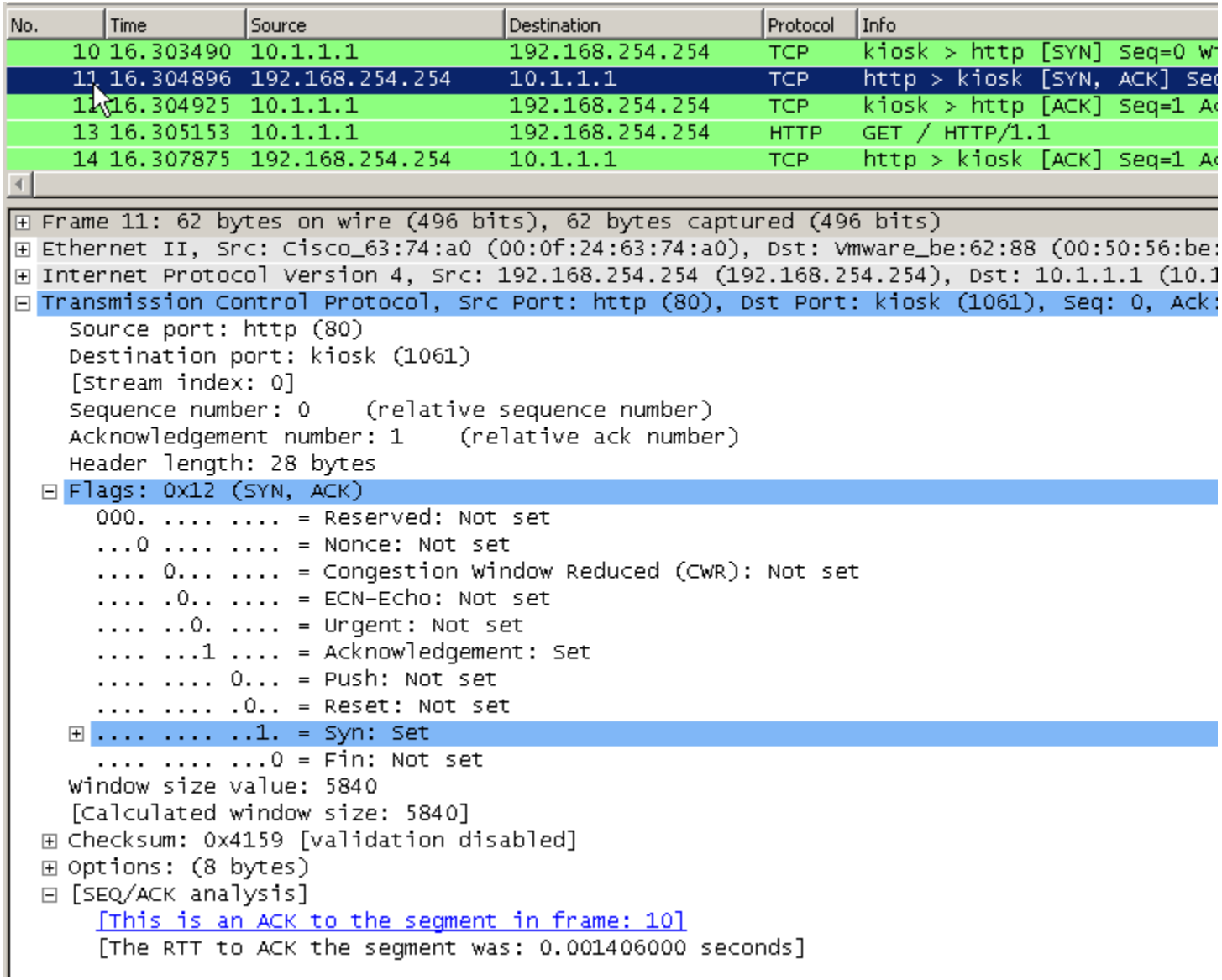
Étape 2 : Le serveur accuse réception de la session de communication client-serveur et demande l'établissement d'une session de communication serveur-client.

Le serveur TCP doit accuser réception du segment SYN provenant du client pour établir la session du client vers le serveur. Pour cela, le serveur renvoie au client un segment accompagné de l'indicateur ACK indiquant que le numéro d'accusé de réception est valide. Grâce à cet indicateur présent dans le segment, le client identifie cela comme un accusé de réception indiquant que le serveur a reçu le SYN du client TCP.

