



תרגיל בית מס' 3.

להגשה עד: 05.03.2024.

חלק ראשון – UDP Pinger (55%).

בחלק זה של התרגיל נממש בעצמו תוכנה לבדיקת תקינות מרחוק של נקודת קצה (למשל שרת מרוחק) – "pinger" תוך שימוש בUDP Sockets, ונשתמש בWireshark על מנת להתבונן בתעבורתה. השם מתבסס על הודעות ping של פרוטוקול ICMP שראינו בכיתה, אך המימוש של יהיה פשוט יותר.

לתוכנה יהיו שני צדדים – Agent וPinger. תוכנת Agent תרוץ על נקודת הקצה ותפקידה לענות להודעות ה"ping" שתקבל וכך לסמן לצד השולח שנקודת הקצה רצה ופועלת בצורה תקינה. תוכנת Pinger תשלח את הודעות ה"ping" לצד המרוחק ותמתין לקבלת תשובות.

הודעות הפרוטוקול:

מבנה הודעת PING Request (אותה הPinger יישלח לAgent):

- שדה Opcode. שדה זה יכול בית אחד שערכו יהיה 0 – קבוע המסמל בקשת ping.
- שדה ID של ההודעה, 4 בתים. יאותחלו ל0 בהודעה הראשונה שתשלח וערכם יעלה ב1 בכל שליחה.
- שדה DATA. שדה זה יהיה בגודל משתנה (כמפורט למטה) ויכלול מידע לבחירתכם (קבוע/אקראי).

מבנה הודעת PING Reply (אותה הAgent יישלח לPinger):

- שדה Opcode. שדה זה יכול בית אחד שערכו יהיה 1 – קבוע המסמל מענה לבקשת ping.
 - שדה ID. שדה זה יכלול 4 בתים ויכיל את ID של ההודעה – יהיה של ID שקיבלנו.
 - שדה DATA. שדה זה יכול את המידע שהגיע בreply.
- זאת אומרת – הודעת המענה תהיה תמונת מראה של הודעת הבקשה שהגיעה, למעט הבית הראשון שישתנה.

שורות הרצה ופרמטרים:

שורת ההרצה של agent:

```
python udp_agent.py -p port
```

כאשר:

- p port • פרמטר אופציונלי המסמן את מספר הפורט הנבחר. הערך הדיפולטי יהיה 1337.

שורת ההרצה של pinger:

```
python udp_pinger.py IP -p port -s size -c count -t timeout
```

כאשר:

- IP • כתובת הIP של נקודת הקצה עליה רץ הudp_agent.
- p port • פרמטר אופציונלי המסמן את מספר הפורט הנבחר. הערך הדיפולטי יהיה 1337.
- s size • פרמטר אופציונלי המסמן את גודל שדה data של ההודעה. הערך הדיפולטי יהיה 100. ניתן להניח שערכו המקסימלי הוא 1400.
- c count •



רשות תקשורת מחשבים
סמסטר א' תשפ"ד

פרמטר אופציונלי המסמן את מספר ההודעות שישלחו עד סיום הריצה. הערך הדיפולטי יהיה 10.

• -t timeout

פרמטר אופציונלי המסמן את כמות הזמן (במילישניות) שיש לחכות מרגע שליחת הודעה, ועד שמחליטים שrequest timed out. הערך הדיפולטי יהיה 1000 (כלומר שנייה אחת).

הדפסות למסך.

במהלך ריצתה, תוכנת udp_pinger תדפיס נתונים אודות ההודעות שנשלחות, בפורמט הבא:

X bytes from IP: seq=ID rtt=t ms

להודעות שהתקבל עליהן אישור בהצלחה. כאשר X הוא גודל ההודעה שהתקבלה (כולל header של הפרוטוקול), IP היא הכתובת ממנה התקבלה ההודעה, ID הוא ID של ההודעה וt הוא משך הזמן שלקח מרגע השליחה ועד קבלת התשובה. עבור הודעות שהלכו לאיבוד יודפס:

request timeout for icmp_seq ID

בסיום הריצה יודפס:

--- IP statistics ---

c packets transmitted, k packets received, pp.pp% packet loss

כאשר c הוא מספר ההודעות הכולל ששלחנו, k הוא מספר ההודעות שקיבלנו עליהן אישור בהצלחה, וpp.pp% הוא אחוז ההודעות שהלכו לאיבוד (מחושב לפי k וc).
תוכנת agentn תהיה שקטה ולא תדפיס דבר.

דוגמאות הרצה:

דוגמא 1:

שורות ההרצה:

```
python udp_agent.py -p 1337
```

```
python udp_pinger.py 1.2.3.4 -p 1337 -s 60 -c 4
```

הפלט של תוכנת pinger:

```
65 bytes from 1.2.3.4: seq=0 rtt=20.288 ms
```

```
65 bytes from 1.2.3.4: seq=1 rtt=19.114 ms
```

```
65 bytes from 1.2.3.4: seq=2 rtt=23.955 ms
```

```
65 bytes from 1.2.3.4: seq=3 rtt=11.175 ms
```

```
--- 1.2.3.4 statistics ---
```

```
4 packets transmitted, 4 packets received, 0.0% packet loss
```

דוגמא 2 (עם איבוד פקטות):

שורות ההרצה:

```
python udp_agent.py -p 1337
```

```
python udp_pinger.py 1.2.3.4 -p 1337 -s 60 -c 3
```

הפלט של תוכנת pinger:

```
65 bytes from 1.2.3.4: seq=0 rtt=20.288 ms
```

```
request timeout for icmp_seq 1
```

```
65 bytes from 1.2.3.4: seq=2 rtt=6.270 ms
```

```
--- 1.2.3.4 statistics ---
```

```
3 packets transmitted, 2 packets received, 33.33% packet loss
```

עבודה עם Wireshark וקובץ Readme.

