

## **סיכון החומר ברשתות מחשבים – סמסטר חורף 2013**

### **מרצה: אלכס פריד**

**נערך ונכתב ע"י שמעון ארזיאן, רון רוזנפלד, גיא פלאג**

הערה חשובה:

החומרים הותאמים למה שנלמד בהרצאות בסמסטר חורף 2013. חלקים מהסיקוםפה תורגמו לשירות מהספר ומוקייפדייה.

**תוכן עניינים:**

עמוד	נושא
2	מבוא
5	שכבת האפליקציה
16	שכבת התעבורה
20	שכבת הרשת
33	שכבת הקישוריות
46	Cognitive network
49	<b>Network Security</b>

## 1. חלק א' מבוא:

### 1.1 מה זה אינטרנט:

האינטרנט היא רשת מחשבים המחברת מאות מיליונים של התקני מחשב בכל העולם. התקנים הם: מחשבים, שרתים, לפטופים, טלוייזיות, קונסולות משחק, טלפונים סולריים, מצלמות רשות, מערכות אבטחה וכו'.

כל התקנים הללו נקראים **hosts** או (end systems ובערבית: תחנות קצה). החיבור בין תחנות הקצה מתבצע באמצעות **Communication Links** (תווור תקשורת) ובאמצעות **Packet Switches**. בכל **Communication Link** יש את התווך הפיזי אליו הוא עובד, וכל אחד יש קצב שידור שונה. \*קצב השידור (transmission rate) של **Communication link** נמדד ביחידות של bits/second (ביטים לשנייה).

באינטרנט באופן כללי, כאשר תחנת קצה אחד שולחת מידע לתחנת קצה השנייה - תחנת קצה השולחת, מחלקת את המידע לחבילות מידע, כלומר ל-packets. למה המידע מוחולק? כדי לא להעיס על התווך תקשורת, אם המידע גדול מדי הוא עלול להעיס על הרשת ולתזקע" אותה.

על מנת שנוכל להעביר מידע באינטרנט אנו זוקקים לשימוש בפרוטוקול. פרוטוקול – הוא מוסכמה בין מכשירים על איך המידע יועבר ביניהם. בעזרתו הפרוטוקול כל host יידע איזה packet שייר ל', בוסף הוא יידע איך לחבר את כל ה-packets זהה על מנת לקבל את חבילת המידע המקורי. Packets - נשלחים דרך הרשת לתחנת היעד ושם מחברים מחדש את ה-packets למילוי המקורי. Packet Switch – תפקידו הוא לחתךpacket המגיע מעתה הכניםות שלו ובהתאם לעד הוא מעביר את ה-Packet - אל היציאה המתאימה.

ה-Packet Switches, Routers וה-link layer switches – הדריך שעושה(packet switches נקראת route או (path ובערבית: מסלול או נתיב).

#### מושגים כללים:

- **Host** - תחנת קצה. יכול להיות מחשבים, שרתים, לפטופים, טלוייזיות, קונסולות משחק, טלפונים סולריים, מצלמות רשות, מערכות אבטחה וכו'. באופן כללי host היא נקודת חיבור אחרון.
- **Communication Link** – הוא התשתיית של הרשת כלומר מקשר בין מכשירים. יכול להיות ממושך באופן פיזי ע"י כבל ויכול להיות גם אלחוטי.
- **Bandwidth** – רוחב פס, כמוות המידע העוברת ליחידת זמן, נמדד ביחידות של ביטים לשנייה.
- **Packet** – חבילת מידע העוברת ברשת.
- **Packet Switch** – מכשירים שתפקידם להעביר את ה-packet ברשת, תוך התחשבות ביעד ה-packet.

### - Circuit switching 1.2

או בעברית "מייתוג מעגליים" - בתקשרות וברשותות מחשבים היא שיטה להתקשרות כך שלפני העברת חבילות מידע בין תחנות המקור לתחנת היעד יש להקים מעגל (או ערך תקשורת) "יעדי" להתקשרות (sessions) זו. כל עוד ההתקשרות ממשיכה, מידע אחר לא יכול לעبور באותו ערך תקשורת. באופן כללי בכך שתתי תחנות ידברו ביניהם עושים קוצר בכבל, וכך הן יכולות לתקשר זו עם זו.

רשתות טלפוןנים ישותן הן דוגמה מובהקת למייתוג מעגליים - היה צריך לבקש מהמרכזיה לחבר את המתקן אל היעד, בין אם ישירות או דרך מרכזיה אחרת, ובכל אופן בסופו של דבר היה צריך ליצור באופן פיזי מעגל חשמלי בין שני מכשירי הטלפון.

כiom רשתות תקשורת המשמשות למייתוג מעגליים חולקות את אותו כבל תקשורת בין מספר ערכאים על ידי אפנון כל אחד מהערךאים בתדר שונה, אך עדין לכל ערך מוקצה תדר ייחודי שמשמש להתקשרות בין שתי תחנות בלבד בכל רגע נתון.

היחסון למייתוג מעגליים הוא בניצול מלאו רוחב הפס לאורק הזמן: כאשר מוקצתה ערך תקשורת לצורך התקשרות, ובמהלך ההתקשרות רוחב הפס לא מנוצל במלואו או שקיים פרקי זמן בהם לא מועבר מידע כלל - אף אחד אחר לא יכול להשתמש בערך התקשרות עד שההתקשרות לא מסתיימת.

### - Message switching 1.3

בשיטתה זו בכל פעם עברת הודעה בשילובתה בין יחידות. כל יחידה שבה יעבור המידע בדרך לתחנת היעד שלו, תמתין עד שתתקבל את כל חבילת המידע בשילובתה. רק כאשר תוכל להמשיך לשילוח את ההודעה תעשה זאת, היא תשמור את ההודעה בكون קשיה מבנה שיש לה.

בהתבה ש**N** צמתים ו**R** הוא קצב ההעברה.

### - TDM (Time Division Multiplexing)

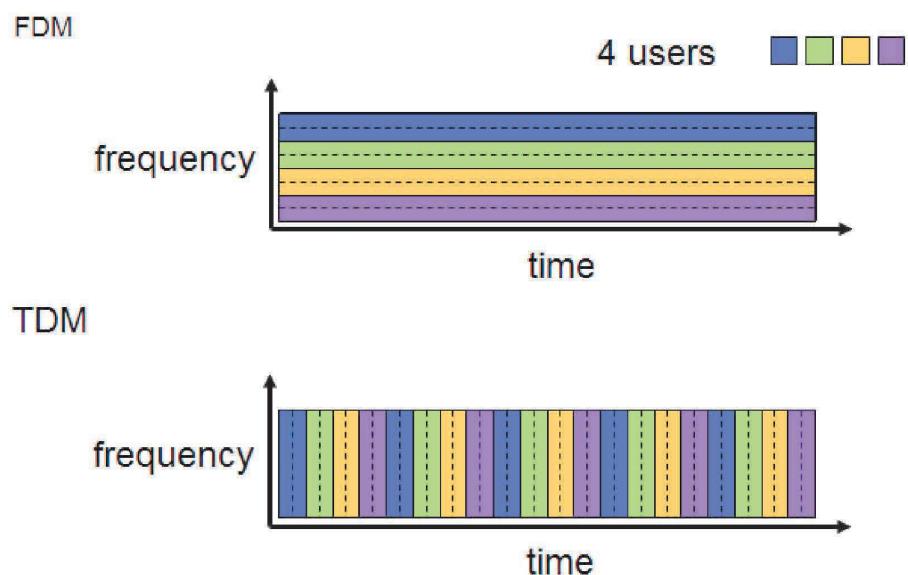
מחליקת את הזמן למוגרות זמן וכל מסגרת כזו מוחולקת ל-**N** סוליטים שמוקצים לכל אחד מ-**N** הצמתים. כאשר לצומת יש הודעה לשילוח, ההודעה מועברת בזמן המועד לצומת זו. כאשר לכל השולחים ניתנה האפשרות לשילוח, הסבב מתחלףשוב. לדוגמא תחנה **X** תסדר תמיד בסלוט מספר 2.

חוchnות:

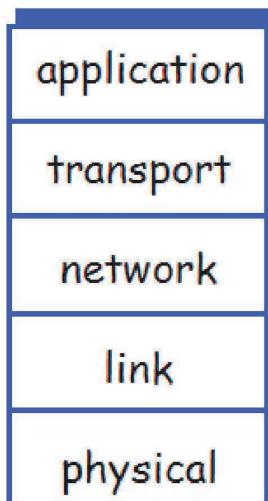
1. יש מצב שהשלוח יחכה הרבה (ולרוב סתום) עד להגעת התוור שלו לשילוח.
2. במצב שיש רק שלוח אחד עדין תהיה העברה בקצב של **N/R** (לא טוב).

### - FDM (frequency Division Multiplexing)

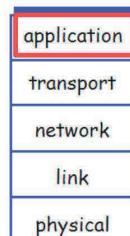
מחליק את **R** לתדריות שונות (כל אחד עם קצב של **N/R**) ומוקצתה כל תדרות לכל אחד מ-**N** הצמתים. בעצם יוצר **N/R** ערכאים חדשים, קטנים יותר מאשר **R** המקורי. גם כאן, אחד החסרונות הוא שצומת מוגבלת לקצב העברה של עד **N/R**.



## חלק ב – שכבות הרשת



### שכבה האפליקציה



#### עקרונות של אפליקציות רשת

הבסיס לכנתיבת אפליקציות הוא כתיבת תכניות שרצות על מערכות קצה שונות וمتקשנות אחת עם השנייה ברשת. לדוגמה: באפליקציה Web יש 2 תכניות שמתקשנות אחת עם השנייה: הדפדפן ב-PC של המשתמש מתקשר עם שרת האינטרנט. הערה: לא ניתן לכתוב תכניות שיוצרים על רכיבי תקשורת (כמו ראותרים וסוויצ'ים) משום שהן לא שייכות לשכבה האפליקציה (החל משכבה הרשת ומטה). ארכיטקטורת האפליקציה נוצרת ע"י יוצר האפליקציה ומכתיבה כיצד האפליקציה בנויה במערכות הקצה השונות.

#### ארQUITקטורות של אפליקציות רשת

שנן 2 ארכיטקטורות בהן יכול לבחור המפתח:

Client Server .1

P2P .2

– Client – Server

תמיד יש "מחשב מאוח" שמשמש כשרת שעונה על בקשות של "מחשבים מאוחים" שהם בעצם הלקוח. כאשר שרת מקבל בקשה לאובייקט מחשב לקוח הוא מגיב ע"י שליחת האובייקט המבוקש למחשב

- לקוח. בשיטה זו, מחשייב לקו"ם לא "mdbrium" אחד עם השני אלא רק דרך שרת. כמו כן, לשרת יש כתובת קבועה. דוגמא לאפליקציות כאלה: E-mail, Web, FTP.
- \* במקרים בהם שרת אחד לא יכול להשתלט על כל הלקוחות נעמיד עוד שירותים. זה נקרא מרכז מידע ולחוב זה נוצר על מנת ליצר שרת וירטואלי חזק.
  - \* אפליקציות מבוססות שיטה זו הן בדרך כלל intensive infrastructure משומש שהן מחייבות את מספק השירותים לרכוש חוות שירותים.

### תהליכי שירות ולוקו"

שכבות הרשת מתקיימות מזוגות של תהליכי ששלוחים הודיעות אחד לשני, לרבות אנו מקטלגים זוג זה כשרת ולוקו", יש לשים לב שבעיר ב-P2P, תהליך יכול להיות גם שרת וגם קו"ם (משתמש יכול גם להוריד וגם להעלות). עיקרונות תהליך ש"יוזם" התקשרות הוא הלוקו"ם והתהליך המגיב הוא השירות. תהליכי מתקשרים (שולחים בקשה ומקבלים תשובה) באמצעות מנשך Socket. (באנלוגיה לבית - אם תהליך הוא בית אז socket הוא הדלת). ה-socket הוא דוגמא ל-Application API (Application Programming Interface). הוא גם שלב מעבר בין האפליקציה לשכבות התעבורה. מפתח היחסום יכול לשולט בפרוטוקול התעבורה.

### שירותי תעבורה לאפליקציות

- העברת מידע אמיןה – אם פרוטוקול מספק תעבורת מידע אמיןה, הוא ערבות לכך ששם מידע לא יאביד בדרך.

### תפוקה (Throughput)

קצב העברת הביטים שהתהליך השולח יכול להעביר לתהליך המתקבל. על הפרוטוקול לשמור על קצב אחיד וקבוע מראש. יישום שיש לו דרישת לקבע מסויים נקרא Bandwidth Sensitive Application. יישום שני לו דרישת לקבע קבוע (תליי בתעבורה שזמןה כרגע) נקרא Elastic Application.

#### נוסחה - נסמן:

X = כמות המידע שיש להעביר.

Y=ס"כ הזמן שלוקח לנו להעביר אותו

$$\text{Throughput} = \frac{X}{Y}$$

- תזמון – פרוטוקול משכבות התעבורה יכול להבטיח תזמונים מסוימים לדוגמא: שביט שנשלח דרך socket יגיע לצד מקבל תוך פחות זמן מסוים.
- שירות אבטחה – תעבורה ניתנת להצפנה לפי הפרוטוקול.

## פרוטוקולים בשכבות האפליקציה

### 1. HTTP (Hypertext Transfer Protocol)

הוא פרוטוקול שכבת האפליקציה. HTTP מישם ב-2 תכניות ללקוח ושרת שמופעלים במערכות קצה שונות ומבדדים אחד עם השני ע"י החלפת הودעות HTTP.

טרמינולוגיית הרשת מורכבת מדף רשת שמורכבים מאובייקטים שהם בעצם קבצים (כמו HTML, JPEG, JAVA, Hypertext markup language) וועוד.

רובה דפי הרשת מורכבים מקובץ HTML בסיסי והתייחסות לאובייקטים אחרים ע"י כתובות URL (Uniform Resource Locator).

לדוגמא, אם דף רשת מכיל HTML ו-5 קבצי JPEG אז לדף הרשת יש 6 אובייקטים.

לכל כתובות URL שני מרכיבים: שם השרת עלייו יושב האובייקט וה-path לאובייקט.

**HTTP משתמש ב-TCP כפרוטוקול העברת הבסיסי שלו (במוקם לעשות זאת באמצעות UDP).**

הלקוח יוזם תקשורת TCP עם הרשת, ברגע שהקשר נוצר תהליכי הלקוח והשרת מתקשרים ב-TCP בעזרת ה-sockets. הלקוח שלוחה הודעת בקשה HTTP לתוך מנשך ה-socket ומתקבל הודעת תגובה HTTP מתוך מנשך ה-socket. וכך גם לגבי הרשת.

מרגע שההודעה יצאת מה-socket של אחד הצדדים, התעבורה עוברת לאחריות TCP.

יש לשים לב שキャッシו של הלקוח לא נשמרת או מאוחסנת בשום מקום, גם אם נעשו כמו בקשות לאותו אובייקט אחת אחריו השניה, התהילה יהיה זהה.

HTTP Stateless Protocol נקרא HTTP ללא שומר מידע קודם על לקוח.

### 2. FTP – העברת קבצים

על מנת שלמשתמש תהיה גישה לחשבון מרוחק הוא צריך לספק מזהה וסיסמה. לאחר מכן המשתמש יוכל להעביר קבצים מתקינה מקומית למערכת הקבצים המרוחקת ולהיפר.

המשתמש מתקשר עם FTP דרך FTP User Agent. הוא מספק hostname של הרשת הרוחק, מה שגורם להיווצרות של חיבור TCP בפורט 21. המזהה והסיסמה נשלחים עם החיבור ומרגע שהשרת אישר את המשתמש יכולה להתחיל העברת הקבצים. FTP משתמש בשני חיבורים מול TCP:

1. **Control Connection** – משמש עבור שליחת אינפורמציה לבקשתה בין שני המחשבים (יוזר, סיסמה ועוד)
2. **Data connection** – משמש לשילוח הקובץ. בغالל ש-FTP משתמש בבקורת חיבור מופרדת, נאמר שהאינפורמציה לבקשתה נשלחת **out of band**. (in-band זה כשהיכל נשלח בחיבור אחד.)

### 3. SMTP - דואר אלקטרוני

מורכב מ-3 מרכיבים עיקריים: משתמש, שירות מייל, SMTP.

- **משתמש (User Agent)** – קורא המייל יכול ליצור ולעורך את תוכנות המייל של הלוקו.
- **שירות מייל** – לכל תוכנת מייל יש שירות משלחה שמנהל את ההודעות שנשלחות ללקוח. מכיל את התיבת עצמה, תור הودעות לשילחה וכו'.

#### (Simple Mail Transfer Protocol) SMTP

דואר מתחילה ב-User Agent של השולח, עבר לשרת הדואר המתאים ומשם מועבר לשרת הדואר המיעוד. מהשרת הוא מועבר ל-User Agent של הצד מקבל.

**SMTP משתמש ב프וטוקול TCP להעברת מיילים.** שירות מייל יכול לשמש גם כלקוח (שלוח) וגם כשרת (מקבל).

smtp משתמש ב-ascii של 7 ביט גם עבור header וגם עבור גופ ההודעה. כל אובייקט חייב להיות מקודד כר.

MIME – מגדיר איך כל אובייקט בהודעה מקודד ובאיזה סוג של אובייקט מדובר (תמונה, טקסט וצדומה).

**דוגמה:** אליס רצתה לשולח מייל לבוב.

אליס פותחת את תוכנת המייל, מספקת את המייל של לבוב, כתבתת הודעה ושולחת.

המוכנה שולחת את ההודעה לתור ההודעות בשרת המייל. צד הלוקו של SMTP, שרצ על שירות smtp של אליס, רואה את ההודעה בתור של ההודעות ויזום חיבור TCP 25 לשרת SMTP שרצ על שירות המייל של לבוב. אחרי SMTP handshaking, לוקו ה- SMTP שלחו את ההודעה של אליס לחיבור ה-TCP.

בשרת המייל של לבוב, צד השירות של SMTP מקבל את ההודעה אז שירות המייל של לבוב ממקם לבוב בתיבת mbox.

לבוב פותח את תוכנת המייל וקורא את ההודעה. \* SMTP משתמש גם בחיבור כמו HTTP .

### השוואה מול HTTP

שניהם מעבירים קבצים תוך שימוש בחיבור **persistent**.

HTTP הוא בעיקר protocol pull – מישוה מעלה אינפורמציה לשרת אינטרנט ויוזרים משתמשים ב-HTTP כדי למשוך את האינפורמציה.

SMTP הוא protocol push – שירות המייל הנשלח "דוחף" את ההודעה לשרת המייל שמתקבל.

ב-HTTP אין הגבלה על קידוד ascii 7 ביט, כמו שיש ב-SMTP .

צורת הטיפול בקובץ טקסט/תמונה: HTTP כמו (encapsulate) כל קובץ צזה עם הودעת תגובה HTTP.

## DNS (Domain name System)

תפקידו העיקרי של DNS הוא לתרגם hostname ל-IP המתאים. מאפייניו ה-DNS:

- מסד נתונים שימושם בהיררכיה של DNS Servers.
- פרוטוקול של שכבות האפליקציה שמאפשר לhost "לחשאל" את מסד הנתונים.

### DNS עובד ב-UDP ורץ על פורט 53

פרוטוקולים כמו HTTP, SMTP שלוחים שאילתת למסד עם ה-Hostname ומקבלים כתשובה את ה-IP של hostname.

#### איך עובד DNS

1. לא בונים שרת DNS אחד מרכזי שיטפל בכל בקשות התרגום (centralized design) משומש ש-Single Point Failure – שרית ייחד יאלץ להתמודד עם כל הבקשות לבד זהה בלתי אפשרי.
2. Traffic Volume – שרית תמיד יהיה רחוק מקומות מסוימים, מה שיגרום לאי-יעילות תמידית במקומות אלו.
3. Distant centralized database – השרת תמיד יהיה רחוק מקומות מסוימים, מה שיגרום לאי-יעילות תמידית במקומות אלו.
4. Maintenance – קשה לתחזק מסד נתונים גדול (כל עדכון host צריך לעדכן במסד נתונים).

- בפועל אין שרת DNS ייחד אלא מס' רב של פזירים בכל העולם. ישנו 4 סוגי שרתי DNS:
- DNS Root Servers – ישנו 13 שרתיים (רובם בצפון אמריקה) אליהם פונים שרתי ה-IP המוקומיים במקורה מהם לא יודעים את כתובת ה-IP של hostname מבוקש. במקרה זה שרת DNS מקומי מתנהגת כמו לquo root. אם הוא יודע אז נשלחת תשובה לשרת המקומי שישלח תשובה לhost.
  - Local DNS Server – שרת-h DNS הקروب ביותר ל-host ששולח את השאילות (אם host בין IP של host אחר שישיך ל-ISP או שרת-h DNS המקומי יספק תגובה מיידית).
  - TLD (Top Level Domain) Servers – אחראים על השירותים com, org, net, edu, gov ועל כל שרתי המדינות כמו fr, jp, uk.
  - Authoritative DNS Servers – כל ארגון עם גישה ציבורית לשירותים חייב לספק רשומות מתחזקת את EDU.

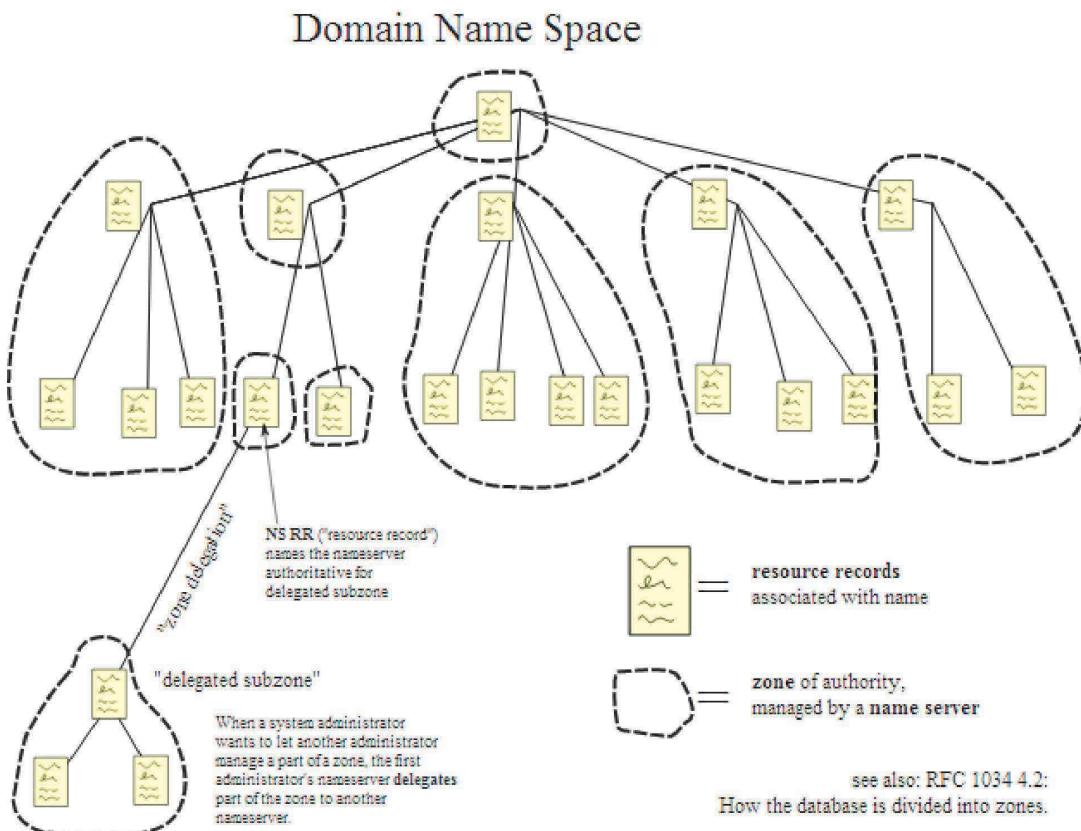
ישן 2 צורות עבודה בין זוג שרתי DNS:  
שאילתת רקורסיבית – כל שרת DNS שואל את השרת שמעליו ומחייב תגובה לשרת שמתוחתי כאשר שרת DNS הראשון עבד רק עם זה שמעליו.

