בס"ד

מטלה גמר תקשורת

אוהד שירזי - 318636693
דביר ביטון - 318765856

1. הנחות יסוד:
FFFFFFFFFFFF - כלומר אנחנו שולחים במצב broadcast.
אנחנו מדברים על שרת צא'ט ששולח הודעות רגילות בין לקוחות.
אנחנו מניחים שזה שרת שאנחנו יודעים את השם שלו. ולכן אנחנו מבקשים DNS.
אנחנו מניחים שההודעה האחרונה שנשלחה(כלומר מהלקוח אל הלקוח השני היא הודעה רגילה שעוברת בTCP אפשר להחליף את ההודעה האחרונה הזאת שהיא עוברת בUDP זה לא משנה. העיקרון הוא שזה עובר מלקוח א' לשרת והשרת ללקוח ב').
(התשובה שלנו מסתמכת על המצגת Chapter 6 copy שיש לנו במודל).

פרוטוקול	מקור IP	יעד IP	PORT מקור	PORT יעד	MAC מקור	יעד MAC	הסבר
DHCP			68	67		FFFFFF FFFFFF	נשלחת הודעת DHCP על מנת שהלקוח שלנו יקבל כתובת IP. ההודעה נעטפת בUDP, IP, ETHER NET 802.3. ההודעה נשלחת במצב BROAD CAST
DHCP	שרת DHCP	עדיין אין	67	68	שרת DHCP	מי שביקש כתובת IP(לקוח)	השרת DHCP מחזיר לנו הודעה ACK שכוללת 4 דברים: IP ללקוח/ IP של הראוטר הראשון שדרכו

							הוא יוצא. שם. של IP שרת DNS
ARP	לקוח (הIP שקיבלתי)	ראוטר			MAC לקוח	FFFFFF FFFFFF	בקשת ARP
ARP	ראוטר	המחשב שביקש			של MAC הראוטר	MAC לקוח	תשובה של הבקשת ARP
DNS	לקוח IP	של IP שרת DNS		53	MAC לקוח	של MAC שרת DNS	בקשת DNS עטוף ב: UDP, IP ETHER NET
DNS	שרת DNS	לקוח IP	53		של MAC שרת DNS	MAC לקוח	תשובת DNS מחזיר כתובת של השרת של הצא'ט
TCP	לקוח IP	של IP השרת של הצא'ט		הפורט של השרת (אליו לקוחות מתחברים)	MAC לקוח	של MAC שרת הצא'ט	בקשת חיבור לשרת צא'ט
TCP	לקוח IP	של IP השרת של הצא'ט		הפורט של השרת (אליו לקוחות מתחברים)	MAC לקוח	של MAC שרת הצא'ט	ההודעה אותה אנחנו רוצים להעביר ללקוח אחר(אנח נו שולחים לשרת צא'ט)
TCP	של IP השרת של הצא'ט	אל הלקוח השני אליו אנחנו רוצים להעביר את ההודעה	הפורט של השרת (אליו לקוחות מתחברים)		של MAC שרת הצא'ט	של MAC הלקוח אליו אנחנו רוצים להעביר את	השרת שולח את ההודעה אל הלקוח אליו רצינו לשלוח את

						ההודעה	ההודעה

.2

CRC בדיקת יתירות מחזורית- 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.

אופן פעולה: משתמש בפולינום המוגדר בפולינום יוצר מדרגה r. סוגים שונים של קוד CRC משתמשים בפולינומים יוצרים שונים.

בהינתן פולינום יוצר מדרגה r ובהינתן הודעה M שברצוננו לקדד, עלינו לבצע את הפעולות הבאות:

1. נוסיף r אפסים מימין להודעה.
2. נחלק בפולינום (תוך שימוש בחילוק של השדה מודולו 2)
3. נחסר את השארית תוך שימוש ב-xor במקום בחיסור רגיל.

נצרף את התוצאה שקיבלנו מימין להודעה המקורית ונשלח.

