רשתות

4.1.1)

4) בהינתן שהמחשב ממנו גולשים לא מכיר כלום על עצמו או על השרת הפנימית אליה הוא מחובר, קודם כל המחשב משיג כתובת IP משרת ה-DHCP (בעזרת ARP) ולאחר מכן הוא שולח בקשת DNS לשרת DNS כלשהו באינטרנט כדי להשיג את כתובת ה-IP אליה הוא מנסה לגלוש . כדי לצאת לאינטרנט הוא צריך להעביר את הבקשה לנתב, וכדי לגלות את כתובת ה-IP שלו הוא משתמש ב-ARP שוב. הנתב יעביר את הבקשה לנתבים הבאים במסלול עד שהיא תגיע לשרת ה-DNS שיחזיר תשובה למחשב. כעת, כנראה שהמחשב יצור תקשורת TCP עם השרת שהוא קיבל את כתובת ה-IP שלו והם יתקשרו בפרוטוקול כלשהו, בדרך כלל HTTP.

4.4.2)

1) GET – לבקש משאב מהשרת. POST – להעלות תוכן לשרת. PUT – לעדכן מידע בשרת.

2) Cookies הן דרך לשמור על session בין הלקוח לשרת. סוג של מילון שמכיל הרבה פרמטרים, ביניהן מחרוזת זיהוי, שעובר בכל הודעה בין הלקוח לשרת כדי לזהות את הלקוח ולשמור עליו מידע נוסף.

3) נשלח DNS Request אל שרת ה-DNS, כאשר בפועל הבקשה תועבר כמה פעמים אל השרתים השונים בהתאם להיררכיה, ונקבל חזרה DNS Response עם כתובת ה-IP של גוגל. התשובה תישמר אצל הלקוח לזמן שיוקצב על ידי ה-TTL של התשובה. בנוסף לכך, ייתכן שתחזור יותר מכתובת IP אחת לשם יתירות.

4.6.2)

2) ההבדל העיקרי הוא שב-TCP יש צורך להקים ערוץ תקשורת בין שתי הישויות בעוד שב-UDP פשוט שולחים מידע ומקבלים מידע, וכל הודעה מכילה את כתובת השולח כדי שנדע מאיפה היא הגיעה. ב-UDP משתמשים כשרוצים להעביר מידע אשר חלקו יכול ללכת לאיבוד בדרך ולא יהיו לזה השלכות רעות מדי, כמו לדוגמה VOIP, בעוד שבTCP משתמשים כאשר רוצים להעביר מידע שחשוב שכולו יגיע ויגיע בסדר הנכון, לדוגמה העברת קובץ. בנוסף לכך, TCP מייעל את התקשורת ומוודא כי כל הפקטות עוברות בסדר הנכון. כאמור, UDP פשוט מעביר פקטות ממקום למקום בלי לוודא יעילות וסדר. ב-Header של פקטות TCP יש מלא דגלים שמטרתם להוריד עומס על הרשת.

4.7.2)

2 + 3) NAT הוא מנגנון אשר מאפשר לרשת פנימית שלמה להיות תחת כתובת IP פומבית אחת. הנתב מחליף ב-Header של כל פקטה את כתובת המקור ככתובת הנתב ורק לאחר מכן מעביר אותה הלאה, ואז יקבל את התשובה ויעביר חזרה למחשב אשר התחיל את הבקשה. הוא נועד לפתור את זה שנגמרו כתובות ה-IP והיה צורך לצמצם את הישויות שנמצאות ברשת.

4) להעתיק זה מלוכלך ולכן <https://en.wikipedia.org/wiki/ICMP_hole_punching>.

4.8.2)

1) ARP מתרגם בין כתובת MAC לכתובת IP. כאשר ישות נדרשת לשלוח פקטה אל כתובת IP מסוימת בתוך הרשת הפנימית שלה, לכרטיס הרשת שלה אין מושג לאן בפועל להעביר את המידע, בשביל זה היא צריכה כתובת MAC. על כן, הישות עושה broadcast ומבקשת שהישות בעלת כתובת ה-IP הנתונה תחזיר את כתובת ה-MAC שלה, כדי שניתן יהיה לבנות את המסגרת של השכבה השנייה.

4.10.1)

1) די בטוח שעניתי על זה בצורה סבירה בשאלה הראשונה עד כדי הרחבות על HTTP וTCP אבל זה תיארתי בחלק 4.4.2. בקצרה, יהיה three way handshake עם פקטות syn, syn+ack, ack בין הלקוח לשרת כדי ליצור ערוץ תקשורת של TCP, ולאחר מכן תשלח בקשת GET כדי לטעון את הדף.

2) הסיבה הכי טובה שמצאתי היא שבכתובת MAC יש שיוך של ייצרן בלבד, כלומר אין קשר בין הכתובת לבין המקום הגאוגרפי או הלוגי בו נמצא הכרטיס. לעומת זאת, כאשר חילקו כתובות IP ביצעו חלוקה גאוגרפית או לוגית (נגיד כל טווח הכתובות X שייך לגוגל), וכך ניתן לזהות מקומות או ארגונים מסוימים באמצעות כתובת ה-IP ולבצע אופטימיזציות.

3) אין לי מושג לא הבנתי.