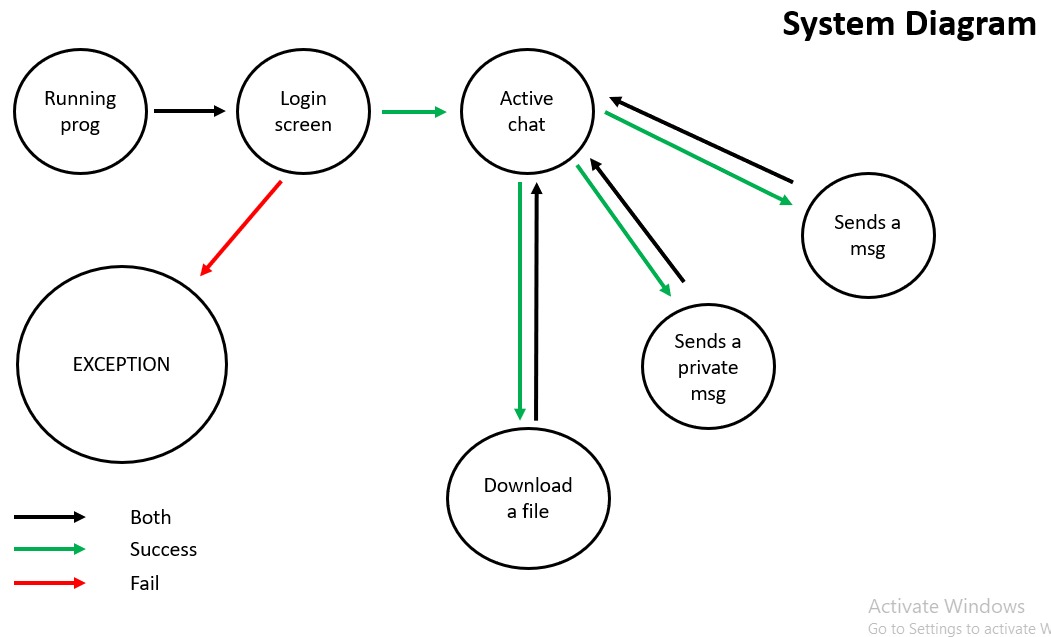
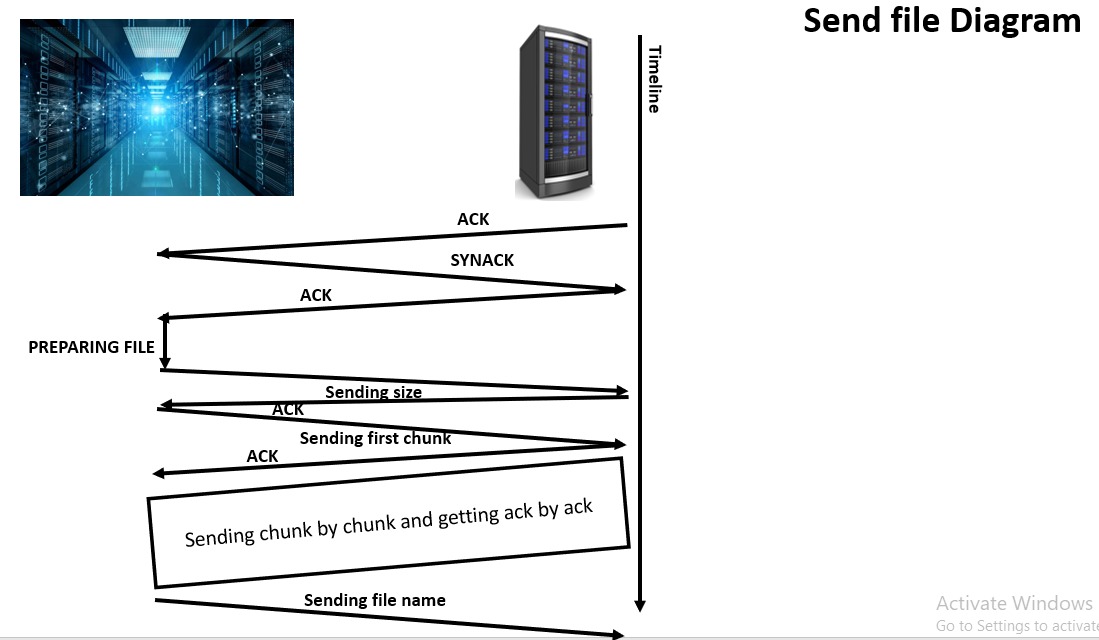
**פרוייקט גמר – תקשורת ומחשוב**

