שלב 3

# תשובות

1. מטרת שכבת האפליקציה היא לאפשר לאפליקציות שונות לתקשר ביניהן ולהעביר ביניהן מידע. עד עכשיו כל השכבות נועדו ליצירת תקשורת, זיהוי רכיבים רשתיים והעברת מידע. בשכבה זו עובר המידע עצמו אותו מבקשות האפליקציות להעביר אחת לשנייה (סרטון ביוטיוב, פוסטים בפייסבוק או כתבות בוויינט).
2. HTTP - TCP  
   DNS - UDP  
   FTP - TCP  
   SMTP - TCP  
   DHCP – UDP
3. DHCP הינו פרוטוקול המאפשר לרכיב ברשת לקבל מידע רשתי על עצמו הנחוץ לו על מנת להתנהל ברשת. לדוגמא, כתובת IP, שרת DNS, subnet mask, default gateway ועוד. הבקשה הראשונה (נקראת גם discover) נשלחת ב-broadcast layer 2 (ל-FF:FF:FF:FF:FF:FF) ללא כתובת IP מקור ויעד (הן מאופסות) ובפורט מקור 67, פורט יעד 68 (סטנדרט). כתובת MAC מקור היא כתובת ה-MAC של הרכיב השולח, כצפוי. לאחר מכן ישנם 3 שלבים נוספים – offer – ישנו רכיב ברשת המציע למחשב לקבל נתונים מסוימים  
   - request – המחשב מעונין ומבקש מאותו שרת DHCP לקבל את הנתונים  
   - ack – השרת מאשר את קבלת הנתונים.
4. הפרוטוקול הוא DNS. שרת ה-DNS נקבע באמצעות תשובת ה-DHCP, בו מחשב מקבל את שרת ה-DNS. כמובן, ניתן גם לקנפג שרת DNS סטטי משלנו ידנית.
5. בקשת HTTP מורכבת מ-3 חלקים מרכזיים. בין כל חלק יש ירידת שורה המורכבת גם מ-carriage return \r וגם מירידת שורה \n). בסוף ישנה ירידת שורה נוספת ללא כל תוכן בשורה כדי לסמל את סיום הבקשה.  
   - חלק ראשון מורכב מהבקשה עצמה. מתודה (GET, POST ועוד), משאב (index.html לדוגמא) וגרסת HTTP.  
   - חלק שני מורכב מ-header-ים – נתונים שונים העוברים בבקשות ה-HTTP המאפיינים את התקשורת.  
   - חלק שלישי מורכב מהמידע העובר בבקשה (בבקשת POST).

GET /index.html HTTP/1.1\r\n  
Host: [www.google.com](http://www.google.com)\r\n  
User-Agent: Chrome96\r\n  
Connection: keep-alive\r\n  
\r\n