# תקשורת נתונים - שיעור 11

## נושא 1: ארכיטקטורות יישומים ברשת - שרת-לקוח מול Peer-to-Peer

**הסבר מקוצר: השוואה בין שני המודלים המרכזיים לתקשורת אפליקטיבית ברשת: מודל ריכוזי (Client-Server) בו שרת מספק שירותים ללקוחות, מול מודל מבוזר (Peer-to-Peer) בו כל תחנה פועלת באופן עצמאי וחולקת מידע ישירות עם תחנות אחרות.**

**הסבר מורחב: במודל שרת-לקוח (Client-Server), קיים שרת מרכזי שחייב להיות זמין ופעיל באופן תמידי. לשרת זה יש כתובת IP קבועה והוא מאזין לבקשות בפורטים ידועים מראש. הלקוח (Client) הוא זה שיוזם את הקשר אל השרת, וכתובת ה-IP שלו יכולה להיות דינמית (להשתנות בהתאם למיקומו). הלקוחות אינם מתקשרים ישירות זה עם זה, אלא כל התקשורת עוברת דרך השרת המרכזי. לעומת זאת, במודל Peer-to-Peer (נקודה לנקודה), אין שרת מרכזי או לקוח מוגדר. כל תחנה (Peer) יכולה לתת שירותים (לתפקד כשרת) ולקבל שירותים (לתפקד כלקוח) בו זמנית. התקשורת מתבצעת ישירות בין התחנות, ואין גורם ניהולי מרכזי, כתובות IP קבועות או פורטים מחייבים.**

**משפטי מפתח והגדרות למבחן:**

- בארכיטקטורת Client-Server רק השרת משתף נתונים ושירותים, בעוד שהלקוח מקבל אותם.
- שרת מאופיין בזמינות תמידית (Always on), כתובת IP קבועה, והאזנה לפורט ספציפי.
- בארכיטקטורת Peer-to-Peer כל תחנה משמשת במקביל גם כשרת וגם כלקוח (מורידה ומשתפת נתונים).
- במודל שרת-לקוח לעולם אין קשר ישיר בין שני לקוחות (לדוגמה, שיחת ווטסאפ או זום עוברת דרך השרת של החברה).

**דוגמאות: מערכת כמו Google Drive, אתר חדשות או פגישת זום עובדים במודל שרת-לקוח. הורדת קבצים דרך טורנטים (Torrents) או תוכנות שיתופיות ישנות (כמו קאזה), כמו גם כריית מטבעות וירטואליים, עובדות במודל Peer-to-Peer.**

**שאלות הבנה: 1. מדוע מחשב לקוח (Client) יכול להסתפק בכתובת IP דינמית במודל שרת-לקוח, בעוד שהשרת זקוק לכתובת קבועה?**

2. מהו החיסרון המרכזי במודל שרת-לקוח כאשר יש תקלה בשרת המרכזי, וכיצד מודל Peer-to-Peer מתגבר על חיסרון זה?
3. כאשר משתמש מעלה קובץ מהמחשב שלו ל-Google Drive, האם הוא פועל בארכיטקטורת Peer-to-Peer? הסבר.

## נושא 2: כתובות MAC, כתובות IP וניתוב חבילות מידע

**הסבר מקוצר: אופן תנועת חבילות מידע (Packets) ברשת, תוך הבחנה בין תפקיד כתובת ה-IP (זיהוי מקצה לקצה) לבין תפקיד כתובת ה-MAC (זיהוי בתוך הרשת המקומית בלבד).**