שאילתת איטרטיבית – כשרת DNS לא יודע תשובה הוא מחזיר לשרת השולח את כתובת ה-IP של השרת הבא בתורו, שאולי יידע את התשובה. התהליך חוזר על עצמו עד שהשרת הראשון מקבל תשובה.

#### DNS Caching

באופן כללי DNS מנצל באופן יעיל את עיקרון caching על מנת לשפר ביצועים ולהפחית את מס' ה-DNS משתתפים בתהליך השאילתת.  
כאשר שרת קיבל תשובה DNS (כתובת IP) הוא מוחסן את המיפוי בזיכרון במקביל לשילוח התגובה. אם מגיעה השאילתת נוספת באותו IP, DNS יספק מיידית את התשובה גם אם הוא לא השרת ה"סמכותי" של hostname.

שיטה זו עזרת בזמן delay ומשחררת את העומס שנוצר בעקבות שליחת שאלות בין שרת ה-DNS השונים.



## P2P applications

ארQUITטורה זו אינה מתבססת על שרת זמן אלא על חיבורם שנוצרים בין משתמשים. כמעט ואין הסמכות על מחשב שרת. במקום זאת, השימוש מנצל תקשורת ישירה (בלי שרת) בין זוגות מחשבים (Peers). דוגמאות לכך: Emule, סקייפ...  
**הערה:** ישנים יישומים משלבים בין שתי השיטות (P2P ו"שרת-לקוח"), לדוגמה תוכנות להעברת מסרים משתמשות בשרת על מנת לאתר IP אך ההודעות עובחות ב-P2P.  
 שיטה זו יותר חסכונית משום שהיא לא דורשת אחיזת שרתים.

## ישנים כמה יישומי P2P: P2P File Distribution

במror התחלת, נדון באפליקציית ה-P2P – BitTorrent.

מבוא: זהו פרוטוקול תקשורת P2P לשיתוף קבצים, המשמש להעברת קבצים ישירה בין מחשבים בטכнологיה P2P וידוע בעיקר בשל היותו כל' שימושי להפצה פיראטית של חומרים המוגנים בזכויות יוצרים.

להורדת קובץ מסוים צריך קודם קודם כל לחפש באינטרנט אחר קובץ torrent. קובץ זה מכיל מידע על שרתי Tracker המכילים מידע על משתמשי "טורנט" אחרים שהקובץ הרצוי, או חלקים ממנו, נמצאים אצלם. את קובץ ה-torrent יש להפעיל באמצעות תוכנת לוח כלהה, כגון BitTorrent. תוכנת הלוקה מבקשת מהרתו ה-Tracker מידע על המשתמשים אצלם נמצא הקובץ הרצוי, וכשהזה מתקיים היא מתחילה להוריד מכל אחד מהם חלק אחר של הקובץ.

היחידות של פרוטוקול torrent הינה ביכולתו להתייחס לכל פרוסה של הקובץ כאלו יחידה נפרדת, ומהרגע בו ירדת אל המחשב פרוסת קובץ אחת, משתמשי torrent אחרים יכולים להוריד מהמחשב זה. בסופו של דבר, תוכנות Torrent שונות מספקות מהירות הורדיה והעלאת קבצים גבוהה בזכות שיטת "שיתוף בחלוקת" זו.

בכל הורדת קובץ ההורדיה מתבצעת סימולטנית מספר משלפנים, ומכל אחד יורט חלק אחר של הקובץ. כך, אם נניח שככל משתמש torrent משתף וחובפס העלה לקובץ של הקובץ ככמה עשרות קילו-בייט לשניה, בסופו של דבר מורידה התוכנה מספר פרוסות של הקובץ ממספר משלפנים במהירות שיכולה להגיע לכמה מאות קילו-בייט לשניה, מה שיכל עקרונית לספק הורדיה של תכנים פופולריים בגודל של כ- 700 MB בשעה.

שלא כמו תוכנות שיתוף מסורתיות, המבצעות את כל הפעולות, כולל איתור מקום הקבצים, דרך חיבור P2P – משתמש פרוטוקול torrent בשרתיהם המנהלים "אינדקס" של המידע על מיקום הפיזי של הקבצים, דבר המסייע למציאת המקורות במהירות גבוהה וליעול תהליך השיתוף.

### פרוטוקול בשכבות התעבורה: TCP

ה-BitTorrent לדוגמא משתמש בשתי גישות בבחירה סדר הקבצים להורד. Random start – בחירה אקראית של הקבצים. שיטה זו עוזרת ליצור שני בין חלקים שמודדים בכל עת ובכך מגדילה את הסיכוי להערכה, מכיוון שכדי שתתבצע הערכה צריך שלמשתמשים יהיו חלקים שונים של הקובץ כדי שייעבירו אחד לשני.

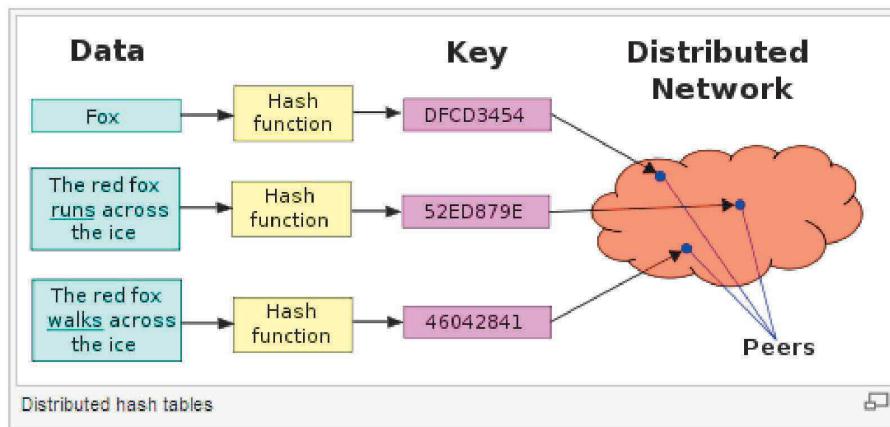
Rarest-first – גישה זו דואגת שהקבצים הנדרים ביותר יופצו קודם ובכך דואגת שהם יהפכו לנפוצים יותר.

### מנוחות כלליים:

חטיות Choking mechanism – הרעיון הוא שמי שטרם רוחב פס יותר גדול ומעלה הרבה יותר חטיות יכול בחזרה הרבה חטיות להורד. מצב זה יגרום לכך שהמשתמש יקבל רוחב פס יותר גדול ובכך הוא תמיד יקבל מידע ולא מתין.

End game mode – הרעיון הוא שגם משתמש עומד לסיים את ההורדיה של הקובץ, אך רוחב הפס שלו יקטן כך שהחטיות שrank לו יש יעבורו למשתמשים אחרים שיוכלו להמשיך הפצה.

## DHT (Distributed Hash Table) .1



DHT היא קבוצה של מערכות מבוזרות אשר ביחד נותנות שירות של מען טבלת גיבוב (hash table). זהו מסד הנתונים של P2P והוא מורכב מזוגות (key, value). כאשר עמית מבצע שאלתה באמצעות ה-key ומסד הנתונים מחזיר את הענן. עיתים גם יכולים לבצע הכנסה של זוגות כ אלה למאגר הנתונים.

\* לכל עמית מוקצה מס' זיהוי ייחודי בטוחה  $[0, 2^n - 1]$  עבור ח קבוע. כל מס' זה מבוטא באמצעות ח ביטים וכל מפתח הוא integer בוואטו טווח. לא כל הערכים המוכנסים הם מס' שלמים ولكن ניתן בטבלת hash על מנת למפות כל מפתח למס' שלם בטוחה הנtent. פונקציית hash-ה היא זמינה לכלם. כאשר אנחנו מתיחסים למפתח אנו מתייחסים למקומו בטבלת hash.

הकצאת מפתחות לעיתים: בהתחשב בעובדה שמס' זהה וערך המ באותו טווח, הגישה הטבעית היא להקצות כל זוג (key, value) לעמית שהמס' הזהה שלו קרוב יותר לkey. (קרוב ביותר = הבא בתור).

דוגמא:

4 – מס' זהה ומפתח בטוחה של [0, 15].  
יש 8 עיתים עם המספרים הזהה 1,3,4,5,8,10,12,15.  
נניח שרוצים להכניס את הזוג (11, JohnyWU) לאחר ו-12 הוא היכי קרוב, נאחסן את הזוג אצלו.  
אם הערך של key הוא בדיק זהה לאחד המס' הזהאים, שם יוחסן הזוג.

## 2. DHT מעגלי

כל עמית מודע רק למי שהוא העוקב שלו, זה מקרה ייחודי של overlay network (רשימת מחשבים שבנוייה על גבי רשת האינטרנט).

\* אם עמית רוצה לדעת איזה עמית אחראי על key מסוים הוא מעביר בקשה לעוקב שלו עד אשר מקבלת תשובה. ברגע שמתקבלת תשובה חיובית, התגובה מגיעה ישירות לעמית המבקש. במקרה הגרוע ייקח לנו  $O(n)$ .

מעגל עם קיצורים – כל עמית יודע מי העוקב שלו וכיום קדם לו. כל הסיבוכיות יורדת מ( $O(n)$ ) ל ( $O(\log n)$ ).

### Peer Churn .3

כאמור ביישומי P2P עמידת יכול לבוא ולילכת מתי שירצה ולכן יש לנקוט זאת בחשבון בזמן יצירת מבנה ה-DHT. נדרש שכל עמידת-ID ידע לעקבות אחרים 2 העוקבים שלו (בדוגמא הנ'ל 4 יcir את 5,8 ומקביל גם שידע אם עוקבים אלו עדין קיימים ("ע"י" שליחת פינג כל הזמן)).

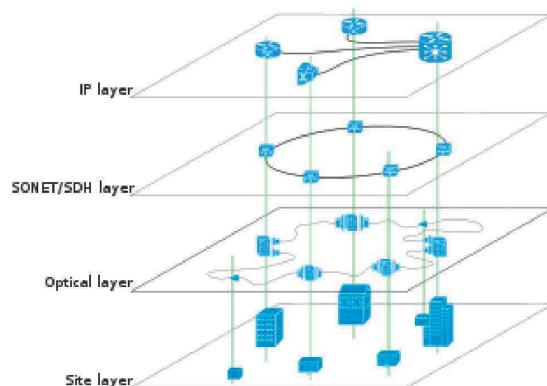
\* כאשר עמידת עוזב, מתבצע קישור כמו בהזאה מרשימה מקושרת. אם עמידת 5 עזב את 4, צריך לתקשר עם 8 ולקבל את כתובת IPו שלו.

\* כהעמיה רצתה להיכנס, מתבצע חיפוש "מי קודם ועקב לו-ח?"  
השאלה עוברת בין העמידים עד לקבלת תשובה חיובית.

### Skype .4

מושגים:

Overlay Network – רשת משנית הבנויה "מעל" רשת האינטרנט. הצלמתים ברשת זו מתנהגים כאילו מחוברות זו לזו למטרות שלא בהכרח קיימן חיבור פיסי בינהן.



- Supernodes - צמתים המשמשים מעין שרתים, ככלمر במקרה שרת אחד כל צומת משמשת להעברת נתונים וניהול המידע. בשיטה זו נחוץ ה צורך בשרת יציב אחד אך כל אחת מהצמתים דורשת יותר רוחב פס ומשאבים.

מבוא: זוגות משתמשים מתקשרים אחד עם השני בזמן אמת. זהוי למעשה רשת overlay היררכית עם Supernodes.

שמות משתמשים ממופים לכטבות IP ומופצים באמצעות supernodes שמחוברים ביניהם, כך שהמשתמש יוכל למצוא משתמש supernode אחר.

- משתמש משתמש בפרוטוקול UDP להעברת הideo, את החיבור נעשה באמצעות ה-TCP

© נערך ונכתב פֵי שמעון אורזיאן, רון רוזנפָּלֶד, גיא פָּלָג

### (Voice over Internet Protocol) VoIP

הו שם כללי לטכנולוגיות ולשיטות שונות להעברת דיבור על-גבי רשתות IP (כדוגמת רשת האינטרנט).  
טכנולוגיות אלו מהוות בסיס למגוון רחב של יישומים:

- שיחות דיבור בתוכנות למסרים מיידיים (IM) כדוגמת ICQ מסנג'ר Skype, ועוד.
- שיחות טלפון על גבי האינטרנט באמצעות מכשירי טלפון "יעודים".
- שיחות מחשבים אל קווי טלפון "רגילים".
- שיחות בין שני קווי טלפון "רגילים" אשר חברות הטלפון בוחרת להעביר אותן על גבי רשתות IP.

יתרונות עיקריים לטכנולוגיות ה-VoIP הם הפרישה הרחבה של רשת האינטרנט ובכך נחסר הצורך לפרוס

תשתיות מיוחדות לטובות הדיבור (כגון קווי טלפון, מכשירי טלפון וכו').

יתרונות נוספים בשימוש ב-IP-להעברת שיחות, הוא השימוש ב프וטוקול סטנדרטי ונווץ. גופים רבים מבצעים

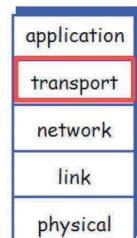
מחקר ופיתוח של כלים שימושיים עבורו: אבטחת מידע, ניהול רשת, ניתוח, אופטימיזציה של הביצועים

וכו', והשימוש ב-VoIP-יכול להתבסס על כל אותו המידע והכלים הקיימים.

משתמש בפרוטוקול – UDP, מכיוון שב-TCP התאוששות מאיבוד packet תיקח זמן רב שיפגע באיכות

השיחה.

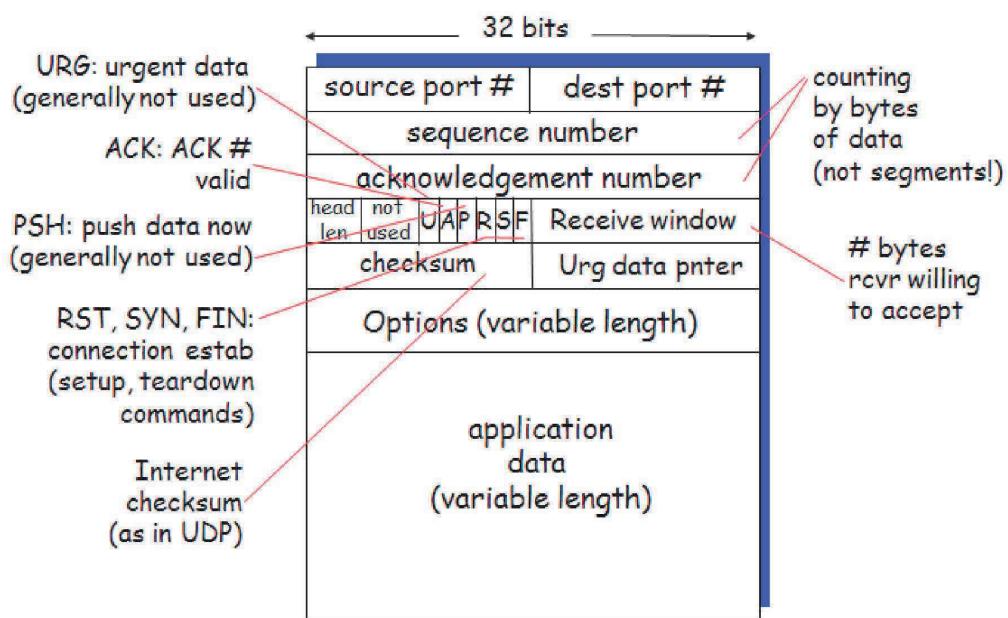
## שכבות התעבורה



### פרוטוקולים:

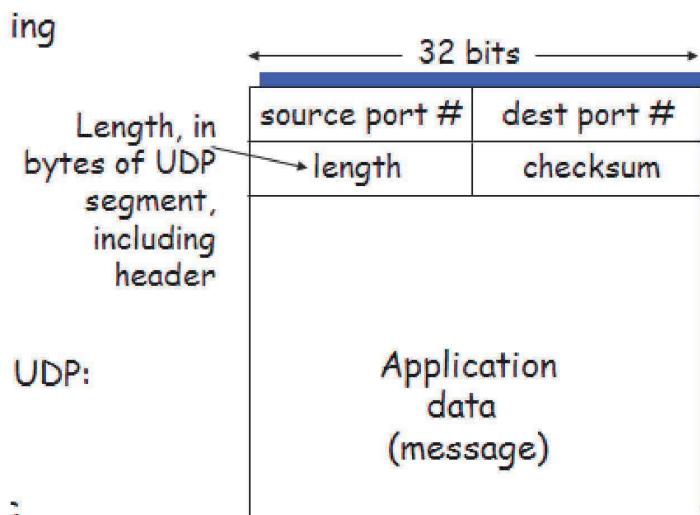
#### - TCP (Transmission Control Protocol)

שירות TCP כולל שירותים מבוססי תקשורת (Connection Oriented) והעברת מידע **אמין**. בשירות מבוססי תקשורת הכוונה היא שיש תעבורת לקווי-שרות לפני שירות האפליקציה מתחילה. לאחר שלב זה אמורים שיש חיבור TCP בין הסוקטים של שני התהילכים. חיבור זה נקרא Handshaking והוא- Full Duplex, כלומר שני התהילכים יכולים לשוחח הודיעות בו זמנית אחד לשני (שירות TCP כולל בתוכו גם מכנים של בקרת עומס, אם הרשות עמוסה אז התהיליך יווט על מנת למנוע עומס נוסף).



### – UDP (User Datagram Protocol)

שירות זה לא כולל הרבה אפשרויות והוא אינו מבוסס תקשורת. לכן גם אין את שלב ה-Handshaking. UDP לא מספק תעבורת מידע אמין והוא אינו כולל בקרת עמס, אך יישום יכול לשלוח הודעות באיזה קצב שהוא רוצה. כמו כן, השירות לא מבטיח delay מיניימלי. מפתחי מערכות זמן אמת (video/audio streaming) יעדיפו את שירות UDP.



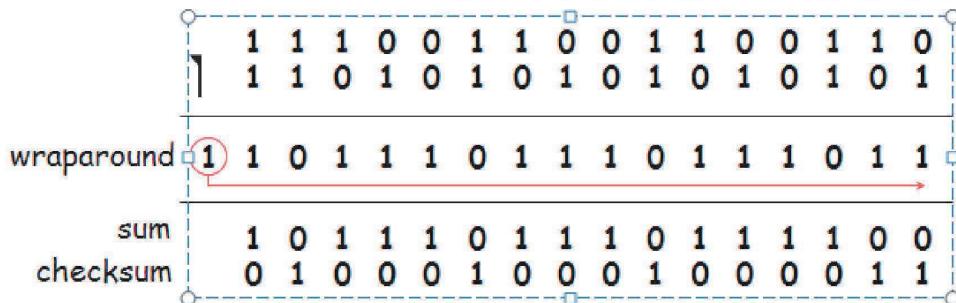
### יתרונות של UDP על TCP:

1. שליטה טובה יותר באשר לאיזה מידע נשלח ומתי: ב-UDP, כൺ-שלחת הودעה היא עוברת ישירות לשכבה הבאה בעוד שב-TCP בഗל מגנון Congestion Control, השילחה עלולה להתעכב ללא ידיעתנו בגלל עומס בחלק מסוים של הרשת.
2. ב-TCP ימשיכו לשלוח עד לקבלת תשובה.
3. אין "מצב חיבור" – על מנת לאפשר באפרים, TCP שומר פרמטרים שונים עבור חיבורים שונים מה שగוזל זמן ומשאבים.
4. UDP – אין שמירת פרמטרים ולכן השרת יכול לתמוך ביותר ב��חות.
5. UDP – Small Packet Header Overhead. TCP מוסיף 20 בתים לכל סגמנט (header). UDP מוסיף 8 בתים לסגמנט. כלומר, TCP יוצר יותר מידע שיש להעביר לעומת UDP.

### **UDP Checksum**

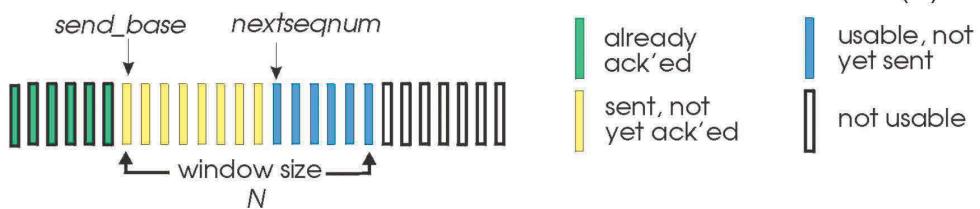
משמש עבור איתור שגיאות (לדוגמה, האם הסגמנט השתנה בזמן התעבורה). **חישוב:** מחברים את כל המיללים באורך 16bit בסגמנט. אם יש carry בחיבור אחרון, הוא מחובר למוצא ומתקבלת מזה מילה באורך 16bit. על מילה זו מפעלים את שיטת המשלים ל-1. אנו צריכים לעשות חישוב זה משומם שפרקתו של UDP לא אמין. UDP רק מזזה את השגיאות אך לא מטפל בהן. בנוסף, יכול להיות מצב שנוצרו שגיאות בסגמנט בזמן ההעברה. לאחר שהיחסנו checksum זה מעודכן מבנה סגמנט UDP.

במחשב המקביל, כל  $4 \times 16$  bit מתווסףם כולל ה-msum. אם לא נמצאו שגיאות אז השדה הזה יהיה רק עם 1-ים. אם אחד הביטים הוא 0 אז נדע שיש טעות.



## Go-Back-N

בפרוטוקול מסוג זה, השולח מורה להעביר כמה חבילות במקביל מבל' לחכות לתגובה אבל מוגבל למס' החבילות ( $N$ ).



משמאל ל-base – חבילה שנשלחה והתקבלה תגובה עבורה.

מימין ל-base – חבילה שנשלחה ועדין ממtingה לתגובה.

מימין ל-nextseqnum – מקום פנוי לחבילה ש摹עמתת לשילוחה.

מימין לשטח  $N$  – מקום שאינו מיועד לשילוח חבילות אלא אם כן ישחרר.

– מס' החבילה הותיקה שעדיין לא התקבלה תגובה עבורה.

nextseqnum – המס' הקטן ביותר הפנוי לשילוח חבילה.

Window size – הוא בעצם חלון בגודל  $N$  שזו לאורך מס' היזיוי ולכן פרוטוקול זה גם נקרא sliding window protocol. המס' הייחודי של החבילה יושב בheader שלה בגודל קבוע.

גודל שדה A ביטים – טווח מספרים אפשריים לחבילה  $[0, 2^k - 1]$

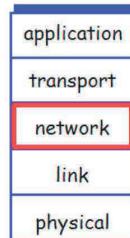
\*TCP הגודל הוא 32 ביט.

