חלק ב'

- המערכת להורדת קבצים מהשרת שבנינו עובדת על UDP אמין בעזרת Go Back N, איך זה עובד?
רוחב הפס (מספר הפקטות שנשלחות בכל פעם) הוא N פקטות, במקרה שלנו N=4. הפקטות נשלחות אחת אחרי השנייה, כאשר פקטה מגיעה אל הלקוח נשלחת בחזרה הודעת ack לשרת שאכן הגיעה הפקטה.
על מנת שהעברת הקבצים תהיה אמינה ונדע האם לשלוח את הפקטות שוב, בתחילת שליחת הפקטות מוגדר מעין סטופר המודד כמה זמן עובר מרגע שליחת הפקטות ועד הרגע שמגיעה הודעת ack.
במידה ועבר הזמן שנקבע ולא התקבלה הודעת ack כל ארבעת החבילות האחרונות תישלחנה שוב והתהליך יחזור על עצמו.
באופן זה אנחנו מתמודדים עם שני דברים:

1. איבוד פקטות- אחרי כל חבילה שנשלחת, נשלחת הודעת ACK מהלקוח לסרבר כדי שידע 'החבילה ששלחת הגיעה', דבר זה מבטיח אמינות, כאשר לא מגיעה הודעת ACK סימן הפקטת מידע נאבדה בדרך וכך הסרבר יודע שעליו לשלוח שוב את 4 הפקטות מהתחלה.
2. בעיית latency- היות שכאשר נשלחות הפקטות מהסרבר ללקוח מתחיל איזשהו טיימר אשר סופר את הזמן שעובר עד שמגיעה הודעת ACK אנחנו בעצם מתמודדים עם העיכוב בזמנים. כלומר, אם עד סיום הזמן שנקבע מראש לא הגיעה הודעת ACK עבור הפקטה, כלומר, לוקח לפקטה יותר מידי זמן להגיע, אז הסרבר שולח מחדש את חבילת הפקטות.

חלק ג'

1. כאשר מחשב רוצה להתחבר לאינטרנט קודם כל הוא בודק מה הכתובת IP שלו, שכן, בשלב זה למחשב יש רק את הכתובת MAC של הכרטיס רשת שלו וזאת כי הנ"ל צרוב לו באופן פיזי על כרטיס הרשת.
כדי לקבל כתובת IP ופרטי רשת הדרך הכי נפוצה בימינו היא על ידי פרוטוקול DHCP.
נחדד שניתן להקצות כתובת IP באופן ידני אך בדרך כלל היא מוקצית אוטומטית על ידי שרת DHCP.
כרטיס הרשת שלנו שולח הודעה DHPC Disscover לכולם (broadcast).
כדי שההודעה תגיע אל שרת הDHCP המחשב והשרת צריכים להיות באותו broadcast domain.
שרת הDHCP רואה את הבקשה שלנו ומחזיר DHCP Offer, הודעה המכילה את כל פרטי הרשת המפורטים בהמשך.

השרת הנ"ל מקצה כתובת עבור לקוח ברשת הלוקאלית לזמן מוגבל (lease time). בין היתר הוא מקצה גם את הsubnet mask, כתובת שרת הDNS, וכתובת הgateway.
המחשב שולח הודעת DHCP Request שמודיעה לשרת DHCP שהוא מקבל את ההצעה.
השרת יחזיר הודעת ack שבין היתר מכילה גם את זמן ההקצאה של הכתובת IP.

DHCP פעול מעל פרוטוקול התעבורה UDP ומעל פרוטוקול הרשת.
השרת משתמש בפורט 67 והלקוח משתמש בפורט 68.

כל עוד הלקוח לא קיבל כתובת IP הוא משתמש באפסים כדי לייצג את הכתובת שלו והוא מזוהה על ידי הכתובת הפיזית שלו (MAC).

כעת, כדי שהמחשב ידע מה הכתובת של האתר אינטרנט בו הוא רוצה להשתמש לצורך הצ'אט, המחשב ישתמש בפרוטוקול DNS וישלח שאילתת DNS לשרת ה-DNS וישאל מה הכתובת IP של הצ'אט.

הקליינט (מחשב) פונה אל שרת הDNS שלו עם כתובת IP הפנימית שלו וכתובת הMAC שלו.
הפקטה נשלחת בפורט 53 בפרוטוקול DNS.

השרת מחזיר תשובה לקליינט עם כתובת הIP של שרת הצאט. הפקטה שנשלחת לשרת יוצאת עם הIP של השרת (כתובת חיצונית) ומוחזרת למחשב שנמצא ברשת הפנימית.

