TCP & UDP

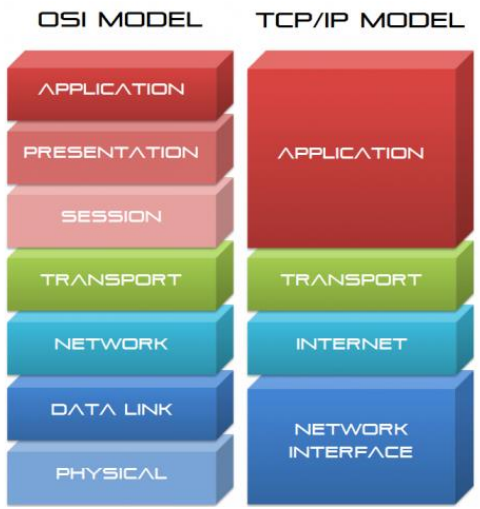
Data Transferring Protocols

Index

- היכרות עם פרוטוקולי העברת נתונים.
- אופן פעולת הפרוטוקולים.
- תהליך 3-Way Handshake
- פורטים לוגיים בשימוש הפרוטוקולים.
- מבנה ה-Header של הפרוטוקולים.

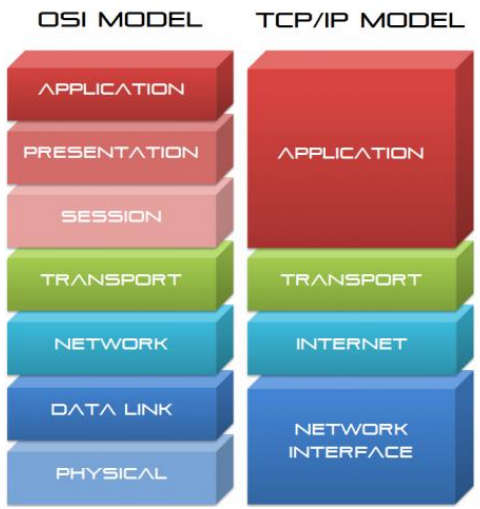
TCP Profile

- שם מלא: Transmission Control Protocol
- תכלית הפרוטוקול: פרוטוקול העברת נתונים אשר מבטיח העברת נתונים אמינה. TCP משיג זאת בכך שהוא דואג לוודא שכל חבילת מידע שנשלחה, התקבלה ע"י היעד. זאת אומרת שכל חבילת מידע שאובדת או נפגמת, נשלחת שוב. TCP דואג בנוסף למספר את חבילות המידע ובכך לעזור ליעד להרכיב אותן בסדר הנכון, ולא בסדר הגעתן.
- במילה אחת אמין ולא מהיר.
- פועל בשכבת ה-Transport.



UDP Profile

- שם מלא: User Datagram Protocol
- תכלית הפרוטוקול: פרוטוקול העברת נתונים שמבטיח העברת נתונים בקצב גבוה, אבל לא העברת נתונים אמינה או שימור סדר כפי שקורה ב-TCP. חבילות המידע בשיטה זו עשויות להתקבל בסדר שונה, פגומות או לא להגיע כלל.
- במילה אחת מהיר ולא אמין.
- פועל בשכבת ה-Transport.



TCP & UDP

- כולם כבר צריכים לדעת שהעברת נתונים לא משנה מאיזה סוג, הכרחית להגשמת קונספט האינטרנט ולשיתוף משאבים ומידע. TCP ו-UDP אומנם שונים אך בעזרת שילוב שני שיטות אלו. אנו יכולים לקיים תקשורת תקינה על פני כל רשת.
- מאפייני הפרוטוקולים:

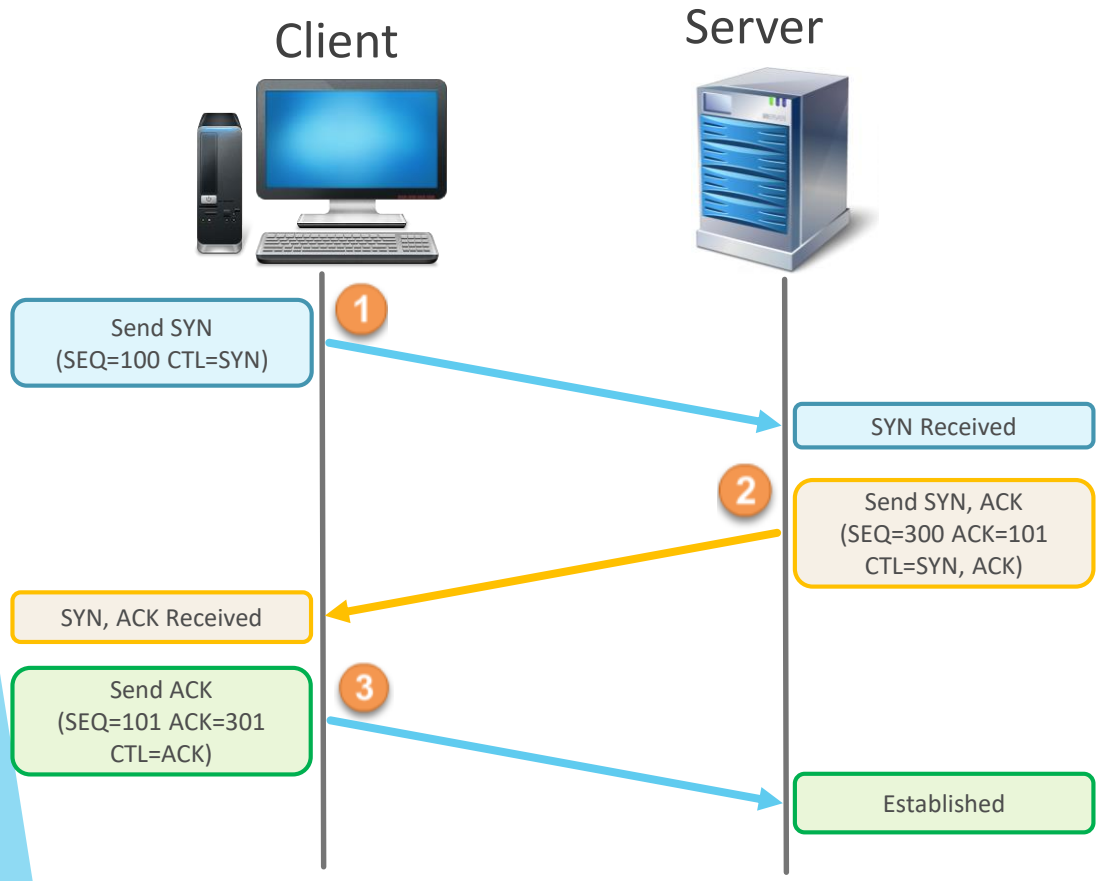
UDP	TCP
Connectionless	Establishing a Session
Unreliable Delivery	Reliable Delivery
No Ordered Data Reconstruction	Same-Order Delivery
No Flow Control	Flow Control

- הסבר מלא על ההבדלים כאן.
- UDP-RFC:768 TCP-RFC:793
- הידעת! ישנם פרוטוקולים נוספים להעברת נתונים: SCTP, RTP ו-DCCP אך הם מחוץ לטווח קורס זה.

How Does TCP work?

