

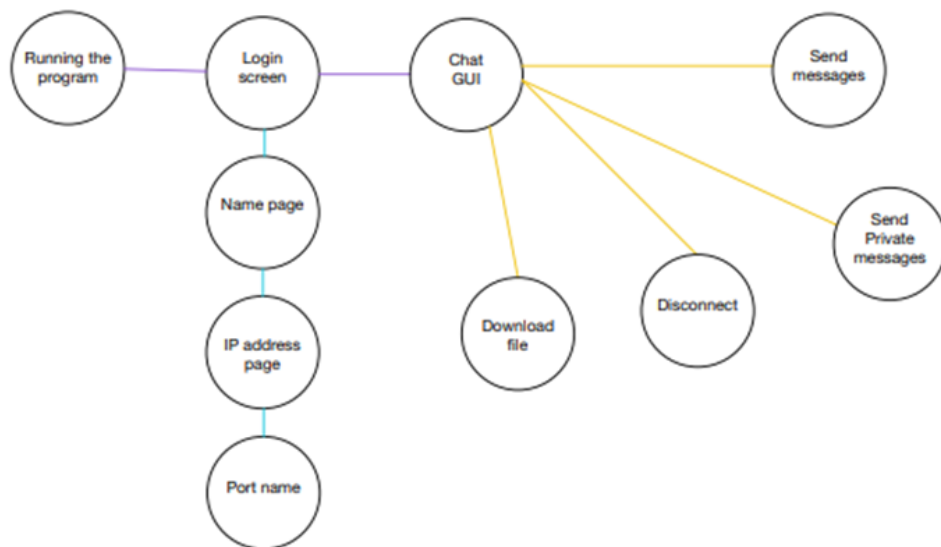
## פרויקט גמר תקשורת ומחשוב


מגשים:

