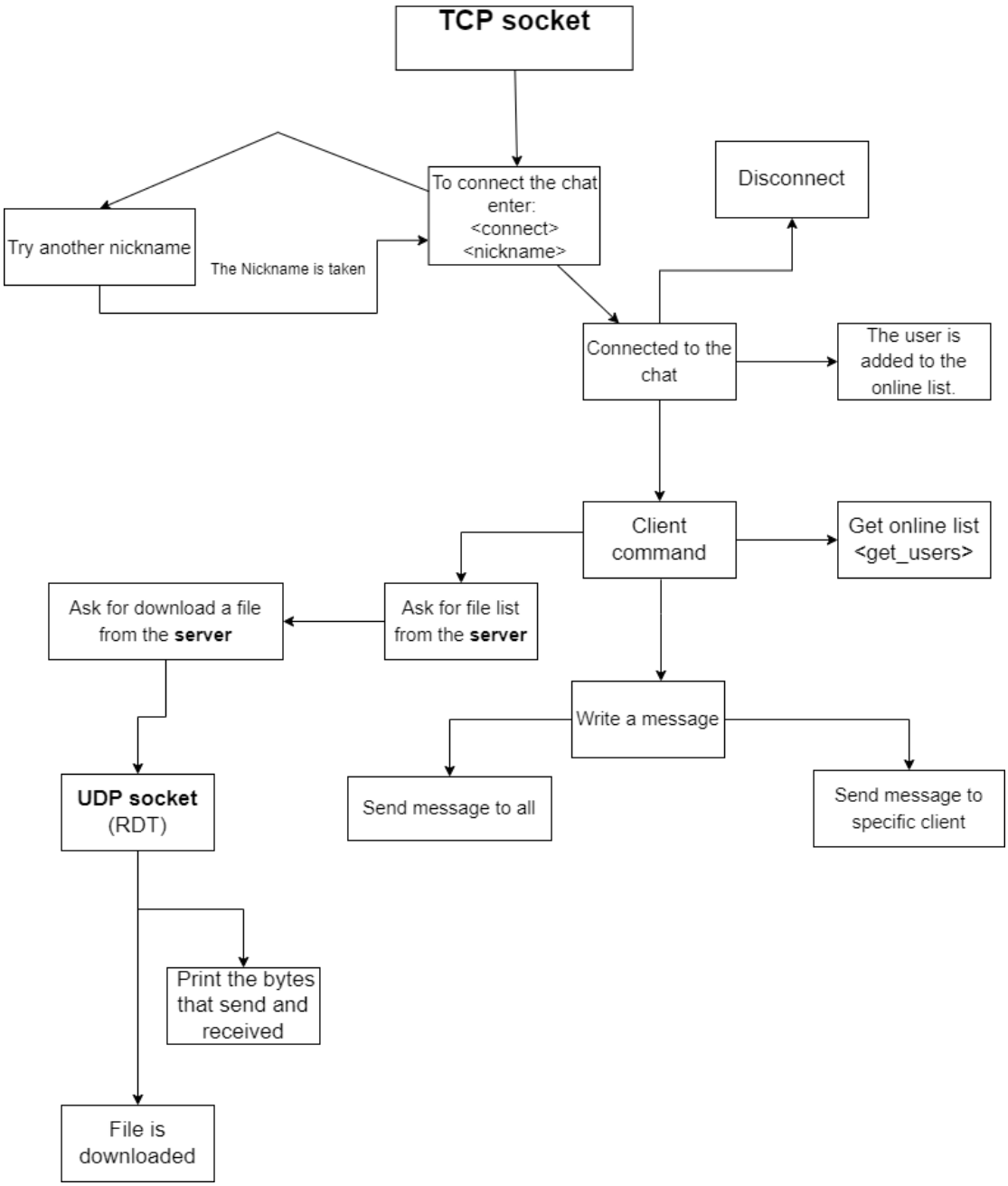
פרויקט סיום – תקשורת ומחשוב

טל אורנן - 209349356

צח יצחק אופיר - 208062943

פרויקט סיום תקשורת ומחשוב – חלק ב'

- ציירו דיאגרמת מצבים בהם המערכת עובדת



- כיצד המערכת מתגברת על איבוד חבילות

על מנת להתגבר על איבוד חבילות , נתנו לכל פאקטה מספר סידורי , וחיכנו ל-התראה שהפקאטה עברה בשלמותה

במידה והיא לא כמו שצריך – היא נשלחת שוב.