La valeur du champ de numéro d'accusé de réception est égale à l'ISN + 1. Cela établit une session du client au serveur. L'indicateur ACK demeure défini pour le reste de la session. Souvenez-vous que la communication entre le client et le serveur est composée de deux sessions unidirectionnelles : une allant du client vers le serveur et l'autre du serveur vers le client. Dans le cadre de la deuxième étape de la connexion en trois étapes, le serveur doit déclencher la réponse au client. Pour lancer cette session, le serveur utilise l'indicateur SYN comme le client l'a fait. Il inclut l'indicateur de contrôle SYN dans l'en-tête pour établir une session du serveur vers le client. L'indicateur SYN précise que la valeur initiale du champ de numéro d'ordre se trouve dans l'en-tête. Cette valeur sert à effectuer le suivi du flux de données dans cette session, du serveur vers le client.

Comme l'illustre la figure 2, les résultats de l'analyseur de protocole montrent que les indicateurs de contrôle ACK et SYN sont définis et les numéros d'ordre relatifs et d'accusé de réception sont affichés.

Figure 2 - Connexion TCP en trois étapes (SYN, ACK)



Analyse de la connexion TCP en trois étapes - Étape 3

Étape 3 : Le client accuse réception de la session de communication serveur-client.

Enfin, le client TCP répond à l'aide d'un segment contenant un ACK qui constitue la réponse au SYN TCP envoyé par le serveur. Ce segment ne contient pas de données de l'utilisateur. La valeur du champ de numéro d'accusé de réception est supérieure de 1 au numéro d'ordre initial reçu du serveur. Quand les deux sessions sont établies entre le client et le serveur, tous les segments supplémentaires échangés dans cette communication comportent l'indicateur ACK défini.

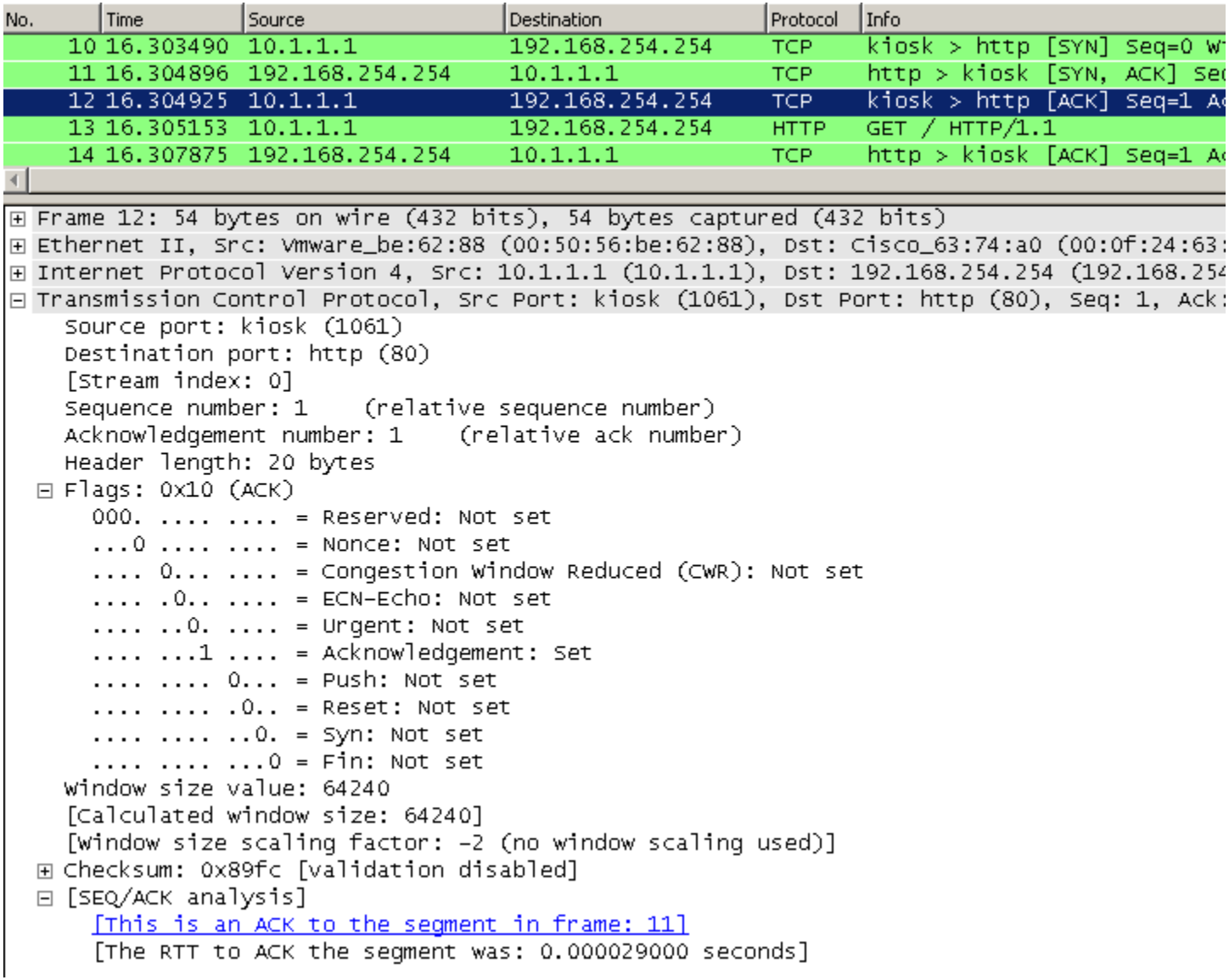
Comme l'illustre la figure 3, les résultats de l'analyseur de protocole montrent l'indicateur de contrôle ACK défini et les numéros d'ordre relatifs et d'accusé de réception.