- ההבדל העיקרי בין TCP ל-UDP הוא אמינות (Reliability). האמינות שלו נובעת מהצורה בה הוא פועל ומעביר נתונים. המאפיין הראשון של TCP הוא יצירת חיבור (Session) עם היעד הרבה לפני תחילת שליחת הנתונים. לאחר יצירת החיבור TCP נמצא בבקרה מלאה (connection-oriented) על כל דבר שמתרחש בחיבור: איזה מידע התקבל, קביעת המהירות של העברת הנתונים, מיון הנתונים לשירותים/פורטים המתאימים.
- TCP משתמש במגוון רחב של הודעות בעזרתן הוא יוצר את החיבור (Establish Sessions), מסמן את סדר החבילות (Sequence), מאשר קבלתן של חבילות (Acknowledgment) ומסיים את החיבור (Terminate Session) לאחר שכל המידע התקבל.
- המנגנון/תהליך שאחראי ליצור את החיבור, נקרא The 3-Way Handshake תהליך זה ועוד פעולות כמו ניהול וסגירת החיבור. עושים שימוש בהודעות מיוחדות (Flags) אותם ניתן למצוא בשדה Code Bits ב-Segment Header. ההודעות השונות:
  - SYN-סנכרון (Synchronize) חיבור ורצף מספרים.
  - ACK-אישור. (Acknowledgment)
  - FIN-סיום. (Finish)
  - RST-איפוס חיבור. (Reset)
  - URG-דחיפות. (Urgent)
  - PSH-דחיפה. (Push)

TCP 3-Way Handshake



Wireshark Example

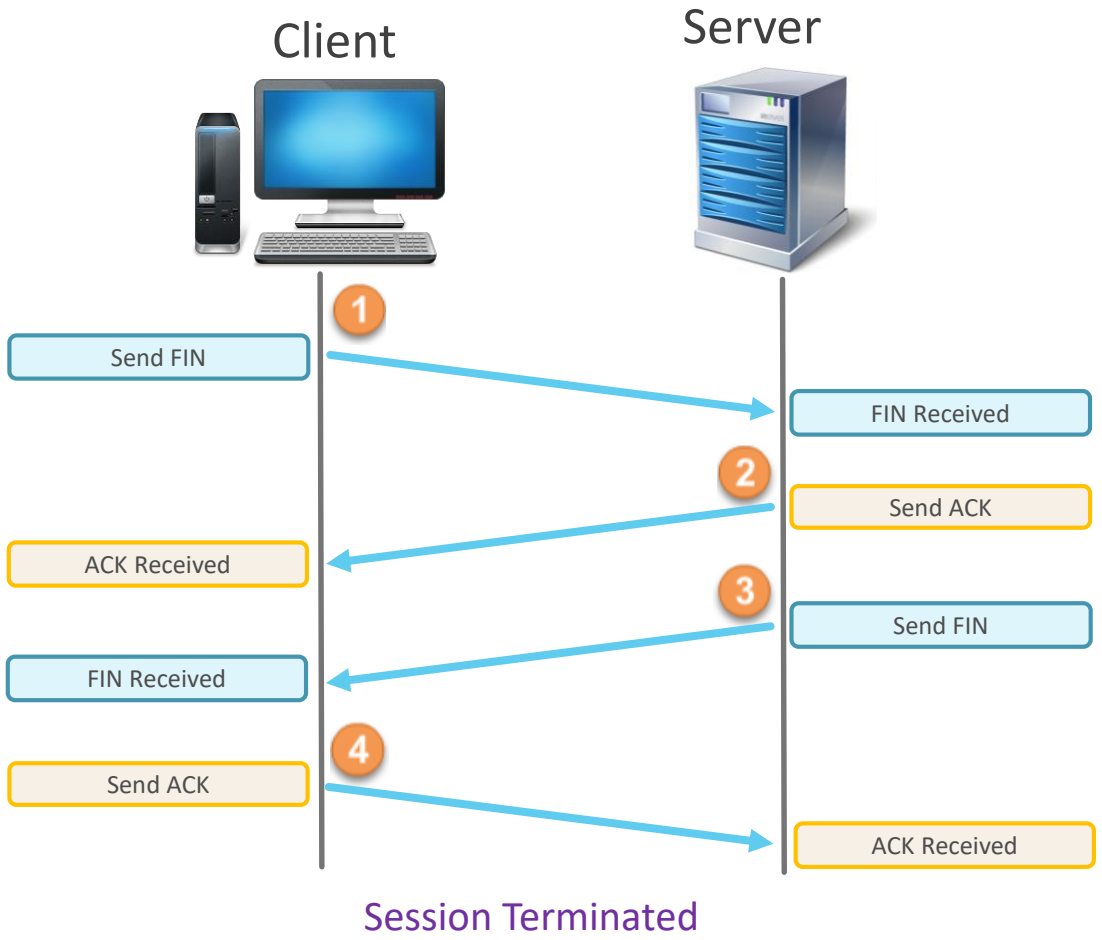
- איור זה מדגים את השלבים ליצירת חיבור:
- בתרבויות מסויימות, כששני אנשים נפגשים הם לוחצים ידיים. הפעולה של לחיצת ידיים מובנת לשני הצדדים, ומסמלת ברכה הדדית. חיבור בעזרת TCP הוא מעט דומה:

1. לחיצת יד ראשונה- התהליך מתחיל בכך שה-Client שולח בקשת התחברות "Client-to-Sever" בעזרת SYN Flag. ל-Segment זה נוסף גם SEQ No ערך שנוצר רנדומלית ותפקידו למספר ולזהות את החבילות.
2. לחיצת היד השנייה- ה-Server מאשר את בקשת ה-Client בעזרת ACK Flag ומייצר בקשת התחברות משלו "Server-to-Client" בעזרת SYN Flag. בשלב זה ערך ACK no משמש לאישור קבלת המידע ו-SEQ no למספור וזיהוי חבילות המידע.
3. לחיצת יד שלישית- ה-Client מאשר את בקשת ה-Server ע"י ACK Flag והחיבור נוצר (Established Session).

- לאחר שלבים אלו, שליחת המידע מתחילה.

Session Termination

- איור זה מדגים את השלבים לסגירת/סיום החיבור לאחר סיום העברת הנתונים:

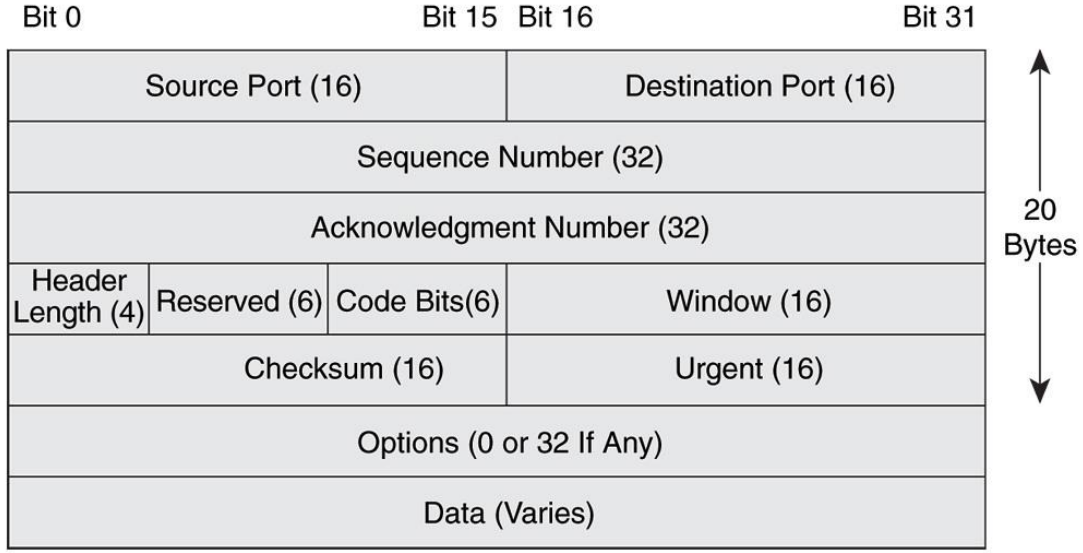


1. ברגע של-Client אין יותר מידע לשלוח, הוא שולח בקשה לסגירת החיבור בעזרת Segment עם FIN Flag.
2. ה-Server שולח הודעת ACK כדי להודיע ולאשר ל-Client על קבלת בקשתו לסגור את החיבור.
3. ה-Server שולח Segment עם FIN Flag כדי לסיים את החיבור מהצד שלו.
4. ה-Client מאשר את בקשת ה-Server בעזרת ACK Flag.