**מגישים:** אריק טטייבסקי 208997056  
 רועי משולם 315635649

**חלק ב':**

****

****

**כיצד המערכת מתגברת על איבוד חבילות / בעיות LATENCY?**

למעשה בגלל שאנחנו משתמשים ב WAIT&STOP אז זו דרך ההתמודדות שלנו עם בעיות אלו.

בכל פעם הSERVER ישלח לCLIENT את החבילה הראשונה (מתוך כל הקבצים) הוא יחכה לקבל מהCLIENT אישור על מספר החבילה.

במידה והוא לא קיבל אישור הוא ינסה לשלוח עוד הפעם ועוד הפעם עד הצלחה.

הCLIENT מבחינתו מקבל בתחילת ההעברת מידע את כמות המידע שהוא צריך לקבל סה"כ ואז הוא פשוט ימשיך לקבל מידע עד שיגיע לכמות שהוא מחכה לה.

ניתן לראות את העברת המידע הזו קוראת בתרשים מעל.

**חלק ג':**

1. בהינתן מחשב חדש המתחבר לרשת נתאר את כל ההודעות שעוברות החל מהחיבור הראשוני ל switch ועד שההודעה מתקבלת בצד השני של הצאט.  
   א) בשלב הראשון המחשב חדש ברשת ועדיין אין לו כתובת IP ולכן הפרוטוקול הראשון שיפעל יהיה פרוטוקול DHCP. המחשב שולח הודעת גילוי לשרת ה- DHCP כאשר כתובת הIP של המחשב יהיה 0.0.0.0 וכתובת היעד תהיה 255.255.255.255 שזו הכתובת האחרונה ברשת, כלומר הוא ישלח הודעה לכל מי שנמצא ברשת על מנת שרק השרת של ה-DHCP יחזיר תשובה. שרת הDHCP מחזיר תשובה עם הודעת הצעה של ה-DHCP, ההודעה תכיל את הכתובת IP שניתן להשתמש בה, וכן פרטים נוספים על הרשת כמו הכתובת של הנתב שנמצא בעצירה הראשונה בדרך אל הלקוח, שם וכתובת הIP של שרת הDNS ומסיכת הרשת. פרוטוקול הDHCP עובד בעזרת UDP מטעמי מהירות.  
     