- כיצד המערכת מתגברת על בעיות latency

על מנת להתגבר על בעיות אלו , ראשית נתנו מספר סידורי לכל פאקטה , ובדקנו אם הגיעה בשלמותה בכל פעם שהועברה פאקטה.

בנוסף , הגדרנו PORT וSOCKET לכל משתמש כך שיוכלו להוריד ולקבל את הפקאטה באותו הזמן ללא הבעיות הללו.

חלק א+ב- מימוש הקוד:

על מנת לממש צ'אט עשינו 3 מחלקות –

1. SERVER : השרת של החלקה – הוא אחראי להקשבה ללקוחות וקיום פקודות כמו שליחת הודעות בין הלקוחות או לכל הלקוחות – הורדת קבצים וכו.

2. client : לקוח – הוא מתחבר לצ'אט ומבקש חיבור מהשרת – הוא שולח הודעות אשר מייצגות דברים שהלקוח יכול לעשות כמו לשלוח בקשה ללקוח אחר , הודעות לכולם וכו. לכל בקשה יש סינטקס קבוע כפי שהתבקש במטלה

3. Const : מחלקת קבועים – על מנת שהדברים יהיו ברורים ומסודרים יותר בקוד השתמשנו במחלקה זו.

בנוסף על מנת לממש את המחלקות השתמשנו בטרדים על מנת שהתהליכים יוכלו לרוץ בו זמנית. כעת נסביר – (באנגלית) על הקוד.

CLIENT:

Function	Explanation
Receive	check if the client ask to download a file from the server if it is get the name of the file and start the process
clientMsg	this function ask from the client to command to do and send it to the server
parse_download_file	Use in test's

down_file	this function ask to download file
close_resoueces	this function close client socket
run_client	Run all clients method's

The main algo is down_file:

this function download file from the server

this function read data from the udp connection, and sending acks for each chunk that has been provided

.in case chunck already send, don't add it

keep doing it until all the file is sent. close the resources .after that

file-name - nameof the file to create

file_size - size of the file

free_port- port to read the file from

free_port_to_send_back - port to send ack to

server:

function	Explanation
sign_in	this function get from the client message to connect
sent_message_to_all	this function send a broadcast message
send_message	this function send s broadcast message or a single message set_msg_all>message> set_msg_all><user_name><me> <ssage

get_users	this function send all the users that are in the chat
user_disconnection	this function removes the chat user from the server metadata
get_files	this function sends a list of files
request_to_download_file	This function gets from the client command to download a file
get_free_port	this function get a port that is not in use
send_file	Send a file to the client
connection_communication	Translate client command and do it

The main algo is send_file :

this function send a file from the server to the client

Sending file data in chunks, attaching serial numbers for .each chunk

this function tries to keep sending the same chunk until .ack it is sent from the client

after all the data is sent, closing the resources

how to run?

1. First enter the at least 2 cmd's.
2. Type cd "path" (where "path" is the location of the files)
3. Type at one cmd "server.py" (for windows) or "python3 ./server.py" (for linux).
4. Repeat it with another cmd but with the command client.py.
5. Connect, and start chatting 😊

פרויקט סיום תקשורת ומחשוב – 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.

פתרון : Switch "לומד" את כתובות ה-MAC של הרכיבים המחוברים לכל כניסה שלו על ידי קריאת נתוני הבקרה בחבילות המידע המגיעות אליו ושמירתן בטבלה פנימית הנקראת טבלת MAC (קיצור של Memory Addressable Content).

כאשר חבילה מגיעה אל הSwitch , הSwitch בודק מה היא כתובת ה־MAC אליה מיועדת החבילה: אם הכתובת מוכרת לSwitch בטבלת ה-MAC הוא יעביר את החבילה אך ורק אל הכניסה שתוביל את החבילה ליעדה, אם הכתובת לא מוכרת הSwitch יעביר את החבילה לכל הכניסות פרט לזו שממנה היא התקבלה.

Switch מאפשר למספר תעבורות שונות להתבצע דרכו במקביל, על ידי חיבור של כל שני קצוות המתקשרים ביניהם בנפרד (microsegmentation).

שיטה זו מקטינה לאפס (כאשר לכל כניסה של הSwitch מחובר מחשב אחד) את מתחמי ההתנגשות (collision domain) ומגדילה את תפוקת הרשת - במיוחד ברשתות עמוסות.