כמו בכל קידוד Checksum, הצד המקבל יבצע את שלבים 1 ו-2 ויוודא ש-r הביטים האחרונים שנשלחו זהים לתוצאה שהתקבלה.

3. HTTP - פרוטוקול שכבת האפליקציה של האינטרנט.

דף האינטרנט מורכב מאובייקטים, כך שכל אחד מהם יכול להיות מאוחסן בשרתי אינטרנט שונים. האובייקט יכול להיות קובץ HTML, תמונת JPEG, יישומון ג'אווה, קובץ שמע, ... דף אינטרנט מורכב מקובץ HTML בסיסי הכולל מספר אובייקטים שהפניה אליהם ניתנת לכתובת אתר.

- מצד הלקוח: דפדפן שמבקש ומקבל הלקוח)באמצעות פרוטוקול HTTP)וגם "מציג "אובייקטים ברשת.
- מצד השרת: שרת האינטרנט שולח)באמצעות פרוטוקול HTTP)את האובייקטים.

ה-HTTP משתמש ב-TCP שמבצע את פעולת החיבור מהלקוח לשרת -)תהליך "לחיצת-היד"(הוא יוצר את הסוקטים בפורט 80. HTTP לא זוכר שום דבר על הלקוח מבחינת הבקשות וכו'...

http 1.0 - פתיחת התחברות TCP, שליחת אובייקט אחד ויחיד ולאחר מכן התנתקות מה-TCP. זה שיטה לא טובה כי יש היום הרבה נתונים ברשת וזה יכול לגרום לעיכובים רבים כי אני כל הזמן פותח וסוגר את ההתחברות.

http 1.1 - פתיחת התחברות TCP, שולח כמה אובייקטים ביחד ולאחר מכן התנתקות מה-TCP. הרבה יותר יעיל ומשפר לי את זמן הגלישה. לגישה זו יש 2 תתי-גישות: (p = parallel מקבילי)

א. NP - אנו נשלח בקשה רק לאחר שקיבלנו את התשובה לבקשה הקודמת.

ב. P - נשלח כמה בקשות ואז נקבל את התשובות ביחד.

http 2.0 - .

מאפשר לשרת "לדחוף" תוכן, כלומר להגיב עם נתונים עבור יותר שאילתות ממה שהלקוח ביקש. זה מאפשר לשרת לספק נתונים שהוא יודע שדפדפן אינטרנט יצטרך לעבד דף אינטרנט, מבלי לחכות שהדפדפן יבחן את התגובה הראשונה, וללא תקורה של מחזור בקשות נוסף.

דחיסת כותרות ותעדוף של בקשות.

QUIC- ישנם כמה שינויים, הראשון הוא להפחית במידה ניקרת את התקורה במהלך החיבור הראשוני "לחיצת היד", בעת החיבור הראשוני השרת שולח גם data ללקוח אשר חוסך זמן בעתיד, מידע זה הוא הוא "המפתחות" וכו' בכדי ליצור הצפנה של הדברים(כאמור אנחנו רוצים הצפנה ולכן במקום לעשות את כל התהליך אנחנו חוסכים זמן בכך שאנחנו עושים זאת חלק השליחה הראשונית בכך אנחנו לא צריכים להגיד חיבור TCP). זה קורה כאשר לקוח פותח חיבור.

שינוי שני הוא, שעובדים מעל udp ולא tcp כמו שאמרנו בשינוי הראשון, למרות שאנו לא פועלים מעל TCP עדיין אנחנו שומרים על אמינות, הצפנה ועוד עקורונות של TCP.

