מטלת סיום תקשורת

Yosef Kahlon - 209011840
Gal Aqua - 207524919

https://github.com/YosefKahlon/Final-Project-Chat-Python.git

סרטון הפעלה

חלק א :

בחלק זה נדרשנו לבנות מערכת מסרים מידיים פרימיטיבית (בדומה ל - messenger) מבוססת על תקשורת .

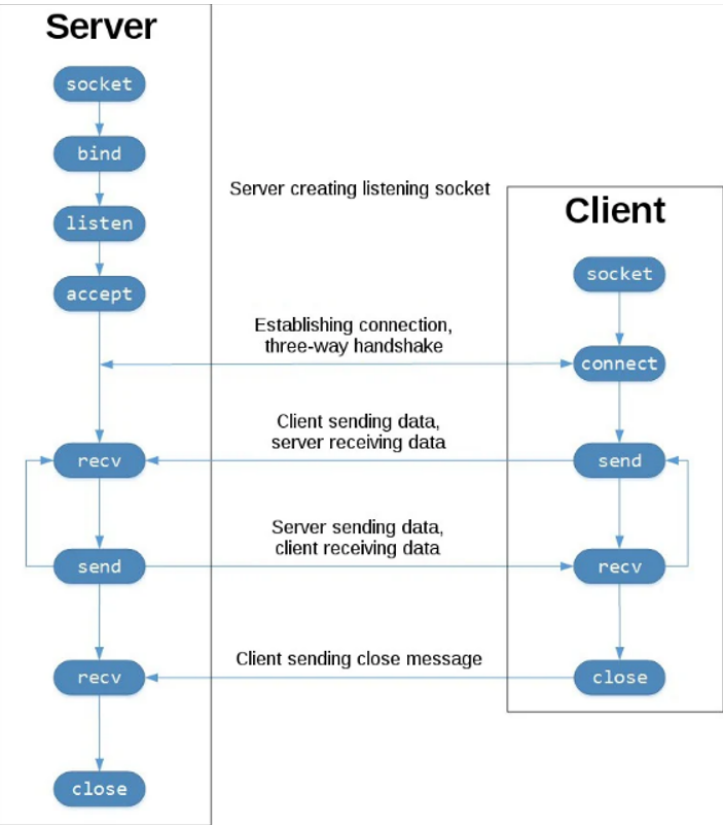
נסביר איך המערכת עובדת :

נגדיר שני סקריפטים Server ו - Client כאשר המטרה היא להגדיר את ה - server כך שהכל עובר דרכו והוא נותן שירות .

ה - Client יהיה המקום בו נגדיר את הממשק משתמש .

כדי לפתוח חיבור בין השרת ללקוח נדרש לפתוח socket TCP כאשר זה יבטיח לנו שירות אמין .

להן דיאגרמה של התהליך בפועל :

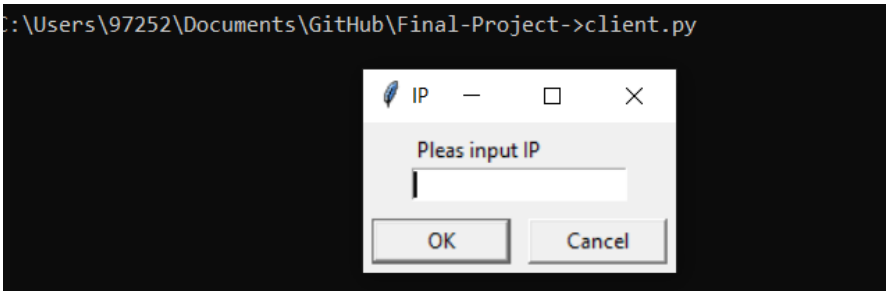


כעת נסביר איך להשתמש : [ניתן לראות כאן]

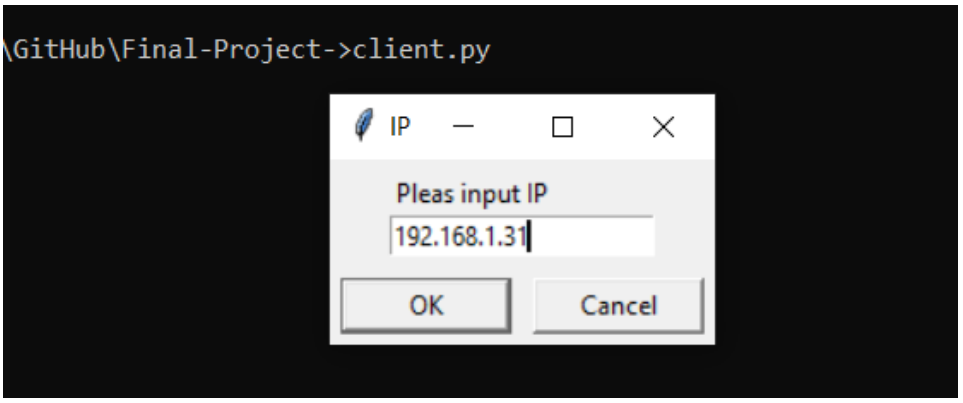
1. הורד את הפרויקט למחשב
2. כנס לשורת הפקודות במקום בו שמור הפרויקט ורשום את הפקודה server.py

```
C:\Users\97252\Documents\GitHub\Final-Project->server.py
192.168.1.31
server running......
```