**הסבר מורחב: כאשר מחשב מעוניין לשלוח חבילת מידע ליעד שנמצא מחוץ לרשת המקומית שלו (למשל, שרת של גוגל), הוא אינו מכיר את כתובת ה-MAC של היעד הסופי. לכן, הוא מציב בשדה כתובת ה-IP את כתובת היעד הסופית (שנשארת קבועה לאורך כל המסלול מקצה לקצה), אך בשדה כתובת ה-MAC ביעד הוא מציב את כתובת ה-MAC של הראוטר המקומי (Default Gateway). הראוטר מקבל את החבילה, קורא את כתובת ה-IP, מזהה שהיעד אינו אצלו, ואז פותח את שכבת ה-Data Link (שכבה 2) ועוטף את החבילה מחדש. בעטיפה החדשה, הראוטר רושם את כתובת ה-MAC של הממשק שלו כמקור, ואת כתובת ה-MAC של התחנה הבאה (הראוטר הבא) כיעד. תהליך זה של "פתיחה ועטיפה מחדש" חוזר על עצמו בכל קפיצה (Hop) עד להגעה ליעד. בתוך שכבת ה-IP קיים שדה בשם TTL (Time To Live) המוגדר למספר מקסימלי של קפיצות (לרוב 30), ויורד ב-1 בכל ראוטר כדי למנוע מחבילות להסתובב לנצח ברשת.**

### משפטי מפתח והגדרות למבחן:

- כתובות IP אחראיות על ניתוב מקצה לקצה (Source to Destination) ונשארות קבועות לאורך המסלול (למעט במקרי NAT).
- כתובות MAC חיות ורלוונטיות אך ורק בתוך הרשת המקומית (LAN) ומתחלפות בכל קפיצה מראוטר לראוטר.
- הראוטר מקלף את 3 השכבות התחתונות, קורא את היעד הלוגי, ועוטף מחדש עם כתובות MAC מעודכנות לשלב הבא במסלול.
- שדה TTL (Time To Live) מונע לולאות אינסופיות ברשת על ידי ספירה לאחור של כמות הקפיצות המותרת.

**דוגמאות: שימוש בפקודת Ping לכתובת 8.8.8.8. החבילה יוצאת מהמחשב עם MAC יעד של הראוטר המקומי. במסך ה-Trace ניתן לראות שהחבילה עוברת כ-8 ראוטרים (קפיצות) בדרך, עד שהיא מגיעה ליעד, כשבכל שלב כתובות ה-MAC מוחלפות לטובת הקפיצה הבאה.**

**שאלות הבנה: 1. מחשב A שכתובתו 111.111.111.111 שולח מידע למחשב B שכתובתו 222.222.222.222. מה תהיה כתובת ה-MAC של היעד בעת יציאת החבילה ממחשב A?**

2. מדוע לא משתמשים בכתובת ה-MAC של השרת של גוגל ישירות כששולחים אליו חבילת מידע?
3. הסבר את המשמעות של הערך TTL ב-Header של חבילת IPv4 ומה קורה כאשר הוא מגיע לאפס.

# נושא 3: שכבת הייצוג - קידוד (Encoding) מול הצפנה (Encryption)

**הסבר מקוצר: שכבה 6 במודל ה-OSI מטפלת בייצוג הנתונים, וכוללת המרה של מידע שפתי לפורמט דיגיטלי (קידוד) וכן הסתרה של המידע למטרות אבטחה ופרטיות (הצפנה).**

**הסבר מורחב: קידוד (Encoding) הוא תהליך של המרת נתונים, אותיות או תמונות לביטים כדי שמחשבים יוכלו לשדר ולעבד אותם. הקידוד המוכר ביותר הוא קוד ASCII, שבו מוקצה בייט אחד (8 ביטים) לכל תו. מכיוון שישנם 8 ביטים, כמות הקומבינציות האפשריות היא 2 בחזקת 8, כלומר 256 תווים שונים. קידוד אינו נועד להסתיר מידע.**

**הצפנה (Encryption)** נועדה לשמור על סודיות המידע על ידי הפיכתו ל"ג'יבריש" באמצעות אלגוריתם, כך שרק מי שמחזיק במפתח יוכל לפענח אותו.
ישנם שני סוגים מרכזיים:

1. **הצפנה סימטרית:** שימוש באותו מפתח גם להצפנה וגם לפענוח. פעולה זו מהירה מאוד וצורכת פחות משאבים, אך יוצרת בעיה של העברת המפתח בסודיות בין הצדדים ללא ציתות.
2. **הצפנה א-סימטרית:** שימוש בזוג מפתחות - מפתח ציבורי (שמחולק לכולם) ומפתח פרטי (שמור בסוד). מידע שמוצפן עם המפתח הציבורי יכול להיות מפוענח אך ורק עם המפתח הפרטי, ולהפך. אלגוריתם זה מורכב ואיטי יותר, אך פותר את בעיית העברת המפתח המאובטחת. לרוב משתמשים בו בתחילת התקשורת כדי להעביר בבטחה מפתח סימטרי

להמשך השיחה (כמו בפרוטוקול HTTPS/SSL).