- בתום תהליך זה, החיבור (Session) נסגר.

TCP Header

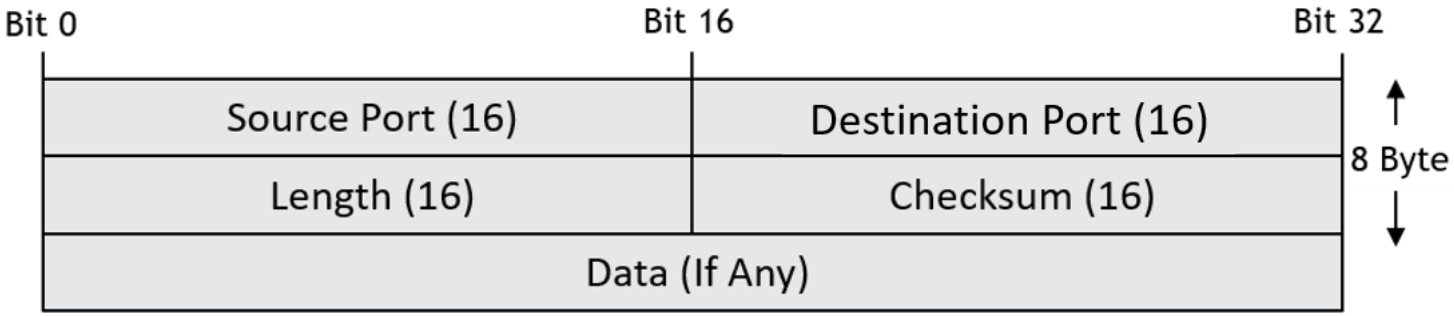
- איור המציג את מבנה ה-Header של פרוטוקול TCP, אשר נוסף לחבילת המידע בתהליך האינקפסולציה. מה שיוצר את ה-Segment.



- משמעות השדות (Fields) ב-Header:
  - Source Port-מספר הפורט של המקור.
  - Destination Port-מספר הפורט של היעד.
  - Sequence Number-משמש להרכבת המידע מחדש.
  - Ack Number-מציין את המידע שהתקבל.
  - Header length-מציין את גודל ה-Header.
  - Reserved-שמור לשימוש עתידי.
  - Control bits-מכיל קוד או סימון של מטרת ותפקיד ה-Segment.
  - Window size-מציין את מספר ה-Segments אשר ניתן לקבל בבת אחת.
  - Checksum-משמש לגילוי שגיאות.
  - Urgent-מציין את רמת הדחיפות.
  - Data-המידע.

UDP Header

- איור המציג את מבנה ה-Header של פרוטוקול UDP, אשר נוסף לחבילת המידע בתהליך האינקפסולציה. מה שיוצר את ה-Segment.



- משמעות השדות (Fields) ב-Header:
  - Source Port-מספר הפורט של המקור.
  - Destination Port-מספר הפורט של היעד.
  - Header length-מציין את גודל ה-Header.
  - Checksum-משמש לגילוי שגיאות.

TCP or UDP ?

- TCP או UDP זו השאלה?
- TCP הוא פרוטוקול העברת נתונים המתאים ליישומים או שירותים להם אמינות ושלמות המידע חשובה מאוד, אבל סובלים עיכובים (Daley) או קצב העברה נמוך. כמו:

SMTP SSH FTP http://

- UDP הוא פרוטוקול העברת נתונים המתאים ליישומים או שירותים להם אמינות ושלמות המידע פחות חשובה, אבל לא סובלים עיכובים וזקוקים לקצב העברה גבוה. כמו:

voip SNMP DNS DHCP

TCP & UDP Port Addressing

- TCP ו-UDP מצליחים ליצור חיבורים שונים (Sessions) ולנהל העברת מידע שונה בין חיבור לחיבור ע"י זיהוי החיבורים בעזרת מספרי פורטים, לדוג' גלישה באינטרנט ומשחק רשת במקביל.
- קיימים 65,535 פורטים לוגים, אשר מחלוקים ל-3 קבוצות:

0-1023	Well-Known Ports	פורטים מוכרים מאוד של שירותים/פרוטוקולים קריטיים לעולם הרשתות.
1024-49,151	Registered Ports	מספרי פורטים רשומים אפליקציות/פרוטוקולים הזקוקות גישה לאינטרנט.
49,152-65,535	Dynamin & Private Ports	פורטים רנדומליים שאינם שייכם לאף שירות/פרוטוקול.