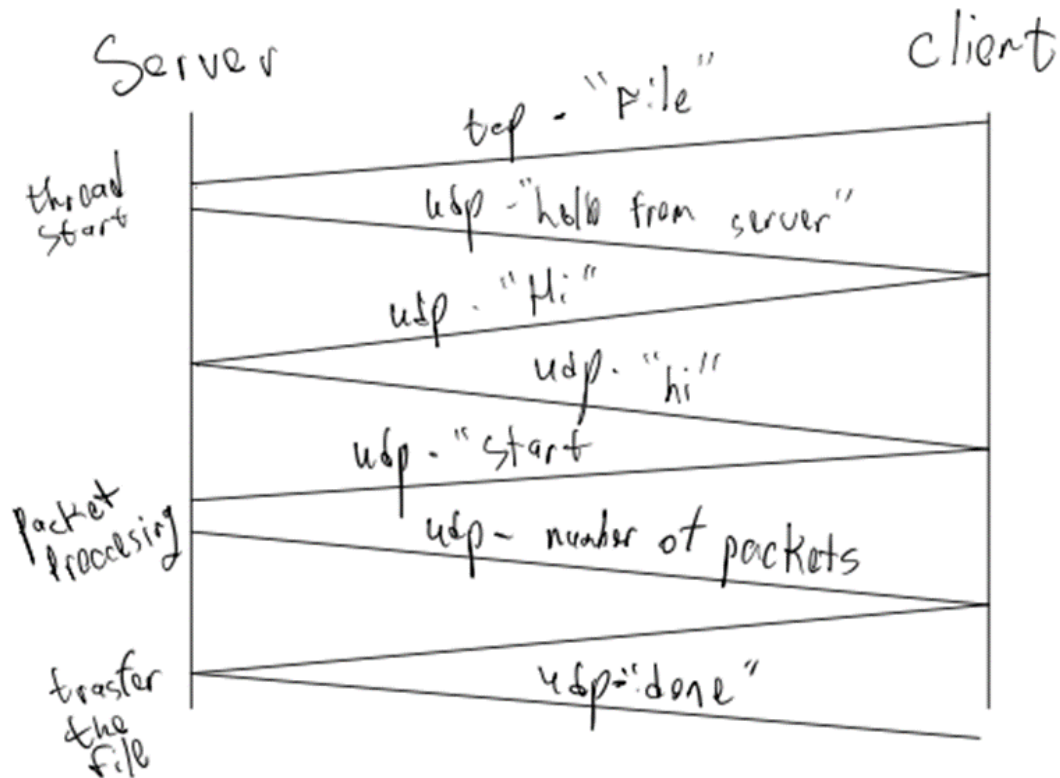
ברק עמרם 209369289

לירוי מלמד 209366970

חלק ב



-  - Continue to
-  - Open this
-  - Options for



כיצד המערכת מתגברת על איבוד חבילות / בעיות LATENCY?  
 כאשר הלקוח לוחץ על "הורדה" נשלחת הודעת tcp אשר מקוטלגת ב"file#", הפונקציה אשר מטפלת בהודעות נכנסת לתנאי של הקיטלוג הנ"ל ומפעילה טרד חדש להעברת הקובץ וחיבור הקוד, ההודעה נשלחת לסרבר עם שם הקובץ ואיזה כתובת רוצה להוריד,  
 הסרבר מחלק את הקובץ למספר הפקטות הרצוי על מנת שנוכל להעביר אותו לאחר מכן הסרבר שולח הודעת UDP לקליינט על תחילת הורדה, הקליינט מקבל את זה ושולח start, הסרבר מקבל start, שולח את מספר הפקטות שהוא עומד לשלוח ומכין את עצמו לקבל מספרים, הקליינט מקבל את גודל הפקטות שאמור לקבל ושולח מספר פקטה לפי הסדר מ0 עד גודל מספר הפקטות. עבור כל מספר שהסרבר מקבל הוא שולח את מספר הפקטה לקליינט, כאשר הקליינט מקבל פקטה הוא שולח לסרבר את מספר הפקטה הבא, כאשר הקליינט לא מקבל פקטה תוך 2 שניות (הגדרה שלנו ל - (settimeout(2) הוא מכניס את מספר הפקטה שלא קיבל ע"counter לרשימת הפקטות שאבדו וממשיך הלאה ושולח את מספר הפקטה הבאה שהוא רוצה לקבל, לאחר שהקליינט סיים לקבל את כל הפקטות כלומר רץ בלולאה לפי גודל הפקטות שאמור לקבל, בודק אם יש פקטות שלא קיבל, כלומר בודק את גודל הרשימה של packet\_lost, אם הוא גדול מ0 הוא נכנס ללולאה נוספת שבה הוא מבקש מהסרבר מספר פקטה שאבדה והסרבר שולח לו, אם הקליינט קיבל את הפקטה הוא מסיר את מספר הפקטה מהרשימה שלו, במידה ועוד פעם נאבדה פקטה הוא ממשיך הלאה, לאחר שסיים לשלוח את כל הפקטות שאבדו הקליינט בודק עוד הפעם האם נשארו פקטות ברשימה, אם כן הוא נכנס עוד הפעם ללולאה, אם לא הוא שולח לסרבר FINISH והסרבר מסיים לשלוח פקטות.  
 כמו כן עבור כל פקטה שהקליינט מקבל הוא מכניס אותה בהתאם לרשימת הפקטות שהתקבלו בהתאמה למספר המייצג את הפקטה כלומר בדיוק למקום שלו ברשימה. לבסוף כל הקובץ עובר והוא ממוין לפי המספר המייצג של הפקטות אשר חילקנו בהתחלה.

## חלק 2

1. בהינתן מחשב חדש המתחבר לרשת נתאר את כל ההודעות שעוברות החל מהחיבור הראשוני ל switch ועד שההודעה מתקבלת בצד השני של הצאט.