   ב) לאחר שהמחשב "הכיר" את הרשת שהוא נמצא בה, הוא צריך את כתובת הIP של המחשב איתו הוא רוצה לתקשר. כדי להשיג כתובת זו המחשב החדש יצטרך לשלוח בקשת DNS. בקשת ה-DNS עטופה בUDP (מטעמי מהירות), עטופה בIP ,שעטוף בEthernet. כדי לשלוח את הבקשת DNS נצטרך להשיג את כתובת הMAC של הנתב הראשון.  
   כדי לעשות זאת, המחשב החדש ישתמש בפרוטוקול ARP. כל מכשיר שיש לו כתובת IP מכיל טבלת ARP. המחשב יאתר את הכתובת של הנתב ע"י שידור של חבילת מידע בשכבת הערוץ עם כתובת הMAC FF:FF:FF:FF:FF:FF בשדה היעד של ה-MAC ואת כתובת הIP של הנתב המבוקש. חבילה זו משודרת אל כל המכשירים המחוברים אל הרשת. המכשיר שיזהה את כתובת הIP שלו בהודעה, ישלח בחזרה הודעה עם כתובת הMAC שלו אל המחשב החדש (ההודעה תשלח ב Unicast).  
   כעת, המחשב החדש יודע מהי כתובת הMAC של הנתב הראשון ויכול לשלוח את בקשת הDNS.   
   \*שלב ב קורה בהנחה שהמחשב החדש לא מכיר את כתובת הIP של המחשב של הצד השני.  
   ג) בשלב זה המחשב החדש מכיר את הIP של המחשב בצד השני של הצא'ט.   
   המחשב החדש שולח הודעת Unicast ל Switch עם כתובת הIP של הצד השני עם כתובת MAC FF:FF:FF:FF:FF:FF כלומר הודעת Broadcast.  
   ה-Switch שולח הודעת broadcast לכל הmac-ים שהוא "מכיר". אם המחשב של הצד השני מחובר לאותו Switch אז הוא ישלח לו את הבקשה מהמחשב החדש. ואם הוא לא מחובר לאותו Switch אז ההודעה מועברת דרך הdefault gateway והRouter בודק בטבלת הARP שלו האם הוא מזהה את כתובת הIP של כתובת היעד. אם הוא לא מוצא הוא עובר לטבלת ניתוב שלו ומעביר את החבילה לפני פרוטוקלי הניתוב (לדוגמה OSPF,RIP,BGP). בסופו של דבר החבילה מגיעה אל הכתובת בצד השני והמידע מגיע אל המחשב שבצד השני ושני המחשבים "מכירים" אחד את השני.  
   ד) כעת המחשב החדש מכיר את כתובת הIP וכתובת הMAC של המחשב בצד השני וכדי להעביר מידע בינהם (הודעות קבצים וכו') המחשב החדש יוצר TCP socket וקורה תהליך "לחיצת ידיים". המחשב החדש שולח SYN ומחכה לתשובה של המחשב בצד השני שישלח לו חזרה SYNACK. לאחר מכן שני המחשבים יוכלו להעביר מידע בינהם.
2. CRC היא שיטה לגילוי שגיאות שידור.   
   לפני ההעברת המידע, מחשבים את הCRC ומוסיפים אותו למידע. לאחר ההעברה , הצד המקבל מאשר באמצעות הCRC שהמידע שהגיע אכן תקין.  
   CRC מבוסס על קוד מחזורי שמקודד הודעות על ידי הוספה של ערך בדיקה בעל גודל קבוע. הוא מבוסס על האיזומורפיזם שבין וקטורים לפולינומים (מסתכלים על כל וקטור באורך n כפולינום שמקדמיו הם קוארדינטות הוקטור).