- TCP ו-UDP מוסיפים 2 כתובות פורטים לכל Segment: מקור ויעד. בדר"כ כתובת המקור נבחרת בצורה רנדומלית מהקבוצה ה-3 וכתובת היעד הוא מספר פורט של שירות מסויים.
  - להלן לינקים לרשימת הפורטים המלאה:
    - IANA
    - WhatsmyIP

TCP & UDP Differences

TCP

- Establishing a Session-הפרוטוקול יוצר חיבור מלא עם היעד, לפני תחילת העברת הנתונים. המקור ויעד מסכמים על פרמטרים במטרה לייעל את תעבורת המידע. (מאפיין זה בר השגה ע"י שדה Code Bits)
- Reliable Delivery-הפרוטוקול מיישם שיטה אשר מסייעת לו, לוודא כי כל חבילות המידע התקבלו. (מאפיין זה בר השגה ע"י שדה Ack No)
- Same-Order Delivery-בגלל שרשתות היום מספקות מספר דרכים (Routes) לכל יעד. המידע יכול להגיע בסדר שונה מזה שנשלח. TCP דואג לחבר את חבילות המידע בסדר הנכון. (מאפיין זה בר השגה ע"י שדה Sequence No).
- Flow Control-לפרוטוקול קיימת יכולת לעלות או להפחית את קצב העברת הנתונים, כלומר מס' או גודל חבילות המידע שנשלחות ברגע נתון. מנגנון זה יעיל במיוחד, משום שלרכיב הקצה ביעד יש משאבים מוגבלים כמו זיכרון או פס-רוחב ו-TCP מתחשב במשאבי היעד וכך דואג להעביר רק את כמות המידע שהיעד יכול לקבל ברגע נתון. יכולת זו מונעת אובדן נתונים מיותר. (מאפיין זה בר השגה ע"י שדה Window size).

UDP

- Connectionless-הפרוטוקול לא יוצר קשר עם היעד לפני העברת הנתונים.
- Unreliable Delivery-הפרוטוקול לא דואג לשדר מחדש חבילות שאבדו או נפגמו.
- No Ordered Data Reconstruction-הפרוטוקול אינו משמר סדר, כלומר לא דואג להרכיב את המידע בסדר הנכון.
- No Flow Control-הפרוטוקול לא מתחשב בכמות המשאבים של היעד. מה שאומר שקצב העברת הנתונים נשאר קבוע. במידה והיעד לא עומד בקצב השליחה, מידע אובד.

Wireshark Example

1

No.	Time	Delta Time	Source	Destination	Protocol	Packet Length	Sequence number	HTTP Delta	Info
53	69.440126000	0.003658000	10.3.1.53	10.1.2.5	TCP	60	0		6713→2000 [SYN] Seq=0 Win=16000 Len=0 MSS=1456
54	69.586965000	0.146839000	10.1.2.5	10.3.1.53	TCP	60	0		2000→6713 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0
55	69.594753000	0.007788000	10.3.1.53	10.1.2.5	TCP	60	1		6713→2000 [ACK] Seq=1 Ack=1 Win=16000 Len=0