**משפטי מפתח והגדרות למבחן:**

- קידוד משנה את צורת הנתונים לפורמט דיגיטלי ללא מטרה אבטחתית, בעוד שהצפנה הופכת את המידע לחסר משמעות ללא מפתח הפענוח.
- גודל טבלת ASCII הבסיסית הוא 256 אפשרויות מכיוון שכל תו מיוצג על ידי בייט אחד (8 ביטים).
- במפתח סימטרי, אותו מפתח משמש להצפנה ולפענוח.
- במפתח א-סימטרי קיים קשר מתמטי בין המפתח הציבורי לפרטי: מה שמוצפן באחד מפוענח בשני.

**דוגמאות: קוד ASCII הממיר אותיות למספרים בין 0 ל-255 הוא דוגמה לקידוד. כניסה לאתר מאובטח (HTTPS) המציג מנעול בדפדפן, בו נעשה שימוש במפתח הציבורי של השרת כדי להצפין נתונים, היא דוגמה להצפנה א-סימטרית.**

**שאלות הבנה: 1. האם קוד ASCII יכול להכיל 300 תווים שונים? הסבר מתמטית מדוע כן או לא.**

2. מהו צוואר הבקבוק המרכזי בשימוש בהצפנה סימטרית בלבד בתקשורת מול גורם חדש (למשל, בנק)?
3. אדם רוצה לשלוח הודעה סודית לחברו תוך שימוש בהצפנה א-סימטרית. באזה מפתח (ציבורי או פרטי) עליו להשתמש כדי להצפין את ההודעה, ובאיזה מפתח חברו ישתמש כדי לפענח אותה?

# נושא 4: ארכיטקטורות רשת קווית (טופולוגיות פיזיות בשכבת ה-Data Link)

**הסבר מקוצר: ארבע השיטות המרכזיות בהן מחברים פיזית רכיבים (Hosts) ברשת קווית: כוכב, טבעת, חיבור מלא וקו.**

**הסבר מורחב: * כוכב (Star): כל המחשבים מחוברים לצומת מרכזי אחד (Switch או Hub). זוהי הארכיטקטורה הנפוצה ביותר כיום. הוספת מחשב פשוטה מאוד (כבל אחד מהמחשב לסוויץ'). החיסרון: אם הרכיב המרכזי נופל, הרשת כולה קורסת. מקסימום קפיצות לתקשורת בין שני מחשבים הוא 2. נדרשים N כבלים עבור N מחשבים.**

- **טבעת (Ring):** כל מחשב מחובר לשני שכניו (מימינו ומשמאלו) ליצירת מעגל סגור. אין רכיב מרכזי בסגנון סוויץ'. נדרשים N כבלים. העברת מידע עוברת דרך המחשבים עד להגעתה ליעד.

- **חיבור מלא (Fully Connected):** כל מחשב מחובר בכבל ישיר לכל מחשב אחר ברשת. שיטה זו (המקבילה הפיזית ל-P2P מול כולם) מציעה אמינות ושרידות מושלמת שכן תמיד יש נתיב ישיר (קפיצה 1). החיסרון הוא עלות כלכלית ופיזית עצומה - נדרשים N*(N-1)/2 כבלים.
- **קו / עורק (Bus/Line):** כל המחשבים מחוברים באופן טורי על אותו כבל מרכזי משותף. יעילות התקשורת גרועה, וניתוק או קרע בכבל במקום אחד מפיל את המשך הרשת.

## משפטי מפתח והגדרות למבחן:

- טופולוגיית כוכב היא הסטנדרט המקובל ברשתות מודרניות מבוססות Switch.
- בטופולוגיית חיבור מלא, כדי לחשב את כמות הכבלים יש לכפול את כמות המחשבים בעצמה פחות 1, ולחלק ב-2.
- נקודת כשל יחידה (Single Point of Failure) בטופולוגיית כוכב היא הרכיב המרכזי.