בשלב הראשון המחשב לא מקבל כתובת IP כי הוא חדש במערכת לכן הפרוטוקול הראשון שיופעל הוא DHCP המחשב שולח הודעה ל-DHCP כאשר המקור הוא 0.0.0.0 והיעד הוא 255.255.255.255 שזה הכתובת האחרונה ברשת כלומר הוא שולח לכל מי שנמצא ברשת כדי שהשרת DHCP יחזיר תשובה, לאחר מכן חוזרת תשובה מהשרת DHCP אשר מציעה כתובת IP הניתנת לשימוש וכן שם וכתובת ה-IP של שרת ה-DNS ומסירת השרת. פרוטוקול ה-DHCP עובד בעזרת UDP מטעמי מהירות. לאחר שהמחשב מכיר את כתובת ה-IP אשר הוא נמצא בה הוא צריך את הכתובת של המחשב שאיתו הוא רוצה לתקשר להשיג כתובת זו המחשב החדש יצטרך לשלוח בקשת DNS. בקשת ה-DNS עטופה ב-UDP (מטעמי מהירות), עטופה ב-IP, שעטוף ב-Ethernet. כדי לשלוח את הבקשת DNS נצטרך להשיג את כתובת ה-MAC של הנתב הראשון. על מנת לעשות זאת המחשב החדש יצטרך להשתמש בטבלת ARP, המחשב יאתר את הכתובת של הנתב ע"י שידור של חבילת מידע בשכבת הערוץ עם כתובת ה-MAC אשר כתובתו FF:FF:FF:FF:FF:FF בשדה היעד של ה-MAC ואת כתובת ה-IP של הנתב המבוקש, חבילה זו משודרת אל כל המכשירים המחוברים אל הרשת. המכשיר שיזהה את כתובת ה-IP שלו בהודעה, ישלח בחזרה הודעה עם כתובת ה-MAC שלו אל המחשב החדש, עכשיו המחשב החדש יודע את כתובת ה-MAC של המחשב ששלח לו הודעה ויכול לשלוח את בקשת ה-DNS. כמובן שכל זה קורה במידה והמחשב לא מכיר את כתובתו של המחשב השני. עכשיו המחשב מכיר את כתובתו של המחשב השני המחשב החדש שולח הודעת Unicast ל-Switch עם כתובת ה-IP של הצד השני עם כתובת ה-MAC FF:FF:FF:FF:FF:FF כלומר הודעת Broadcast. ה-Switch שולח הודעת broadcast לכל ה-mac-ים שהוא "יודע", כאשר שני המחשב מחוברים לאותו Switch אז הוא ישלח לו את הבקשה מהמחשב החדש ואם לא אז ההודעה מועברת דרך ה-default gateway וה-Router בודק בטבלת ה-ARP שלו האם הוא מזהה את כתובת ה-IP של כתובת היעד, אם הוא לא מוצא את הכתובת הוא מעביר את החבילה לפרוטוקולי הניתוב לדוגמה: BGP, RIP, בסוף החבילה מגיעה ליעד ושני המחשבים "יודעים" אחד על השני. לאחר שהם "יודעים" אחד על השני המחשב יוצר TCP סוקט אשר נקרא תהליך "Shaking hand" המחשב החדש שולח SYN ומחכה לתשובה של המחשב השני שישלח לו חזרה SYNACK. לאחר מכן שני המחשבים יוכלו לתקשר ביניהם ולהעביר מידע אחד לשני.

2. CRC זו היא שיטה לאיתור שגיאות שידור בזמן העברת נתונים. ה-CRC מקודד הודעות על ידי הוספה של ערך בדיקה עם גודל קבוע, בשביל איתור שגיאות בעת העברת מידע ברשת. הצד שמקבל את המידע בודק באמצעות קוד ה-CRC שהמידע שקיבל אכן תקין. ה-CRC פועל באופן דומה.

אופן הפעולה:

בהינתן פולינום מדרגה Z והודעה מדרגה: m

(א) הוספה של r אפסים מימין להודעה.

(ב) חילוק בפולינום תוך שימוש בחילוק של מודולו 2.

(ג) חיסור השארית תוך שימוש ב-XOR במקום חיסור רגיל.

(ד) צירוף התוצאה מימין להודעה המקורית ושליחת ההודעה.