המחשב משיג את כתובת ה IP ושאר פרטי הרשת על ידי פרוטוקול DHCP, הודעה נשלחת לכל הישויות ברשת. שרת ה DHCP רואה את הבקשה ומחזיר הודעה הכוללת את פרטי הרשת שלנו.

כעת המחשב שלנו יצטרך לגלות מה כתובת הIP של האתר המבוקש, על מנת שיוכל לשלוח אל השרת של האתר בקשות. המחשב משתמש בפרוטוקול DNS (שרת הDNS שלנו הוא בעצם שרת ה DNS של ספקית האינטרנט שלנו)

לשם כך, הוא יתשאל את שרת ה DNS שלו מה היא כתובת ה IP של האתר.

המחשב יודע שהוא קיבל את כתובת ה IP של שרת ה DNS באצמעות תהליך DHCP, בה הוא קיבל גם את כתובת ה IP שלו.

2. הסבירו מה זה CRC

פתרון :

CRC - Cyclic redundancy check

הינו מנגנון CheckSum לגילוי שגיאות כלומר הוא היא סוג של קוד לאיתור שגיאות המשמש לאיתור שגיאות בהעברת נתונים.

לפני העברת המידע מחושב ה־CRC ומתווסף למידע המועבר. לאחר העברת המידע, הצד המקבל מאשר באמצעות ה־CRC שהמידע הועבר ללא שינויים.

השימוש ב־CRC נפוץ בעיקר בשל קלות המימוש שלו בחומרה בינארית, קלות החישוב המתמטית שלו, ובמיוחד היעילות שלו בגילוי שגיאות נפוצות הנובעות כתוצאה מערוצי תקשורת רועשים.

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

פתרון:

מה זה QUIC:

QUIC הוא פרוטוקול בשכבת התעבורה.

גוגל יצרה את הQUIC: Quick UDP Internet Connection, על מנת לתת מענה לבעיה ב TCP - לא לכל חיבור יש את אותם צרכים וחלק מן המידע הנוסף יכול לשמש לשיפור תחבורה אמינה.

QUIC מאפשר יצירת חיבורים מאובטחים יותר עם שימוש בפחות נסיעות הלוך ושוב, דבר זה מפחית את זמן האחזור של החיבור – במיוחד בעת חיבור מחדש לשרתים שהיו כבר בשימוש בעבר. בנוסף לכך, QUIC מאפשר להביא את המשאבים המרובים הדרושים בדרך כלל לעיבוד דף באינטרנט באמצעות חיבור יחיד תוך כדי שהוא נמנע מהראש של הקו ש TCP יגרום

מה זה HTTP:

פרוטוקול בשכבת היישום.

פרוטוקול זה נועד להעביר דפי HTML ובשכבת היישום של מודל OSI ואת האובייקטים שהם מכילים. התקשורת ב HTTP מתחילה בשיחה בין השרת ללקוח, באמצעות TCP ולאחר מכן יש רצף של בקשות ותשובות (request, responses) שנשלחות הן על ידי הלקוח והן על ידי השרת.

לHTTP יש מספר גרסאות, כיום משתמשים ב HTTP 1.1, גרסא זו תומכת גם בגרסא הקודמת, 1.0. הגרסא זו, בשונה מהגרסא הקודמת, כן שומרים על עקביות הקשר.

לאחר מכן נוצרה גרסאת 2.0, שמשתמשת בריבוב.

הבדלים עיקריים:

QUIC	HTTP 2.0	HTTP 1.1	HTTP 1.0
מאפשר יצירת חיבורים מאובטחים יותר עם שימוש בפחות נסיעות הלוך ושוב, דבר זה מפחית את זמן האחזור של החיבור – במיוחד בעת חיבור מחדש לשרתים שהיו כבר בשימוש בעבר	משתמש בריבו (מולטיפלקינג), שבו משאבי חיבור TCP בודד שיסופקו משולבים ומגיעים ללקוח כמעט באותו זמן. ייתכן שהserver יעשה לנו push וידחוף לנו אובייקטים אחד אחרי השני. אם אובייקט מסוים גדול יכול להיות שנפצל אותו לכמה חלקים.	תומך בשימוש חוזר בחיבור, כלומר לכל חיבור TCP יכולים להיות בקשות ותגובות מרובות, הלקוח יכול לבקש מספר משאבים מהשרת בו זמנית.	לכל חיבור TCP יש רק בקשה אחת ותגובה אחת (request, response)

4. למה צריך מספרי port?