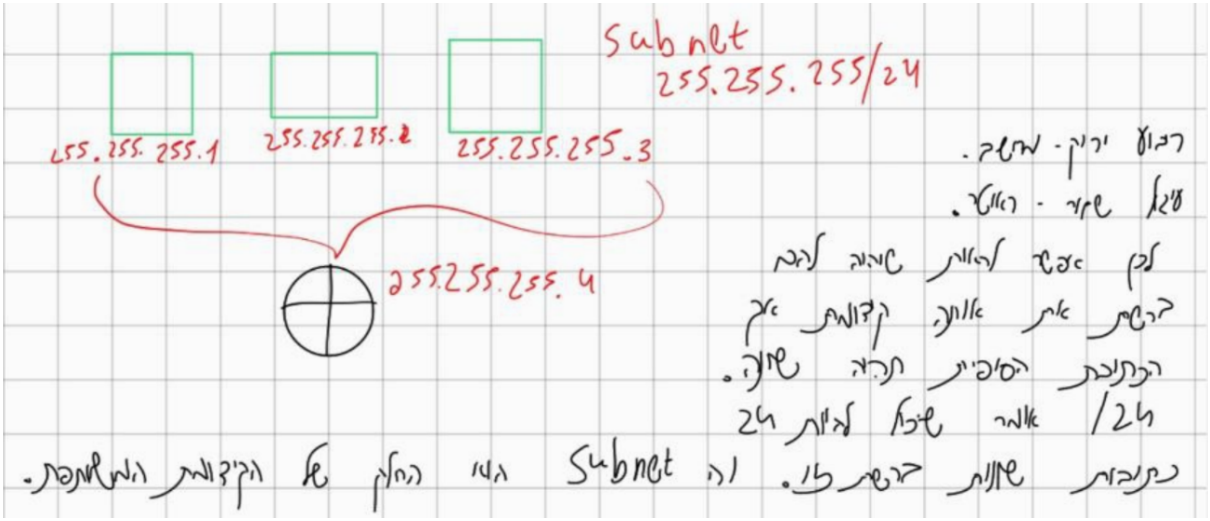
שינוי אחרון הוא, הוא שאנחנו פותחים מקביליות של שליחה, ובכך אנו מונעים את הבעיה הגדולה שהיית ב 2.0 שיכל להיווצר "צוואר בקבוק" ולהאט את השליחה של הקבצים. כאן גם אם יש בעיה באחד הזמרים זה לא

משפיע על שליחה כי זה עובד במקביל.

4.השימוש הנפוץ ביותר בפורט הוא בתקשורת מחשבים במסגרת הפרוטוקולים הנפוצים בשכבת התעבורה: **TCP ו-UDP**. פורט מזוהה לכל כתובת או פרוטוקול מסוים על ידי מספר באורך 16 ביטים היוצר 65536 כתובות אפשריות ל-UDP ו-65535 כתובות אפשריות ל-TCP. כתובת זו נקראת "מספר הפורט".
פורטים מוכרים הם פורטים המשמשים פרוטוקולים מוגדרים כסטנדרט. הצורך בפורטים מוכרים קיים כדי לקבוע סטנדרטים בהתחברות לשרתים המספקים שירותים מסוימים.
<u>לדוגמה</u>: על מנת שהדפדפן יפנה לאתר אינטרנט ב-HTTP, הדפדפן צריך לפנות לפורט פתוח על השרת שיקבל את הפניות אליו ויטפל בהן, והפורט הזה הוא הפורט המוכר לתעבורת HTTP - פורט 80.
הארגון האחראי על תיאום ורישום הפורטים המוכרים הוא IANA (קיצור של Internet Assigned Numbers Authority) המחלק את מרחב הפורטים לשלושה:

- 0 - 1023 - פורטים מוכרים, Well Known Ports
- 1024 - 49151 - פורטים רשומים, Registered Ports
- 49152 - 65535 - פורטים פרטיים או דינמיים, Dynamic or Private Ports

