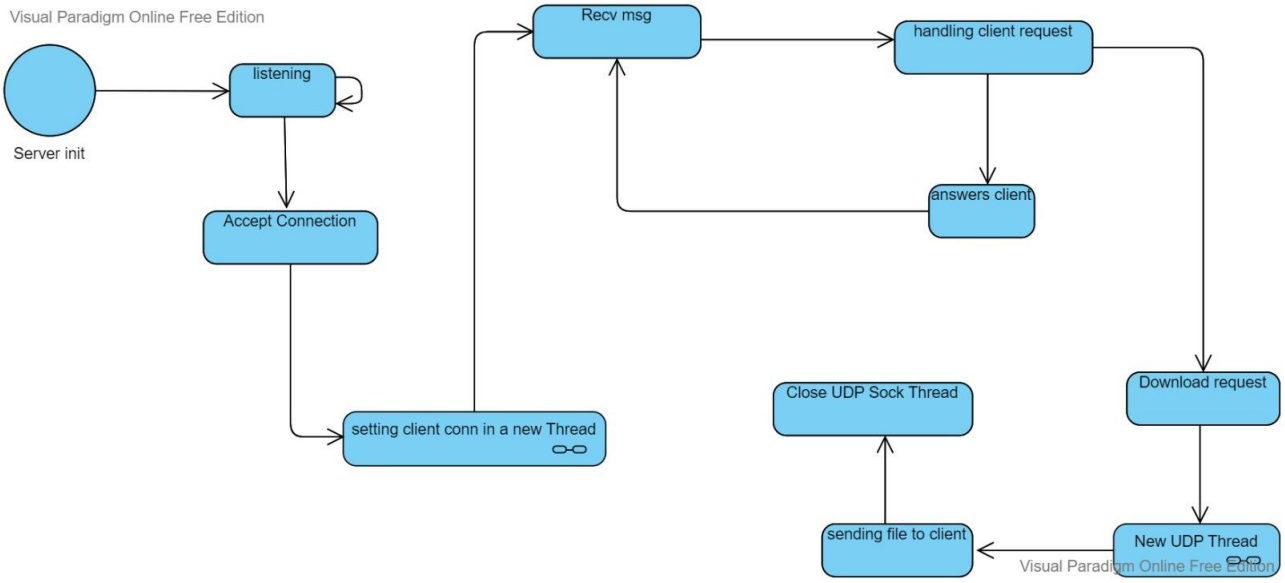
איתי קרואני 311504914
טל מלחי 208278556

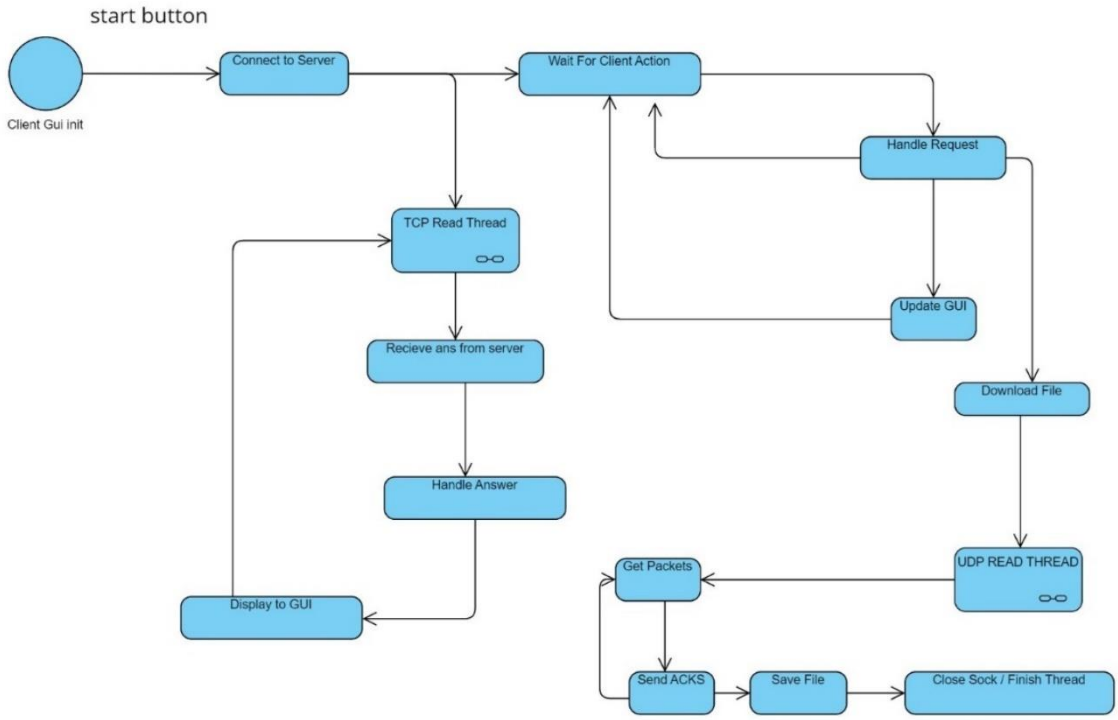
חלק ב':