**אופן הפעולה:**בהינתן פולינום מדרגה rוהודעה 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.

**\*** הצד המקבל יבצע את שלבים א+ב והתוצאה שתתקבל תהיה זהה לr הביטים האחרונים שנשלחו בהודעה.

1. HTTP הוא פרוטוקול בשכבת האפליקציה. פרוטוקול זה מגדיר איך לקוח ושרת מדברים אחד עם השני. כלומר, כאשר הלקוח שולח הודעה לשרת ומבקש דף אינטרנט, השרת מחזיר לו תשובה עם הדף האינטרנט המבוקש. בדרך כלל, דף האינטרנט מכיל אובייקטים נוספים חוץ מקובץ HTML כמו: JPEG, audio , video clip ועוד. ישנם סוגי HTTP שונים ונראה את ההבדלים בין שלושתם:   
   http 1.0 ,http 1.1, http 2.0 וכמו כן נדבר על QUIC ואיך הוא שונה מהם.  
   **http 1.0** עובד בשיטת "**Stop and Wait**", כלומר הוא מבקש מהשרת אובייקט אחרי אובייקט. נניח שהלקוח רוצה לטעון דף אינטרנט מסויים עם כתובת URL כלשהי אזי הלקוח ייצור חיבור TCP עם השרת בפורט 80. לאחר שהחיבור נוצר והתקיימה "לחיצת היד" בין השרת ללקוח, הלקוח ישלח בקשה לקבלת קובץ HTML מהשרת עם ה-PATH המתאים השרת יקבל את הבקשה מהלקוח וישלח את האובייקט המבוקש (הקובץ HTML) ללקוח ולאחר מכן השרת יבקש לסגור את החיבור TCP ביניהם. החיבור עצמו "ימתין" עד שהלקוח יקבל את התגובה של השרת, וכאשר יקבל את התגובה עם הקובץ HTML החיבור ייסגר.  
   אם הקובץ HTML מכיל אובייקטים נוספים כמו שציינו לעיל אזי הלקוח יצטרך לחזור על כל השלבים עבור כל אובייקט. כלומר, עבור כל אובייקט הלקוח יפתח חיבור TCP מחדש.   
   **http 1.1** עובד כמו http 1.0 אך שונה בכמה דברים והם:  
   \* עובד עם קשר TCP אחד בלבד ובשונה מה- http 1.0 לא יפתח קשר TCP עבור כל אובייקט בשיטה שנקראת pipeline.\* http 1.0 לא כולל את הHOST בHEADER מכיוון ש http 1.0 מניח שהסרבר נקשר עם HOST יחיד. לעומת זאת, http 1.1 כולל את הHOST הספציפי בHEADER מכיוון שהסרבר יכול להכיל כמה HOST -ים   
   (MULTI-homed web servers) שחולקים כתובת IP אחת.  
   \* ב- http 1.0 הסרבר שולח את כל האובייקט ללקוח אפילו אם הלקוח צריך רק חלק מהאובייקט דבר הגורם לבזבוז של רוחב פס. לדוגמה אם נרצה להמשיך הורדה של אובייקט וההורדה נכשלה או הופסקה מסיבה כלשהי נצטרך לשלוח את כל האובייקט מחדש. לעומת זאת, ב http 1.1 הלקוח יכול לבקש מהשרת רק חלק מהאובייקט בעזרת הטווח וכך ניתן לחדשש הורדות שהופסקו. לדוגמה נרצה להוריד קובץ של 1024 קילו בייט ועד עכשיו ירד 512 קילו בייט אז נוסיף לHEADER טווח של 512 קילו בייט וכך השרת ידע מה חסר כדי להשלים את הורדת האובייקט.  
   **http 2.0** עובד כמו http 1.1 אך שונה במספר דברים והם:  
   \* http 2.0 מכיל שכבת פורמט בינארי. **\*** לעומת http 1.0 שעובד בעזרת השיטה שהזכרנו לעיל "STOP AND WAIT" שאינה יעילה. ולעומת, http 1.1 שעובד בעזרת PIPELINE כמו שציינו לעיל, שיטה שמשפרת את היעילות של השיטה "STOP AND WAIT" אך עדיין לא מספקת מהסיבה שיכול להיווצר "צוואר בקבוק" שעלול לגרום ל HEAD OF LINE BLOCKING. פירוש הדבר הוא שאם החבילה הבאה בתור לא יכולה להישלח אזי גם כל החבילות שאחריה לא יכולות להישלח. http 2.0 עובד בצורה מקבילית, הלקוח יוצר קשר TCP אחד עם השרת בדומה ל http 1.1 אך שולח את הבקשות במקביל כאשר הFRAMES של החבילה מחולקים ומתוייגים כך ש המידע מגיע ללקוח הוא יודע לשחזר אותו בדיוק כמו שהיה גם אם החבילות לא הגיעו באותו הסדר.  