3. HTTP הוא פרוטוקול בשכבת האפליקציה אשר מגדיר איך לקוח ושרת מעבירים מידע ביניהם. כאשר הלקוח שולח הודעה לשרת, השרת מחזיר את דף האינטרנט אשר ביקש.

1.0 http עובד בשיטת "Stop and Wait", נניח שהלקוח רוצה לטעון דף אינטרנט מסויים עם כתובת URL כלשהי אז הלקוח ייצור חיבור TCP עם השרת בפורט 80. לאחר שהחיבור נוצר בין השרת ללקוח, הלקוח ישלח בקשה לקבלת קובץ HTML מהשרת עם ה-PATH המתאים השרת יקבל את הבקשה מהלקוח וישלח את האובייקט המבוקש ללקוח ולאחר מכן השרת ישלח בקשה לסגירת החיבור ביניהם. החיבור עצמו "מתין" עד שהלקוח יקבל את התגובה של השרת, וכאשר יקבל את התגובה עם הקובץ HTML החיבור ייסגר.

1.1 http עובד כמו עובד עם קשר TCP אחד בלבד ובשונה מה- 1.0 http לא יפתח קשר TCP עבור כל אובייקט.

1.1 http כולל את הhost header מכיוון שהסרבר יכול להכיל HOST -ים שונים שחולקים כתובת IP אחת. לעומת 1.0 http שלא מכיל את הhost header מכיוון שמניח שהסרבר יוצר חיבור עם HOST אחד ויחיד. ב 1.1 http הלקוח יכול לבקש מהשרת רק חלק מהאובייקט בעזרת הטווח וכך ניתן לחדש הורדות שהופסקו. לעומת זאת, ב 1.0 http הסרבר שולח את כל האובייקט ללקוח אפילו אם הלקוח צריך רק חלק מהאובייקט.

2.0 http מכיל שכבת פורמט בינארי.

2.0 http עובד בצורה מקבילית, הלקוח יוצר קשר TCP אחד עם השרת בדומה ל 1.1 http אך שולח את הבקשות במקביל כאשר FRAMES של החבילה מחולקים ומתוויגים כך שכאשר המידע מגיע ללקוח הוא יודע לשחזר אותו בדיוק כמו שהיה גם אם החבילות לא הגיעו באותו הסדר. 2.0 http דוחס את ה FRAMES HEADER וכאשר נבקש את הבקשה השניה הוא יקח מה HEADER של הבקשה הראשונה את כל השדות המשותפים וישלח רק את מה שהשתנה. 2.0 http מציג את ה SERVER PUSH. מכיוון ש 2.0 http תומך בשליחה מקבילית עם חילוק ותיג FRAMES הסרבר יכול לשלוח בנפרד את כל התוספים והלקוח יכול לבחור אם לקבל אותם או לדחות אותם. לעומת, 1.1 http שלוח ללקוח אובייקטים שכנראה יצטרך לפני שהלקוח בכלל מבקש אותם.

QUIC – פרוטוקול אשר פיתחה חברת גוגל. פרוטוקול זה עובד מעל UDP ומאפשר חיבור מאובטח ומהיר בין המשתמש לבין האתר אליו מעוניין להגיע. הוא מאפשר למשתמש שיצר קשר מאובטח מול השרת בעבר, לקבל מידע מהשרת גם ללא בדיקת האתר והמתנה להקמת חיבור מאובטח.

4. בזמן הרצת תוכנית, המחשב מספק נתיב הרצה הנקרא תהליך. לכל תהליך יש שני חיבורים אחד פנימי הנקרא IP, ואחד חיצוני הנקרא PORT. מספרי הפורטים נעים בין 0 ל 65,535, ומספרי הפורטים 0 עד 1023 הם מספרים שמורים לתוכנות ידועות המשתמשות בהם. כאשר שני מכשירים מתקשרים ברשת ושולחים מידע אחד לשני ברשת הם מוצאים אחד את השני בעזרת מזהה יחודי עבור כל מכשיר כלומר ה IP לאחר שמועבר מידע ממכשיר למכשיר, המכשיר שמקבל את המידע יודע מה לעשות עם המידע ולכן לשלוח אותו ברמת האפליקציה בעזרת הפורט אשר אומר למכשיר המקבל את המידע איך להשתמש במידע שהוא קיבל ולאיזה אפליקציה לשלוח אותו.