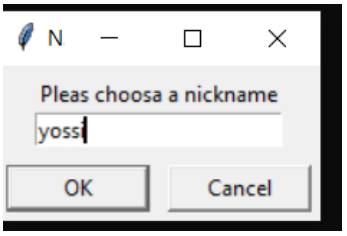
3. קיבלנו את כתובת הIP של השרת שפתחנו
4. כעת נפתח דרך שורת הפקודות את client.py



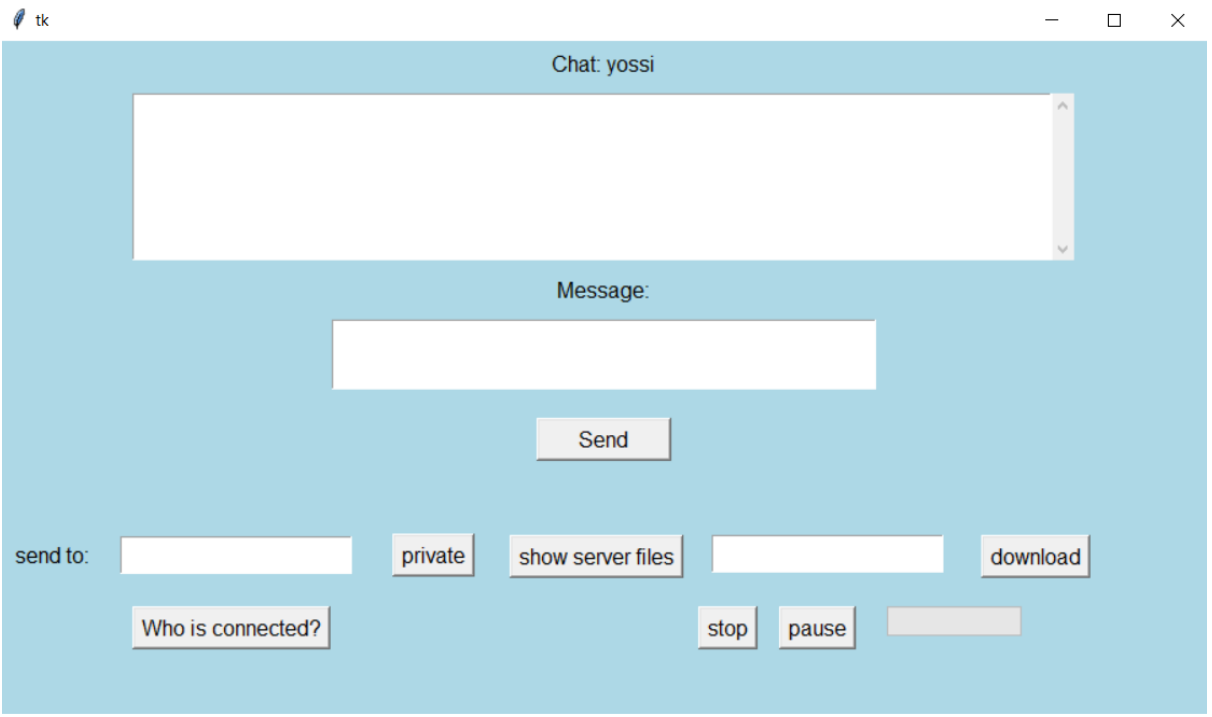
5. נכניס את כתובת IP הנתונה של השרת (ראה 2) .



6. נדרש כעת לבחור את השם שבו יזהו אותנו בחיבור לצאט .



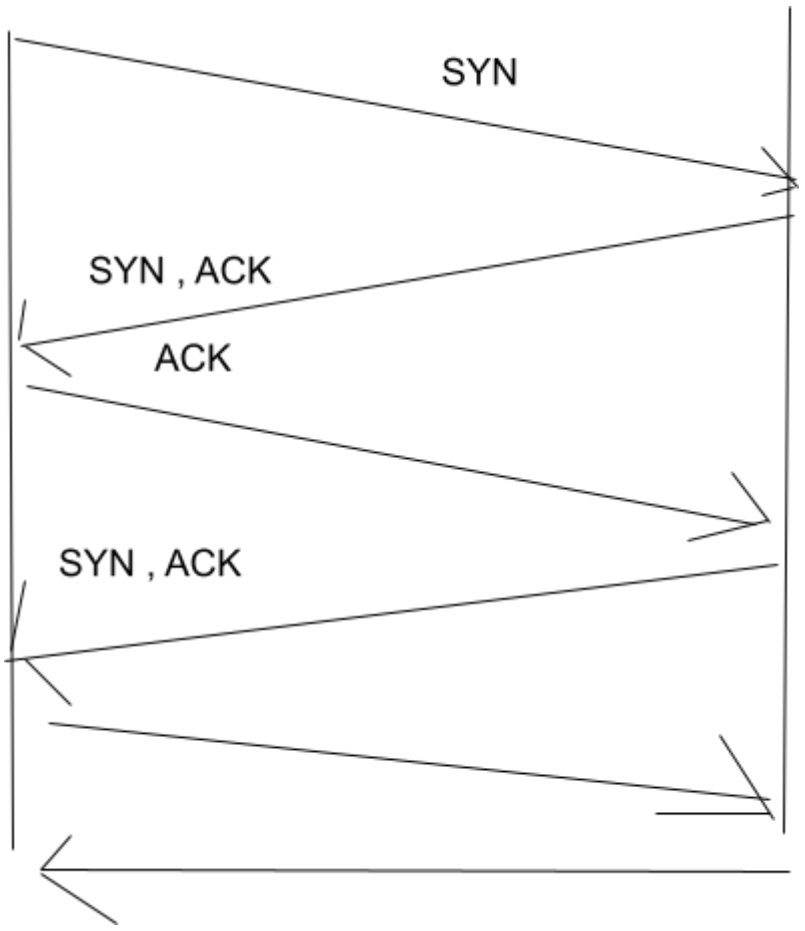
7. נפתח הצאט !