   \* ב http 1.0 ו http 1.1 אם נשלחו שתי בקשות זהות פרט לכמה שינויים קלים עדיין נצטרך לשלוח פעמיים את הבקשה עם הHEADER כולו דבר שתופס מקום. כדי לחסוך את המקום המבוזבז http 2.0 דוחס את ה HEADER FRAMES וכאשר נבקש את הבקשה השניה הוא יקח מה HEADER של הבקשה הראשונה את כל השדות המשותפים וישלח רק את מה שהשתנה.  
   \* SERVER PUSH – http 1.1 משתמש ב RESOURCE LINING כדי לשלוח ללקוח אובייקטים שכנראה יצטרך לפני שהלקוח בכלל מבקש אותם. לדוגמה, אם לקוח מבצע בקשה לקובץ CSS אז השרת ישלח לו אובייקטים נוספים שכנראה יצטרך בתוך הקובץ CSS. דבר זה מוביל לפחות בקשות מצד הלקוח אך גם ישנם חסרונות. השרת שולח את כל התוספים יחד והלקוח לא יכול לפצל אותם או לדחות אותם גם אם אין בהם בכלל צורך. שנית, הקובץ שנשלח הופך להיות גדול יותר וזה גורם להקטנת מהירות החיבור.  
   כדי לשפר את כל זה http 2.0 מציג את ה SERVER PUSH. מכיוון ש http 2.0 תומך בשליחה מקבילית עם חילוק ותיוג לFRAMES הסרבר יכול לשלוח בנפרד את כל התוספים והלקוח יכול לבחור אם לקבל אותם או לדחות 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.
2. כאשר אנחנו מריצים תוכנה במחשב, ההרצה נקראת תהליך. בצורה פנימית לכל תהליך יש מזהה. בצורה חיצונית לכל תהליך שקשור לרשת יש פורט (PORT). מספרי הפורטים נעים בין 0 ל 65,535 , ומספרי הפורטים מ0 עד 1023 הם מספרים שמורים לתוכנות ידועות המשתמשות בהם.   
   כאשר שני מכשירים מתקשרים בינהם ברשת ושולחים מידע אחד לשני ברשת הם מוצאים אחד את השני בעזרת מזהה ייחודי הנקרא IP ADDRESS אחד של השני.כאשר המידע מועבר ממכשיר למכשיר המכשיר שמקבל את המידע צריך לדעת מה לעשות עם המידע ולאן לשלוח אותו ברמת האפליקציה. כאן נכנס התפקיד של הפורט, הפורט בעצם "עושה סדר" ואומר למכשיר המקבל את המידע איך להשתמש במידע שהוא קיבל ולאיזה אפליקציה לשלוח אותו.
3. Subnet זה תת רשת. subnet הופך את הרשת ליותר יעילה, כאשר מחלקים את הרשת הגדולה לתתי רשתות דרכי הרשתות בין מכשירים כלשהם הופכים לקצרים יותר ולחוסכים מעבר לא הכרחי בראוטרים כדי להגיע ליעד שלהם.  
   כמו כן, כתובות IP מחולקות למחלקות שונות A,B,C,D,E (כאשר D,E פחות בשימוש). הצורה שבה כתובות הIP מחולקות מאפשרות לראטורים לנתב אל הרשת בצורה יעילה יותר, אבל בתוך הרשת יכולים להימצא מיליונים של מכשירים שמחוברים אליה כמו לדוגמה בCLASS A. כאן נכנס הsubnet, בשיטה זו נפצל את הרשת הגדולה לתתי רשתות. במקום שכתובת ה- IP תעיד על עצמה לאיזו כתובת היא שייכת, לכתובת יהיה מזהה (מספר) נוסף שיעיד על מספר הביטים ששייכים לרשת. חלוקה זו מאפשרת גמישות בחלוקת הכתובת הכתובת תראה בצורה הבאה: a.b.c.d./x כאשר x מייצג את מספר הביטים השייכים לכתובת של הרשת ושאר הביטים מייצג את החלק של הHOST. X נקרא מסכת רשת משנה (SUBNET MASK).  
   בכל טווח של כתובות IP נשמור 2 כתובות של IP – כתובת IP הראשונה שתהיה שמורה כמזהה של הרשת, כתובת הIP האחרונה ששמורה בשביל שליחת שידור (broadcast).   
   כאשר שולחים חבילה לכתובת IP מסויימת בודקים לאיזה CLASS הכתובת משתייכת והרשת מנתבת את החבילה לרשת הספציפית. ברשת יכולים להיות כמו שאמרנו לעיל הרבה כתובות שונות, כאשר החבילה מגיעה לרשת מתבצע חישוב מתמטי בעזרת הSUBNET MASK ומוצאים את הכתובת הספציפית בעזרת החישוב והחבילה יודעת לאיזה כתובת בתת רשת הזאת היא צריכה ללכת. החבילה נשלחת לראוטר או לשרת ש"אחראי" על כתובת הרשת ומנתב את החבילה לכתובת הספציפית.