- Frame 53: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
- Ethernet II, Src: Cisco_6c:f4:4e (00:07:0e:6c:f4:4e), Dst: Cisco_09:56:42 (00:24:c4:09:56:42)
- Internet Protocol Version 4, Src: 10.3.1.53 (10.3.1.53), Dst: 10.1.2.5 (10.1.2.5)
- Transmission Control Protocol, Src Port: 6713 (6713), Dst Port: 2000 (2000), Seq: 0, Len: 0
  - Source Port: 6713 (6713)
  - Destination Port: 2000 (2000)
  - [Stream index: 1]
  - [TCP Segment Len: 0]
  - Sequence number: 0 (relative sequence number)
  - Acknowledgment number: 86216704
  - Header Length: 24 bytes
  - .... 0000 0000 0010 = Flags: 0x002 (SYN)
  - Window size value: 16000
  - [Calculated window size: 16000]
  - Checksum: 0xf61c [validation disabled]
  - Urgent pointer: 0
  - Options: (4 bytes), Maximum segment size

2

No.	Time	Delta Time	Source	Destination	Protocol	Packet Length	Sequence number	HTTP Delta	Info
53	69.440126000	0.003658000	10.3.1.53	10.1.2.5	TCP	60	0		6713→2000 [SYN] Seq=0 Win=16000 Len=0 MSS=1456
54	69.586965000	0.146839000	10.1.2.5	10.3.1.53	TCP	60	0		2000→6713 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460
55	69.594753000	0.007788000	10.3.1.53	10.1.2.5	TCP	60	1		6713→2000 [ACK] Seq=1 Ack=1 Win=16000 Len=0

- Frame 54: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
- Ethernet II, Src: Cisco_09:56:42 (00:24:c4:09:56:42), Dst: Cisco_6c:f4:4e (00:07:0e:6c:f4:4e)
- Internet Protocol Version 4, Src: 10.1.2.5 (10.1.2.5), Dst: 10.3.1.53 (10.3.1.53)
- Transmission Control Protocol, Src Port: 2000 (2000), Dst Port: 6713 (6713), Seq: 0, Ack: 1, Len: 0
  - Source Port: 2000 (2000)
  - Destination Port: 6713 (6713)
  - [Stream index: 1]
  - [TCP Segment Len: 0]
  - Sequence number: 0 (relative sequence number)
  - Acknowledgment number: 1 (relative ack number)
  - Header Length: 24 bytes
  - .... 0000 0001 0010 = Flags: 0x012 (SYN, ACK)
  - Window size value: 5840
  - [Calculated window size: 5840]
  - Checksum: 0xde0c [validation disabled]
  - Urgent pointer: 0
  - Options: (4 bytes), Maximum segment size
  - [SEQ/ACK analysis]

3

No.	Time	Delta Time	Source	Destination	Protocol	Packet Length	Sequence number	HTTP Delta	Info
53	69.440126000	0.003658000	10.3.1.53	10.1.2.5	TCP	60	0		6713→2000 [SYN] Seq=0 Win=16000 Len=0 MSS=1
54	69.586965000	0.146839000	10.1.2.5	10.3.1.53	TCP	60	0		2000→6713 [SYN, ACK] Seq=0 Ack=1 Win=5840 Le
55	69.594753000	0.007788000	10.3.1.53	10.1.2.5	TCP	60	1		6713→2000 [ACK] Seq=1 Ack=1 Win=16000 Len=0

- Frame 55: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
- Ethernet II, Src: Cisco_6c:f4:4e (00:07:0e:6c:f4:4e), Dst: Cisco_09:56:42 (00:24:c4:09:56:42)
- Internet Protocol Version 4, Src: 10.3.1.53 (10.3.1.53), Dst: 10.1.2.5 (10.1.2.5)
- Transmission Control Protocol, Src Port: 6713 (6713), Dst Port: 2000 (2000), Seq: 1, Ack: 1, Len: 0
  - Source Port: 6713 (6713)
  - Destination Port: 2000 (2000)
  - [Stream index: 1]
  - [TCP Segment Len: 0]
  - Sequence number: 1 (relative sequence number)
  - Acknowledgment number: 1 (relative ack number)
  - Header Length: 20 bytes
  - .... 0000 0001 0000 = Flags: 0x010 (ACK)
  - Window size value: 16000
  - [Calculated window size: 16000]
  - [Window size scaling factor: -2 (no window scaling used)]
  - Checksum: 0xce19 [validation disabled]
  - Urgent pointer: 0
  - [SEQ/ACK analysis]