כך נראה הקלטת של chat

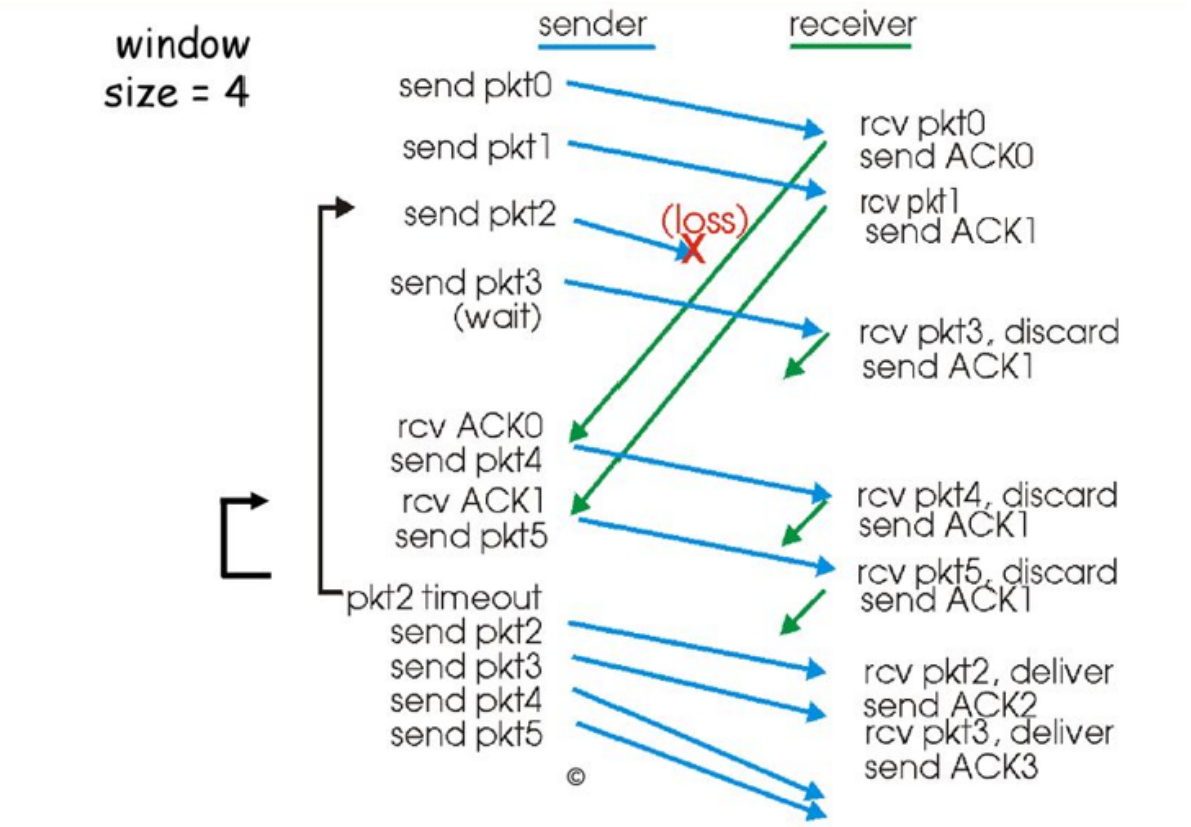
43	13:21:41.25…	192.168.1.31	192.168.1.31	TCP	56	62679 → 50011 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM=
44	13:21:41.25…	192.168.1.31	192.168.1.31	TCP	56	50011 → 62679 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256
45	13:21:41.25…	192.168.1.31	192.168.1.31	TCP	44	62679 → 50011 [ACK] Seq=1 Ack=1 Win=2619648 Len=0
46	13:21:41.25…	192.168.1.31	192.168.1.31	TCP	48	50011 → 62679 [PSH, ACK] Seq=1 Ack=1 Win=2619648 Len=4
47	13:21:41.25…	192.168.1.31	192.168.1.31	TCP	44	62679 → 50011 [ACK] Seq=1 Ack=5 Win=2619648 Len=0
51	13:21:44.35…	192.168.1.31	192.168.1.31	TCP	49	62679 → 50011 [PSH, ACK] Seq=1 Ack=5 Win=2619648 Len=5
52	13:21:44.35…	192.168.1.31	192.168.1.31	TCP	44	50011 → 62679 [ACK] Seq=5 Ack=6 Win=2619648 Len=0
53	13:21:44.35…	192.168.1.31	192.168.1.31	TCP	75	50011 → 62679 [PSH, ACK] Seq=5 Ack=6 Win=2619648 Len=31
54	13:21:44.35…	192.168.1.31	192.168.1.31	TCP	44	62679 → 50011 [ACK] Seq=6 Ack=36 Win=2619648 Len=0
55	13:21:44.35…	192.168.1.31	192.168.1.31	TCP	69	50011 → 62679 [PSH, ACK] Seq=36 Ack=6 Win=2619648 Len=25
56	13:21:44.35…	192.168.1.31	192.168.1.31	TCP	44	62679 → 50011 [ACK] Seq=6 Ack=61 Win=2619648 Len=0
57	13:21:47.59…	192.168.1.31	192.168.1.31	TCP	69	62679 → 50011 [PSH, ACK] Seq=6 Ack=61 Win=2619648 Len=25
58	13:21:47.59…	192.168.1.31	192.168.1.31	TCP	44	50011 → 62679 [ACK] Seq=61 Ack=31 Win=2619648 Len=0
59	13:21:47.59…	192.168.1.31	192.168.1.31	TCP	59	50011 → 62679 [PSH, ACK] Seq=61 Ack=31 Win=2619648 Len=15
60	13:21:47.59…	192.168.1.31	192.168.1.31	TCP	44	62679 → 50011 [ACK] Seq=31 Ack=76 Win=2619648 Len=0
61	13:21:49.85…	192.168.1.31	192.168.1.31	TCP	50	62679 → 50011 [PSH, ACK] Seq=31 Ack=76 Win=2619648 Len=6
62	13:21:49.85…	192.168.1.31	192.168.1.31	TCP	44	50011 → 62679 [ACK] Seq=76 Ack=37 Win=2619648 Len=0
63	13:21:49.85…	192.168.1.31	192.168.1.31	TCP	107	50011 → 62679 [PSH, ACK] Seq=76 Ack=37 Win=2619648 Len=63
64	13:21:49.85…	192.168.1.31	192.168.1.31	TCP	44	62679 → 50011 [ACK] Seq=37 Ack=139 Win=2619648 Len=0
65	13:21:51.31…	192.168.1.31	192.168.1.31	TCP	55	62679 → 50011 [PSH, ACK] Seq=37 Ack=139 Win=2619648 Len=11
66	13:21:51.31…	192.168.1.31	192.168.1.31	TCP	44	50011 → 62679 [ACK] Seq=139 Ack=48 Win=2619648 Len=0
67	13:21:51.31…	192.168.1.31	192.168.1.31	TCP	135	50011 → 62679 [PSH, ACK] Seq=139 Ack=48 Win=2619648 Len=91
68	13:21:51.31…	192.168.1.31	192.168.1.31	TCP	44	62679 → 50011 [ACK] Seq=48 Ack=230 Win=2619392 Len=0
75	13:21:55.57…	192.168.1.31	192.168.1.31	TCP	44	62679 → 50011 [RST, ACK] Seq=48 Ack=230 Win=0 Len=0