שאלה 1

דיאגרמת מצבים:



השרת: מקשיב לחיבורים נכנסים, אם וכאשר הוא מקבל חיבור חדש, הוא פותח סוקט חדש ללקוח ומפעיל טרייד, תוך כדי הוא ממשיך להקשיב לחיבורים נכנסים חדשים.
השרת מחכה לבקשה מהלקוח, מטפל בהתאם לסוג הבקשה ומחזיר את התשובה ללקוח על מה שביקש.
במידה והבקשה היא הורדת קובץ הטיפול יהיה בנפרד. נפתח טרייד חדש שמטפל בהורדת קובץ עם קשר UDP . נשלח את הקובץ המבוקש ללקוח ונסגור את הטרייד.



איתי קרואני 311504914
טל מלחי 208278556

הלקוח: מתחבר לשרת (באמצעות כפתור "start" בממשק הגרפי). כאשר הצליחה ההתחברות באפשרות הלקוח לבצע מספר פעולות, בנוסף נפתח טרייד חדש "read thread" באמצעותו נקבל את ההודעות מהשרת ונטפל בהם בנפרד. בעת ביצוע פעולה מסוימת, נשלחת הודעה לשרת בהתאם לבקשתו של הלקוח והממשק הגרפי מתעדכן בהתאם לאותה הפעולה – נפתחים/נסגרים כפתורים ותיבות טקסט נוספות. את התשובה עבור כל בקשה נראה ב "read thread" , נעדכן את הממשק הגרפי ונציג ללקוח את התגובה.

שאלה 2

כדי להתגבר על איבוד פאקטות השתמשנו במנגנון GO BACK N בנוסף השתמשנו בTimer, בגודל חלון דינמי וב Acknow-ledgment .

מהצד של השרת: הגדרנו חלון התחלתי בגודל 8, בו השרת שולח תחילה ללקוח 8 חבילות. בכל רגע נתון השרת יודע מה החבילה הראשונה בחלון ומה החבילה הבאה שהוא יכול לשדר.

הגדרנו זמן המתנה מקסימלי של 0.5 שנייה. עבור כל חבילה שנשלחה הפעלנו שעון, כל עוד לא הגענו ל Timeout נמשיך לקבל הודעות ACK מהלקוח, ובנוסף, גודל החלון גדל בקצב אקספוננציאלי (פי 2 כל פעם). לאחר ההכפלה נבחר את גודל החלון להיות המינימלי בין גודל החלון הנוכחי לבין כמות הפאקטות נשארו לשלוח. הגדרנו משתנה שסוכם את כל מספרי ההודעות ACK שהתקבלו (counter_packets), ובכל פעם נבדוק אם המספר הסידורי של ההודעה גדול ממשתנה הסכימה, אם אכן התקבלה הפאקטה הנכונה, נעדכן את המשתנה שהגדרנו להיות מס ההודעה של הפאקטה שהתקבלה פלוס אחד (אנחנו צופים מאיזה פאקטה נקבל את ההודעה הבאה) ונעצור את השעון. לכן, במקרה של איבוד פאקטות , כלומר לא קיבלנו ACK עבורן, אך התקבלה הודעת ACK על פאקטה שנשלחה אחריהן, נדע שהלקוח קיבל את אותן פאקטות אך לא הגיעה עבורן הודעת ACK לשרת.

המנגנון עובד עם ACK מצטבר- מאשר את כל החבילות ברצף עד נקודה מסוימת בלי קפיצות.