SESSION TCP
הקליינט פונה אל רכיב הNAT, NAT מפנה את הבקשה אל שרת הצאט.
בשלב זה מתחיל TCP 3wayHandshake כדי לפתוח קשר אמין.
הקשר הוא דו כיווני, הלקוח פונה לשרת עם שאלה "אתה שם?", השרת מחזיר לו תשובת ACK, ואז הלקוח מחזיר לשרת גם כן הודעת ACK.

הקליינט פונה לNAT בעזרת הIP הפנימי שלו והוא מעביר הלאה את ההודעה של הקליינט עם הIP הכללי לאתר האינטרנט. אתר האינטרנט שולח הודעה בחזרה התקשורת בין שניהם היא כמו מעין צ'אט. התקשורת עוברת דרך פרוטוקול HTTP ופורט 80.

נציין, כאשר הלקוח פונה לIP חיצוני הבקשה עוברת דרך רכיב NAT. איך זה מתבצע?
המחשב פונה עם הIP הפנימי שלו לIP הפנימי של הNAT.
הNAT משתמש בכתובת MAC של המחשב שפנה אליו וכן בכתובת הIP הכללית שהוא מביא למחשבים שמחוברים אליו ושולח את הבקשה לאתר אינטרנט המבוקש.

במידה ומדובר בתקשורת בין שני מחשבים שלא נמצאים תחת אותה רשת יש צורך בנתב שינתב את המידע.
במידה ומדובר בתקשורת בין שני מחשבים שנמצאים תחת אותה רשת. המחשב A יודעת את הכתובת IP של מחשב B (באמצעות פרוטוקול DNS) אבל זה לא מספיק, על המחשב לדעת גם את כתובת הMAC של כרטיס הרשת של מחשב B, מכיוון שחבילת הPing תשלח ממחשב A צטויה להיות בנויה משכבת הEthernet, IP, ולבסוף ICMP. כדי לבנות את מסגרת הEthernet על מחשב A לדעת את כתובת היעד של המסגרת שהיא כתובת כרטיס הרשת של מחשב B.
דבר נוסף, כרטיס הרשת של B צריך לדעת שהוא היעד ולכן הוא מסתכל בכתובת היעד של המסגרת. כרטיס הרשת לא מבין כתובות IP כי הוא שייך לשכבת הקו ולכן הכרטיס צריך לראות את כתובת הMAC.

לשם כך, על מחשב A להבין את כתובת הMAC של מחשב B על ידי פרוטוקול ARP.
התפקיד של פרוטוקול זה הוא למפות בין כתובות לוגיות של שכבת הרשת לכתובות הפיזיות של שכבת הקו.
במקרה שלנו, מחשב A ישלח שאלה לכל הרשת (כתובת broadcast)- "למי יש את הכתובת MAC של מחשב B?"
מי שצפוי לענות למחשב A מבין כל המחשבים שקיבלו את ההודעה הוא מחשב B אשר יודע שהכתובת MAC שלו.
באופן זה מחשב A מגלה את כתובת הMAC של מחשב B וכעת הוא יכול לשלוח חבילת Ping ולהמשיך את ההתקשרות.

2. CRC- Cryclic redundancy check דומה במהותו לchecksum כלומר, הוא מנגנון לאיתור שגיאות בעת העברת נתונים.
את הCRC מחשבים באופן הבא-
מדובר בפולינום יוצר, המקדמים של הפולינום מדרגה r (אורך הdata)
- מוסיפים r מימין להודעה, (r כדרגת הפולינום)
- נחלק בפולינום עצמו, תוך שימוש במודולו 2 (כיוון שאנחנו בשדה של 0 או 1)
- נחסר את השארית בעזרת xor
- נצרף את התוצאה שהתקבלה מימין להודעה המקורית

לאחר החישוב, הוא מתווסף לקובץ מידע שאנחנו מעבירים וכאשר מקבל המידע מקבל את הקובץ הוא בודק את הCRC אם אכן הגיע בצורה תקינה, ללא שינויים.

3. QUIC הינו פרוטוקול שנבנה על ידי גוגל. מצוי וניתן לשימוש הרחב בספריות של פייתון.
המטרה שלו הוא להיות פרוטוקול מעל UDP אך עם אמינות כמו של TCP, מדובר בפרוקטוקל שמשפר את היעילות בכ- 75%.

HTTP1.0 פתיחת connection לבקשה ואחרי כל בקשה הוא סוגר את הconnection.
לעומת, HTTP1.1 אשר נשאר פתוח גם לאחר סיום הבקשה, דבר זה גורם לתוצאות וביצועים יותר טובים.
HTTP2.0 מצליח לעשות ניהול על הconnection כלומר, הוא שולח על אותו TCP כמה קבצי

מידע.

לסיכום, QUIC הינו הכי מהיר ויעיל, היחיד שמבוסס על פרוטוקול UDP, השאר מבוססים על פרוטוקול TCP.