ניתן לראות שהצאט מבוסס TCP



חלק ב:

דיאגרמת מצבים :



כיצד המערכת מתגברת על איבוד חבילות :

בחרנו להשתמש בשיטה של go back N

נרחיב עליה : השיטה עובדת בצורה של חלון הזזה כאשר גודל החלון נקבע על ידינו ולמעשה השולח שולח כמה חבילות באתם לגודל החלון שנקבע ואז מחכה להודעת ACK כדי להזיז את החלון ולשלוח חבילות נוספות .

במקרה שלא קיבלנו הודעת ACK עבור חבילה מסוימת כלומר אבדה חבילה השולח יזהה את מספר החבילה האבודה וישלח את כל החבילות מחדש החל מחבילה שאבדה ועד גודל החלון .

כיצד המערכת מתגברת על בעיות latency:

נגדיר חלון זמן שבו מרגע שליחת הפקטה נחכה לקבלת הודעת ACK על אותה מספר פקטה . אם לא התקבלה הודעת ה- ACK אז נשלח שוב את כל הפקטות בחלון החלוקה החל מאותה פקטה שנשלחה ולא קיבלנו ACK בחלון הזמן שהוגדר.לדוגמה: גודל חלון -N=3 ,שלחנו פקטות מספר 0,1,2 קיבלנו ACK בחלון הזמן על 0,2 ולא על 1. מה שיקרה זה שנשלח שוב את פקטות מספר 1,2 ובנוסף את 3. כי גודל החלון=3.

חלק ג' :

1. בהינתן מחשב חדש המתחבר לרשת אנא תארו את כל ההודעות שעוברות החל מהחיבור
הראשוני ל switch ועד שההודעה מתקבלת בצד השני של הצאט. אנא פרטו לפי הפורמט הבא:
a . סוג הודעה, פירוט הודעה והשדות הבאים
i . כתובת IP מקור/יעד, כתובת פורט מקור/יעד, כתובת MAC מקור/יעד,
פרוטוקול שכבת התעבורה.

switch - קופסה עם כמה פורטים אליהם ניתן לחבר כבל רשת . עם זאת, הפונקציונליות שלהם שונה מאוד . לאחר שה-Switch למד את הרשת, הוא מעביר מסגרת מהפורט בה הוא קיבל אותה אל הפורט הרלוונטי בלבד.
היות שכל הודעה מגיעה רק אל היעד שלה , אין פגיעה בפרטיות המשתמשים. בנוסף, הוא חוסך את העומס הרב שהיה נוצר לו היינו משתמשים ב -Hub .כמו -כן , Switch מסייע במניעה של התנגשויות.