## דוגמאות: ראוטר/סוויץ' ביתי שאליו מחוברים בכבלים המחשבים בבית משמש כרשת בתצורת כוכב (Star). מחשבי צבא או מערכות קריטיות עשויים להשתמש בחיבור מלא ליצירת אמינות מוחלטת ללא תלות ברכיב מתווך.

## שאלות הבנה: 1. במשרד ישנם 8 מחשבים. כמה כבלים יידרשו כדי לחבר אותם בתצורת חיבור מלא (Fully Connected)?

2. מדוע טופולוגיית כוכב הופכת את הרשת לרגישה יותר לתקלת חומרה מאשר טופולוגיית חיבור מלא?
3. מהו המספר המקסימלי של קפיצות (Hops) שנדרש כדי להעביר מידע בין שני מחשבים כלשהם ברשת כוכב המבוססת על Switch יחיד?

# נושא 5: שיטות גישה לערוץ התקשורת הפיזי (MAC Protocols)

## הסבר מקוצר: אופן ניהול הערוץ הפיזי והחלטה מי מורשה לשדר נתונים על הקו כדי למנוע ולטפל בהתנגשויות (Collisions), בעיקר בשכבת ה-Data Link.

## הסבר מורחב: קיימות שלוש משפחות מרכזיות של פרוטוקולי גישה לערוץ:

1. **חלוקת ערוצים (Channel Partitions):** לקיחת רוחב הפס הכולל וחלוקתו לקטעים קבועים (לפי זמן או תדר) עבור כל משתמש. אין התנגשויות כלל כי לכל מחשב "נתיב משלו". שיטה זו יעילה מאוד בעומסים גבוהים, אך מבזבזת משאבים אדירים בעומסים נמוכים (מחשב שאינו משדר עדיין "תופס" את חלקו בערוץ).
2. **גישה לפי תור (Taking Turns / Polling / Token):** מחשב פונה לאחרים לפי סדר (Polling) או

שאסימון (Token) עובר בין המחשבים ורק מי שמחזיק בו רשאי לשדר. מונע התנגשויות לחלוטין ויעיל ברוב העומסים, אך יוצר "תקורה" גדולה, זמן המתנה (למשל, העברת אסימון גם למי שאין לו מה לומר), ותלות ברכיב הניהול.

3. **גישה אקראית (Random Access - CSMA/CD):** שיטת העבודה של רשתות ה-Ethernet המודרניות הקוויות. הערוץ פתוח לכולם בכל עת. כשמחשב רוצה לשדר הוא **מאזין** לקו (במשך 9.6 מיקרושניות). אם הקו פנוי - הוא משדר. מכיוון שייתכן ששני מחשבים האזינו ושידרו יחד, קורית **התנגשות (Collision)**. הפרוטוקול יודע לזהות התנגשות: עם הזיהוי, שני המחשבים מפסיקים לשדר מיד, מגרילים מספר אקראי שמהווה את זמן ההמתנה שלהם, שלאחריו הם מאזינים שוב לקו ומנסים לשדר מחדש. יעיל מאוד בעומס נמוך, אך קורס מהתנגשויות מרובות בעומס גבוה.

## משפטי מפתח והגדרות למבחן:

- פרוטוקול CSMA/CD מורכב מ-4 שלבים: האזנה לקו, שידור אם פנוי, זיהוי התנגשות, המתנה לזמן אקראי ושידור חוזר.
- חלוקת ערוצים עדיפה מבחינת יעילות כשהרשת עמוסה במיוחד (מונעת התנגשויות לחלוטין).
- גישה אקראית מנצלת את כל רוחב הפס עבור משתמש יחיד כשהרשת ריקה (עומס נמוך).

## דוגמאות: חיבור כרטיס רשת קווי רגיל (Ethernet LAN) למחשב פועל בפרוטוקול CSMA/CD בגישה אקראית. פרוטוקול Bluetooth הפונה בסבב לאוזניות ולשעון פועל בשיטת גישה לפי תור (Polling).

