**פרויקט בתקשורת מחשבים - Pong**

*יישום תקשורת להעברת מידע בין שני מחשבים המשתמש בPing המקודד לקוד מורס לצורך שליחת המידע.*

**מבוא:**

בפרויקט זה נבצע העברת מידע בין שתי מחשבים, בשביל להשיג זאת בנינו יישום הנקרא Pong. היישום נכתב בשפת C והוא משתמש בICMP ובפקודות Echo request ו Echo replay כלומר Ping.

שתי המחשבים המשתמשים בPong מיוחסים כ Clientוגם כServer.

**מטרת הפרויקט:**

המטרה של Pong היא לייצר תקשורת בין שני מחשבים על ידי שימוש בICMP ובכך התקשורת מתקיימת דרך שכבת התקשורת ומשם לשכבת היישום בלי לעבור דרך שכבת התעבורה כיוון שאין שימוש בTCP או בUDP.

אופי היישום נותן רמה מסוימת של הצפנה והגנה על התוכן בזמן מעבר ברשת כיוון שהתרגום מתבצע בצד הלקוח.

**תיאור:**

היישום Pong משתמש בPing על מנת לתקשר בין מחשבים. התקשורת עצמה מבוססת על קוד מורס ובכך המידע מוצפן.

ההצפנה של המידע הוא על ידי שימוש בגודל בחבילה כתוכן ההודעה, כלומר אנו מרווחים מכך מספר דברים. בגלל שהתוכן של ההודעה הוא גודל ההודעה, אם יש שינוי בהודעה במהלך התקשורת לא נאבד את המידע.

ואם השוני יהיה בכתובת IP ההודעה לא תגיע לנמען אבל ICMP יחזיר לנו בעזרת Echo relay את הכתובת של מי שקיבל את ההודעה ונוכל לשלוח שוב.

לאור העבודה שהתוכן עובר כקוד מורס המידע מוצפן ואם מסתכלים על המידע ללא שימוש ב PONG יהיה ניתן לראות רק Pings שנשלחים ולא תוכן ממשי.

תוכן החבילה משמש בתור ACK של הtcp ובכך אנו יודעים לאיזה Ping להתייחס קודם ובכך לשמור על סדר ההודעה.

קיים קובץ היסטורית הודעות המיועד למתכנתים בלבד שבו נשמר תאריך קבלת הping ותוכן רלוונטי.

כדאי לסיים את הקשר על כול משתמש לרשום exit אשר מודיעה לצד שני על רצון לסיים קשר ועל כך שלא ישלחו עוד הודעות כאשר שתי הצדדים שולחים exit התקשורת נסגרת.

**כלים:**

1. אנו נשתמש בתוכנה WireShark לצורכי השוואה.
2. נשתמש במחשבים מבוססי Linux.
3. תכנות בשפת C.

**תוצאות צפויות:**

אנו נצפה שכאשר נשתמש בPong המידע יעובר בצורה אמינה יותר וכאשר היו הפרעות בתקשורת עדיין נצליח לקבל את המידע שלם.

אנו מצפים לייצר עומס מוגבר ברשת לעומת tcp אבל נרצה לתמוך בכל פונקציונליות של tcp ללא שימוש בשכבת התעבורה. נבצע זאת על ידי חישובים מסובכים בשכבת היישום (PONG).

**הסבר לקוד:**

Pong עובד כשרת ולקוח כאחד. היישום רשום בC ומחולק לשתי חלקים מרכזיים.

החלק הראשון הוא החלק של קליטת מידע כלומר הלקוח. היישום יוצר קשר עם השרת(המחשב השני) ומחכה להתחברות, ברגע שהמחשב מקבל ACK ונוצר קשר, והלקוח ממתין להודעות.

בשלב הראשון, כאשר מפעילים את היישום, רושמים את הip אשר מצפים לקלוט ממנו הודעות. אחרי שהתחברנו ואנו מחכים להודעות, ניתן גם לשלוח הודעות.

הלקוח שולח הודעות והיישום ממיר אותם לתווים המוצפנות לקוד מורס. ההמרה עובדת בצורה שלכול אות בהודעה אנו ממירים לבינארי ומוסיפים עוד ביטים של אפסים עד שמגיעים ל5 ביטים, לפי הכמות ביטים שהשלמנו מוסיפים עוד 4 ביטים אחרי, שאנו סופרים בו את כמות השלמות שעשינו. בצורה זאת אנו נותנים לכול אות ייצוג בינארי ייחודי.

לאחר מכן אנו מכניסים לBitMap בגודל 36 את כול הערכים העשרוניים של הייצוג הבינארי של האותיות.

ככה אנחנו יכולים לגשת בצורה יעילה ומהירה לכול ערך בינארי של אות והוסיף להודעה הסופית. אחרי שתרגמנו את ההודעה לבינארי אנו נשלח Ping, שלפי מורס קוד יש 2 סוגים של הודעה, קו ונקודה וכן אנו נמיר את הבינארי, 0 יהיה נקודה ו1 יהיה קו, בנוסף הוספנו עוד ייצוגים לסוף אות, סוף מילה סוף משפט וסוף התכתבות. ובסוף אנו שולחים את הping לפי הייצוג שלהם לנמען.