4. כתובת הMAC של המכשיר היא כתובת שצרובה בכרטיס רשת של המכשיר. כתובת ה MAC נמצאת בשימוש בפרוטוקול הEthernet הנמצא בשימוש בשכבת הערוץ. כאשר המידע מגיע לשכבת הערוץ, בHEADER נמצא הכתובת MAC של המכשיר השולח וגם של המכשיר המקבל.   
   כאשר מתבצע תהליך של העברת מידע ממכשיר ברשת א' אל מכשיר מרשת ב' כלשהי, בעת שליחת ההודעה, בשכבת הרשת מתווסף למידע כתובת הIP של מכשיר היעד ברשת ב', ובשכבת הערוץ, מתווסף למידע כתובת הMAC של התחנה הבאה בדרך, הנתב ולא של מכשיר היעד. זאת מכיוון שתפקידה של שכבת הערוץ, היא העברת המידע אל התחנה הבאה בדרך עד להגעה אל היעד. לאחר שהמידע יגיע לנתב בעזרת כתובת הMAC שלו , הנתב ישווה את כתובת הMAC של היעד ואת כתובת הIP של היעד ויגלה שהמידע לא מיועד לו.  
   הנתב ימצא את הרשת בה נמצא מכשיר היעד ויעביר את המידע הלאה. כתובת הMAC תשתנה כעת לכתובת של מכשיר היעד והמידע יועבר אל רשת ב' , הרשת של מכשיר היעד שם לפי כתובת הIP המידע יידע לאיזה מכשיר הוא צריך להגיע.  
   לאחר שהסברנו על אופן הפעולה והשימושים של כתובת הMAC והIP נסביר למה לא מספיק לעבוד עם כתובות IP בלבד.  
   כתובות הIP וכתובות הMAC עובדות בשכבות שונות כמו שציינו לעיל. לכל שכבה מטרה משלה ולכן לא נרצה לערבב בינהם.
5. ראשית נסביר מה התפקיד של כל אחד ולאחר מכן נציין את ההבדלים ביניהם.  
   **NAT** הוא פרוטוקול המאפשר למספר מכ­שירים הנמצאים תחת אותה רשת מקומית לחלוק את אותה כתובת IP ודרכה יתקשרו עם "העולם החיצון", כלומר עם כל המכשירים שמחוץ לרשת הזו. כל חבילה שתצא מהרשת ללא קשר מאיזה מכשיר היא יצאה , תהיה בעלת אותה כתובת IP (של הNAT), ההבדל בין המחשבים השונים יתבטא בפורט שממנו יוצאת כל חבילה. החבילות שנשלחות בתוך הרשת נשלחות בצורה רגילה עם כתובת הIP הרגילות של המחשבים ברשת. בסופו של דבר, רשת שלמה של מכשירים חולקים כתובת IP אחת דבר שחוסך כתובות IP.  
   **Router** (נתב) הוא רכיב משמש לניתוב מרשת אחת לרשת אחרת. הנתב פועל בשכבת הרשת. כאשר הנתב מקבל חבילת נתונים, הוא בודק מהי כתובת הIP של היעד, ומסתמך על טבלת הניתוב שלו על מנת לקבוע מאיפה להעביר את החבילה. כדי לבנות טבלה זו, מסתמך הנתב על פרוטוקלי הניתוב (לדוגמה OSPF, RIP) על מנת להחליט מה יש לעשות עם החבילה. במטרה להעבירה לתחנה הבאה בשרשרת ולקרב את החבילה אל היעד עושה הנתב שימוש בסוג נוסף של פרוטוקולים, פרוטוקולים אלה נקראים פרוטוקולים מנותבים, והבולט בהם הוא פרוטוקול האינטרנט -IP.   
   **Switch** (מתג) הוא רכיב שפועל בשכבת הערוץ (LINK). המתג הוא ה"אח הגדול" של Hub. כאשר ישנה רשת מכשירים שמחוברים למתג ונשלחת הודעה ממכשיר כלשהו למכשיר כלשהו אחר ברשת הזו, המתג יודע לשלוח את ההודעה רק למכשיר המבוקש (בניגוד לHub ששולח הודעה לכולם וכולם מתעלמים ממנה חוץ מהמכשיר המבוקש). המתג יודע להבדיל בין המכשירים בעזרת כתובת הMAC שלהם שהסברנו עליה בסעיף הקודם. המתג מקבל את המידע הממכשיר השולח, מעבד את המידע בשכבה הפיזית ובשכבת הערוץ. בשכבת הערוץ , נמצאת כתובת הMAC של מכשיר היעד וכך המתג יודע לנתב את המידע למכשיר הנכון. לאחר שהמכשיר מצא את כתובת הMAC של המכשיר היעד, הוא מעביר את המידע שוב ושוב בשכבת הערוץ ובשכבה הפיזית ומעביר אותו אל המכשיר הנכון.  
     