## שאלות הבנה: 1. במצב של רשת רדומה (ללא תעבורה כלל), מדוע שיטת גישה אקראית (CSMA/CD) יעילה יותר משיטת חלוקת ערוצים?

2. מה מתרחש בפרוטוקול CSMA/CD מיד לאחר ששני מחשבים משדרים בו זמנית ונוצרת התנגשות?
3. ציין חיסרון אחד משמעותי של שיטת העברת אסימון (Token) בניהול תעבורת רשת.

# טבלה מסכמת של הנושאים

| נושא | הגדרות והרחבות מסכמות |
|---|---|
| **ארכיטקטורות יישומים ברשת** | **Client-Server:** שרת מרכזי עם IP קבוע מספק שירותים; תקשורת בין לקוחות עוברת דרכו. **Peer-to-Peer (P2P):** תקשורת ישירה ומבוזרת בין תחנות, ללא שרת מרכזי, בה כל צומת הוא גם לקוח וגם שרת. |
| **כתובות IP, MAC וניתוב** | **IP:** מזהה לוגי המשמש לניתוב מקצה לקצה ונשאר קבוע. **MAC:** מזהה פיזי המשמש לניתוב בתוך רשת ה-LAN בלבד ומתחלף בכל מעבר דרך ראוטר. **TTL:** שדה המגביל את מס' הראוטרים שחבילה יכולה לעבור (מונע לולאות). |
| **קידוד מול הצפנה (שכבת הייצוג)** | **קידוד (Encoding):** המרת נתונים לייצוג דיגיטלי (למשל ASCII: אות = בייט = 256 אפשרויות) ללא כוונת הסתרה. **הצפנה (Encryption):** הסתרת נתונים לקריאה. קיימת בשיטה סימטרית (אותו מפתח, בעיית הפצה) וא-סימטרית (מפתח ציבורי ופרטי, איטי אך בטוח). |
| **ארכיטקטורות רשת פיזיות** | **כוכב (Star):** חיבור כולם לסוויץ' מרכזי (מקסימום 2 קפיצות). **חיבור מלא:** אמינות מירבית אך דורש המון כבלים. **טבעת וקו:** שרשור עם רגישות לנפילת שכנים או עורק ראשי. |
| **שיטות גישה לערוץ פיזי** | **חלוקת ערוצים:** חלוקת משאבים קשיחה (מצוין לעומס גבוה, בזבזני לנמוך). **לפי תור:** אסימון/Polling המונע התנגשות אך יוצר תקורה. **גישה אקראית (CSMA/CD):** שידור לפי דרישה, זיהוי התנגשויות והמתנה אקראית לשידור חוזר (הסטנדרט ב-LAN קווי כיום). |

# נספח: תשובות לשאלות ההבנה

**נושא 1: ארכיטקטורות יישומים ברשת**

1. הלקוח הוא זה שיוזם את הפנייה (Request), ולכן עליו לדעת לאן לפנות. השרת ממוקם בכתובת קבועה כדי שכולם ידעו לאתר אותו באופן אמין. הלקוח לא צריך כתובת קבועה, שכן השרת מגיב ישירות לכתובת ממנה הגיעה הבקשה.
2. במודל שרת-לקוח ישנה נקודת כשל יחידה מרכזית (Single Point of Failure). אם השרת נופל, אף לקוח אינו יכול לקבל שירות או לתקשר. במודל P2P, התקשורת מבוזרת; גם אם תחנה אחת נופלת, שאר התחנות יכולות להמשיך לשתף מידע ביניהן בצורה תקינה.
3. לא. העלאת קובץ ל-Google Drive היא פעולה בארכיטקטורת שרת-לקוח. המשתמש פועל כלקוח שמעביר עותק של המידע אל השרת המרכזי של גוגל לאחסון. משתמשים אחרים שירצו את המידע ייגשו לשרת, ולא למחשב המקורי של הלקוח.