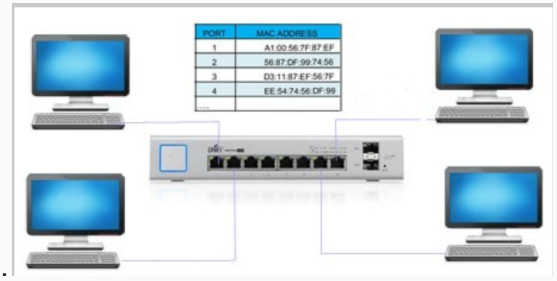
5.מהו subnet - נסביר ונבדיל בין מהו subnet ברעיונו הכללי וכך גם נבין מזה subnet mask שנגזר ממנו.
subnet - זוהי חלוקה לוגית של הרשת, לרשתות. יש לנו את תת הרשת אשר לא צריכה באמת את הראוטר בכדי לתקשר אחד עם השני.
למשל: אם אני מחובר לאינטרנט הביתי שלי ועוד מחשב מחובר לאינטרנט הביתי, ואני שולח לו הודעה אנחנו נתקשר תחת אותו subnet. (נכון זה נשמע מוזר אבל אנחנו לא באמת צריכים אינטרננט כדי לתקשר כי אנו מתקשרים דרך אותה רשת ואנו לא יוצאים לרשת חיצונית).
ה subnet שלנו זה אותה קידומת של כתובות הip ושוני בינינו יהיה רק בחלק האחרון כלומר(בכמות הכתובות השונות שאפשרו לי). נראה דוגמא שתמחיש יותר טוב:



ולכן הsubnet זה הרעיון של החלוקה של הרשת הגדולה לתת רשת מבחינה לוגית
ו subnet mask זהו בעצם הכתובת שמחשב מקבל שיש לה את אותה קידומת.
6. כתובת MAC היא מזהה ייחודי המוטבע על כל רכיב תקשורת לתקשורת נתונים בעת הייצור. כתובת ה-MAC מוטבעת בדרך כלל בכרטיס הרשת של המחשב או במודם.
ברשת הפרטית הביתית שלך, כל המכשירים ברשת מחוברים אך ורק לראוטר, והוא אחראי על ניתוב פקטות בין כולם.
מכיוון שכולם מתחברים אליו - הוא יודע למפות פורט פיזי שלו לכתובת ה MAC של המכשיר שמחובר אליו.
כשאתה רוצה לשלוח פקטה למכשיר מסוים ברשת שלך, אתה לא יכול לעשות זאת ישירות / פיזית - המכשיר היחיד שאתה מחובר אליו פיזית הוא הראוטר (כמו כל שאר המכשירים באותה רשת).
בשביל זה נועדה שכבה 2 - שכבת ה Ethernet .

אתה מעביר את הפקטה לראוטר פיזית עם Header מסוים (מידע בתחילת הפקטה, לפני המידע עצמו שאתה מעוניין להעביר) שאומר מהי כתובת ה MAC שלך, ומהי כתובת ה MAC של המכשיר שאליו אתה רוצה שתגיע הפקטה.
הראוטר מחובר לכל המחשבים ברשת פיזית, ולכן יודע איזה פורט פיזי שלו מחובר המכשיר עם כתובת ה MAC המבוקשת ומעביר את הפקטה אליו.
אז למה אנחנו צריכים בכלל את IP אם אנחנו יכולים להשתמש בMAC, אלא במקום ליצור מיליוני טבלאות כדי שהראוטר ידע לאן לשלוח אנחנו משתמשים ב IP אשר גורם להבדל בין הרשת החיצונית לפנימית ויוצר "רשת של רשתות" ובכך נחסוך הרבה זמן למצוא את היעד אותו אנו רוצים ונחסוך גם הרבה מקום של שמירת טבלאות. שימוש עיקרי של mac מול ip הוא שכאשר ראוטר מקבל פקטה עם כתובת ip והוא רוצה לדעת לאן לשלוח אותה, באמצעות הטבלה עליה הסברנו הוא ימצא את כתובת ה mac וידע לאן לשלוח את הפקטה (אנו מדברים על ראוטר ובמכשיר הנמצאים באותה רשת).