הדגמה של הפרוטוקול בעזרת applet:

<http://www.eecis.udel.edu/~amer/450/TransportApplets/GBN/GBNindex.html>

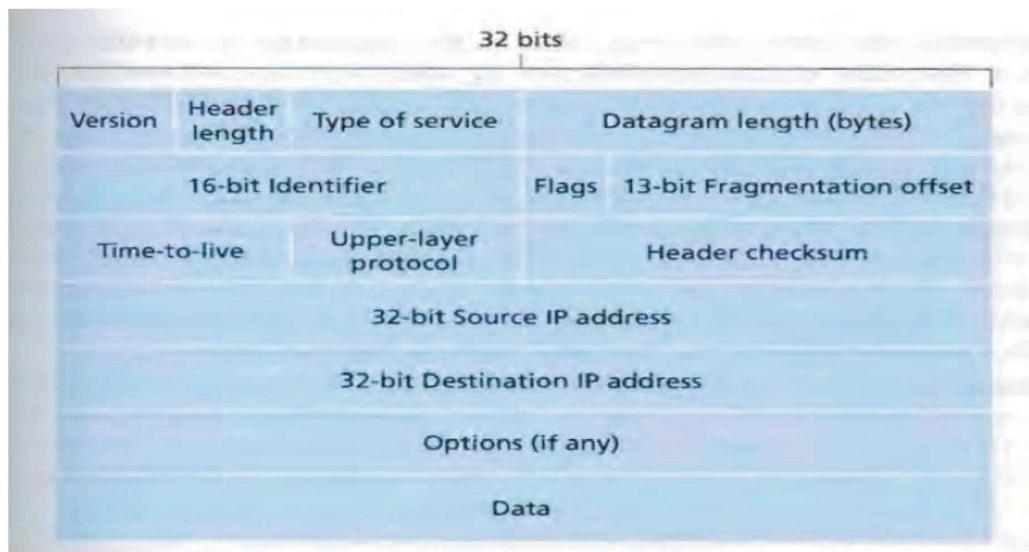
בפרוטוקול זה קיימן חלון בגודל  $N$ . זהו המספר המקורי של packets שמורשים להישלח לפני שהתקבל ACK. כל packet מקבל ACK בנפרד. ברגע שהתקבל ACK לפאקט הראשון בחלון, החלון מתקדם (מכאן השם "sliding windows", בואפן זה נשמר הסדר של השיליחה. במקרה של נפילת packet מסוים, נמתין עד ל-timeout וizzleתו כל הפאקטים שבחלון. אם לאחר נפילת פאקט עדיין יש מקום בחלון עדיין ניתן לשלח, אבל מאחר והפאקטים הקודמים לא קיבלו ACK הוא לא יתקבל והחלון ישלח מחדש.

## שכבה הרשת



## פרוטוקולים: IP Internet Protocol

IPV4



הסבר על השדות:

1. **Version** – 4 ביטים שמצינים את הגירסה של הפאקט. באמצעות זההו הגירסה הרואוטר יכול לקבוע איך לפרש את מה שנשאר מהכתובת. גירסאות שונות של IP משתמשות בפורטים שונים.
2. **Header (4 ביטים)** – קובע איפה פאקט יושב המידע. לרוב פאקט ה-IP אין את שדה 10, כזכור גודל header בפאקט הוא **20 בתים**.
3. **TOS** - 3 ביטים שנכללו על מנת לעשות הפרדה בין סוגי IP, רמת השירות נקבעת ע"י מדיניות הרואוטר שקובע האדמיניסטרטור.

- .data+header – סך גודל ה-IP .4
  5. 16 bit identifier, flags, 13-bit fragmentation offset – שלושת השדות הללו אחראים על פירוק ה-IP.
  6. TTL – מבטיח שהפaket לא יכנס ללופ' בראשת השדה קטן ב-1 עבור כל זמן שהפaket ברואוטר, אם מגיע ל-0, הפaket מופל.
  7. Upper layer protocol – משתמשים בשדה זה רק כאשר הפaket מגיע ליעדו הסופי. הערך של שדה זה מציין באיזה פרוטוקול תעבורת הפaket צריכה לעבור. ערך 6 –TCP, ערך 17 – UDP.
  8. Header checksum – עוזר לרואוטר להבחן בטעות בביטים בפaket ה-IP. מחושב ע"י שימוש בכל 2 בתים מה-header וחיבור שלהם בשיטת המשלים ל-1.
  9. 32 bit source ip address/ 32 bit destination ip address – כתובות ה-IP של המקור והיעד (מודפסות ע"י המissor). לחוב נעשה ע"י DNS.
  10. Options – מאפשר(header) להוות מרחיב.
  11. Data – לרבות מכלל סגמנט של שכבת התעבורה, במקרים מסוימים מכיל גם סוג מידע אחרים (כמו ICMP בהמשך). אם שדה זה מכיל סגמנט TCP של שכבת התעבורה, גודל header יגיע ל-40 בתים.
- כל תא בכתובת בגודל byte אחד

	Start Address	End Address	Max of Networks	Max of Hosts
Class A:	10.0.0.0	126.255.255.255	$128 = 2^7$	$16\ 777\ 215 = 2^{24} - 2$
Class B:	128.0.0.0	191.255.255.255	$16\ 284 = 2^{14}$	$65\ 534 = 2^{16} - 2$
Class C:	192.0.0.0	223.255.255.255	$2\ 097\ 152 = 2^{21}$	$254 = 2^8 - 2$
Class D:	224.0.0.0	239.255.255.255		

## Routing in the internet

### הקדמה:

ניתן לייצג את האינטרנט כגרף כאשר נתב הוא צומת, וקשר תהיה חיבור בין נתבים, כאשר כל קשר תהיה עם משקל וכאשר משקל נקבע לפי עומס הקו (רוחב פס, מידע שעובר בו וכו').

## Routing Protocols

באופן עיקרי כל host מחובר לרואוטר בירתת מחדל (first hop router). בכל פעם ש-host שולח פaket הוא עובר לרואוטר בירתת המחדל - source router (רואוטר בירתת המחדל של היעד נקרא .(destination router).

אלגוריתמי מסלול ימצאו את המסלול הטוב יותר בו פאקט יעבור מה-host ל-host היעד, בעל העלות הנמוכה ביותר. לאלגוריתמים המוכרים והפשוטים נוסף סיבור במקורה שארגון לא רצה לאפשר שפתקים מארגן אחר יעביר דרך ראותר מסוים של הארגון הנ"ל. ניעזר בגרף כאשר הצמתים הם הראותרים והקשתות הם הLINKים. על כל קשת ישנו גם מספר שמצין עלות שנמדדת באורך ה-link, במהירות ה-link והוא מסומן  $(y, x)$ . הגרף הוא לא מכוון, וצמתים לא מחוברים מקבלים ערך  $\infty = (x, y)$ . בראצוננו למצוא את המסלול ה"זול" ביותר מקור לעד. במצב בו כל התלוויות זהות, המסלול שיקבע הוא גם הקצר ביותר.

### סיווג routing algorithms

1. **Global routing algorithm** – מחשב את המסלול ה"זול" ביותר בין המקור לעד ע"י היכרות של כל הרשת. כמובן, האלגוריתם מקבל כקלט את כל הצמתים שקיימים ברשת ומחשב מסלול. החישוב יכול להיעשות בכמה אטרים או באתר אחד. נקרא גם אלגוריתם **Link State (LS)**.
2. **Decentralized routing algorithm** – החישוב נעשה בדרך איטרטיבית. לצמתים אין מידע על ה-links שמחברים ביניהם, עבור כל ראותר מחושב המסלול באופן איטרטיבי. נקרא גם אלגוריתם **Distance Vector (DV)**.
3. a. **Static routing algorithm** – מסלולים משתנים ממש לאט. לרבות משתנים כתוצאה מהתערבות אדם (עrica ידנית של טבלת הgments).  
b. **Dynamic routing algorithm** – המסלול משתנה בהתאם ל"תנוועה" או שינוי טופולוגי של הראותרים.
4. a. **Load sensitive algorithm** – עלות של link משתנה באופן דינامي ויכולת להשפייע על העומס שיש על link מסוים.  
b. **Load insensitive algorithm** – ה
עלות של link לא משפייע באופן ישיר על העומס.

### אלגוריתם Link State

נשלח link state packet link state לכל המחשבים ברשת ובכך לכל הראותרים יש ראייה שווה על כל הראותרים ברשת. כל צומת יכולה להריץ LS ולהשப את המסלול הקצר ביותר. האלגוריתם הוא בעצם **דיאקסטרה**, מחשב מסלול קצר ביותר מזו שאר הצמתים.

באמצעות אלגוריתם זה נבנית טבלת הforwarding עבור כל ראותר.

הסיבוכיות של אלגוריתם זה היא מס' הראותרים בריבוע ( $O(n^2)$ ). ע"י שימוש בערימה, ניתן להציג לסיבוכיות של  $O(\log n)$ .

משמעותה של האלגוריתם מעתנה מאיתרציה לאיתרציה ובגלל חישובי העליות (תלו בתעבורה) ישן הרבה תנודות באlgorigthm.

דרכים להגברת:

1. לא להתייחס לעליות כתליות בתעבורה – לא בא בחשבון משום שאחת המטרות היא להימנע מעומס.

2. להבטיח שלא כל הראותרים מרכיבים LS באותו זמן, ניתן לזמן את תחילת הרצת האלגוריתם.

## אלגוריתם Distance Vector

עבור כל שני צמתים  $x, y$  נקבע

$$d_x(y) = \min_v \{c(x, y) + dv(y)\}$$

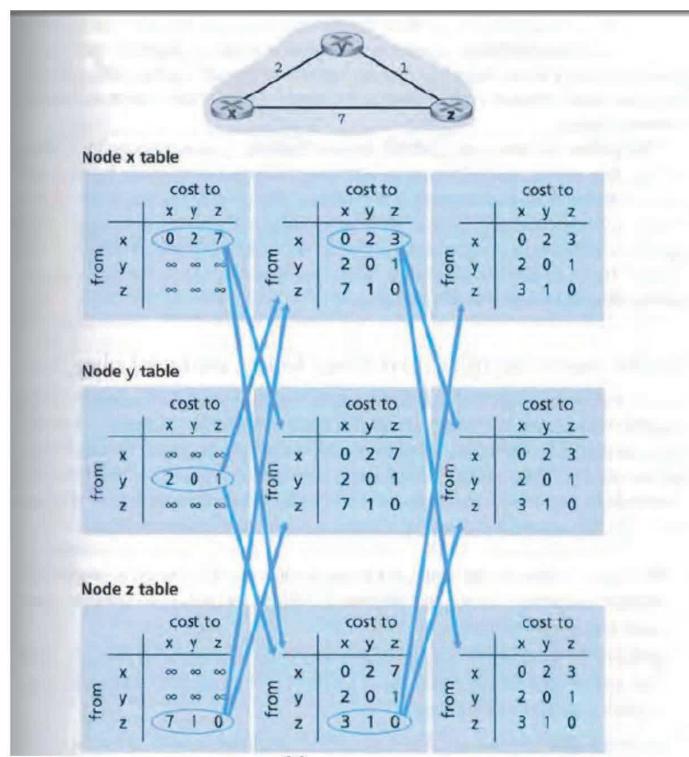
$\min_v$  – נבדק מול כל השכנים של  $x$ .

אלגוריתם זה הוא אלגוריתם **בלמן-פורד**. אלגוריתם איטרטיבי – ממשיך כל עוד המידע בצתמים משתנה. (כל איטרציה עוברים על כל הצתמים ובודקים אם עדין מתקיים  $(v, u) + c(u, v) > d(v)$ ). האלגוריתם עוצר בלבד ומתחילה שוב אם יש שינוי.

- האלגוריתם הוא א-סינכרוני – קדוקדים לא ציריכם להחליף מידע ביניהם.

כל כמה זמן כל צומת שולח עותק של וקטור המרחק שלו לצמתים שכוברים אליו ישירות.

באלגוריתם זה, צומת  $x$  מעדכן את האלגוריתם על כל שינוי בעליות באמצעות מהקשנותו אליה הוא מקשור.



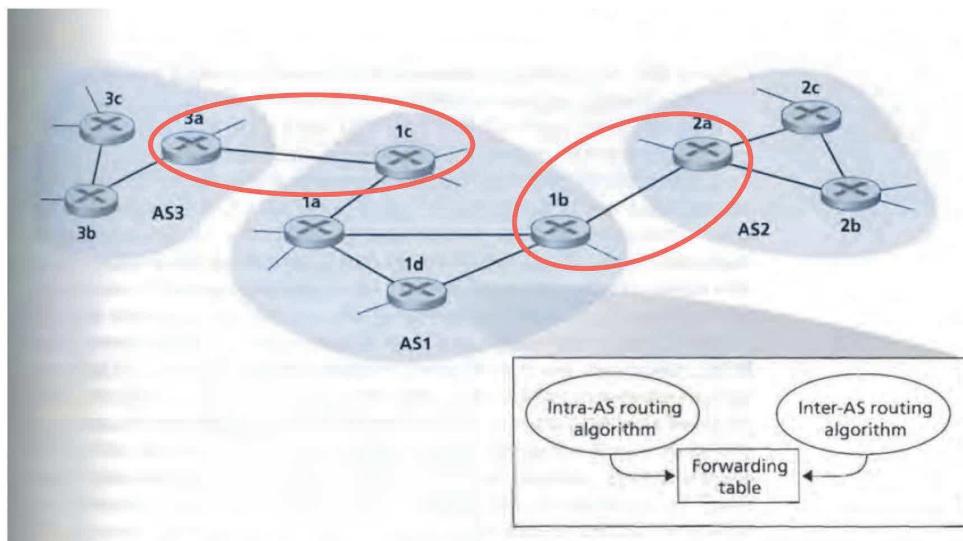
## Hierarchical Routing

הסתמכת על האלגוריתמים שהוצעו כאלגוריתמים שפועלים על ראותרים שכולם זהם היא פשוטה משתי סיבות:

- קונה מידת (scale) – בפועל יש למילה מ-200 מיליון יעדים, לא אמת אפשר לציין את כלם בטבלאות ה-forwarding. האלגוריתם של בלמן-פורד לא יגמר לעולם. צריך לעשות משהו כדי להוריד את הסיבוכיות.
- אוטונומיה ניהולית (administrative autonomy) – כל מנהל רשות רצחה לנוהל את הניתוב ברשות שלו.

2 הסיבות (בעיות) הללו יכולות להיפטר ע"י סידור הראותרים לתוכן Autonomous Systems (AS). כל קבוצת AS מכילה ראותרים שישבים תחת אותו מנהל רשות וכולם מרצים את אותו האלגוריתם ומחזיקים מידע אחד על השני. אלגוריתם שרצ' בתוך AS נקרא **Intra-AS**.

ראותר שנמצא ב-AS, בעל אחריות נוספת. אחראי על שליחת פאקטים מחוץ לרשות הפנימית. **Gateway router**



ניתן לראות שישנם שלושה AS. הקווים העבים יותר מייצגים חיבור ישיר והקווים הדקיקים מייצגים חיבור ל-subnets של מחוברים לראותרים.

ב-AS1 יש ארבעה ראותרים: 1d מרץ intra-AS ו-1c כל אחד מארבעת הראותרים יודע איך להעביר פאקט בדרך היעילה ביותר. ראותרים 3a, 2a, 1c, 1b הם gateway.

אם פאקט צריך לעבור דרך AS<sub>3</sub>, אז הוא עובר דרך gateway. נניח ש-2b קיבל פאקט שהוא שלוי הוא מחוץ ל-AS<sub>2</sub>. 2b יעביר (לפי טבלת ה-forwarding) את הפאקט או ל-2c או ל-2a.

© נערך ונכתב פֵי שמעון אורזיאן, רון רוזנפָּלֶד, גיא פָּלָג

העלות המקסימאלית מוגבלת ל-15 ולכן עבור מערכות אוטונומיות מגבילים ל-15 מעברים.

כל 30 שניות מתבצע עדכון שנייים עבור כל הראותרים באמצעות RIP response message.

הטבלה מכילה עד subnet 25 יעד ב-AS ובנוסף היא מכילה גם את המרחקים מהשולח, לכל אחד מהsubnet. נקרא גם RIP advertisements.

**הערה:** העברת הטבלאות מתבצעת באמצעות פרוטוקול UDP. איך זה יכול לקרות?

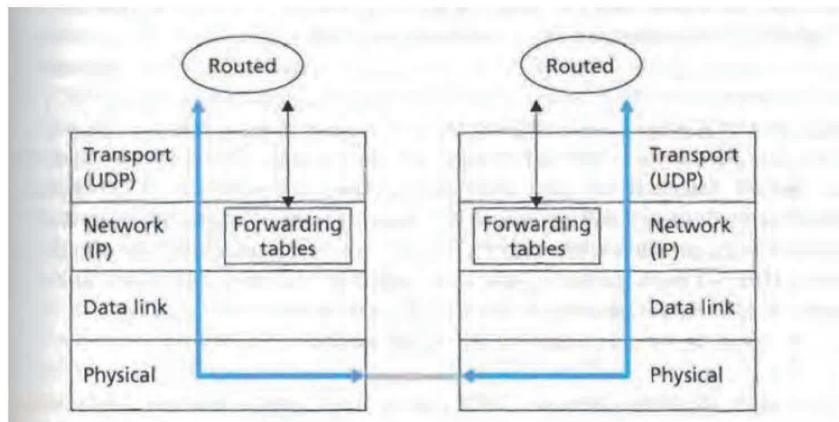
כי אנו עובדים ברמת הרשות והUDP קשור לתעבורה, אז מה שקרה שכל נתב מכיל תוכנה שמשתתפת את הprotokol ולכן בעצם אנו עובדים ברמת האפליקציה של הנתב.

- כל ראותר מחזק routing table שכוללת את ה-DV ואת טבלת ה-forwarding.

בטבלה 3 עמודות: יעד, הראותר הבא במסלול ה照顾, מס' המעברים (hop). נקבל שורה עבור subnet.

עם כל "מודעה" (advertisement) שמקבלת, הטבלאות משתנות. מודעות מגיעות בערך כל 30 שניות. אם במשך 180 שניות לא מגיעה הודעה מראותר שכן, הוא עבר להיות במצב unreachable או שהשך מת או שהחיבור נפל.

במקרה זה RIP משנה את הטבלה ומיפוי את המידע לשאר השכנים ע"י שליחת מודעה. המודעות מועברות באמצעות UDP בפורט מס' 520.

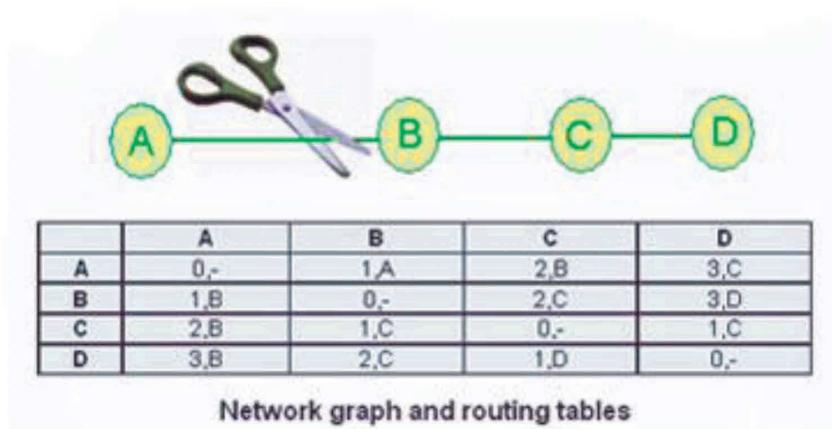


הشرطוט מראה איך RIP מיושם במערכת אוין. תהליך בשם routed מבצע RIP, תחזקה והחלפת מידע עם תהליכי Routed של השכנים. משום ש-RIP מיושם כתהליכי של שכבת האפליקציה, הוא יכול לקבל ולשלוח packets ולהשתמש בפרוטוקול תעבורה סטנדרטי.

### בעיית ה"ספירה לאינסוף" באלגוריתמים מבוסטי DV "count to infinity"

אחת הביעות המשמעויות ביותר באלגוריתמים מבוסטי DV היא בעיית הספירה לאינסוף. נדגים את הבעיה על ידי הדוגמה הבאה הלוקחה מהאתר : <http://computer.howstuffworks.com/routing-algorithm4.htm>

נתבונן ברשת שהגרף שלה נראה כך:



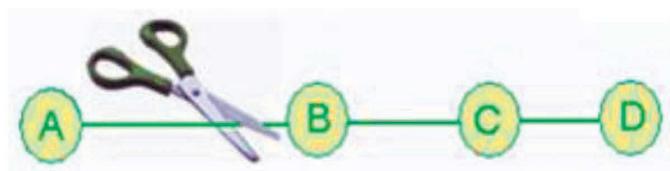
ברשת זו יש חיבור יחיד בין הצמתים A ו-B. כעת נניח כי ברגע מסוים נקטע החיבור בין צומת A ו-B. צומת B מבצעת תיקונים לטבלה שלו, לאחר זמן מה, הרוטרים משדרים אחד לשני את הטבלאות שלהם ולכן צומת B מקבלת את הטבלה של צומת C. מאוחר וצומת C לא מודעת לכך שהחיבור בין A ו-B נקטע, בטבלה שלו יש לינק ל-A עם המשקל 2 (המסלול שהוא דרך B). מכיוון שב-B כבר אין דרך ל-A (марחיק אינסוף) היא מקבלת את המרחק החדש שקיבלת מ-C מוסף לו 1 וכן בשורה של צומת A יהיה כעת המרחק 3. בהחלפת הטבלאות הבאה הדבר מתרחש שוב, כעת צומת C מקבלת את הטבלה מ-B בה יש דרך לצומת A במשקל 3 ומכיון שהוא המסלול הקצר ביותר עבורה, היא חושבת שהיא יכולה להגיע לצומת A באמצעות קפיצות. כך נוצרת למעשה שbullet כל החלטת טבלאות המרחק לצומת A, שbullet כבר מונתקת מהרשות גדל ב-2 כל פעם, עד שmagiu ל-15 (עבורנו זהה האינסוף ומכאן שם הבעיה).

הטבלה הבאה מציגה את הקפיצות ההולכות וועלות עד לאינסוף.

	B	C	D
Sum of weight to A after link cut	$\infty, A$	2,B	3,C
Sum of weight to B after 1st updating	3,C	2,B	3,C
Sum of weight to A after 2nd updating	3,C	4,B	3,C
Sum of weight to A after 3rd updating	5,C	4,B	5,C
Sum of weight to A after 4th updating	5,C	6,B	5,C
Sum of weight to A after 5th updating	7,C	6,B	7,C
Sum of weight to A after nth updating	...	...	...
$\infty$	$\infty$	$\infty$	$\infty$

The "count to infinity" problem

פתרון אפשרי לבעה זו: יהיה לא אפשר לצלתים לשלו מידע, לצטמת שהיא המסלול היחיד לאוינו הידע. במקרה שלנו לא אפשר שצטמת C תשלח לצטמת B מידע על צטמת A מכיוון ש-B היא המסלול היחיד בין C ל-A.



## OSPF (Open Shortest Path First)

ממש פרוטוקול : link state

זהו פרוטוקול של Intra-AS. הוא למעשה המשך של RIP והוא בעל אפשרות מתקדמתה. הפרוטוקול הוא מסווג link state שימוש בדיםטריה.

עם OSPF, הראטור בונה מפה טופולוגית מלאה של כל המערכת. הראטור מרים את דיאקסטרא מקומית וקובע את המסלולים הזולים ביותר לכל subnets. מנהל הרשות יכול לקבוע אם הוא מעדיף להרים לפי עלות נמוכה או מספר מעברים (hop) נמוך (ע"י קביעת כל המשקלים על הקששות ל-1).

ב-OSPF כמו ב-RIP המידע מופץ לכל הראותרים. הראטור שולח עדכנים כל 30 שניות גם אם לא באמת יש. ה"מודעות" בפרוטוקול זה מועברות על גבי IP, כלומר על הפרוטוקול לישם את תכונות פרוטוקול ה-IP. בנוסף הפרוטוקול בודק שקוויים פעילים ומאפשרים לראטור OSPF לאגור מידע ולהפיצו.