**נושא 2: כתובות MAC, כתובות IP וניתוב**

1. כתובת ה-MAC יעד בחבילה שתצא ממחשב A תהיה כתובת ה-MAC של הראוטר המקומי (Default Gateway), שכן היעד הלוגי נמצא מחוץ לרשת ה-LAN המקומית.
2. כתובת MAC מתוכננת ויכולה לתפקד אך ורק בתוך רשת מקומית באותה רמת Broadcast. פרוטוקול ניתוב גלובלי דורש כתובות לוגיות הררכיות (IP), כך שאי אפשר לשלוח חבילה ישירות ל-MAC של יעד מרוחק שנמצא מעבר לראוטר.
3. TTL משמעו Time To Live. זהו מונה המייצג את מספר הקפיצות (Hops - מעברים דרך ראוטרים) שחבילה רשאית לעבור. בכל פעם שהחבילה עוברת בראוטר, המספר יורד ב-1. כשהוא מגיע ל-0, הראוטר משמיד את החבילה (כדי למנוע עומס של חבילות שאבדו או נמצאות בלולאות ניתוב).

**נושא 3: שכבת הייצוג - קידוד והצפנה**

1. לא. קוד ASCII הבסיסי מקצה בייט אחד (8 ביטים) לכל ייצוג של תו. 8 ביטים מאפשרים בדיוק 2 בחזקת 8 קומבינציות, כלומר 256 אפשרויות שונות בלבד.
2. היתרון הוא מהירות עיבוד גבוהה וצריכת משאבים נמוכה. החיסרון הוא שאלת העברת המפתח: מכיוון שאותו מפתח משמש לפתיחת ההצפנה, יש למצוא דרך מאובטחת לחלוטין להעביר את המפתח לצד השני לפני תחילת השיחה (אחרת כל גורם מצותת שיירט את המפתח יוכל לקרוא הכל).
3. כדי שההודעה תהיה חשאית ותקרא רק על ידי החבר, השולח חייב להצפין את ההודעה בעזרת **המפתח הציבורי** של החבר. כשהחבר יקבל את ההודעה המוצפנת, הוא יוכל לפענח אותה אך ורק באמצעות **המפתח הפרטי** (הסודי) שברשותו בלבד.

**נושא 4: ארכיטקטורות רשת פיזיות**

1. הנוסחה היא: N * (N-1) / 2. עבור 8 מחשבים החישוב הוא: 8 * 7 = 56, חלקי 2 שווה 28 כבלים.
2. בטופולוגיית כוכב, כל הנתונים עוברים דרך רכיב משותף מרכזי אחד (הסוויץ'). אם רכיב זה מתקלקל או מאבד חשמל, אין לשום מחשב דרך לדבר עם מחשב אחר. בחיבור מלא, ישירות מוחלטת לכולם מונעת תלות ברכיב מתווך.
3. מספר הקפיצות המקסימלי הוא 2. המידע עושה קפיצה ראשונה מהמחשב השולח אל הסוויץ', וקפיצה שנייה מהסוויץ' אל המחשב היעד.

**נושא 5: שיטות גישה לערוץ התקשורת הפיזי**

1. ברשת ריקה מעומס, שיטת CSMA/CD מאפשרת למחשב לשדר מידית ולנצל את כל רוחב

הפס המלא של הערוץ. לעומת זאת, בשיטת חלוקת ערוצים המחשב יוגבל מראש לחלק יחסי קטן בלבד מרוחב הפס הכולל שהוקצה לו, על אף ששאר הערוצים פנויים לחלוטין.

2. ברגע הזיהוי של התנגשות, שני המחשבים עוצרים את השידור המקורי באופן מיידי. לאחר מכן, כל מחשב מגריל לעצמו מספר אקראי שונה המייצג את משך זמן ההמתנה שעליו להמתין בסבלנות. כשהזמן תם, המחשב יאזין שוב לקו וינסה לשדר מחדש אם הערוץ פנוי.
3. שיטת האסימון יוצרת תקורה ובזבוז זמן מכיוון שהאסימון ממשיך לעבור מתחנה לתחנה גם אם לחלק ניכר מהתחנות אין כלל מידע לשדר באותו רגע, מה שמאט את קצב התגובה הכולל של הרשת למחשבים שכן ממתינים לשדר. מגבלה נוספת היא התלות במחשב שמחזיק את האסימון - אם מחשב נתקע ולא מעביר את האסימון הלאה, הרשת כולה עשויה "לקפוא".