5. Subnet בתרגום לעברית - תת רשת וכשמה היא.

כאשר משתמשים subnet הרשת הופכת ליעילה יותר בעזרת חילוק הרשת לתתי רשתות. כתובות IP מקוטלגות ב A,B,C,D,E (כאשר D,E פחות בשימוש). בעזרת הקיטלוג הראור מנתב אל הרשת בצורה יעילה יותר. לכתובת הקן יהיה מזהה נוסף שקשור למספר הביטים ששייכים לרשת, בכל טווח של כתובות IP נשמור 2 כתובות של IP – כתובת IP הראשונה שתהיה שמורה כמזהה של הרשת, כתובת ה IP האחרונה ששמורה בשביל שליחת שידור. כאשר שולחים חבילה לכתובת IP מסויימת בודקים את הקיטלוג של הכתובת והרשת מנתבת את החבילה לרשת הספציפית, כאשר מוצאים את הכתובת הספציפית בעזרת subnet mask והחבילה נשלחת לכתובת הספציפית בתת רשת. החבילה נשלחת לראוטר או לשרת ש"אחראי" על כתובת הרשת ומנתב את החבילה לכתובת זו.

6. תחילה נסביר שכתובת ה MAC היא כתובת שצורבה בכרטיס רשת של המכשיר, אשר נמצאת בשימוש בפרוטוקול Ethernet הנמצא בשימוש בשכבת הערוץ. המידע אשר מגיע לשכבת הערוץ מכיל ב HEADER את כתובת ה MAC של השולח והמקבל. כאשר מתבצע תהליך של העברת מידע כלשהו, בעת שליחת ההודעה, בשכבת הרשת מתווסף למידע כתובת ה IP של היעד, ובשכבת הערוץ, מתווסף למידע כתובת ה MAC של המיקום הבא לשליחה. לאחר שהמידע יגיע לנתב בעזרת כתובת ה MAC שלו, הנתב ישווה את כתובת ה MAC של היעד ואת כתובת ה IP של היעד ויגלה שהמידע לא מיועד לו ולאחר מכן, יחפש את הרשת בה נמצא מכשיר היעד ויעביר את המידע הלאה. כתובת ה MAC תשתנה כעת לכתובת של מכשיר היעד והמידע יועבר הלאה בדרכו לרשת היעד והכתובת המיועדת. לאחר שהסברנו על אופן הפעולה והשימושים של

כתובת ה-MAC וה-IP נסביר למה לא מספיק לעבוד עם כתובות IP בלבד. כתובות ה-IP וכתובות ה-MAC עובדות בשכבות שונות כאשר לכל שכבה מטרה משלה ולכן לא נערבב ביניהם.

7. NAT הוא פרוטוקול המאפשר למספר מכשירים הנמצאים תחת אותה רשת מקומית לחלוק את אותה כתובת IP ולתקשר עם כל המכשירים שמחוץ לרשת הזו. כל חבילה שתצא מהרשת תהיה בעלת אותה כתובת IP (של ה-NAT) בלי קשר מאיזה מכשיר היא יצאה, ההבדל בין המחשבים השונים יתבטא בפורט שממנו יוצאת כל חבילה. החבילות שנשלחות בתוך הרשת נשלחות בצורה רגילה עם כתובת ה-IP הרגילות של המחשבים ברשת. ובעצם, רשת שלמה של מכשירים חולקים כתובת IP אחת, אשר עוזר לחסוך כתובות IP. Router (נתב) הוא רכיב המשמש לניתוב מרשת אחת לרשת אחרת. הנתב פועל בשכבת הרשת. כשהנתב מקבל חבילת נתונים, הוא בודק מהי כתובת ה-IP של היעד, ומסתמך על טבלת הניתוב שלו על מנת לקבוע מאיפה להעביר את החבילה. כדי לבנות טבלה זו, מסתמך הנתב על פרוטוקולי הניתוב (לדוגמה OSPF, RIP) כדי להחליט מה ניתן לעשות עם החבילה. כדי להעביר את החבילה לתחנה הבאה ולקרב את החבילה אל יעדה, הנתב עושה שימוש בסוג נוסף של פרוטוקולים, פרוטוקולים אלה נקראים פרוטוקולים מנותבים, והבולט בהם הוא פרוטוקול האינטרנט-IP.