### תכונות OSPF:

- security – הخلافות בין ראותרים של OSPF יכולות להיות אמינות, רק ראותרים אמינים יהיו בפרוטוקול.
- multiple same cost path – כאשר לכמה מסלולים יש אותה עלות, מתאפשר ע"י פרוטוקול זה לחלק תנועה בין שניים.
- multicast OSPF – integrated unicast & multicast support (шиידור) מספקת אפשרות פשוטה ל-OSPF, מօסיף סוג חדש של מידע LS.
- support for hierarchy within a single routing domain – היתרונו הכי משמעותי, יכולת להרכיב מערכת אוטונומית בצורה היררכית.

.OSPF מקונגף באופן היררכי לאיזוריים. כל איזור מריצ' LS עבר כל הראותרים שלו. בכל איזור יש ראותר אחד או יותר שאחראים על העברתפקטים החוצה (borders).

איזור אחד מוקצה להיות backbone – מנותב את התנועה בין האיזוריים השונים ב-AS. תמיד מכיל את כל borders של איזור.

בהערכה: תחילת הפקט מנותב מאייזור המקור ל-border וatz מנותב דרך backbone לראותר border באיזור היעד ומשם מנותב לעד.

### BGP (Border Gateway Protocol)

משתמש ב: TCP

פרוטוקול של AS-inter, העברתפקטים בין AS שונים. פרוטוקול זה מספק ל-AS אמצעים ל:

1. קבלת מידע על נגישות ל-subnet מ-AS שכנים.
2. הפצת המידע לכל הראותרים הפנימיים ב-AS.
3. קביעת מסלולים "טוביים" ל-zones בהתבסס על המידע הנ"ל.
4. אפשר לכל subnet להודיע על קיומו לשאר הרשת. מעביר את ההודעה לכלום!

### עקרונות BGP:

בפרוטוקול זה, זוגות של ראותרים מחליפים מידע באמצעות שימוש בחיבורו TCP דרך פורט 179. הזוגות נקראים BGP Peers ולחיבורו TCP Session. חיבור שמקשר בין ASs נקרא IBGP (Internal BGP). נקרא eBGP (External BGP) חיבור שמקשר בין ראותרים ב-AS נקרא IBGP (Internal BGP). פרוטוקול זה מאפשר לכל AS לדעת אילו יעדים נגישים דרך השכנים שלו. BGP עובד על כתובות IP.

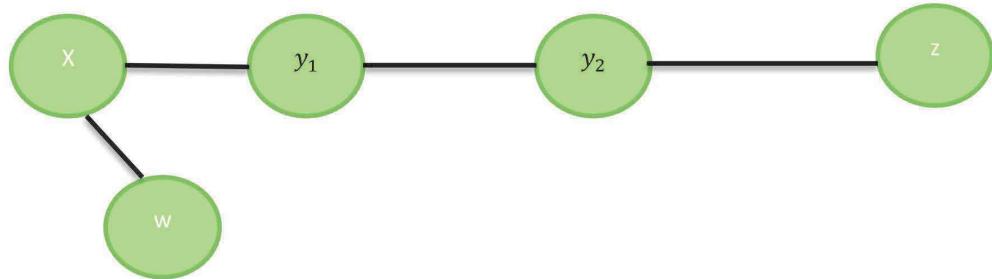
בין כל 2 AS יעבירו פרפייקסים (prefix) על גבי eBGP. כל AS יעביר מידע פרפייקטי לכל הראותרים שבשליטתו על גבי eBGP.

### פרוטוקול זה משתמש באربע פקודות:

OPEN - ברגע שמתחברת צומת חדשה איזה היא משתמש בפקודת `OPEN` היא פותחת link לצמתים שלידה. דרך link זה היא מקבלת את טבלאות הננתונים.

Keep-alive – פקודה זו שולחת את הננתונים (טבלאות) של הצומת לצמתים השכנים. Update – ברגע שימושו השתנה בטבלה של נתב הוא מבצע פקודת `update` ומודיע לכלם על השינוי. ככלומר שולח את הטבלה שלו לכל השכנים.

Notification - אם אחת משלשות הפודות קרתה או אם קرتה טעות כלשהי, או שנסגר חיבור עם צומת כלשהו.



X: מכיל לפני ש- W הцентр Y, z, y<sub>1</sub>, y<sub>2</sub>, z: X

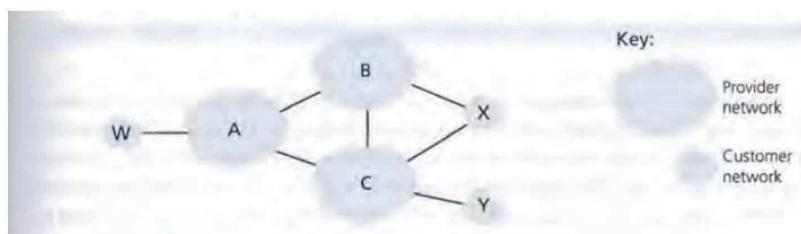
W: עושה open. ל-X נוספה שורה שזה המסלול ל-W.

X: עושה update ושלח ל-W את ההשמה שלו כך שההשמה של W תהיה

W: W, X, y<sub>1</sub>, y<sub>2</sub>, z

אצל X משוחה השתנה בכך הוא עושה update.

### :Routing Policy



בציור ישנן 6 מערכות אוטונומיות מחוברות. נשים לב כי Y, X, W הם ASs ולא רואטרים. ל-X יש 2 ספקי אינטרנט, נקרא **multi homed stub network**. נרצה למנוע מ-X מלהעביר הודעות בין B ל-C. נעשה זאת ע"י שליטה על התמסורות ש-BGP מבצע.

גם אם X יודע על מסלול YCX שmagiu ל-Z, הוא לא יעביר (יפרסם) ל-B אחריו - לא יודע ש-X יכול להגיע ל-Z. הוא לעומת זאת יעביר הודעות ל-Y דרך X.

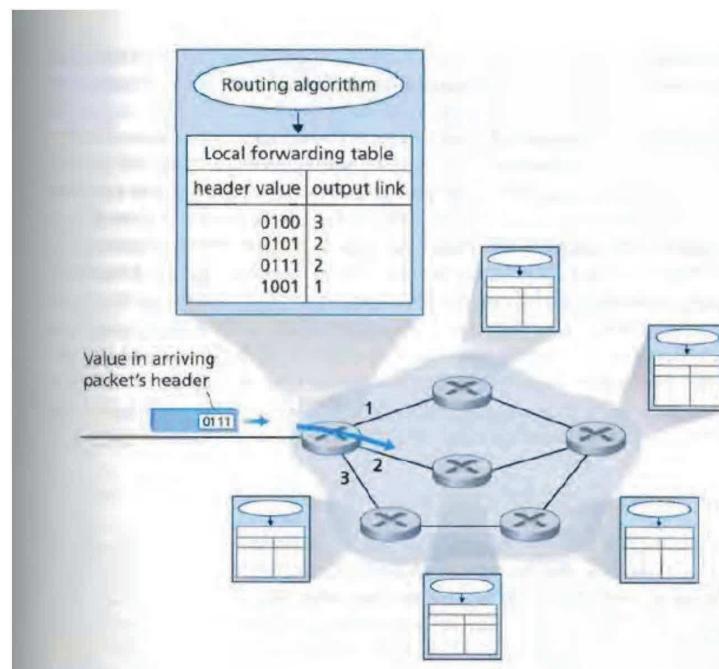
## Forwarding and Routing

התפקיד של שכבה הרכשת הוא פשוט להעביר פאקטים מחשב שלוח למქבל, על מנת לעשות זאת יש צורך בשני מאפיינים פונקציונליים של שכבה זו:

1. Forwarding – כשמתקבל פאקט בקשר הקולט, על הרואוטר להפנותו לקישור הפלט המתאים (כדי שייעבור מידע בין H2H ל-H1).
2. Routing - בחירת מסלול הפאקטים בין המקוור ליעוד.

### Algorithm – Routing algorithms

לכל רואוטר יש טבלת forwarding. רואוטר מעביר פאקט לפי בדיקת הערך (כתובת היעד של הפאקט או אינדיקציה על החיבור אליו הפאקט שייר), יושב בסכבי header ובאמצעות ערך זה מוכנס אינדקס לטבלה. תוצאה הטבלה נותנת אינדיקציה לבחירה של ה-interface אליהם יש לשלוח את החבילות.



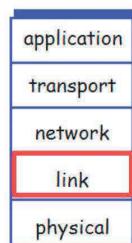
לפי הדוגמא הנ"ל, פאקט עם הערך 0111 מגיע לרואוטר, הרואוטר נכנס לטבלה וקובע שהה-interface 2 הוא מטרתו. לאחר מכן הרואוטר מעביר את הפאקט ל-interface 2.

אלגוריתם routing קובע את הערכים המוכנסים לטבלת forwarding של הראوتر.

### Connection Setup

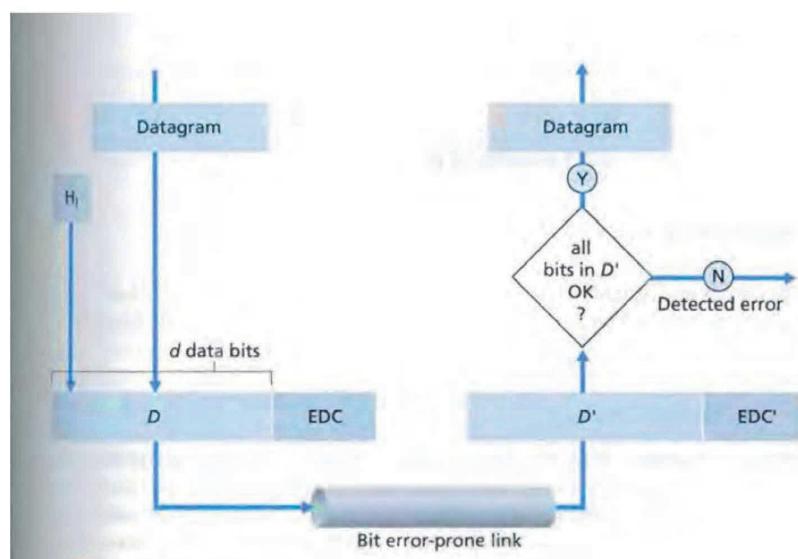
ישנו מאפיין פונקציונלי נוסף, **connection setup**. ישן כמה ארכיטקטורות של שכבת רשת (ATM, Frame-relay, non-internet, …) שבנוסף לבחירה המסלול בין המקור ליעד, דורשות גם שהראוטרים יבצעו "לחיצת ידיים" ביניהם.

## שכבה התקשורת



## Error Detection and Correction Techniques

### שיטות לתקן טעויות:



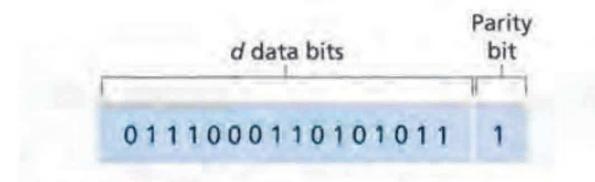
מוסיפים למידע ביטים בסוף המעתפה – קוראים לזה EDC כמהת ביטים זו משתנה מושיטה לשיטה

### ישן 3 טכניקות לישום EDC:

#### Parity Checks

##### :Single parity Check

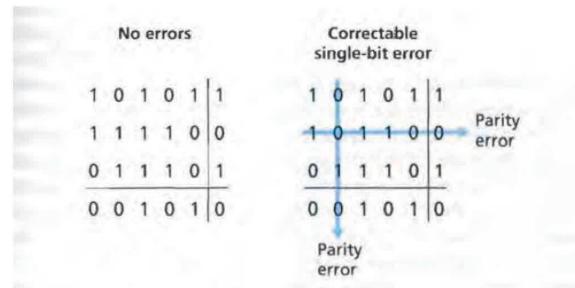
מוספים ביט 1 (גודל של EDC הוא 1) כך שסיבית זו תשלים את כמות האחדים במעטפה הזוגי. שיטה זו מבטיחה שאם תהיה טעות אחת במידע נכון לא יוכל לזהות. בכלליות יוכל לזהות שהייתה טעות באורך اي זוגי אך מכיוון שלא יוכל לזהות טעות באורך 2 נאמר שיכולה להיות היא של טעות אחת.



אם ישים ערך אמין אז שיטה זו טובה.

##### :Two Dimension Bit parity

נחלק את המידע לבLOCKים, לכל BLOCK נוסיף סיבית זוגיות. לאחר שחישבנו את סיבית הזוגיות, נחשב באופן אנכי את סיבית הזוגיות של הטוור. ובנוסף נחשב סיבית זוגיות לכל סיביות הזוגיות. לדוגמה:



אם יש טעות אחת ספציפית ניתן לתקן, בנוסף ניתן לגלוות 2 שגיאות.

היכולת של הצד המקלט של זיהוי ותיקון נקראת FEC (Forward Error Correction) וועזרת להפחית את מס' השילוחות הדרושות ומאפשרת תיקון מיידי.

© נערך ונכתב פֵי שמעון אורזיאן, רון רוזנפָּלֶד, גיא פָּלָג

## Multiple Access Protocols

יש שני סוגים של לינקים:

– שלוח יחיד ומתקבל יחיד בכל קצה לינק (נקרא גם PPP).

– על קו תרבותה יש כמה שלוחים ומקבלים (לדוגמה LAN, Ethernet).

בסוג זה עולה בעיתת הגישה המרובה (multiple access problem), למי מותר לשולח וממי. כאשר יש יותר מצמת אחת ששולחת הודעה כל השאר מקבלים יותר מהודעה אחת ונוצרת התנגשות וההודעות יכולות "להתערבב", וכתוצאה לכך כל הפריים שימושיים בתגובה עלולים להיאבד.

לצורך כך קיימים multiple access protocol שאחראי להשתלט על שליחת ההודעות. קיימים הרבה פרוטוקולים כאלה וניתן חלק אותם לשולשה (במהרשך).

קיימת דרישה מהפרוטוקולים שכאשר יש שלוח אחד, התפקיד תהיה R ועבור M שלוחים M/R במשותף.

## PPP: The Point to Point Protocol

لينك שמחבר ישירות שני צמתים, אחד בכל קצה.

דרישות מ-PPP:

1. **Packet framing** – על השולח להיות בעל יכולת להעביר פאקטים המסמכה (encapsulation)

2. **Transparency** – אסור ל-PPP להגביל את הופעת מידע בפאקט שכבת הרשת.

3. **Multiple network layer protocol** – צריך לתמוך בכמה פרוטוקולים שונים של שכבת הרשת.

4. **Multiple types of links** – צריך לספק טיפוסי לינק שונים להעברות סדרתיות/מקביליות.

5. **Error detection** – חייב לשימוש באם יש בעיה במידע שהועבר (אבל לא חייב לדעת לתקן).

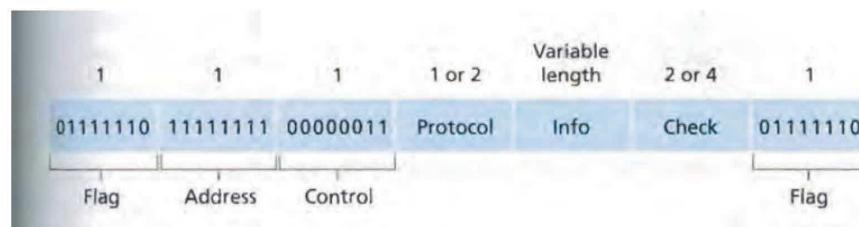
6. **Connection liveness** – חייב לבדוק בನפילת בלינק ובמקרה שיש בעיה להודיע לשכבת הרשת.

7. **Network layer address negotiation** – חייב לספק מנגנון לחיבורו שכבת הרשת, לעדכן/ביבוי רואוטר ועוד.

PPP לא חייב לספק תיקון בעיות, בקרת זרימה וזרימה.

PPP מסכים לקבל לשלוח לא לפי הסדר (Out Of Order) , טיפול בעיות מסווג זה יעשה על ידי שכבת הרשת.

## PPP Data Framing



\* **flag** - כל פריים מתחילה ונגמר על Flag שערך 01111110

\* **address** – הערך האפשרי היחיד 11111111

\* **control** - הערך האפשרי היחיד 00000011

\* **info** – מכיל את הפאקט המוכמס שנשלח ע"י השכבהعلילונה. ערך בירית המחדל המקסימאלי הוא 1500 בתים.

\* **checksum** – על מנת להבחן בטיעות בביטים.

**יתרונות:**

1. חיבור פיזי ממשי גורם לכך שאין הפרעות בשידור.
2. ניצול רוחב הppo.

**חרוגונות:**

1. על מנת לחבר בין host-hostים צריך להשתמש בהרבה חוטים.

## Byte Stuffing 2

עלולה להיווצר בעיה בכך שכשניים 2 בתים על מנת לקבוע את התחלת וסיום הפריים, משומש שיכול להיווצר מצב בו דגל ההתחלה שערך 01111110, יופיע גם כ-byte בשדה ה-data ואז המקלט יכול "להתבלבל". דרך אחת לפטור זאת זה פשוט לאסור על שליחת מידע שמכיל את ערכי הדגלים (אבל זה נוגד אתדרישות ה-PPP). דרך נוספת נקראת **byte stuffing**, PPP מגדרה byte בשם **control escape** עם הערך 01111101. אם הדגלים מופיעים כערך בשדות ה-data מוסיפים את הערך הנ"ל **לפני** ערך הדגל כדי לסייע למקלט לזהות לא דגל אלא חלק מה-data. אם הערך הנ"ל מופיע גם הוא כחלק מה-data הוא יגיע למקלט פעמיים (פעם אחת כdata ולאחריו כescape).

## Local Area Connection

החיבור יעשה על ידי קצר כר שבין כל הhost-ים יהיה חיבור אחד. רשות צזו מתאפסיות ברוחב פס גבוה יחסית ובזמן שהיה נמוך יחסית.

הבעיות:

לא יודעים מי משדר, כל הזמן צריך לעשות סינכרון בין host-ים.  
כאשר מישחו מדבר ושולח הודעות כולם מקשייבים כלומר מקבלים את ההודעות.

יעילות של LAN:

$T_{prop}$  = delay המkosימלי בין 2 צמתים.

$T_{transfer}$  = הזמן שלוקח למעטפה מקסימלית לעبور.

$$efficiency = \frac{1}{1 + 5 \cdot \left( \frac{t_{prop}}{t_{transfer}} \right)}$$

## Random Access Protocols

בפרוטוקולים מסווג random השילחה לא מוגבלת כל עוד אין התנגשות, ברגע שמצויה התנגשות, השולחים שנמצאים בהתנגשות שולחים שוב ושוב את הפריים עד שמבצעת העברת תקינה. השילחה החזרת לא בהכרח נעשית באופן מיידי, השולח ממתיין "זמן עיקוב רנדומלי" שנבחר על ידו.

בדפים הבאים מוצגים שני פרוטוקולים:

\* – נניח את הדברים הבאים:

1. כל הפריים בגודל L ביטים בדיק.
2. R קצב מסויים לשידור.
3. הזמן מחולק לסלוטים של R/L שניות.
4. צמתים מתחילה לשלוח רק בתחילת הסלוט.
5. הצמתים מסונכרים וכל צמת יודעת מתי סלוט מתחילה.
6. אם יש התנגשות, כל הצמתים מודעים לכך לפני סוף הסלוט.
7. P – הסתברות, מס' בין 0 ל-1.

פעולות ה-ALOHA:

1. כשלצומת יש פריים חדש לשילוח הוא ממתיין עד לתחילת הסלוט הבא ואז מעביר את הפריים.
2. אם אין התנגשות הפריים עבר בהצלחה ואין צורך בשילחה מחדש מוחודשת של פריים.

3. אם יש התנגדות, הפריים נשלח שוב ועד סוף הסלוט כל הצמתים מודעים לכך. בהסתברות  $P$  ההודעה נשלחת שוב ושוב עד שתעבור בהצלחה ולא התנגדויות.

יתרונות: אפשר לכל הצמתים להעביר פרימיום בקצב המלא  $R$  (במקורה והצומת היא היחיד ששולחת כרגע).

אם 2 תחנות מסוימות מתחילהות לשדר באותו זמן יכולר יש לנו התנגדות אזי כל תחנה ממחכה  $\frac{L}{R}$  זמן (סלוט אחד) ואז מחליטה לפי הסתברות  $P$  האם לשדר את ההודעה.

**יעילות:**

אם יש  $n$  צמתים כל צומת תשדר בהצלחה ללא התנגדות בהסתברות הבאה:  
קודם כל, הסתברות  $P$  להצלחה שלה. וscal ה –  $1 - P$  הצמתים האחרים לא ישדרו. ככל הסתברות  
שצומת תשדר ללא התנגדות היא:

$$.P(1 - P)^{n-1}$$

$P$ =הסתברות שהשליחה תצליח.

$$(1 - P)^{n-1} = \text{הסתברות ששאר הצמתים לא ישדרו.}$$

היעילות הכללית של המערכת היא הסתברות שמצונו, כפול מספר הצמתים لكن:

$$n \cdot (P(1 - P)^{n-1})$$

נרצה למצוא את ערך המשווה כאשר עבר מספר גדול מאוד של צמתים:

$$\lim_{n \rightarrow \infty} (P(1 - P)^{n-1}) \cdot n = \frac{1}{e} = 0.37$$

קיבלו Ci אחד הנצילות הוא 37%.

**-pure ALOHA**

באלגוריתם זה נרצה לוותר על השימוש בסלוטים של הזמן, כדי לא לדאוג שהצמתים שיופיעו מ遜כנים זה עם זה. הרעיון שכל יחידה תשדר מתי שהיא רוצה, אם תהיה התנגדות אזי בהסתברות  $P$  הצומת תחליט מתי לשדר שוב.

### יעילות:

אם צומת רצה לשדר בהסתברות  $P$  להצלחה שלה. וscal ה – 1- $n$  הצמתים האחרים לא ישדרו. כלומר הסתברות שצומת תשדר ללא התנגשות היא

$$P(1 - P)^{n-1}$$

$P$ =הסתברות שהשליחה תצליח

$$(1 - P)^{n-1} = \text{הסתברות ששאר הצמתים לא ישדרו}$$

נשים לב שאין לנו חפיפה בזמן כיוון שבאלגוריתם זה אין לנו סלוטים, אך לא ניתן לדעת אם בזמן זה יהיו צמתים אחרים שגם ישלחו. לכן בזמן זה שגם משדרים ההסתברות שאחרים לא ישדרו בזמן זה היא גם  $(1 - P)^{n-1}$ . מכיוון שאין חלקה לסלוטים ציריך שף אחד אחר לא התחיל לשדר הוועדה לפני שההועודה שלנו התחליה (כאמור  $(1 - P)^{n-1}$ ) וגם שף הועודה לא התחליה אחרי שההועודה שלנו התחליה (אותה הסתברות  $(1 - P)^{n-1}$ ). לסיום היעילות הכללית של המערכת היא:

$$P(1 - P)^{n-1}(1 - P)^{n-1} \cdot n$$

עבור כל הצמתים :

$$\lim_{n \rightarrow \infty} P(1 - P)^{n-1}(1 - P)^{n-1} \cdot n = \frac{1}{2e} = 0.18$$

אחוז הנצלות הוא 18%.

### CSMA (Carrier Sense Multiple Access)

2 עקרונות למניעת התנגשות:

1. **Carrier sensing** – כמו אצל אנשים, יש להקשיב אז לדבר, צומת "מקשייה" לערוץ לפני השילחה. אם צומת מבחינה שיש פריים שמו עבר כרגע בערוץ, היא תמתין עם השילחה (זמן רנדומלי) ובסיומו שוב "תקשייב".

2. **Collision detection** – אם שני אנשים מתחלים לדבר ביחד, אחד מפסיק. צומת שולחת מקשייה לערוץ בזמן השילחה. אם הצומת מבחינה שצומת אחרית מעבירה פריים, השילחה מופסקת ונקבע זמן לניסיון של שילחה חוזרת.