אם לא התקבלה הודעת ACK עבור הפאקטה הנכונה, נעדכן את משתנה הסכימה להיות **רק** מס ההודעה של הפאקטה וננסה לקבל הודעה עבור אותה פאטקה מחדש (נגדיר משתנה next-pack שיהיה שווה ל counter_packets , באמצעותו נדע שהפאקטה הבאה שאמורה להתקבל היא הפאקטה החוזרת שנשלחה). בדרך זו התקבלו הפאקטות לפי הסדר ונוכל לוודא כי אכן התקבלו כל הפאקטות

מהצד של הלקוח: נגדיר משתנה expected , באמצעותו נוכל לדעת את מספר הפאקטה שאמורה להתקבל. עבור כל פאקטה שאנחנו מקבלים מהשרת נבדוק אם זו אכן הפאקטה הנכונה. במידה וכן נעדכן את expected++ , נוסיף את הפאקטה ל packets_list שהגדרנו עבור כל החבילות שהתקבלו ונשלח לשרת הודעת Acknow-ledgment עבור אותה הפאקטה עם מספרה. אם לא קיבלנו את החבילה שצפינו לקבל, כלומר קיבלנו חבילה לא בסדר, נשלח לשרת הודעת Acknow-ledgment עם מספרה של הפאקטה אחת לפני אחרונה שהתקבלה. בכל הודעת ACK נסתכל על רצף של סדרה שהתקבלה, וננסה לקבל שוב את החבילה שלא קיבלנו. אם לא היה Timeout אך לא הצלחנו לקבל חבילה, נעשה sleep וננסה לקבל בשנית.

בדרך זו, יהיה לנו מעקב עבור כל פאקטה שנשלחה והתקבלה ונבטיח שאם לא קיבלנו הודעת ACK מהלקוח עבור פאקטה בזמן שהוגדר, או שהתקבלה מהשרת פאקטה שמספרה הסידורי לא כפי שצפינו, נדע לשדר מחדש משני הכיוונים.

איתי קרואני 311504914
טל מלחי 208278556

שאלה 3

ה latency מושפע ממספר גורמים: עיבוד ואיבוד הפאקטות, עומס ברשת, תורים בראוטר ועוד.
מבחינת התמודדות עם ה latency , נקטנו במספר גישות:
1. עבור כל לקוח שמתחבר, נפתח טרד בשרת המטפל בבקשותיו. הטיפול נעשה במקביל לטיפול בבקשות של הלקוחות האחרים המחוברים גם כן. בכך נאפשר מקביליות בטיפול הבקשות והורדה מעומס של סך כל הפאקטות הנשלחות ברשת.
2. עבור הורדת קבצים נעשה שימוש בפרוטוקול UDP שהינו מהיר יותר ומצריך כמות קטנה יותר של פאקטות שנשלחות. בכך מוריד עומס מהרשת וחוסך זמן של לחיצות יד מיותרות, בהשוואה לפרוטוקול TCP.
3. חלון שליחה דינמי בשרת עבור שליחת קבצים ללקוח. החלון מגביל את השרת בשליחת ההודעות ללקוח בהתאם לאיבוד פאקטות וTimeout , ובכך מאפשר שליטה בקצב שליחת וקבלת הפאקטות.
4. בצד הלקוח, כאשר מתקבלת פאקטה היא נשמרת בזיכרון (מערך) ורק לאחר קבלת כל הפאקטות נכתוב את המידע לקובץ- בכך נפחית את זמן הOverhead והעיבוד של כל הפאקטות יחדיו יהיה מהיר יותר.

איתי קרואני 311504914
טל מלחי 208278556

חלק ג':

שאלה 1

כאשר מחשב מתחבר לרשת, כלומר ל SWITCH, יש לו רק כתובת MAC אותה קיבל מהיצרן, לכן המחשב צריך לקבל כתובת IP כדי שיוכל להתחבר לרשת. כלומר צריך שיהיה ברשת שרת DHCP שממנו נוכל לקבל את השירות. נשלחות 4 הודעות broadcast כפי שמצורף הנ"ל:

Info	Length	Protocol	Destination	Source	Time
DHCP Discover - Transaction ID 0x13cae6a0	331	DHCP	255.255.255.255	0.0.0.0	…0.000000 1
DHCP Offer - Transaction ID 0x13cae6a0	590	DHCP	255.255.255.255	10.0.2.3	…0.000000 2
DHCP Request - Transaction ID 0x13cae6a0	343	DHCP	255.255.255.255	0.0.0.0	…0.007312 3
DHCP ACK - Transaction ID 0x13cae6a0	590	DHCP	255.255.255.255	10.0.2.3	…0.011571 4

הודעה ראשונה- כתובת המחשב שולח הודעת broadcast שבודקת אם קיים שרת DHCP ברשת.
סוג ההודעה: (DHCP-DISCOVER UDP)
פירוט ההודעה: הלקוח שולח TRANSACTION ID , דגל עם הIP שלו, הודעת DISCOVER לשרת ה DHCP ואת ה- HOST NAME שלו.
IP/PORT מקור - 0.0.0.0:68
IP/PORT יעד- 255.255.255.255:67
כתובות MAC מקור- 08:00:27:b8:da:c8
כתובות MAC יעד- ff:ff:ff:ff:ff:ff

הודעה שניה- השרת DHCP עונה בהודעת broadcast ומציע כתובת IP ללקוח
סוג ההודעה: (DHCP-OFFER UDP)
פירוט ההודעה: TRANSACTION ID, כתובת IP שמוצעת ללקוח, MACADRESS של הלקוח, כתובת של השרת DHCP, SUBNETMASK , כתובת IP של הראוטר , כתובת IP של הDNS וזמן "השכרה".
IP/PORT מקור – 10.0.2.3:67
IP/PORT יעד- 255.255.255.255:68
כתובות MAC מקור- 08:00:27:a6:31:83
כתובות MAC יעד- ff:ff:ff:ff:ff:ff

הודעה שלישית – המחשב שולח הודעת broadcast לשרת ומבקש את הכתובת שהוקצתה בהודעה השנייה או כתובת חלופית על פי דרישה
סוג ההודעה: (DHCP-REQUEST UDP)
פירוט ההודעה TRANSACTION ID , בקשה לרשימת פרמטרים הנחוצים לחיבור, כתובת הIP המבוקשת, כתובת של השרת DHCP ממנו הוא מבוקש, HOSTNAME
IP/PORT מקור - 0.0.0.0:68
IP/PORT יעד- 255.255.255.255:67
כתובות MAC מקור- 08:00:27:b8:da:c8
כתובות MAC יעד- ff:ff:ff:ff:ff:ff

איתי קרואני 311504914
טל מלחי 208278556

הודעה רביעית- שרת ה DHCP שולח הודעת broadcast למחשב שמאשרת את כתובת הIP
סוג ההודעה: (DHCP-ACK UDP)
פירוט ההודעה: TRANSACTION ID ,כתובת IP שמוצעת ללקוח, MACADRESS של הלקוח, כתובת של השרת DHCP , SUBNETMASK , כתובת IP של הראוטר , כתובת IP של הDNS וזמן "השכרה".
IP/PORT מקור - 0.0.0.0:67
IP/PORT יעד- 255.255.255.255:68
כתובות MAC מקור- 08:00:27:a6:31:83
כתובות MAC יעד- ff:ff:ff:ff:ff:ff