7. **סוויץ** - כאשר התקשורת נעשית בתוך הרשת הביתית אין צורך בשימוש בראוטר ,
נהיה זקוקים לראוטר רק שנרצה לחבר את הרשת הביתית לאינטרנט או לרשת אחרת חיצונית.
הסוויץ הוא בעצם מעביר תקשורת בין רכיבי התקשורת באותה רשת , תקשורת זו מבוצעת ע"י כתובת MAC address כפי שהסברנו לעיל(שאלה 6).
לכל סוויץ יש טבלה שבה יש שיוך בין פורט פיזי לבין כתובת MAC address , הסוויץ לומד את כתובות ה MAC ושומר אותם בטבלה. וזהו ההבדל העיקרי שבעצם הוא לומד תוך כדי, ולכן בהתחלה הסוויץ לא יודע כלום!
תקשורת זו נעשית בשכבה 2 - שכבת ה Ethernet.
תמונה להמחשה איך עובד סוויץ.



NAT - טבלת NAT ממירה את הכתובת הפנימית ל IP חיצוני(ובך מונעת מצוקה של כתובות IP כי בעצם ראוטר מייצג הרבה כתובות פנימיות ע"י כתובת אחת) , כתובת היעד ופורט היעד תמיד נשמרים גם שהמידע יעבור דרך ראוטרים אחרים הנמצאים ברשת האינטרנט(המידע הזה נשמר בטבלה).
תקשורת ברמת IP ופורטים ,ולכן התקשורת הזאת בשכבה מס' 3 , שכבת התעבורה עוברת ע"י כתובת IP ופורטים.
ראוטר- בעברית נתב(כמו שאמרנו בכל השיעורים "נתב" תפקידו לנתב את ההודעות). אין הבדל בינו לבין האחרים כי הוא משתמש בהם.

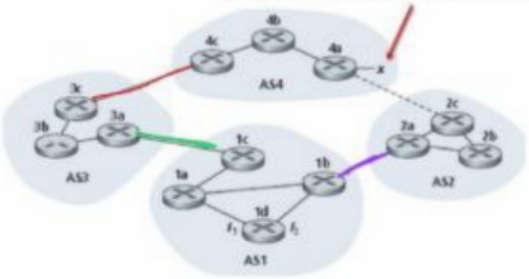
8. בגלל שיש לנו כל כך הרבה מכשירים מתחילות להיגמר לנו כתובות הip. ולכן היו כמה שיפורים שנוצרו.
יש לנו את הNAT עליו הסברנו בשאלה 7 שגורם לכך שנצטרך להשתמש בפחות כתובות.
כל רעיון הרשת הוא לפרק לרשת של רשתות ובכך בתוך רשת פנימית אין לנו בעיה מול הרשתות החיצוניות ונוכל להשתמש בכתובות הפנימיות והחיצוניות על מנת לחסוך כתובות.
יש לנו גם את ipv6 שזו עכשיו הגרסה החדשה של הכתובות במקום 32 סיביות אנחנו משתמשים ב ב128 מה שנתן לנו עוד מרחב כתובות ענק. הבעיה שנוצרה לנו היא שעכשיו לא כל אחד רוצה להשקיע את הכסף והמשאבים בכדי לעבור מ4 ל6. אז המציאו שיטה אשר עוזר לדעת מהי סוג הכתובת שלך כאשר אתה עובר מסוג IP לסוג השני, אנחנו יוצרים סוג של מעטפת ובכך אין לנו את הבעיה של המעבר בין סוגי הכתובות.