CSMA מבוסס על שני עקרונות אלה (למעשה CSMA וgil לא מכיל את אופציית Collision Detection אבל יצא לו שדרוג שנקרא CSMA/CD, אשר תומך גם בעיקון השני).

זה עשוי להיות בעייתי כאשר יש זמן השהייה בין תשובות בין תחנות קרובות לרוחקות, כלומר התמונה הרוחקה תחשב שף אחד לא משדר למרחוק שתחנה אחרת שרחוקה ממנה כבר משדרת. (מקרה להטנגשות).

## Taking Turns Protocols

לכל צומת לשלוח מס' מקסימאלי קבוע של פריטים. השליפה מתבצעת באופן ציקלי.

### חסרונות:

1. Billing delay – הזמן שלוקח להודיע לצומת שהיא רשאית לשלוח פריטים.
2. אם צומת המאסטר נופלת אז כל התהילך נפגע (חמור).

– מגדרים פריטים מסוימים כטוקן (token) שמוחלף בין הצומטים בסדר קבוע.

צומת 1 תמיד יעביר טוקן ל-2... צומת ח תעביר ל-1. כאשר צומת מקבל טוקן, היא מחזיקה בו אם יש לה מה לשלוח, לאחר מכן עובר לצומת הבא. אם אין יש מה לשלוח, נשלח המס' המקסימאלי האפשרי של פריטים ואז הטוקן עובר לצומת הבא.

## Mac Address

מורכב מ-48 ביט ובשונה מ-IP אין לו היררכיה لكن לא ניתן לתחזק לו טבלאות. לכל רכיב תקשורת יש mac שלו שנמצא אצל בחומרה. ניתן לשנות ערך זה, אך לאחר ניתוקו מהחומרה הוא יחזור לערך המקורי שהוא לו לפני השינוי. ישנו גוף שאחראי על חלוקת כתובות אלו באופן הבא:

מעבר כל יצירנית תקשורת שמייצרת רכיב תקשורת, הגוף בוחר בשבילה את 24 הביטים הראשונים של ה-MAC. כתובות זו תציג את החברה שאר ה-24 ביט מיועדים לרכיבים שהחברה מייצרת. לרוב ה-MAC נכתב בבסיס hexa.

## :ARP

כל תחנה מתחזקת טבלת ARP, תפקיד טבלה זו היא להתאים בין כתובת IP לכתובת MAC.

IP	MAC	זמן שלוקח להודיע TTL להגיע לחידה
תחנה שולחת מעטפת ARP, שכוללת		

IP sender	MAC sender	IP dest	Mac Dest
אם בטבלת ARP של התחנה, לא קיימת רשומה של ה-IP DEST כלומר התחנה לא מכירה את כתובת mac של היעד. אך ברשומה mac הוא broadcast(FFFFF....F) מכיוון שאנו לא ידעים לאן לשלוח את המעטפה. בדרך זו המעטפה מועברת לכלום: אם היא מגיעה לתחנת קטנה בעלת כתובת mac שונה, אז המעטפה תופל. אם המעטפה מגיעה לעד איז חוזרת מעטפה חדשה מיעד למקור עם כתובת mac של היעד. כל הראוטרים בדרך יפעלו בהתאם.			

נראה דוגמא:

אם A רוצה לשלוח ל- B הודעה ואין לו את כתובת ה MAC של B הוא ישלח הודעה ב-broadcast אם A – כל התחנות:

IP A	MAC A	Ip B	BroadCast = FFFF...F
------	-------	------	----------------------

לאחר שהמעטפה מגיעה ל-B, B ישלח ל A את המעטפה:

IP B	MAC B	Ip A	Mac A
------	-------	------	-------

A ו- B יעדכו את טבלאות ARP שלהם כאשר הם מקבלים את המעתפות.

אם A רוצה לשלוח ל C הודעה אבל C לא נמצא באותה רשת של A, אז A ישלח מעטפה ARP. הרואוטר שאחראי על רשת C. לרואוטר ש后排負責這封郵件。

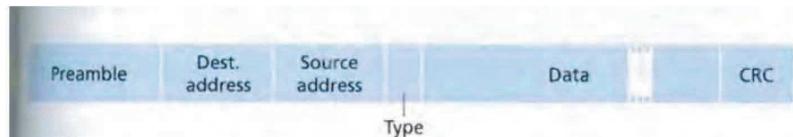
IP A	MAC A	Ip C	Mac Broadcast
------	-------	------	---------------

ההודעה של A מגיע לרואוטר של C, הרואוטר פותח את ההודעה ומחדיר ל A תשובה.

IP Gateway	MAC Router	Ip A	Mac A
------------	------------	------	-------

## Ethernet

### Ethernet Frame Structure



נסביר באמצעות דוגמא: נתאר שימוש datagram של IP המשמש למשתמש כאשר 2 המשתמשים באוטו LAN.

לאדרט (Adapter) של השולח יש כתובת MAC, AA-AA-AA-AA-AA-AA, ולקבל אותה כטובת רק עם B.

- **8 בתים (preamble)** – 7 הבטים הראשונים משתמשים להערת"ם המקבל על מנת לסנן את החסמים. 7 הבטים הם בעלי ערך 10101010 ויאלו הביט האחרון הוא בעלי ערך 10101011, שני הביטים האחרונים בשדה זה מיועדיםליידע את B שמתחליה העברה.

- **6 בתים (Dest. Address)** – מכיל כתובת MAC של היעד.
- **6 בתים (Source Address)** – מכיל כתובת MAC של המקור.

- **2 בתים (type)** – סוג פרוטוקולי שכבת הרשת.
- **1500-46 bytes (Data)** – נושא את העברת המידע מהקסימאלית (MTU) Ethernet היא 1500 והמינימאלית היא 46 בתים.

#### • CRC (4 בתים) – זיהוי תקלות.

Ethernet משתמש בהברת **baseband** – הא댑טר שלוח סיגנל דיגיטלי ישירות לערזץ השידור. הוא עושה שימוש ב**קידוד מנצ'סטר**, כל בית מורה על דבר אחר: בית 0 מורה על תזוזה מלמטה למעלה ובית 1 מלמעלה למטה. ברגע ששען הא댑טר של המקלט מסונכרן, המקלט יכול לשרטט כל בית ולקבוע האם הוא 0 או 1. הקידוד הוא פועלה של השכבה הפיזית אך הוא בעל שימוש רב ב-Ethernet.

#### An Unreliable Connectionless Service

Ethernet לא מספקת אמינות בנושא העברת הודעות, שלא כמו ב-IP אין "לחיצת ידיהם" ואין ACK או NAK. כשהודעה מ-A ל-B לא עוברת טוב, B פשוט מזריך (discard) אותה ומחשב A לא יידע על כך. הדבר יכול לגרום לחורים ברכף העברת.

כאשר הודעה לא עוברת את בדיקת CRC היא מופלת ולא נשלחת שום אינדיקציה לא댑טר השולח. לכן יכולים להיווצר חורים ברמת הקישוריות. אם החבילה עטופה בرمת התעבורה בפרוטוקול UDP אז אכן לצד המקלט יהיו "חורים" במדיה. אם השתמשנו ב프וטוקול TCP, אז הטיעות תתגלת כאשר נקלף את החבילות עד רמת התעבורה ושם בספיית החבילות נגלה כי חסירה חבילה. במקרה זה תישלח הודעה שגיאה וחビルת ה-TCP תישלח שוב. נשים לב כי במקרה זה אומנם המידע שנשלח בשנייה עבר שוב דרך הא댑טר, אבל הוא לא ידע להבחין בין מידע ישן שנשלח שוב למידע חדש, וכך נומר שהא댑טר אינו שלוח את המידע החדש.

Ethernet משתמש בפרוטוקול CSMA/CD ע"י מדידת רמות המתח לפני ובמשך השליחה.

כל א댑טר מרים את ה-CSMA/CD ללא תיאום מפורש עם שאר האדים.

במקרה ובמהלך השליחה, הא댑טר מבוחן (sense) שיש הודעה אחרת שמווערת, השליחה שלו תתבטל ווישלח סיגנל jam (בגודל 48 בית), על מנת לידע את כולם שהייתת התנגשות.

לאחר שזוזתה ההתנגשות, הא댑טר ממחה זמן רנדומלי. זמן רנדומלי זה משתנה באופן אקספוננציאלי, באופן הבא: לאחר כל שליחה הא댑טר ממתין זמן של  $512bit \cdot k$  אשר  $k$  הוא פרמטר עליון נרחב מיד,  $512bit \cdot k$  זה הזמן שלוקח לא댑טר לשלח 512 בית (תלוי ברוחב פס).

ה- $k$  נבחר באופן רנדומלי מתחום שהולך וגדל באופן אקספוננציאלי. כלומר  $2^n \leq k \leq 0$ , כאשר  $n$  הוא מספר היכישנות. אשר  $n$  מגע לערך התקorra 10 הוא לא עלה יותר.

## Link Layer Switches

### Switches Vs. Routers

אם switches ו-gm routers מועמדים להיות רכיבי תקשורת שמקשרים בין כמה hosts ורכיבים נוספים. נראה יתרונות וחסרונות של כל אחד.

#### Switches

##### יתרונות:

- switches הם plug and play – לאחר החיבור switch לומד בעצמו את המבנה הרשות ללא צורך בהגדרת.
- בעלי שיעור מיון ופילטר גבוהים (מעבדים עד לשכבה 2 ואילו routers עד שכבה 3).

##### חסרונות:

- טופולוגיה הקישור מוגבלת ל"עץ פורש" (על מנת למנוע מעגלים) ולכן דיווקה נמצאת ב المسلול המינימלי.
- הרבה switches – הרבה טבלאות ARP. תנוצת שאלות ARP תהיה רבה לעומת זאת רוטר שבקשת ARP למשתמש ברשת שלו תיעזר עצמו.
- לא מגנים בפני "סערות" בערזץ, אם host שולח רשיימה בלתי נגמרת של הודעות הסוויץ' יעביר את כלן הלאו ויסתום את כל הרשות.

#### Routers

##### יתרונות:

- הטופולוגיה לא מוגבלת ל"עץ פורש" וניתן להשתמש במסלול הקצר ביותר, מה שמאפשר לאינטרנט להבנות עם טופולוגיה עשרה בעריצים. חשוב להבין כי בغالל הגדרות לא נוכנות של הרוטר, פריטם datagram יכול להסתובב בממעלים, אבל קיימים לה שדה ב-header שתפקידו למנוע זאת, זה שדה TTL – Time to Live (ראה מעתפת IP).
- מספקים הגנת firewall 2. יכול להגן נגד "סערת" של שידור Ethernet על ידי חסימת מקור שמשדר ללא הגבלה.

##### חסרונות:

- לא plug and play – חייב IP על מנת להגדיר.
- זמן עיבוד גבוה יותר בغالל שmagiu עד לשכבה 3.

**מתי נשמש במה?**

עבור רשתות קטנות עם כמה מאות hosts מוסף switch אבל עבור רשתות גדולות עם כמה אלפי hosts ניאלץ להשתמש ב-router בנוסף לסוריצ'ים.

**:HUB**

מחבר בין 2 מערכות LAN. מקבל מעטפה ועביר אותה לכל מי שמחובר אליו.

## Wireless Network

### :CDMA

רעיון הוא כך: כולם משדרים כל הזמן על אותם התדרים. השדר שעובר בך הוא חיבור של כל השדרים שכרגע משודרים ברשת. מי שאחראי על פיענוח השדר הוא הרכיב שקולט אותו בפרוטוקול, רק מי שאמור לקבל את המידע בסופו של דבר יוכל אותו.

תהליך בין host-ים שմדברים בניהם, יהיה תבנית ידועה כלשהי. כל בית שנרצה לשלוח נכפיל אותו ב-

גודל התבנית ואת מה שנשלח נכפיל בהתאם בתבנית.

לדוגמא:

התבנית תהיה:

1	1	1	-1	1	-1	-1	-1
נרצה לשלוח							

1	-1	1	-1
---	----	---	----

כל בית בשדר שנרצה לשלוח נכפיל אותו ב 8 כגודל התבנית,

1	1	1	1	1	1	1	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

נכפל כל בית שכפלנו, בהתאם.

1	1	1	1	1	1	1	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

X

1	1	1	-	1	-	-	1	1	1	-	1	-	-	1	1	1	-	1	-	-	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

=

1	1	1	-	1	-	-	1	1	1	-	1	-	-	1	1	1	-	1	-	-	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

את הטליה האחורונה נשלח ל HOST היעד. host מקבל את השדר, כופל אותו בהתאם בתבנית  
לכל 8 בית (בית 1 מקורי) עושה סכום של הכפל. ומתקבל את הביט המתאים.

**לדוגמא:**

## לבית מספר 1:

שדר	1	1	1	-1	1	-1	-1	-1	-1
	X								

תבנית	1	1	1	-1	1	-1	-1	-1	-1
-------	---	---	---	----	---	----	----	----	----

הסכום : 8 ונחלק את הסכום ב-8 ונקבל את הבית המבוקש שהוא 1.

מגנום

- קצב הרשות צריך להיות פי כמה ממה שהוא רוצה לשדר כתלות באורך התבנית.
  - צריך לדעת איך אנחנו יוצרים את הקודים ולא כל כך פשוט למצוא אותם כי הם צריכים להיות אוטוגונליים.
  - מספר היחידות המשדרות הם כמספר הקודים הקיימים בראשת.
  - אין הרבה קודים שונים לשימוש בהם.

#### **יתרונות:**

- כלום משדרים בו זמינות על גבי כל התדרים.
  - אין צורך בסync'רין בין התחנות.
  - מי שאינו לו תבנית חושב שף אחד לא מדובר עכשו ברשות.

# Cognitive network

תאיון לכל התדרים עד שМОוצאת תדר פניו לשדר בו. ברגע שעוד מישחו משדר בתדר שלה, תחפש תדר אמצעי

## Channel Partitioning Protocols

. TDM ו FDM : (broadcast channel) שידור ערוץ אחד לחלקת שיטות שתי דובר על פרק בפרק 1.

- TDM** - מחקקת את הזמן למסגרות זמן וכל מסגרת כזו מחולקת לנ' סוליטים שמדוברים לכל אחד מן הצלמים. כאשר לצמת יש הודעה לשולח, ההודעה מועברת בזמן המועד לצמת זו. כאשר לכל השולחים ניתן האפשרות לשולח, הסביב מתחילה שוב.

מפרט:

1. יש מצב שהשליח יחכה הרבה (ולרוב סתם) עד להגעת המור שלו לשולח.

2. במצב שיש רק שלוח אחד עדין תהיה העברת בקצב של N/R (לא טוב).
- **FDM** – מחלק את R לתדריות שונות (כל אחד עם קצב של N/R) ומקצה כל תדרות לכל אחד מ-N הצמתים. בעצם יוצר N/R ערוצים חדשים, קטנים יותר מעורץ R המקורי. גם אחד החסרונות שצומת מוגבלת לקצב של עד N/R.
  - **CDMA** – מקצה קוד שונה לכל צומת. לכל צומת יש קוד שונה לקידוד הביטים הנשלחים וכך מאפשרים לכל host לשלוח כרצונו.

Trudy



## Network Security

### מוטיבציה:

נרצה שרק מי שליח את הודעה ומילא אותה י賓ו אותה. כדי לעשות זאת צריך קודם לבצע אימות בין השולח למקבל.

### הפולש:

- יכול להציג לכל הודעות ברשת ב-LAN הרו כולם מקבלים את כל הודעות.
- הפולש יכול להכניס הודעות שלו לתוך הרשת ויכול להכניס הודעות שכယול לא שיוכת אליו לרשת כלומר לשולח הודעות ולשנות להם את ה src IP.
- יכול להתחזות למשהו ברשת .
- ניתן למנוע שירות ממישחו - ניתן להעמס שרת מסוים בבקשתו ואז להשבית אותו.

### מונחים כללים:

- $K$  שם קיצור למפתח.
- $K_A$  מפתח של אליס.
- $K_B$  מפתח של בוב.
- $M$  הודעה לא מוצפנת .
- $K(M)$  הודעה מוצפנת.

**מפתח סימטרי** - מפתח המשותף רק לשני משתמשים בלבד אותו ניתן לקודד או לפענן את הודעות.

### דוגמאות לקידוד בעזרת מפתחות סימטריים:

(1) התמרת אותיות – נחליף כל אות באות אחרת.

בשיטת זו לבוב ואליס חיב להיות את אותו המפתח.

### חרוגונות השיטה:

- אם יודעים מהו לגבי הטקסט המקורי אז קל לדעת את ערבות האותיות, לדוגמה: אם מכתב מתחל תמיד ב"שלום" אז ניתן לשער את הערבול לאותיות הנ"ל.
  - ניתן לעשות התקפה סטטיסטית – ידוע כי באנגלית האותיות השכיחות ביותר הן t,e,a,g נוצר גרף שייצג את מספר האותיות המקודדות, האותיות בעלות המקסימום בגרף הן האותיות השכיחות ביותר וכן הלאה.
- (2) שיפור להתרמת אותיות – נוצר ח ערבלים שונים של כל אותות כל אות במילה נקודד ערובל שונה. בקידוד זה לא ניתן לעשות התקפה סטטיסטית.

© נערך ונכתב פֵי שמעון אורזיאן, רון רוזנפָּלֶד, גיא פָּלָג

© נערך ונכתב פֵי שמעון אורזיאן, רון רוזנפָּלֶד, גיא פָּלָג

## מפתח פומבי

את מפתח זה יוצר משתמש מסוים משתמש זה יוצר מפתח ציבורי שאותו הוא מפרסם לכלום ובנוסח הוא יוצר מפתח פרטי שייהי שיר רק לו.

- **מפתח ציבורי** – מפתח זה ידוע לכל משתמש ומשמש להצפנה המידע (את מפתח זה בדרך כלל נשיג certification authority)
- **מפתח פרטי** – מפתח זה שיר רק למשתמש אחד ואיתו ניתן לפענה את המידע שהצפינו בעזרתו המפתח הציבורי.

### דוגמאות לקידודים בעזרת מפתחות פומביים:

:RSA

הרעין של RSA עובד על חשבון מודולרי, הרעיון הוא זהה:

אנו מסתכלים על כל בלוק בהודעה כמספר שלם.

- (1) בוחרים 2 מספרים שונים ראשוניים p,q.
- (2) מחשבים :  $q^k \equiv h \pmod{p-1}$ .
- (3) נבחר מספר נוסף  $e < h$  כך ש  $e$  זר ל  $p-1$ . כלומר אין להם מכנה משותף.
- (4) בחר מספר  $d$  כך ש  $ed \equiv 1 \pmod{p-1}$ .

$$ed - 1 \equiv 0 \pmod{p-1}$$

$$ed \equiv 1 \pmod{p-1}$$

- (5) מפרסמים לכלום מפתח פומבי את  $h$  ואת  $e$  ואילו המפתח פרטי יהיה  $d$  ו-p.

כיצד לשЛОח הودעה:

מחלקים את ההודעה לרצפים של ביטים כך שלכל רצף שנבחר נתייחס כמספר עשרוני. חשוב לשים לב שמספר זה תמיד יהיה קטן מ-p עבור כל רצף צזה, נקרא לו m. נניח  $n \equiv c^d \pmod{p-1}$  כאשר c הוא המספר המקורי.

כדי לפענה הודעה:

היעד מקבל את c ומחשב  $m = c^d \pmod{p-1}$ .

את שיטת הצפנה זו עדין לא פרצו. וזאת מכיוון ש- d ו-p מספרים ראשוניים גדולים מאוד, (גודל של 1024 ביט לפחות) ולכן קשה למצאו מספרים כאלה.

© נערך ונכתב פֵי שמעון אורזיאן, רון רוזנפָּלֶד, גיא פָּלָג

### **– Integrity**

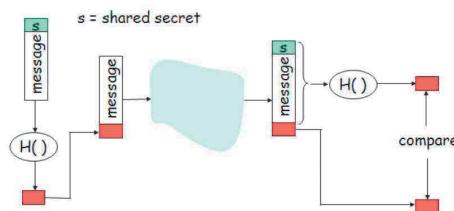
בכדי להבטיח שancockה נכתבה ע"י משתמש כלשהו נוספת חתימה מסויימת לכל משתמש הייחודית אֶרְ וּךְ לו. בנוספַף קיימת פונקציית *hash* ציבורית הממפה מידע כלשהו למידע אחר.

לאילס נוספה חתימה מסויימת שבוב מכיר, ישנה פונקציית *HASH* פומבית אותה יש לצלם, פונקציה זו מודדת שהמיפוי שלה תמיד יתן את אותה תוצאה כלומר על כל *input* נקלט אותו *output*, אבל לא להיפר. אם אליס רוצה לדבר לבוב :

- אליס שולחת לבוב הודעה – למידע שהוא רוצה להעביר היא תשריר מקדים מה את החתימה שלה, ותעביר אותה בפונקציית  $hash$  עברורה תקבל ערך כלשהו. אליס תשלח לבוב הודעה כך שהמידע ישלח יחד עם תשובה פונקציית  $hash$ .

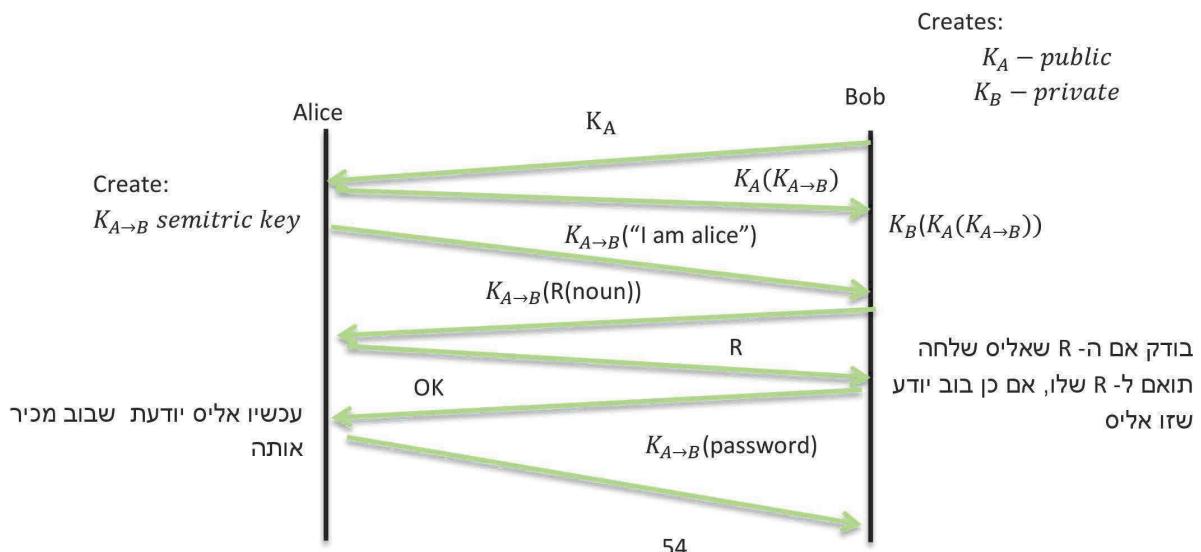
הודעה	$\text{Hash}(\text{signature}/\text{message})$
-------	--

2. בוב קיבל את ההודעה, ומפרק אותה ל<sup>2</sup> אט להודעה ולערך של פונקציית *hash*. בוב מכיר את החתימה של אליס, אך להודעה שהוא קיבל הוא מוסיף משמאלו את החתימה של אליס ומעבר אותה לפונקציית *HASH* לאחר מכן משווה עם מה שאליסשלח לו. אם יוצאת תשובה נכונה אז אליס זאת שכתבה את ההודעה.