איך switch פועל?
ל-Switch יש טבלה שעליו למלא בזמן ריצה. הטבלה תמפה בין כתובת MAC לבין הפורט הפיזי הרלוונטי . לכל פורט פיזי יש מספר המאפשר לזהות אותו .
בהתחלה טבלת הSwitch תהיה ריקה ,לכן כאשר תהיה שליחת הודעה ממחשב A למחשב B הSwitch לא ידע למי לשלוח את ההודעה אין לו את כתובת הMAC של כרטיס הרשת הרשת של מחשב B לכן הוא יתנהג בדומה לHUB ויעביר את ההודעה לכל הפורטים חוץ מהפורט שממנו היא נשלחה.

בנוסף Switch ראה את ההודעה שהגיעה ממחשב A מגיעה מפורט x, ויכול לקרוא את ההודעה ולראות את כתובת הMAC של כרטיס הרשת של מחשב A מחובר לפורט x. ומוסיף את הנתונים האלה לטבלה (עמודות :PORT , MAC). ימשיך להוסיף פורטים וכתובות MAC לטבלה רק הפעם בצורה שונה.

מחשב B שולח מסגרת אל מחשב A ה-Switch מסתכל בטבלה , ורואה שהוא מכיר את כתובת ה-MAC אליה המסגרת נשלחת , והיא מקושרת לפורט מספר x .אי לכך , ה- Switch מעביר את המסגרת רק אל פורט x ,ולא לאף פורט אחר. בנוסף, הוא מסתכל במסגרת ורואה שבשדה כתובת המקור נמצאת כתובת ה-MAC של כרטיס הרשת של מחשב B היות שהמסגרת נשלחה מפורט מספר y ,הוא יכול להוסיף את מידע זה לטבלה.

כתובת ה-IP שלו עצמו מחשב A – שכן הוא יודע מה הכתובת שלו, למשל באמצעות DHCP.
- כתובת ה-MAC שלו עצמו מחשב A – שכן הוא יודע מה הכתובת שלו , שהרי היא צרובה על הכרטיס.
- כתובת ה-IP של מחשב B – שהוא גילה, למשל , באמצעות DNS .

- כתובת ה-MAC של מחשב B – שהוא גילה באמצעות פרוטוקול ARP

פרוטוקול – ARP -Protocol Resolution Address בשכבת הENTERNET. פרוטוקול זה ממפה בין כתובות לוגיות של שכבת הרשת לכתובות פיזיות של שכבת הקו .

פרוטוקול DHCP -Dynamic Host Configuration Protocol
הוא פרוטוקול תקשורת המשמש להקצאה של כתובות IP ייחודיות למחשבים ברשת מקומית.

פרוטוקול DNS - Domain Name System
פרוטוקול ה-DNS הומצא על מנת להקל את השימוש של אנשים ברשתות תקשורת. בני אדם זוכרים בקלות שמות, אך לא כתובות מספריות דוגמת כתובות IP. ה-DNS מגשר על הפער הזה על ידי ביצוע המרה בין הכתובת המילולית, אותה זוכר המשתמש, לבין כתובת ה-IP בה למעשה משתמש המחשב על-מנת לתקשר עם היעד.

2. הסבירו מה זה CRC

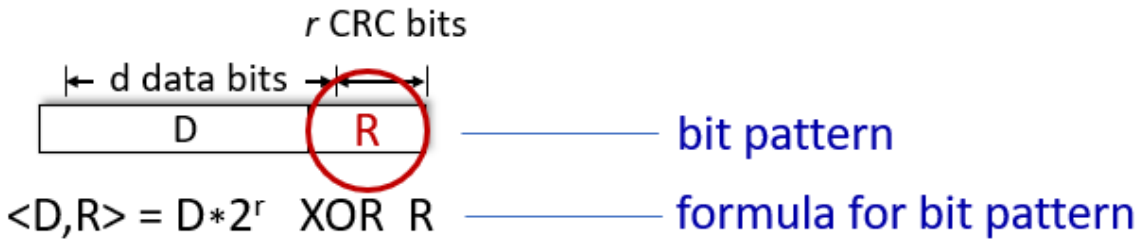
תשובה : Cyclic Redundancy Check (CRC)
שיטה לזיהוי רצף שגיאות בהעברת נתונים על ידי טריקים מתמטים של חילוק פולינום ופעולת הזזה .
השיטה פועלת על שכבת הקו ומשתמשים בה בעיקר ב Ethernet, 802.11 WiFi .

אז איך היא עובדת :

בהינתן פולינום יוצר מדרגה r ובהינתן הודעה M שברצוננו לקודד, עלינו לבצע את הפעולות הבאות:

1. נוסיף R אפסים מימין להודעה.
2. נחלק בפולינום (תוך שימוש בחילוק של השדה מודולו 2)
3. נחסר את השארית תוך שימוש ב-Xor במקום בחיסור רגיל.

נצרף את התוצאה שקיבלנו מימין להודעה המקורית ונשלח.