No.	Time	Source	Destination	Protocol	Length	Info
5	7.977348202	10.0.2.14	10.0.2.13	TCP	74	57062 → 55000 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=2058555891 TSecr=0 WS=128
6	7.978065586	10.0.2.13	10.0.2.14	TCP	74	55000 → 57062 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM=1 TSval=1717961920 TSecr=2058555891 WS=128
7	7.978130816	10.0.2.14	10.0.2.13	TCP	66	57062 → 55000 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=2058555892 TSecr=1717961920
8	7.979987773	10.0.2.13	10.0.2.14	TCP	95	55000 → 57062 [PSH, ACK] Seq=1 Ack=1 Win=65280 Len=29 TSval=1717961921 TSecr=2058555892
9	7.980061204	10.0.2.14	10.0.2.13	TCP	66	57062 → 55000 [ACK] Seq=1 Ack=30 Win=64256 Len=0 TSval=2058555894 TSecr=1717961921
10	13.658126278	10.0.2.14	10.0.2.13	TCP	79	57062 → 55000 [PSH, ACK] Seq=1 Ack=30 Win=64256 Len=13 TSval=2058561572 TSecr=1717961921
11	13.658781786	10.0.2.13	10.0.2.14	TCP	66	55000 → 57062 [ACK] Seq=30 Ack=14 Win=65280 Len=0 TSval=1717967600 TSecr=2058561572
12	13.659440823	10.0.2.13	10.0.2.14	TCP	97	55000 → 57062 [PSH, ACK] Seq=30 Ack=14 Win=65280 Len=31 TSval=1717967601 TSecr=2058561572
13	13.659448911	10.0.2.14	10.0.2.13	TCP	66	57062 → 55000 [ACK] Seq=14 Ack=61 Win=64256 Len=0 TSval=2058561573 TSecr=1717967601
14	22.129964274	10.0.2.14	10.0.2.13	TCP	84	57062 → 55000 [PSH, ACK] Seq=14 Ack=61 Win=64256 Len=18 TSval=2058570044 TSecr=1717967601
15	22.130768684	10.0.2.13	10.0.2.14	TCP	66	55000 → 57062 [ACK] Seq=61 Ack=32 Win=65280 Len=0 TSval=1717976072 TSecr=2058570044
16	22.131116113	10.0.2.13	10.0.2.14	TCP	84	55000 → 57062 [PSH, ACK] Seq=61 Ack=32 Win=65280 Len=18 TSval=1717976073 TSecr=2058570044
17	22.131124460	10.0.2.14	10.0.2.13	TCP	66	57062 → 55000 [ACK] Seq=32 Ack=79 Win=64256 Len=0 TSval=2058570045 TSecr=1717976073

הודעה חמישית- הלקוח שולח בקשת חיבור לשרת. עם מספר רצף התחלתי
סוג ההודעה: (TCP-SYN)
פירוט ההודעה: סוג הפרוטוקול, CHECK-SUM, SEQ-NUM , חותמות זמן, גודל הפקטה המקסימילי, גודל החלון, דגל של SYN
IP/PORT מקור – 10.0.2.14:57034
IP/PORT יעד- 10.0.2.13:55000
כתובות MAC מקור- 08:00:27:b8:da:c8
כתובות MAC יעד- 08:00:27:96:d0:ed

הודעה שישית- השרת עונה ללקוח על הבקשה לחיבור, ומחזיר אישור שקיבל את הבקשה
סוג ההודעה: (TCP-SYN,ACK)
פירוט ההודעה: סוג הפרוטוקול, SEQ-NUM , CHECK-SUM,ACK-NUM , חותמות זמן, גודל הפקטה המקסימילי, גודל החלון, דגל של SYN , דגל ACK
IP/PORT מקור –10.0.2.13:55000
IP/PORT יעד- 10.0.2.14:57034
כתובות MAC מקור- 08:00:27:96:d0:ed
כתובות MAC יעד- 08:00:27:b8:da:c8

הודעה שביעית- הלקוח מחזיר הודעה לשרת שקיבל את האישור לחיבור
סוג ההודעה: (TCP-ACK)
פירוט ההודעה: סוג הפרוטוקול, SEQ-NUM , CHECK-SUM,ACK-NUM, חותמות זמן, גודל הפקטה המקסימילי, גודל החלון , דגל ACK
IP/PORT מקור – 10.0.2.14:57034
IP/PORT יעד- 10.0.2.13:55000
כתובות MAC מקור- 08:00:27:b8:da:c8
כתובות MAC יעד- 08:00:27:96:d0:ed

איתי קרואני 311504914
טל מלחי 208278556

הודעה שמינית – הלקוח שולח הודעה לצ'ט דרך השרת
סוג ההודעה: (TCP-PUSH)
פירוט ההודעה: סוג הפרוטוקול, CHECK-SUM,SEQ-NUM , חותמות זמן, גודל החלון, דגל PUSH , גודל הDATA, HEADER
IP/PORT מקור – 10.0.2.14:57034
IP/PORT יעד- 10.0.2.13:55000
כתובות MAC מקור- 08:00:27:b8:da:c8
כתובות MAC יעד- 08:00:27:96:d0:ed