צרפו קובץ Readme עם תיאור קצר של פרוטוקול udp_pinger שלכם שמועבר שם.

בדומה לתרגיל הראשון, הריצו את התוכנה שכתבתם והתבוננו על הפקטות בWireshark. בחרו פקטת request אחת ופקטת reply אחת, צרפו אותן לקובץ Readmen (כprint screen), והראו את ההבדל ביניהן בהתאם לשדות פרוטוקול התוכנה שלכם.



חלק שני – ICMP, כתובות IP, ואלגוריתמי ניתוב ברשת (45%).

שאלה 1 – ICMP (5%):

איזה שירות פקטות ה-PING מספקות למנהל הרשת? מתי נרצה להשתמש בהן? תארו תרחיש בו שימוש בהן יוכל לעזור לניהולה התקין של הרשת.

שאלה 2 – כתובות IP (20%):

מנהלת רשת חילקה את רשת הארגון למספר תת רשתות. נתב בארגון מכיל את טבלת הניתוב הבאה:

Network Name	Network	Exit port
A	192.10.12.128/26	1
B	192.10.12.64/26	2
C	192.10.12.96/27	3
D	192.10.12.160/28	4
E	0.0.0.0/0	5

א. מה התפקיד של שורה E?

ב. מצאו היררכיה בין הרשתות. לכל שתי רשתות X ו-Y עבורן X מוכלת ב-Y, ציינו זאת.

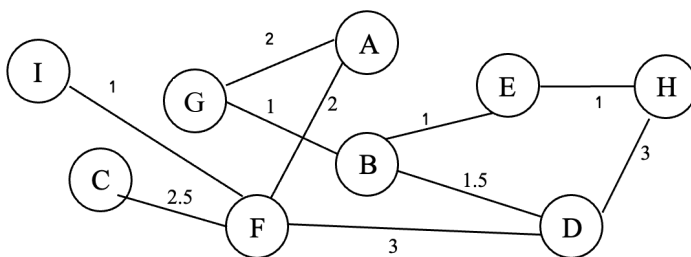
ג. חשבו עבור 5 החבילות P-T לאיזו יציאות הן ינותבו.

Datagram Name	Dest IP Address
P	192.10.12.134
Q	192.10.12.173
R	192.10.12.103
S	192.10.12.94
T	192.10.12.215

ד. (סעיף זה אינו קשור לסעיפים הקודמים) נתונה הכתובת 192.10.0.160. מהי מסכת הרשת (network mask) הקצרה ביותר עבורה כתובת זו יכולה לשמש ככתובת רשת (network address)? עבור מסכה זו, מה תהיה כתובת ה-broadcast של הרשת?

שאלה 3 – אלגוריתמי ניתוב ברשת (20%):

נתונה הרשת הבאה (באיור), כולל המרחקים, "עלויות", בין הצמתים השונים.



א. נניח ונשתמש באלגוריתם ה-Distance Vector למציאת מסלולי ניתוב ברשת, בצורתו הנאיבית (ללא הגנות

מתקדמות). מהי טבלת הניתוב בצומת C (כלומר – מרחקיו משאר הרשת) בתחילת התהליך? ובסיום התהליך (לאחר התייצבות האלגוריתם)? הציגו גם את שדה ה-next hop.

ב. נניח כעת כי נותקה הקשת A-F. האם האלגוריתם יתייצב? אם כן, מה יהיו כעת המרחקים בצומת C? אם לא, כיצד תראה טבלת המרחקים ב-D לאורך הזמן?

ג. בהמשך לסעיף הקודם, נניח כעת כי בנוסף נותקה גם הקשת F-D. האם האלגוריתם יתייצב? אם כן, מה יהיו כעת המרחקים בצומת C? אם לא, כיצד תראה טבלת המרחקים ב-D לאורך הזמן?

ד. נניח כי היינו משתמשים ב-link state routing במקום. הריצו על הרשת (ללא ניתוקים) סימולציה של מציאת המסלולים הקצרים ביותר מ-B לכלל הצמתים ברשת תארו את שלבי האלגוריתם.



הנחיות הגשה (לתרגיל בשלמותו) –

- הגישו את התרגיל כקובץ ZIP אחד, המכיל בתוכו:
(1) קובץ ZIP נוסף עבור החלק הראשון (המעשי) של התרגיל, שיכיל את קבצי הקוד, וקובץ Readmen עם חלק ה-Wireshark.
(2) התשובות לחלק התאורטי של התרגיל (בסריקה או מוקלד, לבחירתכם).
- לZIP הראשי קראו ID.zip אם הגשתם את התרגיל לבד, או ID1_ID2.zip אם הגשתם בזוג.