איך נדע שאין שגיאות ?
הצד המקבל יבצע את שלבים 1 ו-2 ויוודא ש-r הביטים האחרונים שנשלחו זהים לתוצאה שהתקבלה .
במילים אחרות השארית היא אפס .
הערות :

שיטה יותר חזקה לגילוי error detection coding מ checksum אבל גם העלות שלה יותר יקרה ולכן עושים אותה רק בשכבת הקו .
נשים לב שבגלל שבגלל שעושים אותה רק בשכבת הקו על כל המידע IPv6 יכול לוותר על cheksum .

3. מה ההבדל בין http 1.0 ,http 1.1, http 2.0, QUIC

HTTP הוא פרוטוקול תקשורת שפועל בשכבת היישום ונועד להעברת דפי WEB המורכבים מאובייקטים כמו תמונות, קבצי קול, סרטוני פלאש.
התקשורת בין השרת ללקוח ב-HTTP נעשית באמצעות בקשות ששולח הלקוח ותשובות שמחזיר השרת.

נגדיר RTT (round-trip time, זמן הלוך-חזור): זמן שלוקח לשלוח חבילה קטנה מלקוח לשרת ובחזרה (מעבר של החבילה דרך כל הקווים, המתנה בתורים של הנתבים, זמן טיפול בנתבים, זמן שידור בנתבים, זמן טיפול בשרת כולל בניית תגובה קצרה והדרך חזרה)

HTTP 1.0

חיבור לא עקבי , הקשר נסגר לאחר טיפול בבקשה
לכל היותר אובייקט אחד נשלח על גבי חיבור TCP.

HTTP 1.1

חיבור עקבי , שולח כמה אובייקטים ביחד
ולאחר מכן סוגר את החיבור TCP .
עבור גרסה זאת יש שני סוגי עם pipelining ובלי pipelining .

עקביים ללא pipelining:
לקוח שולח בקשה חדשה רק כאשר תגובה לבקשה קודמת נתקבלה.
עקב כך, הודעות HTTP עוקבות בין אותם לקוח/שרת נשלחות על גבי חיבור TCP פתוח. RTT אחד לכל אובייקט אליו יש הפניה + זמן שידור אובייקט

עקביים עם pipelining:
לקוח שולח בקשות ברגע שהוא נתקל באובייקט אליו יש הפנייה.
יכול להגיע גם ל-RTT אחד לכל האובייקטים אליהם יש הפניות+זמן שידור כל האובייקטים.

HTTP 2.0

חיבור עקבי , הקשר נשמר פתוח לאחר טיפול בבקשה
אובייקטים רבים יכולים להישלח על גבי חיבור TCP יחיד.

QUIC

http 3 נעזר בquick באמצעות חיבור udp
הבדל בין http 3

HTTP Header מאפשרת ללקוח ולשרת להעביר מידע נוסף עם בקשת HTTP או תגובה.

אז אם http 1.0 או 1.1 הם גרסאות בהם הhttp יוצר הרבה חיבורים במקביל , לא עובד בסדרי עדיפויות ובנוסף לא דוחס את ה- header .

הגרסה של Http 2.0 דוחס את ה-header, ומאפשר לנו puse שזה בעצם היכולת של השרת לשלוח תגובות מרובות לבקשת לקוח יחיד, ומפסיק לו התחברות TCP אחת .
בנוסף גרסה זאת מחלקת את האובייקטים ל frames וכך התעבורה עוברת מהר יותר .

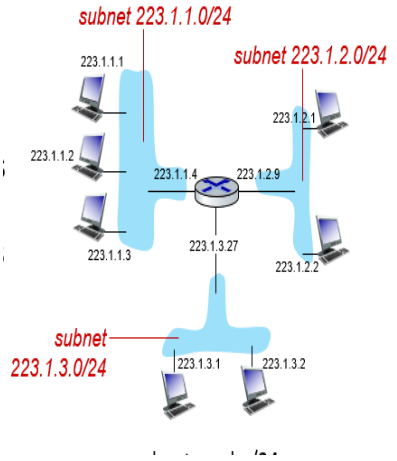
הבדל בין Http 2.0 לבין http 3 השימוש ב Udp לעומת TCP , יש לחיצת יד מהירה יותר בנוסף http 3 מאובטח יותר ומבקר כל שגיאת אובייקט ועומס.

4. למה צריך מספרי port ?

כדי שתהליך יכול לשלוח או לקבל מידע מהרשת נהיה חייבים ליצור Socket
(שילוב של IP, PORT) .
לרוב במחשב עובדים מספר יישומים .
כדי שהמחשב יוכל לסווג את הנתונים המתקבלים ליישומים השונים פועלים בו זמנית
לא מספיק רק כתובת ה IP אלה גם Port , סוג של כתובת פנימי שממפה אותי לשירות אליו אני רוצה לגשת.
לדוגמא ב port 80 נעבוד עם http .

5. מה זה subnet ולמה צריך את זה?

זה תת- רשת שלא עוברת דרך ראוטר, כלומר בין מחשבים מאותה תת רשת אין צורך להשתמש בראוטר כדי לתקשר האחד עם שני .
נשים לב שכתובות IP מאותו subnet יש בדיוק את אותה כתובת ip רק שקיים הבדל במיקום האחרון של המספרים 1 - 24 .
מדוע צריך את זה :
ברשת מסיומת יכולים להיות מיליוני מכשירים מחוברים ויכול לקחת זמן עד שהנתונים ימצאו את הכתובת המתאימה ולכן נוצר התת רשת שתצמצם את טווח החיפוש לכתובות ה IP .