הודעה תשיעית- השרת שולח את ההודעה של הלקוח ללקוח עצמו ולשאר הלקוחות שמחוברים לצ'ט, ואישור על קבלת ההודעה הקודמת.
סוג ההודעה: (TCP-PUSH ACK)
פירוט ההודעה: סוג הפרוטוקול, SEQ-NUM , CHECK-SUM,ACK-NUM , חותמות זמן, גודל החלון , דגל ACK , דגל PUSH , גודל הDATA, HEADER
IP/PORT מקור –10.0.2.13:55000
IP/PORT יעד- 10.0.2.14:57034
כתובות MAC מקור- 08:00:27:96:d0:ed
כתובות MAC יעד- 08:00:27:b8:da:c8

הודעה עשירית- הלקוח מחזיר הודעה לשרת שקיבל את ההודעה
סוג ההודעה: (TCP-ACK)
פירוט ההודעה: סוג הפרוטוקול, SEQ-NUM , CHECK-SUM,ACK-NUM ,חותמות זמן, גודל הפקטה המקסימילי, גודל החלון , דגל ACK
IP/PORT מקור – 10.0.2.14:57034
IP/PORT יעד- 10.0.2.13:55000
כתובות MAC מקור- 08:00:27:b8:da:c8
כתובות MAC יעד- 08:00:27:96:d0:ed

שאלה 2

CRC – סוג של קוד לאיתור שגיאות כאשר מעבירים נתונים. הקוד מקודד הודעות ע"י הוספה של ערך בדיקה על גודל קבוע. את הערך הקבוע מחשבים ומוסיפים לפני שליחת ההודעה בצד של השרת. לאחר שליחת ההודעה הלקוח מחשב את ה CRC מהצד שלו.
אם קיימת התאמה בין ה CRC של השרת לשל הלקוח, ההודעה התקבלה בשלמותה וללא שגיאות.

שאלה 3

HTTP 1.0 - פתיחת קשר TCP , שליחת אובייקט אחד, ולאחר מכן מתבצעת התנתקות מה TCPכלומר עבור כל בקשה אחת נפתח קשר אחד. השיטה לא טובה כיוון שיכולה לגרום לעיכובים כי מתבצעת פתיחת קשר מחדש. לא דוחס את ה header ולא עובד בסדרי עדיפויות

HTTP 1.1 - פתיחת קשר TCP, שליחת כמה אובייקטים יחדיו, ולאחר מכן מתבצעת התנתקות מהTCP .
שיטה זו יעילה ומשפרת את זמן הגלישה. קיימות 2 גישות:
NP- נשלח בקשה רק לאחר שקיבלנו את התשובה לבקשה הקודמת.
P- נשלח כמה בקשות ואז נקבל את התשובות ביחד
השיטה מאפשרת כמה בקשות עם התחברות TCP יחידה.

איתי קרואני 311504914
טל מלחי 208278556

HTTP 2.0– פתיחת קשר TCP יחיד. שיטה זו מאפשרת לשרת לשלוח תגובות מרובות לבקשת לקוח. השיטה מחלקת את האובייקטים ל frames וכך התעבורה עוברת יותר מהר. חסרון לשיטה- אם משהו הולך לאיבוד יכולים להיווצר עיכובים כיוון שה TCP-צריך לשדר את זה עוד פעם(TCP כמו צינור שמכניס לתוכו הרבה streams). דוחסת את ה header ומאפשרת לשרת לשלוח תגובות מרובות לבקשה יחידה (push)

QUIC- ההבדלים המרכזיים בין פרוטוקול QUIC ל HTTP 2.0 , הם ש QUIC פועל על גבי חיבור UDP ומאפשר לטפל באיבוד פאקטות ללא הגבלה של שאר הזרמים בחיבור, לעומת HTTP2, ובכך מונעת עיכובים ופגיעה בשירות. בנוסף, בחיבור TCP נדרשת לחיצת יד לעומת, פרוטוקול QUIC על גבי UDP ובכך נחסך זמן רב של יצירת החיבור

שאלה 4

למכשיר קיימת כתובת IP אשר דרכה אנחנו מקבלים נתונים. על מנת שנוכל לדעת לאיזה שירות במכשיר לקשר את ההודעה שהתקבלה, קיימים מספרי פורטים אשר באמצעותם המכשיר יידע לאיזה שירות לנתב את ההודעה, ואליו לגשת. באמצעות מספרי הפורטים שבבת התעבורה תדע לאיזה אפליקציה לתת שירות ולאן לגשת

שאלה 5

חלוקה של כתובת IP לתתי רשתות.