4. קיימת דרישה למספרי port על מנת שהמידע ידע לאיפה לעבור ולאיפה להגיע בתוך המחשב כלומר, port הוא מעין ערוץ תקשורת בין רשתות שונות.
הרעיון מאחורי הפורט היה לייצר דרך שמספר תוכניות יוכלו להשתמש באותה כתובת IP.
מספרי port נעים בין 1 ל- 65000.

5. תפקיד ה-subnet היא לסווג באיזה סוג רשת מדובר.
subnet mask מגדיר כמה ביטים מתוך כתובת הIP מייצגים את מזהה הרשת. מדובר במספר בינארי בעל 32 סיביות המשמש לקביעת מזהה הרשת ומזהה היישות של כתובת הIP.
כתובת תת־הרשת שתיווצר היא הכתובת שתתקבל לאחר הפעולה הבינארית בין ה-IP ולבין ה-Subnet mask.
מתוך הכתובת שתיווצר, הביטים הדולקים מהווים את מזהה הרשת של הIP והכבויים מהווים את מזה הישות.

6. כתובת MAC וכן כתובת IP משמשות למזהה של מכשיר ברשת האינטרנט.
כתובת MAC הינה מזהה ייחודי שניתן לכל מכשיר אלקטרוני שמוטבע בכרטיס הרשת בשעת הייצור- נצרבת פיזית על כרטיס הרשת של המכשיר, לעומת כתובת IP שזו כתובת אשר מהווה מעין "תעודת זהות" לרשת הנוכחית- מספר המשמש לזיהוי נקודות קצה, אשר משתנה בין רשת לרשת כלומר, היא לא כתובת קבועה (יכולה להיות אותה כתובת IP לשני מחשבים שונים שלא נמצאים באותה רשת, דבר שלא יכול לקרות עם MAC).
ההבדלים:
- כתובת MAC מורכבת מ6 בתים האקסה דצימלים.
לעומת כתובת IP שנעה בין 4 בתים אם מדובר בIPV4 או ב6 בתים, אם מדובר בIPV6.
- מכשיר יכול בעזרת הכתובת MAC לשלוף מידע דרך פרוטוקול ARP.
לעומת כתובת IP שאפשר לשלוף בעזרתה מידע דרך פרוטוקול RARP.
- כתובת הMAC פועלת בשכבת הdata link.
לעומת כתובת הIP שפועלת בשכבת הnetwork.
- כתובת MAC היא המזהה (הכתובת) הפיזי של המחשב.
לעומת כתובת IP שהיא מזהה (כתובת) לוגי של המחשב.
- כתובת MAC של מכשיר היא אחת ויחידה, אין עוד מכשיר שמיוצר עם כתובת זו.
לעומת כתובת IP, יכול להיות שמספר מכשירים שונים מיוצגים על ידי אותה כתובת IP.

7. ראוטר הינו רכיב תקשורת שמטרתו לנתב את ההפצה של חבילות נתונים- פקטות ברשתות תקשורת נתונים. כשמו כן הוא, נתב שמנתב את המידע מרשת אחת לרשת אחרת בהתבסס על הכתובת שלהן. לדוג, מרשת ביתית לרשת האינטרנט.
הנתב עובד בשכבת הרשת.
לראוטר הרבה ממשקים אליהם הוא מעביר את הפקטות. הוא עושה זאת בעזרת טבלת הניתוב המקומית כדי שידע "מה לעשות" עם המידע שהתקבל.
דבר זה יכול להקבע באופן סטטיסטי או באופן מחושב בצורה דינאמית, על ידי פרוטוקולי ניתוב כגון: BGP, OSPF.

SWITCH הינו רכיב המחבר בין צמתים שונים ברשת, כגון מכשירי קצה או רכיבי רשת בסיסיים.
עובד בשכבה הפיזית, עובד לפי , כתובת MAC כלומר הוא מבצע העברה של נתונים לפי כתובות הMAC של המכשיר.
הswitch לומד את כתובות הMAC של הרכיבים שמחוברים אליו על ידי קריאת נתוני הבקרה בפקטות שמגיעות אליו. את הכתובות הוא שומר בטבלה פנימית (טבלת MAC).
כאשר פקטה מגיעה אל הswitch הוא בודק מה הכתובת MAC אליה מיועדת הפקטה.
אם הוא כבר מכיר את הכתובת הוא יעביר את הפקטה לport ששייך לרכיב של כתובת הMAC המדוברת.
אם הוא לא מכיר את הכתובת MAC הוא ישלח לכולם (broadcast) את הפקטה, חוץ מלפורט ששלח את ההפקטה, הport שממנו תחזור ההודעה שמדובר בו יתווסף אל טבלת הMAC.