6. למה צריך כתובות mac למה לא מספיק לעבוד עם כתובות ip ?

כתובת mac זוהי כתובת פיזית שנמצאת על החומרה עצמה (כרטיס רשת) כך שכל כרטיס שמיוצר מקבל כתובת ייחודית לו והיא למעשה עונה על מי אתה .

בעוד שכתובת IP הוא דבר לא קבוע ומשתנה בין רשתות והיא למעשה עונה על המיקום הנוכחי של המכשיר .

אז כדי שההודעה שנשלחה תהיה אמינה נצטרך לדעת שהגענו למכשיר המתאים ברשת הנכונה .

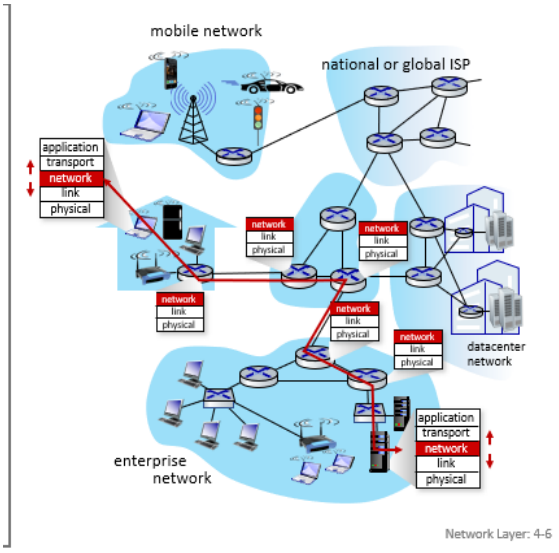
7. מה ההבדל בין Router Switch Nat?

Router נמצא בשכבת הרשת , תפקידו לחבר רשתות שונות ולקבוע את הנתיבים בעזרת אלגוריתם כמו דייקסטרה ,

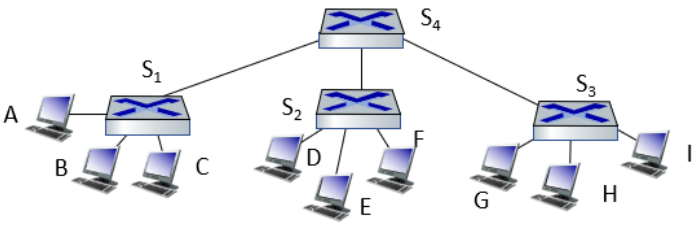
Switch נמצא בשכבת הלינק תפקידו לחבר מכשירים שונים ברשת

Sr. No.	Key	Router	Switch
1	Objective	Router main objective is to connect various networks.	Switch main objective is to connect various devices in a network.
2	Layer	Router works in Network Layer.	Switch works in Data Link Layer.
3	Usage	Router is used in LAN and MAN.	Switch is used only in LAN.
4	Data Format	Router sends data in form of packets.	Switch sends data in form of packets and frames.
5	Mode of Transmission	Router follows duplex mode of transmission.	Switch also follows duplex mode of transmission.
6	Collision	Less collision in case of Router.	In full duplex mode, no collision happens in switch too.
7	NAT Compatability	Compatible with NAT.	Not compatible with NAT.
8	Type	Routing type is Adaptive and Non-adaptive routing.	Switching type is Circuit, Packet and Message switching.

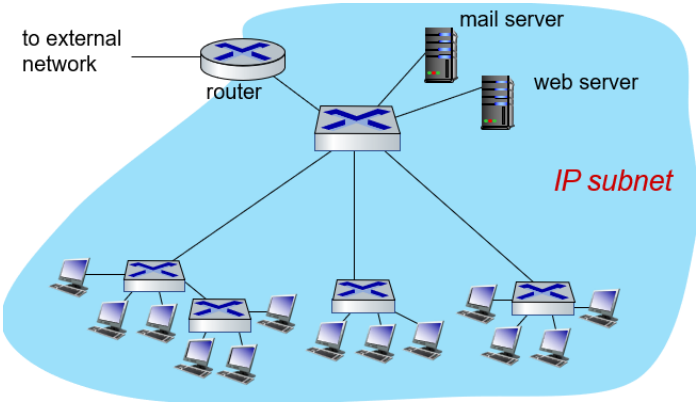
from



Routing protocols



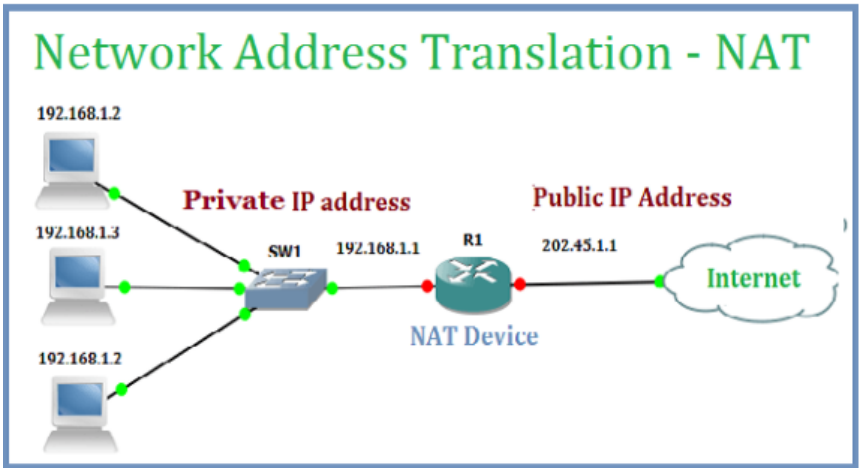
Switch



8. שיטות להתגבר על המחסור ב IPv4 ולפרט?