#### **השלבים לשילוח הودעה מאובטחת:**

1. הצפנה באמצעות RSA.
  2. אימות באמצעות noun.
  3. בדיקת תקינות ההודעה (integrity) באמצעות shared key (החותימה מקודם)



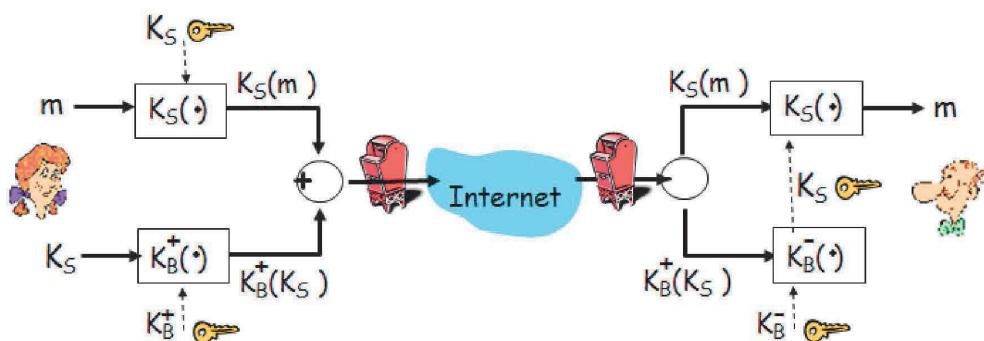
## בטחה במייל

אליס רוצה לשלוח הודעה סודית במייל לבוב:

- אליס יוצרת מפתח סימטרי רנדומלי  $K_S$
- את ההודעה שהיא רוצה לשלוח היא מקודדת באמצעות המפתח הסימטרי.
- בנוסף באמצעות המפתח הפומבי של בוב היא מקודדת את המפתח הסימטרי שהיא יצרה.
- אליס מחברת בין מה שהיא יצרה, ושלחת לבוב.

בוב:

- משתמש במפתח הפרטי שלו ומפענח את המפתח הסימטרי שאלייס שלחה לו
- בעזרתו המפתח הסימטרי הוא מפענח את ההודעה.



אליס רוצה להבטיח לבוב שacky היא כתבה את המיל

. אליס מעבירה את ההודעה בפונקציית  $hash$ .

- את תוצאה הפונקציה היא מקודדת באמצעות המפתח הפרטי שלה ובכך היא למעשה חותמת על ההודעה.

لتצאה זו היא משלרת את ההודעה ללא כל קידוד ושולחת לבוב.

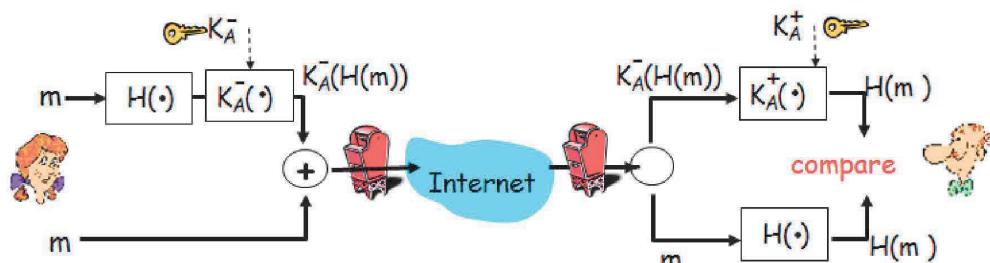
**בוב:**

מפריד בין החלק המוצפן לחלק הרגיל.

- את החלק המוצפן הוא מפענה באמצעות המפתח הפומבי של אליס. בכך הוא מקבל את ההודעה לאחר

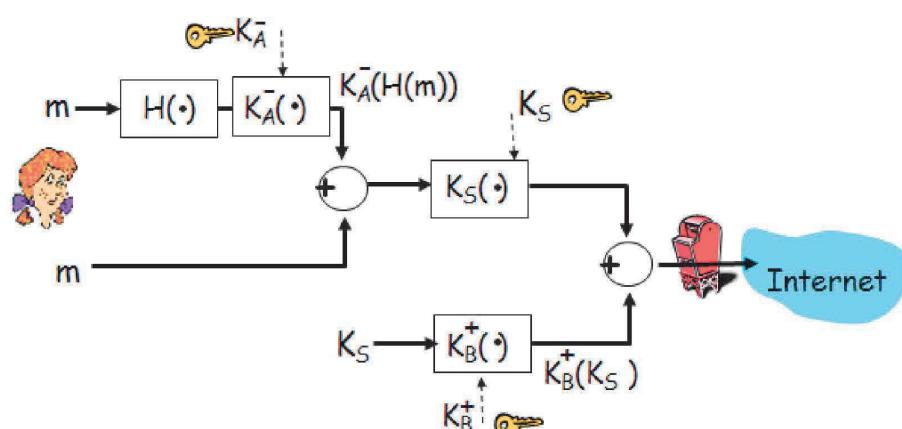
שברה את פונקציית  $hash$ .

- את ההודעה ללא מוצפן הוא מעביר דרך פונקציית  $hash$  ומשווה ביניהם.



שילוב של שתי השיטות יחד:

קודם אליס עושה את שיטה 2, ולאחר מכן עושה את שיטה 1.



## סolutions:

חידות המריה:

Multiples of bits			V · T · E	
SI decimal prefixes		Binary usage	IEC binary prefixes	
Name (Symbol)	Value		Name (Symbol)	Value
kilobit (kbit)	$10^3$	$2^{10}$	kibibit (Kibit)	$2^{10}$
<b>megabit</b> (Mbit)	$10^6$	$2^{20}$	mebibit (Mibit)	$2^{20}$
gigabit (Gbit)	$10^9$	$2^{30}$	gibibit (Gibit)	$2^{30}$
terabit (Tbit)	$10^{12}$	$2^{40}$	tebibit (Tibit)	$2^{40}$
petabit (Pbit)	$10^{15}$	$2^{50}$	pebibit (Pibit)	$2^{50}$
exabit (Ebit)	$10^{18}$	$2^{60}$	exbibit (Eibit)	$2^{60}$
zettabit (Zbit)	$10^{21}$	$2^{70}$	zebibit (Zibit)	$2^{70}$
yottabit (Ybit)	$10^{24}$	$2^{80}$	yobibit (Yibit)	$2^{80}$

See also: Nibble · Byte · Multiples of bytes  
Orders of magnitude of data

## אוסף שאלות ל מבחן ברשותות

מרצה: אלכס פריד

נכתב ע"י

גיא פלג, שמעון ארזואן, רון רוזנפלד.

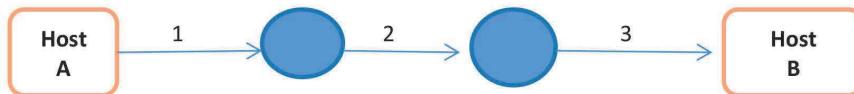
### חלק א Message switch & .Packet switch

שניהם ונריצה לשולח הודעה בגודל 7.5 Mbit מ-A host ל-B host. נניח כי בין host הנו יல ישנו עוד 2 תחנות מעבר.

נתנו לנו רוחב פס 1.5 Mbit/Sec

תוך כמה זמן תגיע הודעה מ-A host ל-B host ?

ענה על השאלה תוך שימוש ב-Message switch ובשימוש ב-Packet switch  
כשיש לנו סה"כ 5000 חבילות.



ת:

שיטת Message switch : נשים לב בכל חיבור שיש לנו יכול לעבוד שנייה רק 1.5Mbit ,  
לכן את החבילות מידע שלמו נעביר ב-5 חבילות לכל אחת בגודל bit 1.5Mbit , לכן מעבר ב-  
כל פס ייקח לנו 5 שניות , ולכן נעביר את כל המידע ב-15 שניות.

שיטת Packet switch: ראיינו כי במקרה כל המידע יכול לעבור דרך הפס בזמן של 5  
שניות .

נhapus את גודל ה-packet , סה"כ יש לנו 5000 חבילות לכך כל חבילה תהיה בגודל של :

$$\frac{7.5 * 10^6}{5000} = 1500 \text{ bit}$$

כל packet תהיה בגודל 1500bit לכן כל packet תעבור בכל פס הזמן של :

$$\frac{1500 \text{ bit}}{1.5 \frac{\text{Mbit}}{\text{Sec}}} = 0.001 \text{ Sec}$$

לכן חתיכת ראשונה תגיע ל-B host תוך 0.003 Sec ומשם כל הpacket יגיעו בתור אחד  
אחרי השניה, כלומר תוך 5.003 שניות תגעה כל החבילה ליעד .

נכתב ע"י שמעון ארזואן, גיא פלג, רון רוזנפלד

ש: נתון הודעה בגודל של bit 750,000, נתון רוחב פס 1.536 Mbit/Sec כל חיבורעובד בשיטת TDM שבו יש לנו 24 סלוטים. לכל שליחה יש השהייה של 0.5Sec כמה זמן יקח לשולח את ההודעה?

ת: מכיוון שיש לנו 24 סלוטים המהירות האפקטיבית תהיה  $\frac{1.536 \text{ Mbit}}{24} = 0.064 \left( \frac{\text{Mbit}}{\text{Sec}} \right)$  נצטרך לשולח את ההודעה בחלקים קטנים שכטול חלק הוא בגודל 0.064Mbit, לכן נצטרך לעשות :

$$\left[ \frac{750,000 \text{ bit}}{0.064 \text{ Mbit}} \right] = \left[ \frac{750,000}{0.064 * 10^6} \right] = [11.71] = 12$$

קיבלנו שנצטרך לשימוש 12 פעמים בסלוטים שלמו. נשים לב שבין כל סלוט יש לו 0.5 שניות השהייה, לכן סך שיקח לנו לשולח את ההודעה הוא:  
 $11 * 0.5 + 12 = 17.5 \text{ Sec}$

ש: נניח שימושיים חולקים רוחב פס של 100Kbit/sec ו声称 משמש ציריך 1Mbit/sec בשימוש בו- Circuit switching בכמה משתמשים אפשר לתמוך?  
 ת: ניתן לאפשר שימוש ל:

$$\frac{1 \text{ Mbit}}{\frac{\text{Sec}}{100 \text{ Kbit}}} = 10 \text{ Users}$$

## חלק ב שכבות האפליקציה :

### 1. תן דוגמא ל 4 אפליקציות ופרוטוקולים מшибכבות האפליקציה:

אפליקציה	פרוטוקול
דף	HTTP
uTorrent	Torrent – P2P
outlock	SMTP
fileZila – קבצים	FTP

### 2. מה אינפורמציה שהתהליך ב- Host משתמש בה על מנת לדבר עם תהליך בשרת?

הוא משתמש בכתובת ה- IP של השרת ומספר הפורט שאנו חזו יוצאים דרכו.

### 3. מודיע הפרוטוקולים : Http,Ftp,SmtpTCP ?

מכיוון שפרוטוקול TCP הוא פרוטוקול אמין, ואילו הפרוטוקולים הנ"ל מחובבים בכל המידע שהם שלוחים יגיע במלואו. לדוגמא: לא נרצה לשולח מייל שרק חלק ממנו הגיע ליעד. נזכיר כי UDP לא מבטיח שהמידע יגיע ליעד.

**4. מה האפשרות לימוש מעקב אחרי Host בפרוטוקול Http?**

- א. Cookies, host שומר מידע כך שבחריבור הבא לאתר, ישלח באופן אוטומטי אמצעי זהה של אותו host . בכל פעם שניתכנו לאתר נגידיל את Counter שסופר את מספר הפעמים שנכנסו לאתר ב-1.
- ב. HTTP Authentication – בכל פעם כניסה ידנית את אמצעי זהה ואז כשהאתר יעלה נגידיל את ה-Counter.

**5. האם 2 דפי Html שונים יכולים לעבור דרך אותו Connection ?**

לפעמים,

לדוגמא ישנו 2 דפי html שונים על אותו שרת:

www.walla.com\login.html

www.walla.com\welcoom.html

לשני הדפים יהיה אותו Connection לשרת של האתר walla אבל לשניהם דפים שונים.

**6. בשימוש ב-Pipeline מה חוסכים?**

אנחנו חוסכים את RTT, זמן המתנה לתשובה. לא צריך לחכות לתשובה.

- 7. למה אפשר להגיד ש프וטוקול FTP שולח את המידע שלו מחוץ לעורץ?**
- FTP פותח 2 ערכאים: אחד עבור Data ואחד עבור Control(מידע). לכן התקשרות היא מחוץ לעורץ ששולח מידע. ערך – כאשר פותחים זוקם בעזרת Socket.

**8. כל בקשה של DNS ? , לוקח  $RTT_i$  , בקשה לאתר לוקח  $RTT_0$  . כמה זמן לוקח עד לכינוסה לאתר?**

בשביל לחיצת ידיים קיבל  $RTT_0$

בשביל קבלת המידע מהאתר קיבל  $RTT_0$

בשביל בקשת DNS , נניח כי יש לנו ח DNS עד שנגיע ל-SNS של האתר:

$$\sum_{i=1}^n RTT_i$$

סה"כ קיבל:

$$2RTT_0 + \sum_{i=1}^n RTT_i$$

- 9. נניח ונרצה להעביר קובץ בגודל 1MB מהירות העורץ היא 1GB/SEC כאשר  $RTT=100Ms$ . מה התפוקה של העורץ?**

X = כמות המידע שרצינו להעביר = 1MB

Y=סה"כ הזמן שלוקח לנו להעביר אותו אוטומטית

$$\frac{1MB}{1(\frac{GB}{Sec})} = \frac{1 * 2^{20}}{1 * 2^{30} * \left(\frac{1}{Sec}\right)} = \frac{1}{2^{10}} Sec = 0.001 Sec$$

כתב ע"י שמעון ארזיאן, גיא פלאג, רון רוזנפלד

$$\text{Throughput} = \frac{1MB}{0.101Sec} = \frac{2^{20}}{0.101Sec} = 10381940.59 \frac{1}{Sec} \\ = 9.9MB/Sec$$

**10.** רוצים להעביר ברשת סרטון וידאו ברזולוציה של **640X640** פקסלים. כאשר כל פיקסל לוקח 3 בתים, ואת התמונות רוצים להראות 30 פעמים בשנית. מה צריכה להיות מהירות הערץ בשבייל זה?

נחשב את מספר הבתים שנצרך להעביר בשנית:

$$640 * 640 * 3 * 30 = 36864000 = 35.15 MB$$

כלומר נדרש שמהירות הערץ תהיה:  $35.15 \frac{MB}{Sec}$

הערה – חשוב להבין כי מהירות זו גבוהה מאוד, אין הרבה ערכאים שיכולים לעבוד ב מהירות כ אלה, لكن מקווצים את התמונה.

**11.** רוצים לשולח פקס **10 X 8** אינץ' של מידע שהוא שחור לבן והרזולוציה של הדפסה היא **72 פיקסלים לאינץ'** (פיקסל שוקל ביט) כמה זמן יקח להעביר את הפקס אם הוא נשלח דרך מודם שמהירותו  $14.4 \frac{(kb)}{sec}$ .  
נחשב את גודל המידע

$$720 byte = 5760 bit = 8 * 10 * 72$$

$$\frac{5760 bit}{14.4 \left( \frac{kb}{sec} \right)} = 0.4 Sec$$

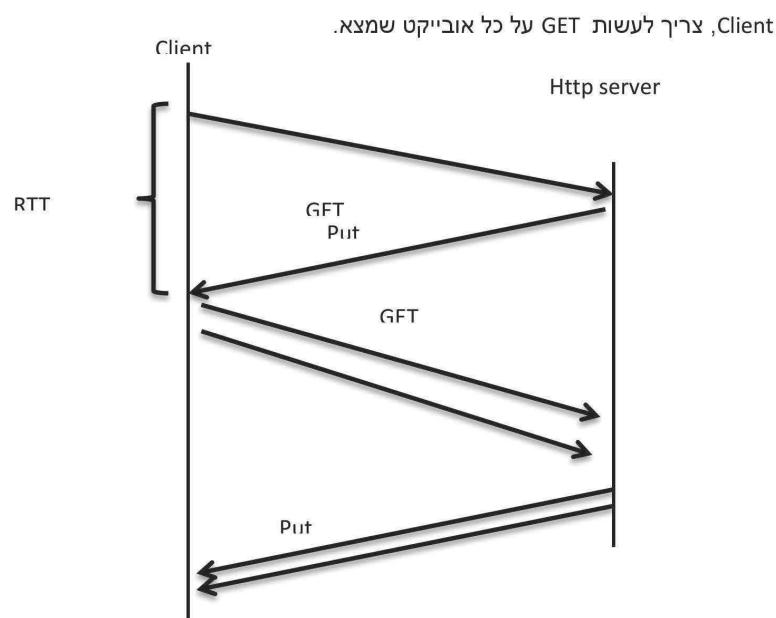
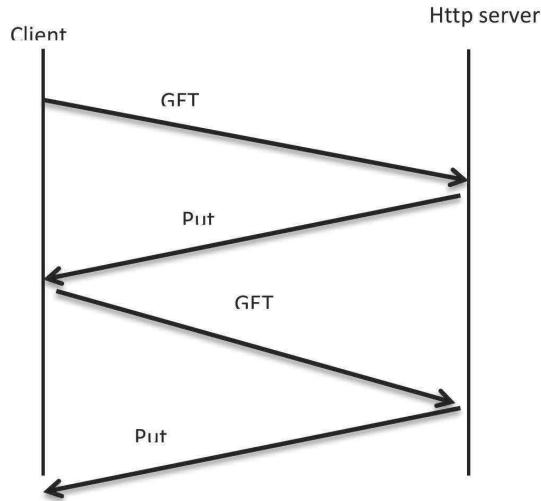
**12.** האם פרוטוקול **Http** מתבסס על **Out-Of-Band**?  
לא, הוא משתמש רק בערך אחד (בדרך כלל **port 80**).  
ה-**Ftp** משתמש פרוטוקול **Out-Of-Band**.

**13.** האם **DNS** משתמש ב-**UDP**?  
כן, DNS אחראי לתרגום כתובת **URL** לכתובת IP. אם **host** פנה אליו ורצה ללחוץ כתובת **URL** אז ה-**DNS** שולח את ה-IP ולא אכפת לו אם ה-**host** קיבל אותו או לא, כי אם לדוגמא הוא לא יקבל אותו הוא ישלח בקשה שוב.

**14.** אם כל שרת ה-**DNS** נפלן, האם לא ניתן לגלוש?  
כן, ניתן להיכנס לאתר באמצעות IP

**15.** נניח שהמשתמש מבקש דף אינטרנט שמכיל טקסט כלשהו ושני תמונות:  
שתי האופציות האפשריות הן:

כתב ע"י שמעון ארזיאן, גיא פלג, רון רוזנפולד



מקרה 2 נקרהpipelining. הדפסן שלח 2 בקשות get, על מנת לקבל. כלומר שוב לכל Client, צריך לעשות GET על כל אובייקט שמצוא. . חיב להיות בקשה get.

לדוגמא: בפעם הראשונה נשלח בקשה לאתר, חוזר לנו האובייקט שמכיל את הtekסט של האתר + 2 לינקים חדשים לתמונות שהאתר מכיל, בעזרה הלינקים נשלח בקשה לתמונות (זיכרון תמונה זה אובייקט), מקבל את התמונות ונוצרו אותן לדף של האתר.

כתב עי' שמעון ארזיאן, גיא פלאג, רון רוזנפלד

## 16. למה שרת DNS מיושם כהיררכיה של שירותי ולא שירות אחד גדול?

ב כדי להימנע מ:

- א. Single Point Failure – אם ה-DNS ייפול אז כל האינטרנט ייפול.
- ב. Traffic Volume – שירות יחיד יאלץ להתמודד עם כל הבקשות בלבד וזה בלתי אפשרי.
- ג. Distant centralized database – השירות תמיד יהיה רוחק מקומות מסוימים, מה שייגרם לאיות תמידית במקומות אלו.
- ד. Maintenance – קשה לתזק מסד נתונים גדול (כל עדכון host צריך לעדכן במסד נתונים).

## 17. בשרת DNS, מנה את היתרונות והחסרונות של שיטת חיפוש וקורסיבית ושיטת חיפוש איטרטיבית?

שיטה	יתרונות	חסרונות
שיטה רקורסיבית	<ul style="list-style-type: none"> <li>• אם ה-<i>recursive query</i> לא מעודכו, אזי פניה לכטובת מסוימת תגרום לטעות.</li> <li>• כל שירות זקוק לנפק איחסון <i>caching</i> בשบาล ה-<i>caching</i>.</li> </ul>	<ul style="list-style-type: none"> <li>• חוסר בקשות DNS לשירותים, ע"י שימוש ב-<i>caching</i> שאליה בודדת תהיה מהירה יותר במקורה שהיא נמצאת-ב-<i>caching</i> של השירות המקומי.</li> <li>• תהליך מציאת הכתובת שקוֹף מבחינת המשתמש, ככלומר הוא שולח בקשה אחת ועובדת מקבל תשובה.</li> <li>• תהליך מהיר יחסית.</li> <li>• אם שירות שמחזיר בכתובת נופל, אזי אם יש שירות נוסף שמחזיר ב-<i>Cash</i> שלו את הכתובת המבוקשת, אזי המשתמש עדיין יוכל לקבל כתובת זו.</li> </ul>
שיטה איטרטיבית	<ul style="list-style-type: none"> <li>• איתי יחסית.</li> <li>• גורר בקשות מרחבות לשירותים בכך יוצר עומס ברשת.</li> <li>• אם שירות נופל אז לא ניתן לקבל רק ממנו את המידע.</li> </ul>	<ul style="list-style-type: none"> <li>• תמיד נקבל מידע עדכני ונכון לגבי הכתובת המבוקשת.</li> <li>• זכות בלעדית לשרת על הכתובות שיש בו, הכוונה כדי להציג ללקוחות ספציפית נצטרך לבצע רק ממנו את המידע.</li> </ul>

כתב ע"י שמעון ארזואן, גיא פלאג, רון רוזנפלד

## חלק ג שכבת התעבורה :

1. נתון 3 מערכים של ביטים בגודל 8 ( נניח כי נבדוק עם *CheckSum* שעובד עם מערכים בגודל 8 במקומות 16 .  
 01010101,01110000,01001101  
 הראה חישוב של ה-*CheckSum* בעדרת המשלימים ל-1.  
 הצד השולח, יבצע את החישוב הבא:

<i>C</i>	1	0	1	1	1	1	1	
<i>B0</i>	0	1	0	1	0	1	0	1
<i>B1</i>	0	1	1	1	0	0	0	0
<i>B2</i>	0	1	0	0	1	1	0	1
<i>Res</i>	0	0	0	1	0	0	1	0
<i>Carry</i>	0	0	0	1	0	0	1	1

בשיטת המשלימים ל-1 כל ביט יוחלף בהפרש שלו, لكن נקבל את המספר:  
 11101100

את ה-*CheckSum* הוא ניתן לטור מעטפת ה-*UPD* וישלח אותה ליעד. היעד יקבל את המעטפה, יחבר את 4 המספרים (3 המספרים וה-*CheckSum*), אם בתוצאה שהוא קיבל ישנו ביט שאינו 1, אז נפללה טעות.

<i>C</i>	1	0	1	1	1	1	1	
<i>B0</i>	0	1	0	1	0	1	0	1
<i>B1</i>	0	1	1	1	0	0	0	0
<i>B2</i>	0	1	0	0	1	1	0	1
<i>CheckSum</i>	1	1	1	0	1	1	0	0
<i>Res</i>	1	1	1	1	1	1	1	0
<i>Carry</i>	1	1	1	1	1	1	1	1

2. מדוע *UPD* משתמש במשלימים ל-1 ולא בסכום עצמו ?

