

מבוא לתקשורת מחשבים – תרגיל מעשי 2

1. שמות המגישים ות"ז

תומר סרוסי 209130269
בן כהן 208685784

2. הוראות שימוש והרצה

את הפרויקט כתבנו בשפת c. בקובץ המכוון שלנו ישנם 2 קבצי קוד רלוונטיים, 2 קבצי הרצה, קובץ הרצה ליצירת קבצי הלוגים עבור המדידות, תכניות הפיתוח לחישובי הסטטיסטיקה מהמדידות, וקובץ ה-readme:
islip.c, tr_gen.c, islip.exe, tr_gen.exe, create_logs.bat, statistics_calc.py, readme.pdf

על מנת להריץ את התכניות יש לבצע:

א. על מנת להריץ את tr_gen.exe מהcmd ניתן לבצע את שורת הפקודה הבאה:
tr_gen N T seed p [-d] > in_script.txt

ב. על מנת להריץ את islip.exe מהcmd ניתן לבצע את שורת הפקודה הבאה:
islip N k r < in_script.txt > out_script.txt

ג. על מנת להריץ את שתי התכניות יחד מהcmd ניתן לבצע את שורת הפקודה הבאה:
tr_gen N T seed p - d | islip N k r > out_script.txt

3. תיאור הפרויקט ומבנה הקוד

בפרויקט זה מימשנו תזמון מסוג iSLIP למתג NxN עם virtual output queuing. מימוש זה נכתב באמצעות:

- מחולל תעבורה (tr_gen): תכנית המייצרת קבצים עם תסריטי הגעת חבילות למתג. קבצים אלה ישמשו קלט לישום השני;
- מודול תזמון (islip): תכנית המקבלת תסריט הגעת חבילות ומייצרת תסריט עזיבת חבילות עפ"י iSLIP כפי שראינו בהרצאה, אשר מבוסס מדיניות Round-Robin

נפרט אודות המימושים השונים תוך הצגת captures של הקוד שלנו ונציין כי הקוד תועד במלואו בקבצי ה-c.

Traffic generator (tr_gen)

התכנית מייצרת קובץ ע"י כתיבה ל stdout המתאר את התעבורה המגיעה למתג.

- לאחר אתחול המשתנים בתחילת פונקציית main, מתקיימת בדיקת תקינות הארגומנטים. במידה וקיימת טעות בכמות הארגומנטים/ בפורמט ארגומנט מסוים, מודפסת הודעת שגיאה באמצעות perror.
- לאחר מכן (כמתואר בקוד המצורף), התכנית מייצרת חבילות לכל פורט כניסה באופן בלתי תלוי ע"פ ניסוי ברנולי עם פרמטר p בכל צעד. יעד החבילה יוגרל אקראית כתלות בקיום או העדר הדגל $-d$ –
 - אם הדגל לא ניתן בשורת הפקודה, התפלגות היעדים היא אחידה, כלומר ההסתברות שיעד החבילה הוא פורט j היא $\frac{1}{N}$ לכל j ולכל חבילה.

```
for (int t = 0; t < T; t++)
{
    /*iterate over n*/
    for (int n = 0; n < N; n++)
    {
        cur_p = (double)rand()/(double)RAND_MAX;
        if (cur_p < p)
        {
            if (d) /*if d flag, use prob as mentioned*/
            {
                cur_p = (double)rand()/(double)RAND_MAX;
                if (cur_p < (2.0/3))
                {
                    out_port = n;
                }
                else
                {
                    out_port = (n + 1) % N;
                }
            }
            else /*if not d flag, use regular prob*/
            {
                cur_p = (double)rand()/(double)RAND_MAX;
                out_port = (int)(cur_p * N) % N;
            }
            printf("%d %d %d\n", t, n, out_port);
        }
    }
}
```

- אם הדגל d – ניתן בשורת הפקודה, התפלגות היעדים נקראת "אלכסונית", והיא מוגדרת לחבילה עם פורט הגעה i כדלהלן:

$$\Pr[\text{destination port} = j \mid \text{arrival port} = i]$$

$$= \begin{cases} \frac{2}{3} & \text{if } j = i \\ \frac{1}{3} & \text{if } j = 1 + i \bmod N \\ 0 & \text{otherwise} \end{cases}$$

Scheduler(islip)

תפקידו העיקרי של מודול המתזמן הוא לקרוא את תסריט הגעת החבילות מ `stdin`, לחשב תזמון, ולכתוב את תסריט עזיבת חבילות ב `stdout`. המתזמן מקבל בשורת בפקודה את מימד המתג N , מספר איטרציות ה `iSLIP` המסומן k , ומספר מזהה ריצה r . בנוסף, המתזמן כותב קובץ בשם `log.r` המתאר נתונים שונים של הריצה.