   לאחר שהבנו מה אופן הפעולה של כל אחד מהם נבדיל בינהם:  
   1. NAT הוא פרוטוקול לעומת Router ו Switch שהם רכיבים פיזיים.  
   2. Switch עובד עם כתובות MAC לעומת Router שעובד כתובות IP.  
   3. NAT לא מתעסק כלל בתעבורה של חבילה ליעד כלשהו, לעומת הRouter וה Switch כמו שהראנו לעיל.
6. כתובת IP ב-IPv4 בנויה מ-32 סיביות , וכך מאפשרת באופן תאורטי 4,294,967,296 כתובות שונות, אך באופן מעשי המספר קטן יותר.  
   מכיוון שטווחים מסויימים של כתובות במסגרת הפרוטוקול שמורים למטרות מיוחדות, מספר הכתובות השמישות בפועל קטן יותר, ועומד על כ-4 מיליארד כתובות. כמות כתובות זו, בראשית השימוש באינטרנט, נשמעה רבה ומספקת לכל השימושים, אך ברבות השנים נוצר מחסור בכתובות ברמה העולמית. זאת, בעיקר עקב ההחלטה לפתוח את האינטרנט לשימוש של כלל המגזרים בעולם ולכלל האוכלוסייה. ההתקדמות הטכנולוגית המשמעותית וחיבור מגוון רחב ורב של ציודי קצה לאינטרנט, כגון מחשבים נייחים, מחשבים ניידים, טלפונים סלולריים, מחשבי כף יד, אביזרי ניווט, שעונים, אביזרי צילום וכד'.  
   לאור המחסור בכתובת IPv4 פיתחו שיטות להתגבר על המחסור בכתובות.  
   ראשית, מתבצע מעבר לכתובות גדולות יותר שנקראות IPv6 שמורכבות מ128 ביטים. החידוש העיקרי בגרסה 6 הוא הגדלה משמעותית בכמות כתובות הIP האפשריות. השימוש ב-128 סיביות, המאפשרות עד 3.4\* 10^38 כתובות שונות מגדיל את כמות הכתובות באופן משמעותי ומתגבר על המחסור בכתובות IP שהיה בIPv4.  
   דרך נוספת, היא באמצעות NAT- Network Address Translation . NAT מאפשר למספר מכ­שירים הנמצאים תחת אותה רשת מקומית לחלוק את אותה כתובת IPv4 ודרכה יתקשרו עם "העולם החיצון", כלומר עם כל המכשירים שמחוץ לרשת הזו. כל חבילה שתצא מהרשת ללא קשר מאיזה מכשיר היא יצאה , תהיה בעלת אותה כתובת IP (של הNAT), ההבדל בין המחשבים השונים יתבטא בפורט שממנו יוצאת כל חבילה. החבילות שנשלחות בתוך הרשת נשלחות בצורה רגילה עם כתובת הIP הרגילות של המחשבים ברשת. בסופו של דבר, רשת שלמה של מכשירים חולקים כתובת IP אחת דבר שחוסך כתובות IP ועוזר להתגבר על המחסור בכתובות IPv4.
7. e) הנתב 3c לומד על תת הרשת x באמצעות פרוטוקול הניתוב BGP.   
   f) הנתב 3a לומד על תת הרשת x באמצעות פרוטוקול הניתוב OSPF.  
   G) הנתב c1 לומד על תת הרשת x באמצעות פרוטוקול הניתוב BGP.  
   H) הנתב c2 לומד על תת הרשת x באמצעות פרוטוקול הניתוב OSPF.