3. לקוח ושרת פתחו קשר *TCP*. בהמשך שלוח הלקוח את החבילות 25, 26, 27, 29 ו-30 עד החבילה 31. השרת קיבל את החבילות 25, 26, ולאחר מכן את החבילות 29, 30, 31.   
 א. תאר את פעולות *TCP* מצד השרת בקבלה החבילות *TCP* 27 מצד השרת.   
 ב. תאר את פעולות *TCP* מצד הלקוח כתוצאה מהתנהגות *TCP* 27 מצד השרת. אם נדרש לכם נתונים נוספים, הניחו את ההנחות המתאימות וציינו זאת במפורש בתשובתכם.

- א. אם *TCP* מצד השרת יש נתונים אפליקציה לשלוח הוא יעים על חבילות הנתונים אישורים לחבילות 25 ו-26, ולאחר מכן יחזיר ויעmis אישורים חוזרים לקבלת חבילה 27, כל עוד לא קיבל את חבילה 27. אם *TCP* מצד השרת אין נתונים אפליקציה

נכתב ע"י שמעון ארזואן, גיא פרג, רון רוזנפלד

למשלות, הוא ימתין קמעא (לא יותר מ 300 מילישניות) ואם עדין לא יהיה לו נתנו אפליקציה לשלוח הוא ישלח חבילות ריקות עם האישורים ל hutut חבילות 25 ו 26, וימשייר לשולח אישורים לקבלת 26, כל עוד לא קיבל את חבילה 27.

ב. ה- TCP בצד הלקוח, ברגע שהגינו אליו 3 אישורים חוזרים של חבילה 26, בין שיש חסימה בדרך - חבילה 27 (ואולי חבילות נוספות) אבדו, אך החסימה אינה מוחלטת (עובדת שמשהו בכל זאת הגע TCP בצד השרת וכן גם האישור), ולכן הוא יתחל לשולח מחדש את החבילות החל מחבילה 27.

4. חבילות נשלחות משרת אל לקוח הקשורים באמצעות UDP. החבילות עוברות שבירה (פרגמנטציה) באחד מנתבי הביניים.

א. תאר מבני נתונים אפשריים אצל IP בצד הלקוח אשר משמש אותו בתהיליך האיסוף של השבירים. دون ביתרונו והיחסון של כל אחד מלאה.

ב. מוצע שיפור IP: כאשר IP בצד האוסף את השבירים מזדהה כי אוסף מסוים משמשו של רצף שבירים, יעביר אותם לUDP שימושי (שמילא מצפה כל פעם לחבילה אחת, בגודל לא מוגדר מראש, ואינו שולח אישורים). מהיקן IP יודיע שמדובר בחבילה UDP? دون ביתרונות וחסרונות של שיפור זה.

א. IP בצד האוסף שבירים יכול לתזק רשותה מקוונת של השבירים, מומינת לפ' סדר הופעתם בחבילה המקורית. היתרון הוא שזה מבנה נתונים פשוט. החיסון הוא שבכל רגע בו מופיע שבר על IP האוסף לסרוק את הרשותה ולמצוא את המיקום. אפשר לשפר על ידי תחזוקת טבלת hash שהאנידקסים שליה מצביעים למקומות מסוימים בחבילה המקורית (שגדלה ידוע), וכל כניסה בטבלה מצביעה על רשותה מקוונת קטנה של שבירים.

ב. ראשית, IP יידע שזו חבילה UDP שכן זה רשום לו בשדה ה-Protocol. היתרון לכך הוא שם רצף השבירים יועבר לUDP אז הוא יעביר אותו ב מהירות לאפליקציה. אבל ככל לא בטוח ש UDP ידע למי להעביר את רצף השבירים, שכן מספר הפורט של האפליקציה המאזינה רשום ב כתובת של UDP, וכותרת זו נמצאת רק בשבר הראשוני UDP. יוצרך להמתין עד שיקבל רצף שבירים המתחילה בשבר הראשוני. את הרצף זהה יוכל להעביר לאפליקציה. אבל במקרה זה UDP יוצרך לשמור את המידע אוזות הרצף שהועבר, על מנת ליזהות את הרצף הבא. בקיצור UDP יוצרך לשמור הרבה מידע, שמילא רשום גם בIP. מעבר לכך, שיפור IP המוצע רק לחבילות הרבה מידע (ולא TCP) מסביר את IP והוא פרט לפחות כללי. השיפור הזה אינו שיפור.

5. לקוח ושרת פתחו קשר TCP. ערוץ התקשרות ביניהם מאפשר משלוח מיידי בקצב  $R Mbps$  לכל כיוון. בזמן פתיחת הקשר הם סכמו על גודל סגמנט  $S$  בתים, והלקוח סימן לשרת שגודלו החלון שלו  $W$  סגמנטים. השרת שולח ללקוח חבילות בעלות גודל ממוצע של  $k$  סגמנטים כל אחת. זמן סרב (Round Trip Time) הוא RTT שניות. בטאת את זמן הסרב RTT (בשניות), כפונקציה של כל או חלק מהפרמטרים האחרים, בהנחה שהשרות שולח את החבילות בקצב המרבי אך מנצל רק מחצי החלון. לאחר מכן חשב את זמן הסרב עבור

כתב ע"י שמעון ארזאון, גיא פלג, רון רוזנפולד

$$k = 4, R = 1 Mbps, S = 500 Bytes, W = 50 Seg$$

גודל כל סגמנט  $S = 8$  ביטים. חלון שלם פירשו  $WS = 8$  ביטים. ניצול מחצית הchlון פירשו שבמשך  $RTT$  שניות, ישלחו ממחצית מהביטים המותרים על פי הchlון, ככלומר  $WS/4$  ביטים. מצד שני, הביטים נשלחים בקצב  $R$  (מיליוני בית לשניה), ולכן במשך זמן  $RTT$  (בשניות) ישלחו  $R * 10^6 * RTT$  ביטים. לכן  $R * RTT * 10^6 = 4WS$

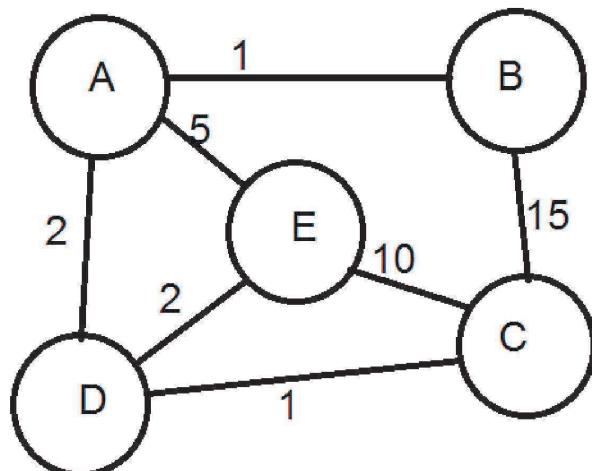
מכאן

$$RTT = (4WS/R) * 10^{-6}$$

$$RTT = (4 * 50 * 500/1) * 10^{-6} = 0.1 \text{ Sec}$$

## חלק ד שכבת הרשת :

1. נתן הגרף הבא:



מלא בכל שלב את טבלאות ה DV.

זמן 0 :

	A	B	C	D	E
A	0	1	$\infty$	2	5
B	1	0	15	$\infty$	$\infty$
C	$\infty$	15	0	1	10
D	2	$\infty$	1	0	2
E	5	$\infty$	10	2	0

כתב ע"י שמעון ארזיאן, גיא פלאג, רון רוזנפלד

זמן 1: - בזמן זה כל צומת שולח לשכן שלו את הטבלה שלו

	A	B	C	D	E
A	0	1	3	2	5
B	1	0	15	3	6
C	3	15	0	1	10
D	2	3	1	0	2
E	5	6	10	2	0

זמן 2: שוב פעם כל צומת תביא לשכן שלו את הטבלה שלו מה שיקרר באותו זמן זה, יהיה שיפור מרחוקים.

	A	B	C	D	E
A	0	1	3	2	<b>4</b>
B	1	0	<b>4</b>	3	6
C	3	<b>4</b>	0	1	<b>3</b>
D	2	3	1	0	2
E	<b>4</b>	6	<b>3</b>	2	0

2. נתון ה IP הבא ,**123.20.31.0**

- לאיזה מחלקה שייך ? A
- כמה host יש לו ? –  $2^{24} - 2 = 16,774$

3. נתון ה IP הבא **199.30.24.7**, נתב שקיבל את הכתובת האם הוא יכול לבנות **320 host** ?

לא הוא שייך למחלקה C ולכן הוא יכול לבנות רק  $2^8 - 2 = 250$ .

4. נתונה הכתובת **P/I** הבאה **.202.84.45.14**

- א. לאיזה מחלקה שייכת הכתובת ? **Class C**
- ב. מה ה-**Network Mask** של המחלקה ? **255.255.255.0**
- ג. מה ה-**ID Network** של המחלקה ? **202.84.45.0**
- ד. מה כתובה תחנת הנקה (**Host id**) ? **0.0.0.14**

5. נניח שיש לנו כתובת **0.0.0.176.19.0.0**,

א. כמה subnet-subnetים ניתן לבנות מכתובת זו ?

מדובר ב-class B, לכן ישנו 16 סיביות שניות להשתמש בהן. נשים לב כי לכל subnet צריך להיות לפחות 4 כתובות של host שהוא יכול לתרmor בהם (כי אם לא יהיה לו כתובות אז הרשות מבחןינו סתמית) כלומר ניתן לאפשר

כתב עי' שמעון ארזיאן, גיא פלאג, רון רוזנפלד

$$2^{16-2} = 2^{14} \text{ subnets}$$

ב. נרצה לחלק ל-650 subnets כמה hostים תוכל כל רשות להחזיק.

נחפש את החזקה של 2 כדי קרובה ל-650. נשים לב כי החזקה心仪的 קרובה היא 10 אך נדרשים 10 ביטים להקים 650 subnets.

כלומר ישאר לנו 6 ביטים לכל subnet תחזיק  $2^6 - 2 = 62$ . הורדנו 2 כי אנו צריכים כתובת אחת לhost Broadcast ואחת ל-ID Network.

ג. מה יהיה subnet mask ? מכיוון שהוא משתמשים ב-10 ביטים ל subnet אזי :

255.255.255.192

.6

ארגוני קיבל את גוש הכתובות 193.1.1.0/24 ויש לו צורך ב-6 תות רשות. מת הרשות הנגדולה ביותר כולל 25 מחשבים.

א. קבע את ה netmask (או אורך הקידומת)

ב. קבע את מספרי תות הרשות

ג. כמה כתובות מחשבים תהינה בכל תות רשות? כתוב את כתובות המחשבים בתת הרשות השנייה (כתובתה היא הנמוכה מכל כתובות תת הרשות, פרט לאחת)

ד. מהי כתובת ה broadcast של תת הרשות השנייה?

.255.255.255.224 א. או /27

ב. מספר תותי רשות 8 כתובות שהתחולן:

193.1.1.0 .1

193.1.1.32 .2

193.1.1.64 .3

193.1.1.96 .4

193.1.1.128 .5

193.1.1.160 .6

193.1.1.192 .7

193.1.1.224 .8

ג. 193.1.1.33-62 2^5 -2=30

ד. 193.1.1.63

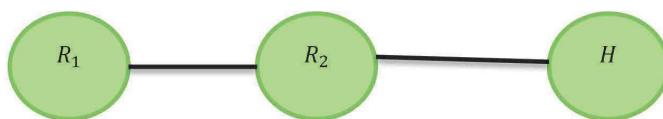
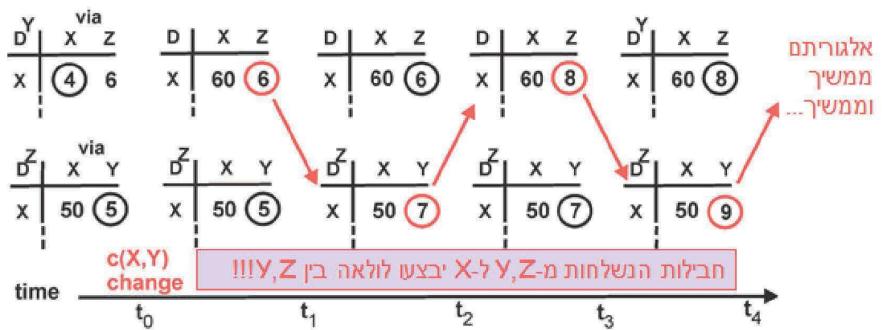
כתב ע"י שמעון ארזיאן, גיא פלאג, רון רוזנפולד

7.ospf הוא אלגוריתם שמתבסס על link state ?

נכון מכיוון שאלגוריתם זה עובד במציאת המסלול הקצר ביותר בין רשתות

8. מה היא בעיית הספירה לאינסוף Counting to infinity (פרוטוקול RIP)?

[הסבר בסיכום](#)



ה MTU של  $R_1$  הוא  $1100\text{bytes}$  ו  $R_2$  הוא  $1500\text{bytes}$ .

היחידה הגדולה ביותר הניננת להעברה MTU – maximum Transfer unit

רואה להעביר  $R_1$  של ip בגודל 4000 byte

א. כמה מעתופות צריך  $R_1$  לחלק את המידע על מנת להעביר אותו ל $R_2$ ?

ת) נזכיר כי ה overhead של מעתפה ip הוא 20 byte. לכן בכל פעם יוכל לשלוות

$$1500 - 20 = 1480\text{bytes}$$

1480 בתים של מידע. נדרש לחלק את המידע ל 4 מעתופות שונות כך:

	size	offset
1	1480	0
2	1480	1480
3	1040	2960

נכתב ע"י שמעון ארזיאן, גיא פלאג, רון רוזנפלד

ב) גרצה להעביר את המידע מR2 לhost. לכמה מעטפות علينا לחלק את המידע?  
נשים לב כי המידע שמתקיים בR2 הוא כל המעטפות שנשלחו מR1 מלהם 3 מעטפות גם הפעם יש לנו תקורה של 20 בתים. لكن בכל מעטפה נוכל לשולח:

$$1100 - 20 = 1080 \text{ bytes}$$

של מידע.

	size	offset
1	1080	0
2	420	1080
3	1080	0
4	420	1080
5	1040	0

#### 10. מה ההבדל המרכזי בין *distance vector* ל *link state*?

*Distance vector* בכל פעם יעביר רק לשכנים שלו את הטבלאות *hopsets* שלו ורק להם ואילו *link state* משתף את כל הalamatים במידע שלו. הבדל נוסף הוא *link state* משתמש באלגוריתם *bellman-ford* ואילו *distance vector* משתמש בדיקוסטרה.

#### 11. תן דוגמא ל프וטוקול שעובד בשיטת DV ו LS?

.DV – RIP

.LS – OSPF

#### 12. ניקח בחשבון את ה-SubNet 101.101.101.64/26 הבא:

א. תן דוגמא לכתובת IP ברשות זו?

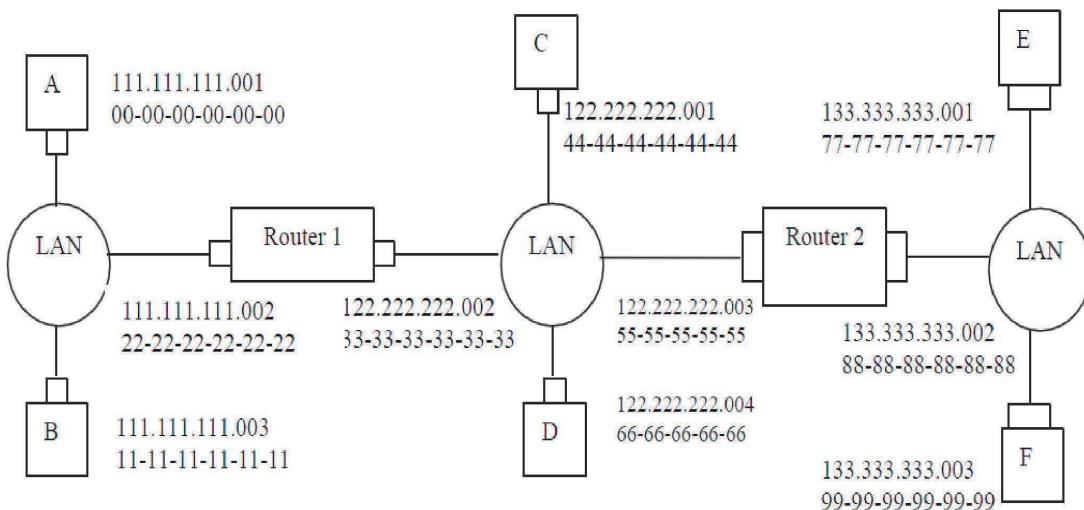
ב. נניח, שلسפק התקשורת שיכים הכתובות הבאות: 101.101.128/17 . עלייך  
לייצור 4 subnets מבлок זה, שלכל רשת יהיה אותה כמות host-ים. מה  
הכתובות כולן prefixes של התת רשות?

א. נשים לב כי אסור לנו לתת את הכתובת 101.101.101.64 מכיוון שהוא ה-*NetWork ID*, בנוסף אסור לנו לתת את הכתובת 101.101.101.127 כי זהה  
הכתובת של ה-*Broadcast*, لكن נוכל לתת כתובות מהטוויה:  
101.101.101.65-126

ב. נשים לב כי ה-*NetWork ID* היא 255.255.128.0 , עליו לחלק ל-4 subnets, וכך שכל subnet יוכל להחזיק  $2^{13} - 2$

ונכתב ע"י שמעון ארזואן, גיא פלאג, רון רוזנפולד

-  
- *Hosts*, כלומר כל *subnet* יתמור ב- $2^{13}$  כתובות, עכשו ה-ID-ה-*Hosts* של ה-*Subnets* תהייה:  
255.255.224.0



- from A to left router: Source MAC address: 00-00-00-00-00-00  
Destination MAC address: 22-22-22-22-22-22  
Source IP: 111.111.111.001  
  
Destination IP: 133.333.333.003
- from the left router to the right router: Source MAC address: 33-33-33-33-33-33  
Destination MAC address: 55-55-55-55-55-55  
Source IP: 111.111.111.001  
Destination IP: 133.333.333.003
- from the right router to F: Source MAC address: 88-88-88-88-88-88  
Destination MAC address: 99-99-99-99-99-99  
Source IP: 111.111.111.001  
Destination IP: 133.333.333.003

כתב ע"י שמעון ארזיאן, גיא פלג, רון רוזנפלד

## חלק ה שכבות הקיישוריות :

1) האם יש יתרות בין שכבות כלומר יש הרבה שכבות שעושות את אותן פעולות?

כן, שדה CRC בדיקת Sağiot – בשכבה הרגשת ישב פרוטוקול IP שעושה בדיקת checksum ובדיקת הקישוריות יש את אלגוריתם CRC שם הוא עושה בדיקת Sağiot. אך יש לנו יתרות וכך CRC בודק שאגיה שאם תהיה אז מיותרת SHA IP יבודק אותה.

2) האם LAN היא תשתית אמינה – הכוונה להודעות נשלחו והיעד קיבל אותן ובודאות?

לא מכיוון שבשכבה Data link אנו מחלקים את ההודעות לחטיבות קטנות ושולחים אותן משתמשים בפרוטוקול UDP כדי נחכה לACK ולכך לא כל הודעה מתקבלת אצל היעד בודאות.

3) האם הפרוטוקול שמשתמש בתווך ring protocols (Taking Turns Protocols)UIL אם הרשות גדולה?

בשימוש בפרוטוקול שמשתמש בתווך ring יש מעיטה שעוברת בסדר קבוע בין תחנות, כל תחנה שהטופון אצלה מדרצה. אם הרשות גדולה יקח הרבה מאוד זמן בין זמני שידור של תחנה, ככלומר לאחר השידור הראשון יעבור זמן רב עד שיגיע התור שלה לשדר שוב (ה-token צריך לעבור בכל שאר התחנות שברשת).

4) למה בקשה ARP נשלחת בצורה broadcast

מכיוון שאנו לא יודעים את כתובת MAC של היעד, אך מטרת שידור broadcast הוא לשלוח את המעיטה לכלם בכך שהיא ייחזר לנו תשובה עם כתובת MAC שלה.

5) נניח ש HOST קיבל את השדר הבא:

101010101010101011

רשום את השדר בצורה מטריצה בשיטת two dimesion parity check כאשר כל בלוק הוא 3 ביטים האם יש שגיאה בבלוק?

0	0	1	0
1	0	1	0
1	0	1	0
1	0	1	0
1	0	1	1

ישנה שגיאה. מכיוון שהбитים הבולטים לא נכונים, לכן ישנה טעות בבייט [2][0]. ניתן לתקן את השדר כך שנשנה את הביט ל - 0.

לשדר זה הראה הוספה two dimension parity check כרך שכל בלוק הוא 3 ביט

נכטב ע"י שמעון ארזואן, גיא פלג, רון רוזנפלד

0	0	1	1
0	1	0	1
1	0	1	0
0	1	0	1
1	0	1	0
0	1	1	0
<hr/>			1
0	1	0	1

6) ישנו 2 hostים A ו- B שמחוברים ב link אחד:

- link זה יכול להעביר R ביט בשניה ( $R \left[ \frac{bit}{sec} \right]$ )
  - אורך הلينק הוא  $d$ . ( $d[m]$ )
  - הזמן שלוקח לשדר לעבר מרחוק זה הוא  $\left[ \frac{m}{sec} \right]$
- א. A שולח מעטפה ל B בגודל  $s$  ביט מה ההשניה שיש ברשות?

$$T_f = \frac{s}{R}$$

$$t_p = \frac{d}{p}$$

כמה זמן לוקח לשדר לעבר. (זמן שלוקח לעבר בתוך הリンק)

$t_p$  הוא ההשניה שיש ברשות. מכיוון בזמן שלוקח לשדר לעבר לשדר את המ�פה לא יכול לשוח נטענים\מעטפות לכך זה זמן ההשניה.

הערה: הזמן שלוקח לשוח בפועל את המ�פה הוא זמן שציריך לטעון את המ�פה לlienk + זמן העברת השולחן בlienk.

ב) אם A מתחילה לשדר ב  $T=0$  מה קורה למעטפה ב  $t_f$

בזמן  $t_f$  host A שם את הביט האחרון בlienk.

ג) נניח ש  $t_p < t_f$  מה קורה בזמן  $T=t_f$

הbeit האחרון יוצא מ-A. ו B ממשיר לקבל ביטים.

7) מה המשמעות של פרוטוקול ?CSMA

פרוטוקול מבצע האזנות לרשות האם הרשות פנוייה והוא מעוניין לשדר אזי הוא משדר.

כתב ע"י שמעון ארזיאן, גיא פלג, רון רוזנפלד

.8

א. מחשבים ברשת אלחוטית שלחמים מסגורות בנות 8000 בית, בממוצע כל 200 שניות. קצב השידור הוא 100 Kbps.

המחשבים משתמשים ב프וטוקול ALOHA להחלטה על מועד השידור. מהו מספר המחשבים ברשת, לכל היותר?

ב. נסעה בין ALOHA לשל Slotted Pure ALOHA כאשר העומס ברשת נזוק (כלומר למחשבים אין הרבה מה לשדר).

באיזה פרוטוקול יהיה השימוש מרצע בו מסגרת מוכנה לשידור עד לרגע בו תשודר בהצלחה, בממוצע, קטן יותר?

א. ראשית נבין כי בפרוטוקול זה נחשב את קצב השידור האפקטיבי לפי הניצילות של הפרוטוקול. במקרה הנוכחי ייקח את הניצילות של Pure Aloha שהוא 0.184.

$$\text{מגן רוחב הפס הוא: } 100Kbps * 0.184 = 18.4Kbps$$

$$\text{נחשב בממוצע את גודל החבילה שמחשב שלוח: } \frac{8000bit}{200Sec} = 40bit/\text{Sec}$$

$$\text{לכן סך המחשבים ברשת יהיה } \frac{18.4 * 10^3}{40} = 460$$

ב. ב – PURE ALOHA השידור יכול להתחילה מיד. בקצב נמוך לא תהיה התנגשות וסביר להניח שהשידור יצילח.