Il est possible de sécuriser le réseau de données en :

- Refusant l'établissement de sessions TCP ;
- Autorisant uniquement l'établissement de sessions pour des services spécifiques ;
- Autorisant uniquement le trafic faisant déjà partie de sessions établies.

Ces mesures de sécurité peuvent être implémentées pour toutes les sessions TCP ou uniquement pour certaines sessions.

Figure 3 - Connexion TCP en trois étapes (ACK)



No.	Time	Source	Destination	Protocol	Info
10	16.303490	10.1.1.1	192.168.254.254	TCP	kiosk > http [SYN] Seq=0 W
11	16.304896	192.168.254.254	10.1.1.1	TCP	http > kiosk [SYN, ACK] Se
12	16.304925	10.1.1.1	192.168.254.254	TCP	kiosk > http [ACK] Seq=1 A
13	16.305153	10.1.1.1	192.168.254.254	HTTP	GET / HTTP/1.1
14	16.307875	192.168.254.254	10.1.1.1	TCP	http > kiosk [ACK] Seq=1 A

```
⊞ Frame 12: 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 (192.168.254
⊟ Transmission Control Protocol, Src Port: kiosk (1061), Dst Port: http (80), Seq: 1, Ack:
    Source port: kiosk (1061)
    Destination port: http (80)
    [Stream index: 0]
    Sequence number: 1    (relative sequence number)
    Acknowledgement number: 1    (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: 64240
    [Calculated window size: 64240]
    [Window size scaling factor: -2 (no window scaling used)]
  ⊞ Checksum: 0x89fc [validation disabled]
  ⊟ [SEQ/ACK analysis]
      [This is an ACK to the segment in frame: 11]
      [The RTT to ACK the segment was: 0.000029000 seconds]
```

Analyse de l'interruption d'une session TCP

Pour mettre fin à une connexion, l'indicateur de contrôle FIN (Finish) doit être défini dans l'en-tête de segment. Pour mettre fin à chaque session TCP unidirectionnelle, on utilise un échange en deux étapes, constitué d'un segment FIN et d'un segment ACK. Par conséquent, pour mettre fin à une seule conversation TCP, quatre échanges sont nécessaires pour mettre fin aux deux sessions (voir la Figure 1).

Remarque : Les termes client et serveur sont utilisés ici pour simplifier l'explication, mais le processus d'interruption peut être initié par n'importe lequel des deux hôtes ayant une session ouverte :

Étape 1 : Quand le client n'a plus de données à envoyer dans le flux, il envoie un segment dont l'indicateur FIN est défini.

Étape 2 : Le serveur envoie un segment ACK pour informer de la bonne réception du segment FIN afin de fermer la session du client au serveur.

Étape 3 : Le serveur envoie un segment FIN au client pour mettre fin à la session du serveur au client.

Étape 4 : Le client répond à l'aide d'un segment ACK pour accuser réception du segment FIN envoyé par le serveur.

Quand le client n'a plus aucune donnée à transférer, il définit l'indicateur FIN dans l'en-tête d'un segment. Ensuite, le serveur de la connexion envoie un segment normal contenant des données dont l'indicateur ACK est défini en utilisant le numéro d'accusé de réception, confirmant ainsi que tous les octets de données ont été reçus. Quand la réception de tous les segments a été confirmée, la session est fermée.