9. עשינו שאלה זאת בטאבלט כדי שיהיה ברור הסימונים.
קודם נשים הסברים על אופן הפעולה של OSPF, BGP, RIP כדי שיהיה יותר מובן למה שכתבתי את התשובות כך.
BGP - לכל מערכת אוטונומית (AS - Autonomous System) לניהול רשת באינטרנט מקוצה מספר מזהה ייחודי (ASN - Autonomous System Number). כל AS מהווה צומת ניתוב שמשתמש בפרוטוקול BGP כדי לבנות מסלולי ניתוב דינמיים כדי להתממשק מול AS-ים אחרים[2].
כדי לבנות מסלולים, כל AS מפרסם את הASN ורשימת תחומי כתובות (Prefix) שיש ברשותו והוא יכול להעביר אליהם תעבורה בחבילת BGP Announcement לשכנים שלו. כל שכן מקבל את ההודעה, אוגר את המידע ומשקלל אותו עם המסלולים ששמורים אצלו. לאחר מכן השכן מחלחל את המידע הלאה לשכנים שלו על ידי שרשור ה-ASN של עצמו להודעה. השכנים החדשים מקבלים את תחומי הכתובות יחד עם רשימת ה-ASN-ים שעליהם לעבור כדי להגיע לאותה קבוצת כתובות.
כדי להימנע ממעגלים, AS-ים מפילים הודעות BGP שמכילות את ה-ASN של עצמם.
OSPF - פרוטוקול ניתוב להעברת נתונים בין ראוטרים שונים הנמצאים באותה מערכת אוטונומית. משתמש באלגוריתם דייקסטרה, המשמש לחישוב עץ המרחק הקצר ביותר. OSPF משתמש בעלות העברת הנתונים (מספר המציין את גודל רוחב הפס) לצורך חישוב המרחק, ותמיד יבחר את הנתיב הזול ביותר להעברת חבילה מהמקור אל היעד.
נתב המנתב חבילות בהתבסס על OSPF מנהל רישום של כל הנתבים שהוא "מכיר", והנתיבים אליהם. כאשר מגיעה אליו חבילה הוא מעביר אותה אל נתב היעד דרך הנתיב בו עלות התעבורה היא הזולה ביותר.
RIP - נתב המשתמש ב־RIP מנהל רישום של כל הנתבים אותם הוא "מכיר", הרשתות המחוברות אליהן, וכמות הצעדים בכל נתיב לכל יעד. כאשר מגיעה חבילה אל הנתב הוא יעביר אותה בנתיב בו היא תעבור מינימום צעדים עד לרשת היעד, שיטה זו מכונה ספירת צעדים (hop count). הנתב מבקש עדכונים לגבי שינויים בטופולוגית הרשת מהנתבים המחוברים אליו כל שלושים שניות, וכך הוא נשאר מעודכן לגבי שינויים בנתיבים המובילים אל היעד, ומקבל מידע על נתבים חדשים שחוברו אל הרשת.

לשים לב שכאשר אני כותב" חיבור ישיר" הכוונה היא בין מערכות אוטונומיות AS.

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 הת רשת x
f. בעזרת איזה פרוטוקול לומד הנתב 3a על הת רשת x
g. בעזרת איזה פרוטוקול לומד הנתב 1c על הת רשת x
h. בעזרת איזה פרוטוקול לומד הנתב 2c על הת רשת x



e. X נמצא ברשת AS4 יש לנו קו ישיר בין 3c ל- 4c (מומן בקצה בכל שבין על גבי קו מקבר). כלומר יש לנו חיבור ישיר בין 3c ל- 4c. בין AS4 ל-AS3 לכן הנתב לומד את הדרך לפי פרוטוקול BGP.

f. אם הנתון ב- a מרצים OSPF מאיזון ש AS3 מריץ OSPF אם הנתון. אלא 3c יקבור מידע ל- 3a כדי להגיע ל- X. ולכן 3a לומד את הדרך לפי OSPF.

g. 1c לומד את הדרך לפי BGP. מכיוון שיש לנו חיבור ישיר בין AS1 ל- AS3 מאמין בירוק. ולפי הסעיפים הקודמים נקבל שבגלל החיבור הישיר ל- 3a שלומד את הדרך ל- X. 1c יתנהג ב-BGP אם מתר להגיע אליו.

h. נתון אם a AS2 מריץ פרוטוקול OSPF. לכן אנו מקבלים מידע ש 2a מחובר ל- 1b (מסומן בכחול) בחיבור ישיר 2a מקבל מידע מ- 1b באמצעות BGP. 1b מקבל מידע מ- 1c אם פרוטוקול RIP (נתון ב- b). בסעיף g הסברנו איך 1c מקבל את המידע. ולכן 2c לומד את הדרך ל- X באמצעות OSPF.