NAT מהווה חיבור מחשבים רבים הנמצאים באותה הרשת המקומית לרשת האינטרנט באמצעות כתובת IP אחת בלבד. יישום זה שימושי לצורך צמצום כתובות ה-IP בעולם, שהרי במקום שלכל מחשב תינתן כתובת IP חיצונית למול האינטרנט, כל המחשבים באותה הרשת הלוקאלית מיוצגים על ידי כתובת אחת בלבד.
הנתב שמתחבר לאינטרנט גם מקבל מהחיבור כתובת IP חוקית ציבורית אחת. אותו נתב בדרך כלל גם יבצע את הניתוב עם NAT, ועל כן יקרא שער הגישה (gateway) של הרשת הפנימית אל רשת האינטרנט.
כלומר, למול האינטרנט וכל הרכיבים שלא נמצאים ברשת המקומית המחשבים מיוצגים עם אותה כתובת IP גלובלית ואילו בתוך הרשת המקומית לכל מחשב יש כתובת IP ייחודית לו.

דוגמא לאיך הNAT עובד:
כאשר מתקבלת פאקטת מידע מהאינטרנט אשר מיועדת לאחד ממחשבי הרשת הלוקאלית: כתובת היעד מתורגמת כאילו הפאקטה נשלחה אל המחשב המתאים עם הכתובת הפנימית, הפאקטה מנותבת מחיבור האינטרנט אל החיבור של הרשת הלוקאלית בצירוף מספר הפורט שאיתו יצאה הפאקטה מהרשת הפנימית.
דבר דומה קורה בכיוון ההפוך, כאשר מתקבלת פאקטת מידע מהרשת הלוקאלית, המיועדת לרשת האינטרנט.

NAT הינו רכיב לוגי, לעומת האחרים שלרוב הם רכיבים פיזיים.
בסופו של דבר, מדובר בתרגום בין הכתובת הפרטית לבין הכתובת באינטרנט. במהלך השנים, נוצר מצב שברשתות IPV4 נהיה מחסור בכתובות ולכן השימוש בNAT גדל והפך לפופולרי במיוחד.

7. קיימות 3 דרכים אשר מתמודדות עם המחסור בכתובות בIPV4-
הראשונה, IPV6, מורכב מ128 סיביות. 64 הסיביות הראשונות משמשות לזיהוי תת הרשת והשאר משמשות כמזהה רשת. בשונה מ32 סיביות שיש לIPV4.
היות שיש יותר סיביות, יש יותר כתובות.
השנייה, CIDR הרעיון הוא לחלק את הרווחים של הכתובת IP לחמישה חלקים, המבוססים על 4 הביטים הראשונים של הכתובת.
השלישית, NAT.

8. e- נתב הרשת 3c לומד על תת רשת x דרך פרוטוקל eBGP כיוון שמדובר בניתוב בין שני נתבי BGP השייכים ל- Asים שונים אך כן שכנים.
f- נתב הרשת 3a לומד על תת הרשת X דרך פרוטוקול iBGP כלומר, BGP פנימי כיוון שמדובר בניתוב בין שני נתבי BGP השייכים לאותו AS (נחדד, המעבר בין 4c ל-3c הוא חיצוני אך ברגע שעברנו מ3c ל-3b ועד ל3a המעבר הוא כבר פנימי בתוך אותה תת רשת).
g- נתב הרשת 1c לומד על תת הרשת X דרך פרוטוקול eBGP כיוון שמדובר בניתוב בין שני נתבי BGP השייכים ל- Asים שונים אך כן שכנים.
h- נתב הרשת 2c לומד על תת הרשת x דרך פרוטוקול iBGP.
x לומד על 4c בעזרת פרוטוקל- iBGP, כי הם נמצאים תחת אותו AS
4c לומד על 3c בעזרת פרוטוקול- eBGP
3c לומד על 3a בעזרת פרוטוקול iBGP, היות שמדובר במעבר פנימי
3a לומד על 1c בעזרת פרוטוקול eBGP היות שמדובר בניתוב בין שני נתבי BGP השייכים ל- Asים שונים אך כן שכנים.
1c לומד על 1b בעזרת פרוטוקול iBGP, כי מדובר בניתוב בין שני נתבי BGP השייכים לאותו AS.
2a לומד על 1b בעזרת פרוטוקול eBGP היות שמדובר בניתוב בין שני נתבי BGP השייכים ל- Asים שונים אך כן שכנים.
2c לומד על 2a בעזרת פרוטוקול iBGP, כי מדובר בניתוב בין שני נתבי BGP השייכים לאותו AS.
ולכן, בשורה התחתונה 2c לומד על x בעזרת פרוטוקול iBGP.