תחילה, נסביר מהו פורט:

שני מכשירים ירצו לתקשר אחד עם השני באמצעות TCP הם יצטרכו להחליט באיזה ערוץ להעביר את המידע, זאת בשביל שמכשיר בצד אחד ידע שהמידע שהוא שולח אכן מגיע ליישום הנכון במכשיר בצד השני, כנ"ל לגבי איך ידע המכשיר בצד השני שהמידע שהוא מקבל אכן הגיע מהמכשיר בצד האחד, כמו כן, הוא יצטרך מידע לאיזה יישום להעלות אותו (שכבת האפליקציה). בין שני המכשירים המתקשרים נוצר Session (במידה והיישום משתמש בפרוטוקול TCP). ישנם הרבה סוגי מידע והשימוש ב port עוזר למכשיר להבין מה לעשות עם הנתונים שהם מקבלים.

5. מה זה subnet ולמה צריך את זה?

מהו Subnet ? כתובת הIP מחולקת לשני חלקים

- מזהה רשת: כלומר לאיזה רשת משויכת כתובת הIP

-מזהה ישות: זה ה-host, לאיזה כרטיס רשת בתוך הרשת שייכת כתובת הip הזו.

מה זה subnet mask?

כתובת IP אחת מייצגת שתי כתובות:

כתובת ראשונה - כתובת של הרשת.

כתובת שניה - כתובת של המחשב בתוך אותה הרשת.

תפקידו של subnet mask הוא לסמן את הסוף של כתובת אחת ואת תחילתה של הכתובת השנייה.

כי מבחינת המחשב, ה IP זאת כתובת אחת ארוכה, וצריך לדעת לחלק אותה לשני חלקים.

המטרה של חישוב Subnet:

*אם היעד שלו נמצא ברשת שלו אז אפשר לשלוח לו את המידע ישירות.

*אם המחשב נמצא ברשת אחרת אז נשלח את המידע לראוטר כדי שינתב ויעביר את המידע החוצה.

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

פתרון : כתובת ה mac הינה כתובת פיזית שלא ניתנת לשינוי והיא ייחודית.

כתובת ה Ip הינה כתובת לוגית, כלומר היא לא ברמת החומרה או הקושחה, היא ברמת התוכנה.

לכן בניגוד לכתובת הפיזית mac, היא ניתנת לשינוי והיא ניתנת על ידי מנהל הרשת או השרת והיא לא ייחודית, כלומר אפשר להשתמש באותה כתובת כמה פעמים (במצבים מסוימים), כלומר באופן כללי היא לא ייחודית.

כתובת ה MAC - הפיזית משמשת כמזהה של כרטיס הרשת בתוך הרשת המקומית ה - LAN.
כתובת ה- IP - הלוגית משמשת כמזהה של המחשב בין רשתות שונות. באמצעות כתובת הIP ניתן לעבור בין רשתות, כתובת ה- MAC לבדה אינה מאפשרת זאת.

7. מה ההבדל בין Router Switch Nat?

Switch	Router
חבר התקנים מרובים ברשת.	הוא מחבר switchים מרובים והרשתות המתאימות להם.
זה עובד על שכבת קישור הנתונים של מודל OSI.	זה עובד על שכבת הרשת של מודל OSI.
הוא משמש בתוך LAN.	ניתן להשתמש בו ב-LAN או MAN.
Switch לא יכול לבצע NAT או תרגום כתובות רשת.	נתב יכול לבצע תרגום כתובות רשת.
הSwitch לוקח יותר זמן תוך קבלת החלטות ניתוב מסובכות.	נתב יכול לקבל החלטת ניתוב הרבה יותר מהר מאשר Switch.
זה מספק רק port אבטחה	הוא מספק אמצעי אבטחה להגנה על הרשת מפני איומי אבטחה.
Swichים יכולים לעבוד רק עם הרשת הקווית.	נתבים יכולים לעבוד גם עם רשתות קוויות וגם עם רשתות אלחוטיות.
switchים זמינים עם פורטים שונים, כגון 8, 16, 24, 48 ו-64.	נתב מכיל שתי יציאות כברירת מחדל, כגון Fast Ethernet Port. אבל אנחנו יכולים גם להוסיף את הפורטים במפורש.
הוא משתמש בCAM (זיכרון שניתן לכתובת תוכן) עבור כתובת ה-MAC של המקור והיעד.	הוא משתמש בטבלת הניתוב כדי לקבל את המסלול הטוב ביותר עבור כתובת ה-IP של היעד.