למה צריך את זה?
- מניעת עומסים- בארגונים גדולים, קיימים הרבה host-ים, קיימת הרבה תעבורה, אשר כל המחשבים ברשת נחשפים אלייה. ביצירת sunbet אנחנו מקטינים את הרשת הפנימית ובכך קיימים פחות משתמשים באותה רשת והתעבורה קטנה כלפי כל לקוח.

- בטיחות- לקוח יכול לראות את התעבורה רק בתת רשת שהוא נמצא בה, לכן בעת פריצה לרשת, פורץ שהצליח להיכנס לאותה subnet לא יהיה מסוגל לגשת לחלקים אחרים של האירגון היושבים בסאבנטים אחרים.

- חוסך בכתובות IP- במקום שהספק יצטרך לספק לנו כתובות IP נוספות, כיוון שכתובות ה IPv4 חסומות (2 בחזקת 32 כתובות, מפורט בשאלה 8), יהיה לנו כתובות נוספות ע"י חלוקת כל כתובת IP יחידה לסבאנטים. בדרך זו ניצור כתובות IP נוספות תחת אותה תת רשת.

שאלה 6

למכשיר שמתחבר לרשת יש כתובת פיזית קבועה (MAC), בחיבור ראשוני לרשת, מתבצעת הקצאה של IP עבור אותו מכיר ע"י שירות DHCP . מכיוון שלמחשב אין עדין כתובת IP , על מנת שהשירות יידע לאן להחזיר את הכתובת IP שהוקצתה, הוא משייך לפי הכתובת MAC של המכשיר.
כתובת הMAC הנה ייחודית עבור כל מכשיר לעומת כתובת IP , וכבה נוכל להבטיח כי המידע המועבר יגיע למקום הנכון (דוגמא: כמה מחשבים ברשת שמאזינים לאותו פורט)

איתי קרואני 311504914
טל מלחי 208278556

שאלה 7

על מנת לתקשר ביו התקני תקשורת ברשת פנימית נשתמש בSWITCH, התקשורת תהיה מהירה יותר כיוון שאין צורך להמיר כתובות. תקשורת בין רשת חיצונית לרשת שלנו מצריכה ROUTER וכדי להמיר את הכתובת הפנימית לחיצונית כדי שנוכל לתקשר נשתמש בNAT.

SWITCH- מעביר תקשורת בין מכשירים שנמצאים באותה הרשת. התקשורת מתבצעת עי כתובות MAC ADDRESS (מס מזהה ייחודי עבור כל מכשיר שמגיע מהיצרן. כתובת זו מורכבת ומוטבעת על כל מכשיר שמחובר לרשת ומורכבת משדות של מספרים ואותיות).
לכל SWITCH יש טבלה פנימית שבה הוא רושם את ה- MAC ADDRESS של כל התקן שמתחבר לאחד הפורטים שלו, ובכך הוא יודע לתת לשני פורטים לדבר ביניהם כאשר מחשב אחד מחפש מחשב אחר ברשת לפי ה MAC. עובד בשכבת הLINK

ROUTER- כאשר הרשת הפנימית צריכה להתחבר לאינטרנט היא תשתמש בROUTER. כלומר הROUTER עובד עם כתובות IP ויכול לקשר בין רשתות שונות ובדרך זו הרשת שלנו תוכל "לצאת" ולדבר עם רשתות אחרות. כלומר תפקידו העיקרי לנתב תעבורה מרשת אחת לאחרת. עובד בשכבת ה NETWORK

NAT- ממיר כתובת IP פנימית לחיצונית. כדי שה ROUTER יוכל לנתב תעבורה הוא צריך להמיר את כתובת הIP הפנימית לכתובת חיצונית והוא יעשה זו באמצעות ה NAT.
כתובת ופורט היעד תמיד נשמרים גם שהמידע יעבור דרך ראוטרים אחרים הנמצאים ברשת.