כידוע IPV4 הוא בגודל 32 ביט ומיוצגת באמצעות 4 מספרים עשרוניים המופרדים בנקודה. כל מספר מהווה מקבץ של 8 ביט וגודלו נע בין 0 ל-255.

כלומר נוצר מצב שבו יש מספר מוגבל של כתובות (256^4) מספק כמעט 3.4 מיליארד כתובות שונות.
נמצא פתרון שנקרא לו NAT-Translation Address Network .
לפי רעיון זה , כל הישויות בתוך הרשת יקבלו כתובות פרטיות – כלומר כתובות שיזהו אותן בתוך הרשת בלבד , ולא בעולם החיצוני . כתובות אלו אינן ניתנות לניתוב – כלומר, נתב באינטרנט שרואה חבילה שמיועדת לכתובת שכזו עתיד " לזרוק " אותה.



פתרון נוסף- IPv6.(IPv6 - היא גרסה בגודל 128 ביט .) פותרת את הבעיה של מחסור בכתובות אבל יקח זמן שעד פרוטוקול IPV6 יעבוד בצורה חלקה.
בפועל, רוב מהמשתמשים באינטרנט לא נתקלים בבעיות בעת ביקור באתרים שהופעל בהם פרוטוקול האינטרנט IPv6. עם זאת, קיימים מקרים לא רבים בהם IPv6 יופעל במחשב של משתמש, אך לא יתפקד כהלכה.
רוב המחשבים היום מגיעים כך שברירת המחדל שלהם היא תקשורת IPv6, ואם היא אינה אפשרית לעבור ל IPv4 (dual-stack). בעית קישוריות של IPv6 בדרך כלל תגרם כתוצאה מתצורה שגויה או מתוכנה לא מעודכנת של נתבים ביתיים, מבאגים במערכות הפעלה או מבעיות הקשורות לספק האינטרנט.
בדכ"ל למי שרוצה לעבוד ב IPv6 רצוי לשדרג לגירסה העדכנית ביותר של הדפדפן, לגירסה העכשווית של מערכות הפעלה, לעדכן את תוכנת הנתב וכמובן לודא שספק אינטרנט אכן תומך ב IPv6 .

9. נתונה הרשת הבאה.

a . AS2, AS3 מריצים OSPF

b . AS1, AS4 מריצים RIP

c . בין ה- Ass רץ BGP

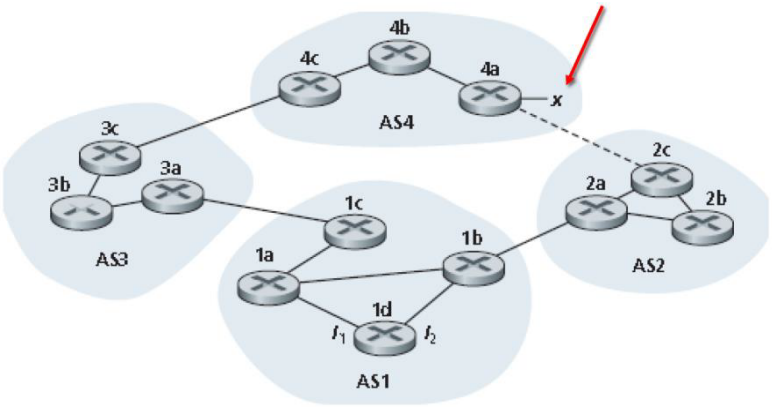
d . אין חיבור פיזי בין AS4 , AS2

e . בעזרת איזה פרוטוקול לומד הנתב 3c על תת רשת x

f . בעזרת איזה פרוטוקול לומד הנתב 3a על תת רשת x

g . בעזרת איזה פרוטוקול לומד הנתב 1c על תת רשת x

h . בעזרת איזה פרוטוקול לומד הנתב 2c על תת רשת x



iBGP או BGP פנימי, הוא פרטוקול ניתוב בין שני נתבי BGP בתוך אותו AS. מטרת הפרוטוקול היא לספק מידע לנתבים פנימיים ברשת, כמו רשימת כתובות IP הנמצאות מחוץ ל-AS.
eBGP או BGP חיצוני, הוא פרוטוקול ניתוב בין שני נתבי BGP השייכים לAS-ים שונים ושכנים זה לזה (BGP peers). מטרת הפרוטוקול היא לאתר את המסלול האופטימלי ברשת באמצעות עדכון AS-ים אודות כתובות ה-IP עליהן כל נתב אחראי, כדי שהם ינתבו אליו מידע רלוונטי.

e) iBGP-לממנו יש את היציאה לAS4
f)eBGP- אין ממנו יציאה לAS4
g)iBGP-(הרצנו RIP) אפשר להתייחס לזה כאילו הם באותה רשת
h)eBGP- אין חיבור פיזי בין השניים לכן זה