בצד של השרת, מתבצע אותו תהליך, מכניסים את הip ממנו רוצים לקלוט ומחכים להודעות. ברגע שמתקבל Ping מהIP אשר אנו מצפים לקלוט ממנו השרת מפרק את רכיבי הPing לגודל חבילה שהתקבלה פחות 42 כדי לדלג על שאר המידע שמגיע עם הping, הip של השולח ותוכן ההודעה(משמש בתור ACK).

לפי הACK ניתן לדעת את סדר הקריאה של ההודעה ובכך נדע איזה ping לתרגם קודם. מתחילים לרוץ על הpings ולשמור אותם עד אשר מתקבל ping של סוף אות, ברגע שמתקבל הping לסוף אות אנו לקוחים את הpings אשר התקבלו ומתרגמים אותם לאות אשר נשמרת במערך.

פעולה זאת מתמשכת עד אשר מגיע ping לסוף הודעה וברגע זה היישום מדפיס למשתמש את ההודעה אשר התקבלה.

המרת הpings לאותיות מתבצעת בצורה הבאה:

מכניסים לBitMap בגודל 512 את כול האותיות, כך שהתא שלהם בbitmap הוא הערך העשרוני של הערך הבינארי של האות וכך אנו יכולים לבדוק איזה אות התקבלה במהירות ויעלות גבוה.

כאשר רוצים לסיים תקשורת, על שני הצדדים לשלוח ping ייחודי אשר מודיע לצד השני על אי שליחת הודעות נוספות וממתין ברקע להודעה מהצד השני על רצון לסיום קשר.

**תוצאות שהתקבלו: (עם הפתעות)**

הפתעה: תפסנו Ping! ☺

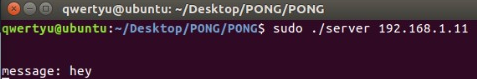
הצלחנו לייצר תקשורת תקינה בשימוש ב3 שכבות בלבד(פיזית, רשת, יישום), כפי שציפינו נוצר עומס קל ברשת עקב שליחת Pings מרובה.

ניתן לראות דוגמה לשליחת הודעה:

פה רואים שנשלחה הודעה עם התוכן "hey" מהמחשב הראשון שכתובת הip שלו היא 192.168.1.11

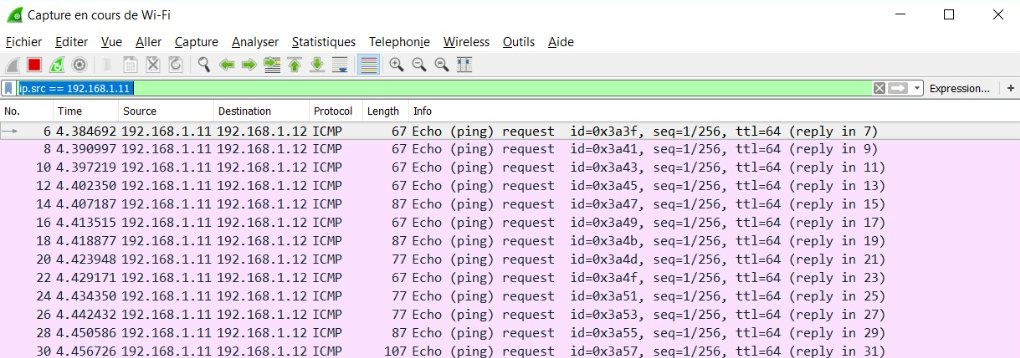


כעט במחשב השני בעל הip 192.168.1.12 ניתן לראות שהתקבלה ההודעה "hey"



כדי לבדוק שמטרתנו הושגה רצינו לראות את הנתונים הנשלחים לוודא שאכן התקשורת נעשית רק על ידי ICMP וללא שימוש בTCP או UDP.

בעזרת התוכנה Wireshark בדקנו מה נשלח והתוצאות לא היו מפתיעות, כול התקשורת התנהלה בicmp



ניתן לראות בשחור על גבי ורוד שכול התקשורת התבצעה על ידי ICMP.

אם נסתכל על הגדלים ניתן לראות את כול סוגי הpings:

Ping בגודל 67: גודל ההודעה נטו שנשלחה הינו 42 בטים פחות כלומר 25 שמייצג נקודה בקוד מורס.

Ping בגודל 77: מייצג קו בקוד מורס וגודלו האמיתי 35.

Ping בגודל 87: מייצג סוף מילה וגודלו האמיתי 45.

Ping בגודל 107: מייצג סוף הודעה וגודלו האמיתי 65.

הפתעה נוספת היא שלמדנו על ה DHCP, נתקלנו בזה כאשר חקרנו את תוצאות של Pong, כאשר השארנו את הקשר בין מחשבים דלוק בלי לשלוח מידע ראינו בעזרת הWireShark שעדיין הייתה תנועה. מחקר קצר על הנושא הבנו שהDHCP נשלח בצורה עצמאית ומטרתו לבדוק האם הקישוריות עדיין קיימת.

אחרי התכתבות קצרה בין שני המחשבים, רצינו לראות כמה הודעות(pings) נשלחו, ניתן היה לראות בקובץ מעקב כ900 pings.

**מקורות:**

1. <https://en.wikipedia.org/wiki/Ping_(networkring_utility)>
2. <https://linux.die.net/man/8/ping>
3. <https://en.wikipedia.org/wiki/Internet_Control_Message_Protocol>
4. <https://www.wireshark.org/download.html>