לעומת זאת עם SLOTTED ALOHA הוא יצטרך לחכות ל-SLOT

הבא וכן ההתחילה שלו תהיה ב-Delay של חצייה מה – SLOT TIME. וכך.

הפרוטוקול עם הממוצע הקטן ביותר יהיה של PURE ALOHA.

כתב ע"י שמעון ארזיאן, גיא פלאג, רון רוזנפלד

## **Wireless Network**

(1) האם CDMA משתמש בשיטת random accses בשכבה הילינק.

לא, מכיוון שהוא שולח כל הזמן, אין לו תלות בהשזהים אחרים.

## :Network Security

.1 RSA. נתן

$$q = 7, p = 5$$

נרצה להצפין את המילה Bob. ( אסן 98-111-66) כיצד נעשה זאת?

בשיטת RSA ראשית נחשב:

$$n = q \cdot p = 5 \cdot 7 = 35$$

$$z = (p - 1)(q - 1) = 6 \cdot 4 = 24$$

נבחר  $e$  כך ש- $e$  ובנוסף אין לו מחלק משותף עם  $-z$  (24). נבחר  $e=5$

נבחר  $d$  כך  $d \cdot e \equiv 1 \pmod{24}$ , כלומר שקיימים את המשוואה:

$$5d \equiv 1 \pmod{24}$$

קל לראות ש- $d=5$  מתאים לנו. מפסרים את  $e$  ואת  $d$ .

נצפין את 66-111-98

$$c = m^e \pmod{n}$$

$$c = 6^5 \pmod{35} = 6$$

$$c = 1^5 \pmod{35} = 1$$

$$c = 9^5 \pmod{35} = 4$$

$$c = 8^5 \pmod{35} = 8$$

לכן המסר יצא:

לאחר הצפנה 66-111-48

מקורי 66-111-98

המשתמש רוצה לפענוח:

בעזרת המפתח הפרט נצליח לעונח את השדר לפי הנוסחה הבאה:

$$m = c^d \pmod{n}$$

$$c = 6^5 \pmod{35} = 6$$

$$c = 1^5 \pmod{35} = 1$$

נכתב ע"י שמעון ארזיאן, גיא פלג, רון רוזנפלד

$$c = 4^5 \bmod 35 = 9$$

$$c = 8^5 \bmod 35 = 8$$

2. נניח ש  $N$  משתמשים רוצים לדבר כל אחד עם כל אחד כמה מפתחות סימטריים הם צריכים?

כל זוג צריך שייהי לו מפתח משותף ולכן נדרש מספר מפתחו כמספר הזוגות השונים ובסך הכל:  $\binom{n}{2}$   
ויר הכל  $\frac{n(n-1)}{2}$

3. אם רוצים לעשות אותו דבר באמצעות מפתחות פומביים?

כל משתמש צריך ליצור מפתח פומבי + פרטי קלומר N.

4. בוב ואליוס מדברים ביניהם באמצעות מפתח סימטרי

- a. אליס מקבלת ההודעה מוצפנת מצילחה לפענה אותה, האם היא יכולה להיות בטוחה שבוב הצפין את ההודעה? כן כי רק שבוב יש את המפתח הסימטרי.
- b. אליס רוצה לשכנע את אמא שלה שהיא קיבלה את ההודעה מבוב? לא כי טרור יכול לשלווח שבוב את ההודעה שבוב שלח אחרון (PlayBack Attack). בនוסף גם אליס יכולה ליזור את ההודעה.

5. בשיטת הצפנה "התמרת אוטיות" כמה אפשרויות יש להצפן א"ב שמורכב מ<sup>26!</sup> אותיות ? !<sup>26</sup>.

6. אם טרורדי ידעת את המיפוי של 7 מהאותיות של הא"ב בכמה קטן מרחיב האפשרויות שלה?

נשארו 19 אותיות לא מפוענחות ולכן נשאר לה !<sup>19!</sup> אפשרויות לפיענוח. אז היא צמיצה ב-

7. איזה בעיה פוטר השימוש בHASH?

הבעיה שפותר HASH היא האמינות (Integrity) כי ה HASH מוסיף חתימה דיגיטלית. ככלומר נוכל לדעת בוודאות מי יצר את ההודעה.

8. מה המטרה של noun , בauthentication פרוטוקול?

נווד לזמן האימות בלבד כיון שהוא חד פומי ולכן לא ניתן להשתמש בו שוב ולהתחזות אליו. playback attack

9. על מה מתבסס הפרוטוקול RSA?

2 מספרים ראשוניים מאוד גדולים שאפשר לענה אותם. חשבו מודלי.

10. מה זה (CA) certification authority ?

זהו הסמכתם האחראית על אמינות המפתחות הפומביים. בשורת שלהם ישבים כל המפתחות הפומביים.

כתב ע"י שמעון ארזיאן, גיא פלאג, רון רוזנפלד

**11. למה יש כל כך הרבה שלבים בDES?**

כדי שהפלט שלו יהיה קרוב כמה שיותר לרנדומלי. בנוסף ללמידה מסוימת בכל פעם הקידוד יהיה בעל פלט שונה.

**12. האם שיטות ההצפנה יכולות להיות לא בשכבת האפליקציה?**

תשובה של אלכס: כל תוכנה רוצה להצפין את המידע לצריכה לדאוג להצפנה בשכבה האפליקציה. אם היינו ממשים את האבטחה בשכבה הפיזית אז כל המידע שהוא י יצא מ"המכונת" החזאת היה מאובטח החסכנות הם שהרכיבים הפיזיים יצרכו לבצע שימוש במפתחות. גם אם היינו עושים את ההצפנה בשכבת התקשרות היינו צריכים לשנות את המבנה של עבודה הריאוטרים.  
תשובה שלנו: שיטות ההצפנה יכולות להיות גם בשכבות האחרות, לדוגמה: בשכבת הרשת פרוטוקול *OSPF* משתמש בהצפנה של *Hash* עם *MDS*.

נכתב ע"י שמעון ארזואן, גיא פלאג, רון רוזנפלד

## שאלות מספר הלימוד computer networking top down approach

### חלק א -

1. מה הבדל בין הוסט לבין תחנת קצה? מנה סוגים שונים של תחנות קצה? האם שרת WEB הוא תחנת קצה?

אין הבדל ביניהם, שני המושגים מתארים מכשירים שמחוברים לרשת האינטרנט ומקבלים ממנה שירותים. דוגמאות לתחנות קצה: מחשבים, טלפונים סולריים, קונסולות משחקים, Webcam וכו'. כן, שרת אינטרנט הוא אכן תחנת קצה.

2. מהו *client* ומהו *server*. האם *server* מבקש וambil שירותים מ-*client*.

תוכנת ליקוי היא תוכנה שלוחב יוצרת קשר עם שרת (אך לא בהכרח) ומנהלת אותו יחסיו גומלין של העברת מידע. תוכנת שרת היא אותו דבר רק שתפקידה הוא בעיקר לספק ללקוח שירותים (ז"א מידע עפ"י בקשה שתוכנת הלקוח מבקשת. העיקרי תפקידי של השרת הוא לקבל בבקשת פרטיהם מה לקוח ולבסוף השירותים ללקוח, אך ניקח לדוגמא מקרים מסוימים כמו למשל בבקשת פרטיים מסוימים אצל לקוח *cookie* אשר אפשר להגדיר שהשרת בิกש ממנו מהקווק.

List six access technologies. Classify each one as residential access, company access, or mobile access.

3. מנה שיש טכנולוגיות גישה. סוויג אותן.  
Namna camha tecnologiyot gisha shebamatzuton ish gisha lainternet (Access  
Residential access Dial Up : Technologies DSL ז"א gisha mahavit, DSL גם Company Residential Access גם FTTTH, Residential Cable, Residential Mobile Access Wi-Fi ,Access

4. מהו קצב השידור של LAN ? Ethernet LAN? עברו קצב תעבורת נתון, האם כל שימושים ברשת יכולים להעביר מידע באופן רציף בנסיבות זו?

קצב השידור של Ethernet הוא בדר"כ  $100Mbps$  ויכול לקבל קצבים של  $1Gbps$  או  $10Gbps$ . העניין של התנוגויות הוא הקשר לטופולוגיה מכיוון שם יש לנו Star Topology שמחובר עם HUB אז בהחלט יתכו התנוגויות.

5. מנה כמה מידות פיזיות של Ethernet יכול לרחוץ עליה? זוג מוליכי נחושת, כבל קוואקסיאלי, אופטי, גלי רדיו.

כתב ע"י שמעון ארזאון, גיא פלאג, רון רוזנפלד

6. א) מה היתרון של *packet-switched* על פני *circuit-switched* ברשת?

ב) מה היתרון של *TDM* על *FDM* ב-*circuit-switched* ?

א) ב *circuit* רוחב הפס הוא שמור, ככלומר לא יתכן שהוא יהיה תפוס ע"י משתמש אחר וולך כל הודעה תגעה בשולמתה ברצף בתחום של רוחב הפס. ב *packet* רוחב הפס לא שמור ולכן ניתן שחייבות ייחכו אם הרוחב פס תפוס עד אשר יתפנה מקום אז הם ישלו.

ב) ב- *TDM* כאשר משתמש מתקבל את זמן השיליחה הוא ישלח בכל רוחב הפס. لكن שליחת הודעה יכול להיות מהיר יותר מאשר *FDM* אשר משתמש רק בחלק יחסית מרוחב הפס.

7. מדובר נאמר כי *Packet-Switching* משתמש בירוב סטטיסטי. השווה בין ריבוב סטטיסטי לזה שנעשה ב- *TDM*.

ב *packet switching* אנחנו מזמנים חיבור בכל פעם שיש צורך שיש בקשה, ככלומר כשיש משתמשים שוחצים לשולחabilities איז החיבור היה משותף עבורם במקביל ונעביר חבילת אחריה חבילת ולכן זה דומה לשיטוף משאבים לדוגמה אם משתמש מסוים שולח פתאום  $1000 \text{ bps}$  בזמן שאחרים לא שולחים כלום אז הוא יוכל ללקחת את כל הרוחב פס ולהעביר את החבילת מהר, לעומת זאת ב- *TDM* עם  $10 \text{ slots}$  בכל מסגרת וכל אחד מהם ברוחב  $10 \text{ bps}$  אז אם משתמש אחד ישלח 1000הוא לא יוכל ללקחת את כל הסלוטים כי יוספק לו סלוט אחד מכל מסגרת ולכム למראות ש 9 סלוטים לא יהיו מנוצלים הוא ישמש באותו רוחב פס.

נכתב ע"י שמעון ארזיאן, גיא פלג, רון רוזנפלד

## רשתות סיכון לבחן מעבדה

### Lab1:

#### עבודה על הרשות ב shell של windows

Run->cmd->nestsh

הגדרת כתובת IP עבור интерface

set address name="<Interface name>" source=static address=<IP address>  
mask=<network mask>

.Local area connection интерface הוא הרשות  
לדוגמא:

set address name=" Local Area Connection" source=dhcp  
set address name="Local Area Connection" source=static address=192.168.1.10  
mask=255.255.255.0

```
C:\Users\Guy>netsh
netsh>interface
netsh interface>ipv4
netsh interface ipv4>set address name="Local area connection" source=static address=192.168.1.10 mask 255.255.255.0
```

הגדרת DNS:

setdnsservers "<Interface name>" static <DNS IP address> primary

לדוגמא:

```
netsh>interface
netsh interface>ipv4
netsh interface ipv4>set dnsservers "Local Area Connection" static 192.168.222.254 primary
```

#### הבנת נתונים מ-IP CONFIG

משמעות	מה רשום
שם וכותרת הרשות ( connection )	שם וכותרת הרשות ( connection )
MAC Address	(physical address)
קבוע אם הכתובות מחולקות אוטומטית או לא Yes = automatic No – manually / static ip	DHCP enabled

הצגת הכתובות של המחשב מחובר אליו.

```
C:\Users\Guy>netsh
netsh>interface
netsh interface>show interface
Admin State      State        Type          Interface Name
-----          -----        -----        -----
Enabled          Connected    Dedicated    Wireless Network Connection
```

**אם לדוגמא ישאלו אותנו:**

**Q:**You need to set IP address 172.16.102.6, network mask 255.255.0.0, default gateway 172.16.102.1 and DNS server 62.38.238.8 for “Local Area Connection” network interface on a computer. Write a command netsh for setting up this address.

**A:**netsh>interface  
netsh interface>ipv4  
netsh interface ipv4> set address name="Local Area Connection" source=static  
address=172.16.102.6 mask=255.255.0.0 gateway 172.16.102.1  
netsh interface ipv4>set dnsservers "Local Area Connection" static 62.38.238.8 primary

איך בודקים איך מתאפשר כתובות ה-ip:



**Q:**How IP address is assigned to network interface, and how do you determine what it is.

**A:**The ip assigned automatic, because the “DHCP Enabled” is yes.

## Lab2:

אם שולחים packet במצב סימולציה. נראה מעתפה. בלחיצה כפולה על המעתפה יפתח מסך עם לשוניות.

Osi model: מציג את שכבות ההודעה.

Outbound pdu:

מכל את המעתפה של ethernet.

Dest mac –

כתובת mac אליה מיועדת המעתפה. לדוגמא:

FFFF.FFFF.FFFF – אומר שידור, (broadcasting) עדין לא יודעים למי מ יודעת המעתפה לכז  
היא תלו לcoliן עד אשר נמצא target ip שווה. (ע"מ 11 בPDF).

Target mac –

הכתובת הפיזית אליה אנו אמורים לשלוח את הפקטה. אם היא 0 או עדין לא יודעים.  
OPCODE:

סוג הפקודה שיש לביצוע:

1 – outgoing packet

2- reply

בפקודה בcmd הפקודה a – arp

תיתן לנו את טבלת החסרים המוכרים. דוגמא:

**Command Prompt**

```

Packet Tracer PC Command Line 1.0
PC>ping 192.168.1.23

Pinging 192.168.1.23 with 32 bytes of data:

Reply from 192.168.1.23: bytes=32 time=19ms TTL=128
Reply from 192.168.1.23: bytes=32 time=8ms TTL=128
Reply from 192.168.1.23: bytes=32 time=9ms TTL=128
Reply from 192.168.1.23: bytes=32 time=9ms TTL=128

Ping statistics for 192.168.1.23:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 8ms, Maximum = 19ms, Average = 11ms

PC>arp -a
      Internet Address          Physical Address          Type
      192.168.1.23              0005.5e3b.1157        dynamic

PC>

```

הבדל בשליית פקודות בין hub ל switch.

Hub- עובד בשכבה 1 בלבד. תמיד שולח לכלום חוץ ממי שהוא קיבל.  
 Switch- עובד בשכבה 2 , אם זה שידור(dest mac = **FFFF.FFFF.FFFF**) הוא שולח לכלום ,  
 אחרת שולח לhost ספציפי.

## Lab 3:

IP –

	Start Address	End Address	Max of Networks	Max of Hosts
Class A:	10.0.0.0	126.255.255.255	$128 = 2^7$	$16\ 777\ 215 = 2^{24} - 2$
Class B:	128.0.0.0	191.255.255.255	$16\ 284 = 2^{14}$	$65\ 534 = 2^{16} - 2$
Class C:	192.0.0.0	223.255.255.255	$2\ 097\ 152 = 2^{21}$	$254 = 2^8 - 2$
Class D:	224.0.0.0	239.255.255.255		

ע"נ lab2 – pdf 11 ע

How many bits are required to extend the network prefix =  $2^7 \approx \# \text{ of networks}$

הסביר ניסף להקצת כתובות ip (host'ים ורשתות) מפורט בפתרון מעבדה 3.  
שאלה לדוגמא:

Network interface has IP address 202.84.45.14, this is class full address.

**Q:**What is the IP address class for this address?

**A:** Class C

**Q:**What is the Network mask for this class?

**A:**255.255.255.0

**Q:**What is the Network ID for this IP address?

**A:**202.84.45.0

**Q:**What is the Host ID for this IP address?

**A:**0.0.0.14

**Q:**Write the IP address 212.89.56.34 and the network mask 255.255.255.128 in CIDR notation.

**A:**CIDR notation: 212.89.56.34\25

טבלת פקודות לעבודה עם CLI בסימולציה של Cisco

Command (with prompt)	Entered EXEC Mode	Mode Prompt
R-Site-A#disable	User Exec Mode	R-Site-A>
R-Site-A>enable	Privileged EXEC Mode	R-Site-A#
R-Site-A#configure terminal	Global Configuration Mode	R-Site-A(config)#
R-Site-A(config)# interface GigabitEthernet 0/0	Interface Configuration Mode	R-Site-A (config-if)#
R-Site-A (config-if)#exit	Back to "Global Configuration Mode"	R-Site-A(config)#
R-Site-A(config)#router rip	Router Configuration Mode	Router(config-router)#
Router(config-router)#exit	Back to "Global Configuration Mode"	R-Site-A(config)#
R-Site-A(config)#exit	Back to "Privilege Exec Mode"	R-Site-A#
R-Site-A#disable	Back to "User Exec Mode"	R-Site-A>

## Lab 4:

**אם נרצה לשנות שם רואוטר:**

```
Router1>enable  
Router1#configure terminal  
Enter configuration commands, one per line. End with CNTL/Z.  
Router1(config)#hostname R-Site-A  
R-Site-A(config)#{
```

**אם נרצה להגדיר IP ל interface של router:**

```
R-Site-A(config)#interface Gig 0/0 // enter interface config mode  
R-Site-A(config-if)#ip address 192.168.1.1 255.255.255.0 //give the interface ip  
R-Site-A(config-if)#description Network-A //give it name  
R-Site-A(config-if)#no shutdown //start working
```

**אם נרצה לדעת האם router הוא של interface DCE או DTE:**

```
R-Site-A>show controllers Se 9/0  
Interface Serial9/0  
Hardware is PowerQUICC MPC860
```

**DTE V.35 clocks stopped.**

The Option of the router: DTE – Data terminal Equipment

יש להגדיר שעון-> DCE- Data Communication Equipment

**הגדרת מושך בין רואוטרים :**

```
InternetGateway(config)#interface se 9/0  
InternetGateway(config-if)#ip address 192.168.4.6 255.255.255.252 //ip of interface  
InternetGateway(config-if)#clock rate 56000 // if DCE else no need  
InternetGateway(config-if)#no shutdown
```

**הערה : כדי לבדוק שאכן הגדרנו הכל :**

```
R-Site-A#show ip interface brief
```

**על מנת להראות את כל החיבורים ל interface וראות כי השעון אכן מוגדר:**

```
InternetGateway>enable  
InternetGateway# show running-config
```

**שאלה לדוגמא:**

**Q:**You are working on router in the Interface Configuration Mode (router(config-if)#) and you need to launch the command *show iproute*, that is working in the Privilege Exec mode. What are the commands that you need to run to launch this command *show iproute*.

**A:**router (config-if)# exit  
router (config)#exit  
router# *show iproute*

**Q:**You are working in the User Exec Mode (router>) and you need to work in the Privilege Exec Mode (router#) with Privilege Level 5. What is the command that you need to run to enter the Privilege Exec Mode with Privilege Level 5?

**A:**router> enable 5  
router #

## Lab 5:

הגדרת routing בין רשתות זרות (לא מחוברות ישירות):

<i>ip route</i>	<i>remote_network_ID</i> 192.168.3.0	<i>network_mask</i> 255.255.255.0	<i>gateway_IP</i> 192.168.101.2
שם הפקודה	כתובת IP של הרשת <b>network</b> המורוקה ( ) <b>id</b> לאחר (masking)	מס' רשת	כתובת interface של הנטב המחבר הקרוב אליו. (בדרך כלל (serial

דוגמא:

```
R-Site-A>enable
R-Site-A#configure terminal
R-Site-A(config)#ip route 192.168.3.0 255.255.255.0 192.168.101.2
```

הערה:

כדי לבטל route علينا לרשום את הפקודה הבא:

```
ip route remote_network_ID network_mask
```

אם נרצה להגדיר קשר בין רשתות ובצע את הדבר הבא:

כפי שראינו לעילנו לרשום את הפקודה : ip route

- כתובות ה- ip הר馮ון שנכתוב הוא ה- network id של הרשת עליה נרצה להתחבר, ip זה יהיה ה- interface שמננו יוצא החיבור לרשת.
- כתובות ה- ip השני יהיה ה- network mask של הרשת שאליה נרצה להתחבר.
- כתובות ה- ip השלישי יהיה כתובות ה- interface שאליו מחבר ה- router של הרשת שלנו.

כדי להגדיר routing דיפולטי علينا להשתמש בפקודה הבא:

```
ip route 0.0.0.0 0.0.0.0 Serial x/x
```

כאשר x/x הוא מספר המנשך.

דוגמא:

```
R-Site-A(config)#ip route 0.0.0.0 0.0.0.0 Se9/0
```

להגדירה דינמית של routing :

```
R-Site-A>enable
R-Site-A#configure terminal
InternetGateway(config)#router rip
```

וכעת מגדירים בעזרה הפקודה

```
Network connected_network_id
      . .
      .
R-Site-A(config)#router rip
R-Site-A(config-router)#network 192.168.101.0
R-Site-A(config-router)#network 192.168.1.0
R-Site-A(config-router)#network 192.168.2.0
```

שאלה לדוגמא:

**Q:**You are configuring a router and need to set the IP address 192.168.23.254/24 (CIDR notation). You are working in the Privilege Exec Mode (router#). What are the commands that you need to run to set this IP address to interface GigabitEthernet 0/1 and bring this interface to status *up* from status *down*?

```
A:router >enable
router #configure terminal
router (config)#interface GigabitEthernet 0/1
router (config-if)#ip address 192.168.23.254 255.255.255.0
router (config-if)#no shutdown
```

## Lab 6:

.access-list standard יצירת

```
R-Braude>enable
R-Braude#configure terminal
R-Braude(config)#access-list y deny deny_ip reverse_mask
.....
R-Braude(config)#access-list y permit any
R-Braude(config)#interface gig x/x
R-Braude(config-if)#ip access-group y in\out
```

שייר לאחד היפנים את הרשימה:

Y = 1....99

ולגמאנ:

```
R-Braude>enable
R-Braude#configure terminal
R-Braude(config)#access-list 1 deny 192.114.40.11 0.0.0.0
R-Braude(config)#access-list 1 permit any
R-Braude(config)#interface gig0/0
R-Braude(config-if)#ip access-group 1 in
R-Site-A(config-if)#no ip access-group y out
R-Braude(config) no ip access- list y out
```

מחיקת שייר של access list

Router(config)#	מצב הנוכחי
access-list <access-list-number>	מספר הרשימה נוע בין 100 ל 1997
<action>	Permit/deny
<protocol>	tcp, udp, ip or icmp
<source>	Ip source
<s-port>	Optional for tcp or udp else omitted
<destination>	Dest ip
<d-port>	Optional for tcp or udp else omitted

כתב עי' : שמעון ארזיאן & ניר פרג

לציין חסימת ה portים ניתן להשתמש ב:

- lt n* – All port numbers less than n
- gt n* – All port numbers greater than n
- eq n* – Port n
- neq n* – All ports except for n
- range n m* – All ports from n through m, inclusive

דוגמה:

```
R-Braude(config)#access-list 111 permit tcp any host 192.114.40.9 eq 80
R-Braude(config)#access-list 111 permit tcp any host 192.114.40.10 eq ftp
R-Braude(config)#access-list 111 permit tcp any host 192.114.40.10 range 1030 1175
R-Braude(config)#access-list 111 permit icmp any any
R-Braude(config)#access-list 111 deny tcp any any
R-Braude(config)#interface gig0/0
R-Braude(config-if)#ip access-group 111 out
```

הערות:

*192.168.23.254/24* זה של כמות הביטים של המסיכה CIDR notation (1