- לכל צעד זמן, התכנית קוראת מ `stdin` את כל החבילות שהגיעו בצעד זה ומציבה כל חבילה בחוצץ המתאים עפ"י `virtual output queuing`.

```
fgets(cur_input, MAX_INPUT, stdin); /*Read the first packet from stdin*/
sscanf(cur_input, "%d %d %d", &arr_time, &arr_port, &dest_port); /*Parse the read packet into relevant variables*/

while (!queues_empty || !eof) /*Loop until all queues are empty, and stdin has reached EOF*/
{
    while (!eof && arr_time <= cur_time) /*read until eof reached or last read packet's arrival time is bigger than current time
    {
        enqueue(queues[arr_port][dest_port], arr_time); /*Enqueue the last read packet*/
        if(fgets(cur_input, MAX_INPUT, stdin) == NULL) /* Try reading the next packet from stdin*/
        {
            eof = true; /*If the reading failed, eof of stdin has been reached*/
            break;
        }
        sscanf(cur_input, "%d %d %d", &arr_time, &arr_port, &dest_port); /* Parse the read packet into relevant variables*/
    }

    queues_empty = true; /*Initialize queues_empty as true*/
    for (int i = 0; i < N; i++)
        /*Initialize all queues as not linked*/
    {
        ports[i].input_linked = false;
        ports[i].output_linked = false;
    }
}
```

- לאחר מכן התכנית מריצה k איטרציות של `iSLIP` הכוללות `grant`, `accept`, `request`, בכל איטרציה כדי למצוא פרמוטציה חלקית למימוש בצעד זה. למשך k איטרציות, ניתן לראות את ה `Requests`:

```
for (int cur_output = 0; cur_output < N; cur_output++) /*Iterate through the output ports*/
{
    if (!ports[cur_output].output_linked) /*If the current output isn't linked yet*/
    {
        for (int cur_input = 0; cur_input < N; cur_input++) /*Iterate through the input ports*/
        {
            if (!ports[cur_input].input_linked) /*if the current input isn't linked yet*/
            {
                if(queues[cur_input][cur_output]->size > 0) /*If the queue for the current input and output isn't empty*/
                {
                    queues_empty = false; /*Mark that the queues aren't empty*/
                    ports[cur_output].requests[cur_input] = true; /*Raise a request from the current input port to the current output port*/
                }
            }
        }
    }
}
```

כמו כן, את הGrants, Accepts: בשלב זה גם התוכנית כותבת, לכל חבילה, מתי יצאה מאיזה פורט. כך, בתסריט הפלט הנכתב לstdout יש שורה לכל חבילה

```
for (int i = 0; i < N; i++)
{
    int cur_input = (ports[cur_output].request_priority + 1) % N; /*Iterate through all input ports, starting from the port with most priority*/
    if (ports[cur_output].requests[cur_input]) /*If the current input port sent a request*/
    {
        ports[cur_input].grants[cur_output] = true; /*Send a grant to the current input port*/
        ports[cur_output].request_priority = cur_input; /*Mark the current input port as the one with most priority in case this output port wasn't accepted*/
        break; /*Found a port to grant to*/
    }
}

for (int cur_input = 0; cur_input < N; cur_input++) /*Iterate through the input ports*/
{
    if (!ports[cur_input].input_linked) /*If the current input isn't linked yet*/
    {
        /*Choosing an output port to accept*/
        for (int i = 0; i < N; i++)
        {
            int cur_output = (ports[cur_input].grant_priority + 1) % N; /*Iterate through all output ports, starting from the port with most priority*/
            if (ports[cur_input].grants[cur_output]) /*If the current output port sent a grant*/
            {
                ports[cur_input].input_linked = true; /*Set the current input port as linked*/
                ports[cur_output].output_linked = true; /*Set the current output port as linked*/
                ports[cur_input].grant_priority = (cur_output + 1) % N; /*Set the current input port as the one with least priority*/
                ports[cur_output].request_priority = (cur_input + 1) % N; /*Set the current output port as the one with least priority*/
                printf("%d %d %d %d\n", dequeue(queues[cur_input][cur_output]), cur_input, cur_output, cur_time); /*Print the current link information*/
                break;
            }
        }
    }
}

cur_time++; /*Increase the time by 1*/
```

- לבסוף מתבצעות ההדפסות לקובץ r.log את גודל כל חוצץ לאחר כל צעד זמן כדרוש:

```
if (!queues_empty || !eof) /*If it is not the final loop*/
{
    for (int cur_input = 0; cur_input < N; cur_input++) /*Iterate through all input ports*/
    {
        for (int cur_output = 0; cur_output < N; cur_output++) /*Iterate through all output ports*/
        {
            fprintf(r_log, "%d %d %d %d\n", cur_time, cur_input, cur_output, queues[cur_input][cur_output]->size);
        }
    }
}

cur_time++; /*Increase the time by 1*/
```

- בשלב האחרון מתבצע שחרור זיכרון למערכים שהוקצו.

מקורות שנעזרנו בהם:

- נעזרנו במתודות תורים שכתבנו בתרגיל 1 לטובת התורים
- כתבנו תוכניות Python באמצעות הספרייה Statistics על מנת לממש את חלק הניסוי והגרפים (אשר מצורפים בקובץ RESULTS)

4. באגים ידועים ומגבלות שימוש:

בבדיקת פורמט הפרמטרים, השתמשנו בפונקציה atoi() על מנת להמיר מstring לint. התייחסנו לכך שatoi לא מתקבל מספר עשרוני ואם כן הוא מעוגל (לדוג' קלט של "3.5" יעוגל מטה ל3)