La session dans l'autre sens est fermée selon le même processus. Le récepteur indique qu'il n'y a plus de données à envoyer en définissant l'indicateur FIN dans l'en-tête d'un segment envoyé à la source. Un accusé de réception confirme que tous les octets de données ont été reçus et que cette session, à son tour, se ferme.

Reportez-vous aux figures 4 et 5 pour voir les indicateurs de contrôle FIN et ACK définis dans l'en-tête de segment, et qui permettent ainsi l'interruption d'une session HTTP.

Il est également possible de fermer la connexion à l'aide d'une connexion en trois étapes. Quand le client n'a plus de données à envoyer, il envoie un segment FIN au serveur. Si le serveur n'a plus de données à envoyer, il peut répondre en définissant les indicateurs FIN et ACK simultanément et en combinant ainsi deux étapes en une. Le client répond par un segment ACK.

Figure 4 - Fin de session TCP (FIN)

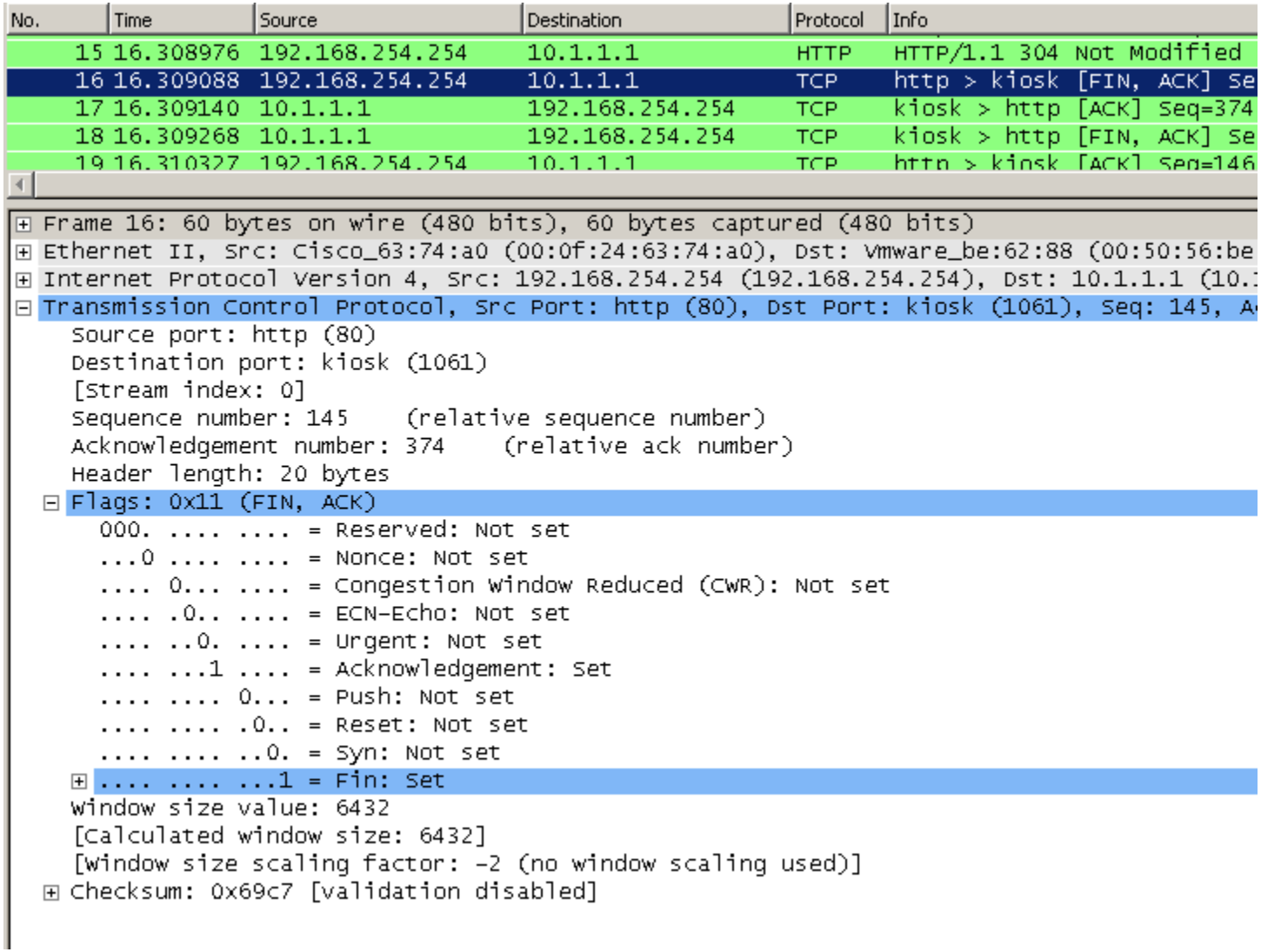
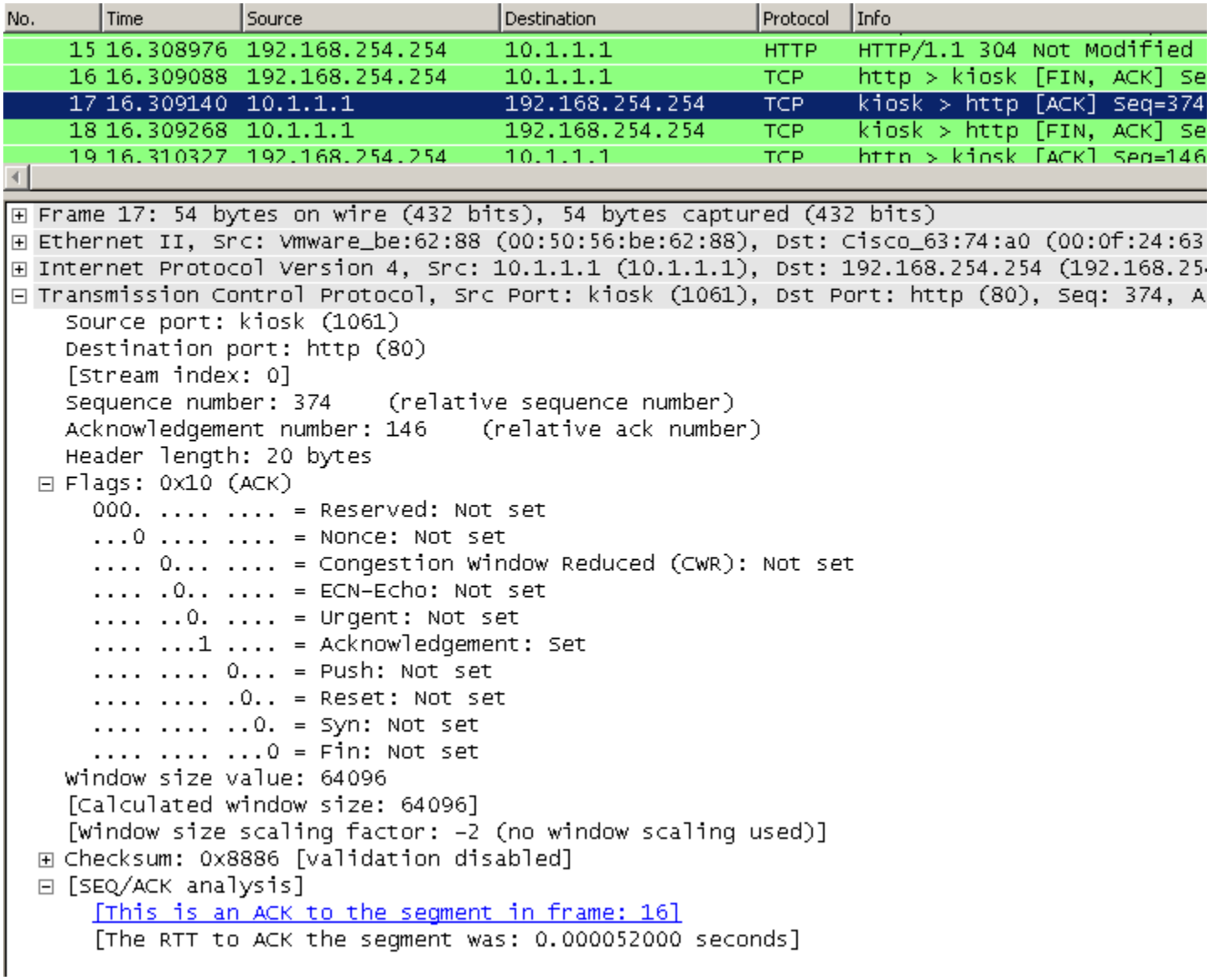


Figure 5 - Fin de session TCP (ACK)



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]
```