- NAT לא קשורה לפה, Router ו Switch הינם רכיבים בעוד NAT הינה טכניקה.

8. שיטות להתגבר על המחסור בIPv4 ולפרט?

תחילה, נסביר מהו המחסור ב IPv4: כל מכשיר שמתחבר לאינטרנט, זקוק לכתובת IP. כרגע, רשת האינטרנט מבוססת על IPv4. כתובת IP זו בנויה מ 32 ביטים, לכן מאפשרת כ4.3 מיליארד כתובות שונות, בפועל המספר קטן יותר ובהתחשב בתפוצה של עשרות מיליארדי מכשירים המתחברים לאינטרנט, לכן יש חוסר בכתובות IPv4.

שיטה להתגבר על המחסור הינה שימוש ב NAT. NAT מפצל כתובת IPv4 ציבורית אחת למספר כתובות IPv4 פרטיות. פתרון זה נחשב לפתרון לטווח קצר מכיוון שהוא לא מאפשר בדרך כלל חיבורים נכנסים ללא תצורה ידנית של port forwarding.

פתרון נוסף אך לא אפקטיבי הוא שימוש בפרוטוקול נוסף, חדש יותר, IPv6 הפרוטוקול מאפשר להציע כתובת בעלת 38 ספרות שונות (מה שיפתור את בעיית המחסור לצמיתות), אך פתרון זה לא אפקטיבי מכיוון שיש הרבה ציוד תקשורת שאינו תומך, כלומר נניח ונתב במרכזיה תומך רק IPv4, הוא יכול להתקיים ברשת IPv6 רק כNAT, אבל אם אין לו פונקציה כזו אז המכשיר שמעליו ברשת צריך לעשות את זה וכל הרשת שמתחתיו תשאר IPv4, שלא לדבר על מודמים ביתיים. בשורה התחתונה, IPv6 לא תוכנן להתקיים לצד IPv4 אלא כמחליף, ולהחליף את כל הציוד בעולם ביום אחד לא קרה ולא יקרה.

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 AS2, AS4
   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

e - הנתב 3c לומד על תת רשת x דרך פרוטוקול RIP וBGP בגלל שעל מנת ש 3c יקבל מידע על תת רשת x הוא יצטרך לעבור דרך AS4 והנתב שמחבר בין 3c ו AS4 הוא 4c. ידוע לנו שבין ה ASS רץ BGP ולכן ישתמש בו לעבור בתוך AS4, נשתמש ב RIP לעבור מ 4c ל x.

BGP → RIP

f – כמו סעיף קודם, נתב 3a יקבל מידע על תת רשת x דרך פרוטוקול RIP ו BGP, אך בנוסף העברה תעשה ב AS3 על מנת להגיע ל 3c שמחובר ל 4c ולבסוף יגיע ל AS4. לשם כך, נשתמש ב OSPF בגלל שידוע לנו שהוא רץ ב AS3 ואז נשתמש ב BGP שרץ בין ה ASS ולבסוף RIP רץ ב AS4.

OSPF → BGP → RIP

g - בשביל שנתב 1c ילמד על תת רשת x הוא יעבור מ AS1 ל AS3 בעזרת BGP, שם הוא יעבור מ 3a ל 3c בעזרת OSPF. לאחר מכן הוא יעבור מ 3c ל 4c עם BGP ולבסוף יגיע ל x בעזרת RIP.

BGP → OSPF → BGP → RIP

h - 2c יצטרך לזוז בתוך AS2 על ידי שימוש ב OSPF ל 2a, לאחר מכן לזוז ל AS1 דרך BGP ל 1b. בתוך AS1 תהיה החלפה ל 1c עם RIP, ואז נצטרך ללכת ל 3a עם RIP. מ 3a ל 3c עם OSPF, לבסוף הוא ישתמש ב BGP לזוז ל 4c ואז שוב ישתמש ב RIP להיכנס ל AS4 לx.

OSPF → BGP → RIP → BGP → OSPF → BGP → RIP