שאלה 8

אחת השיטות להתגבר על מחסור בכתובות IPv4 היא כתובות הIPv6 שמכילה 128 ביטים ובכך מגדילה משמעותית את כמות כתובות הIP האפשריות. כתובות הIPv6 מיוצגת באמצעות סדרה של 8 מספרים (4 ספרות כל אחד) בבסיס הקסדצימלי, שכל אחד מהם מייצג 16 ביטים.
דרך נוספת להתגבר על המחסור הוא הNAT . מאפשר למספר מכשירים שנמצאים תחת אותה רשת מקומית לחלוק את אותה כתובת IPv4 ודרכה יתקשרו עם העולם החיצון- כלומר מה שנמצא מחוץ לרשת.
לכל חבילה שתצא מהרשת ללא קשר למחשב שהיא יצאה ממנו, תהיה בעלת אותה כתובת IP של הNAT וההבדל בין המחשבים יהיה בפורט שממנו יוצאת הפאקטה.
הפאקטות שנשלחות ברשת נשלחות בצורה רגילה עם כתובת הIP הרגילה של המחשבים ברשת.
בנוסף: * שימוש ברשת פרטית- מאפשרת שימוש בחיבורים בתוך הרשת, ללא כתובת (אך כל יציאה לאינטרנט מחייבת לעשות שימוש בכתובת השער)
* שרת מאחר וירטואלי – מאפשר לעשות שימוש בכתובת של השרת המארח

שאלה 9

e - ניתן לראות ברשת שנתונה כי הנתב 3c שנמצא ב AS3, מחובר בקשר פיזי ישיר ל AS4 .
כידוע, התקשורת בין הרשתות מתבסס על פרוטוקול BGP , אשר מאפשר לכל רשת משנה (subnet) לפרסם את קיומה ואת היעדים אליהם ניתן להגיע בעזרתה. הפרוטוקול מאפשר לכל AS :
eBGP – אוסף את היעדים אליהם ניתן להגיע מרשתות AS סמוכות.
iBGP – מפיץ מידע נגיש לכל הנתבים הפנימים של הAS, ומאפשר לקבוע מסלולים "טובים" לרשתות אחרות.
כיוון שבין כל הAS-ים רץ פרוטוקול BGP , ניתן להגיד שהנתב 3c לומד על תת רשת X באמצעות פרוטוקול eBGP.

f- הנתב 3a שנמצא ב AS3, לא מחובר בקשר ישיר ל AS4. כפי שנאמר בשאלה על AS3 מריצים OSPF – אלגוריתם ניתוב שמבוסס על אלגוריתם link-state. ניתוב זה הוא גלובלי ועובד בצורת הצפה. יודעים את הטופולוגיה השלמה של הרשת ואת העלות של כל מסלול ישיר בן נתבים.

איתי קרואני 311504914
טל מלחי 208278556

בכל שינוי כל נתב מציף את המידע שברשותו אל הרשת כולה וכך המידע מסתנכרן בין כל הנתבים. לכן אפשר להגיד שהנתב 3a, לומד על תת רשת X באמצעות פרוטוקול הניתוב, iBGP, אשר מ פיץ מידע נגיש לכל הנתבים הפנימיים של הAS, כלומר הוא לומד באמצעות פרוטוקול הניתוב OSPF שרץ באותו AS .

g- הנתב 1c שנמצא ב AS1, מחובר בקשר פיזי ישיר ל AS3 .
כפי שנאמר, בין כל הAS-ים רץ פרוטוקול BGP . לכן כפי שהוסבר בסעיף e, ניתן להגיד ש 1c, לומד על תת רשת X דרך AS3 , (???שלומד על תת רשת X דרך AS4 ???) כלומר באמצעות פרוטוקול eBGP.

h – ניתן לראות כי הנתב 2c שנמצא ב AS2, לא מחובר בקשר ישיר ל AS4. כפי שנאמר AS2 מריצים OSPF – אלגוריתם ניתוב שמבוסס על אלגוריתם link-state כפי שהוסבר בסעיף f . לכן ניתן להגיד שבעת שינוי מתבצעת הצפה של הרשת. כל הנתבים שקשורים לנתב שהשתנה מעבירים את המידע הלאה לכל הנתבים איתם הם שכנים, וכן הלאה עד שכל הרשת מודעת לשינוי. כל נתב מחזיק את אותו המידע כמו כל הנתבים. לכן אפשר להגיד שהנתב 2c, לומד על תת רשת X באמצעות פרוטוקול הניתוב iBGP, כלומר באמצעות פרוטוקול OSPF שרץ באותו אזור AS .