Switch (מתג) הוא רכיב שפועל בשכבת הערוץ (LINK). ה-switch הוא ה"אח הגדול" של Hub. כאשר ישנה רשת מכשירים שמחוברים ל-switch ונשלחת הודעה ממכשיר כלשהו למכשיר אחר ברשת הזו, ה-switch ישלח את ההודעה רק למכשיר המבוקש (בניגוד לHub ששולח הודעה לכולם וכולם מתעלמים ממנה חוץ מהמכשיר המבוקש). ה-switch יודע להבחין בהבדל בין המכשירים בעזרת כתובת ה-MAC שלהם. המתג מקבל את המידע מהמכשיר השולח, מעבד את המידע בשכבה הפיזית ובשכבת הערוץ. בשכבת הערוץ, נמצאת כתובת ה-MAC של מכשיר היעד וכך המתג יודע להעביר את המידע למכשיר הנכון. לאחר שהמכשיר מצא את כתובת ה-MAC של המכשיר היעד, הוא מעביר את המידע שוב ושוב בשכבת הערוץ ובשכבה הפיזית ומעביר אותו אל המכשיר הנכון.

ישנם מספר הבדלים בין המרכיבים הללו:

1. Switch ו Router הם רכיבים פיזיים לעומת NAT שהוא פרוטוקול.
2. Switch עובד עם כתובות MAC לעומת Router שעובד כתובות IP.
3. NAT לא מתעסק בהעברה של חבילה ליעד כלשהו, לעומת Router ו Switch כמו שהראנו לעיל.

8. כתובת IP ב-IPv4 בנויה מ-32 סיביות, אשר מהווים באופן עקרוני 4,294,967,296 כתובות שונות, אך המספר קטן יותר, מכיוון שטווחים מסויימים של כתובות במסגרת הפרוטוקול שמורים למטרות אחרות, מספר הכתובות השמישות בפועל קטן יותר, ועומד על כ-4 מיליארד כתובות, אך נוצר מחסור בכתובות בעקבות ההתקדמות הטכנולוגית המשמעותית וחיבור מגוון רחב ורב של מכשירים לאינטרנט. לאור המחסור בכתובות IPv4 פיתחו שיטות להתגבר על המחסור בכתובות. אחת מהן היא מעבר לכתובות גדולות יותר שנקראות IPv6 שמורכבות מ-128 ביטים אשר מהווים עד  $3.4 \times 10^{38}$  כתובות שונות מגדיל את כמות הכתובות באופן משמעותי ומתגבר על המחסור בכתובות IP שהיה ב-IPv4. דרך נוספת, היא באמצעות NAT- Network Address Translation אשר מאפשר למספר מכשירים באותה רשת מקומית לחלוק את אותה כתובת IPv4, כלומר עם כל המכשירים שמחוץ לרשת הזו.

9.

- (e) הנתב 3c לומד על תת הרשת x בעזרת פרוטוקול הניתוב BGP.
- (f) הנתב 3a לומד על תת הרשת x בעזרת פרוטוקול הניתוב OSPF.
- (G) הנתב c1 לומד על תת הרשת x בעזרת פרוטוקול הניתוב BGP.
- (H) הנתב c2 לומד על תת הרשת x בעזרת פרוטוקול הניתוב OSPF.

