עבודת גמר בקורס תקשורת מחשבים(הגשה עד 13.08.20) גרסה 1.1

הבחינה מחולקת ל2 חלקים: (1) שאלות תכנות עם שאלות תיאורטיות (2) בחינה בעל פה.

הוראות:

1. העבודה צריכה להתבצע לבד ללא עזרה. כל השאלות על העבודה אפשר לשאול בוואטסאפ או בגוגל דוק של הקורס.
2. אם שאלת התכנות לא מתקמפלת אתם מקבלים ציון אפסי על שאלו זו.

שאלות:

1. אפיון חבילות (50 נק')

א. בחלק הזה אתם תממשו טבלת חיפוש של כתובות IP, הטבלה מיוצגת על ידי עץ בינארי. המימוש יתמוך ב שלוש פעולות: ADD/REMOVE, שמוסיפה/מוחקת פרפיקס לעץ המייצג והפקודה שצריכה להתבצע עבורו ופעולה FIND מחפשת פרפיקס הנתון. הקלט נקרא מהקובץ בפורמט הבא: action, prefix, result

Input file:
ADD 255.255.255.0/24 A
FIND 255.255.255.255
REMOVE 255.255.255.0/24 A

Expected output:
Added 255.255.255/24 A at the depth <depth in the binary trie>, total nodes <notal nodes in binary trie>
Found 255.255.255.255 <found action> at the depth <depth in the binary trie>
Removed 255.255.255/24 A at the depth <depth in the binary trie> total nodes <notal nodes in binary trie>

ב. בינתן אופטימיזציה הבאה: אם 2 פרפיקסים נדבלים רק בביט אחרון ויש להם את אותה פעולה (ACTION) אז שני פרפיקסים יכולים להיות מוחלפים על ידי פרפיקס בודד עם אותה פעולה.

Example: Two prefixes 1111* -> A 1110*-> A can be replaced by 111**->A.

דוגמות של מקרה רגיל ומשופר תסתכל פה.

תוסיפו האופטימיזציה זו לאימפלימינטציה הקודמת של טבלאות חיפוש, הפעולה FIND צריכה להחזיר פרפיקס משופר עם הפעולה(ACTION) הנכונה.

ג. באימפלמנטציות מסיף א' זמן חיפוש במקרה הגרוע שווה לאורך המקסימלי של הפרפיקס. נניח שקיימת הגבלה מערכתית על זמן חיפוש ב- FIND. למשל 8 חיפושים בטבלת הפרפיקסים באורך עד 32 ביט. תציעו שיטות מימוש של טבלת חיפוש במקרה הזה.

ד. תציעו דרכים אלטרנטיביים לייצוג של טבלת הפרפיקסים עם מקבלות שונות על זמן חיפוש וזיכרון מוקצה. תשווה יתרונות וחסרונות של הצעות, הצעות יותר מעניינות יקבלו הארכה טובה יותר,אפשר להשתמש לא רק בעצים בינאריים.

מה צריך לעשות:

1. לממש 2 ייצוגים של טבלאות של פרפיקסיים כמו שמפורט בסעיפים א' וב'. בקובץ MAKEFILE הטרגט עבור סעיף א' צריך להיות **prefix_table** ועבור סעיף ב' **prefix_table_opt**. תוסיפו קובץ README שמסביר איך להריץ שני מימושים. תוסיפו קובץ **sample_inpu**t שהשתמשתם הרצאה.
2. ענו על שאלות המפורטות בסעיפים גי וד' בקובץ נפרד 1_<your_id>.pdf>.
3. כל הקוד שמפותח וקובץ 1_<your_id>.pdf> תכניסו לספרייה נפרדת בשם 1_yourid.

לבחינה בעל פה:

בנוסף למה שמימשתם בשאלה 1 אתם צריכים לדעת את כל הפרוטוקולים ניטוב שלמדנו: RIP, OSPF, IS-IS, BGP ויתרונות וחסרונות שלהם.

2. טבע של הגודש (50 נק')

אחת השאלות המרכזיות בתקשורת מחשבים היא איך להתעסק בגודש שנוצר. כמו שלמדנו הגודש ברשתות של חבילות בלתי נמנע אם אנו רוצים להשתמש במושבים של הרשת בצורה יעילה. בזמן ההרצאות אנחנו דיברנו על בקרת הגודש איפה שהלוגיקה ממומשת בקצוות (endhosts). הסוג הזה של בקרת הגודש מסתמך על איבוד של חבילות ולא מניח ידע על המצב בתוך הרשת (end-to-end design principle). אם יעילות נוספת נדרשת principle end-to-end design צריך תמיכה מהרשת (ניהול חוצצים). במקרה הזה בקרת הגודש זה קומפוזיציה של לוגיקת הבקרה ממומשת בקצוות אלגוריתמים לניהול חוצצים ממומשים בתוך הרשת. בשאלה זו אנחנו מנסים להבין השפעות של 2 אלגוריתמים לניהול חוצצים: Early-Deadline-First ו Bounded-Delay אנחנו נגדיר אותם מיד למטה. כמו שלמדנו האלגוריתם לניהול חוצצים שמנהל תור בודד מוגדר על ידי admission policy שמגדיר איזה חבילות יכולות להתקבל לתוך התור ו-processing policy שמגדיר באיזה סדר החבילות מאובדות.

קלט .אנחנו מניחים שלכל חבילה יש 2 פרמטריים: s(p) packet slack שמגדיר מרווח זמן שהבילה צריכה להיות מאובדת v(p packet value שמגדיר את הערך של החבילה. אתם יכולים להניח שכל החבילות עם expired slack נמחקות מהתור בצורה אוטומטית.

מודל. אנחנו מניחים הארכיטקטורה של תור בודד. הזמן מחולק לסלוטים, כל סלוט מחולק ל-2 פאזות.

1) פאזה הגעה (arrival phase) איפה ש-admission policy מחליט איזה חבילות מוכנסות לתור.
2) פאזה איבוד (processing phase) איפה שחבילה בודדת שנבחרה על ידי processing policy נשלחת והערך שלה התווסף לערך הכללי.

הערכים של סלאקים של כל החבילות שנשארו אחרי פאזה הגעה מוקטנים באחד. כל החבילות עם ערך של סלק שווה אפס נמחקים מהתור בלי להוסיף לערך הכללי של פונקציה מטרה.

פונקציית מטרה זה להגדיל ערך כולל של כל החבילות שנשלחות אם סלאק שלו פג תוקף.

אלגוריתם EDF:
פאזה הגעה: לכל חבילה נכנסת p אם התור לא מלא לקבל p. אחרת(תור מלא) אם בתוך התור קיימת חבילה q s(q)<s(p) נמחק מהתור q ו-p מתקבל. אחרת p נזרק.
פאזה איבוד: בין כל החבילות בתור לשלוח חבילה עם ערך של הסלק קטן ביותר.

אלגוריתם BOUNDED-DELAY:
פאזה הגעה: לכל חבילה p נכנסת אם התור לא מלא לקבל את p. אחרת(תור מלא) אם בתוך התור קיימת חבילה q עם ערך קטן יותר מהערך של p אז q נמחקת מהתור ו-p מתקבלת. אחרת p נזרקת.
פאזה איבוד: בין כל החבילות בתוך התור לשלוח חבילה עם ערך המקסימלי.

מה צריך לעשות:

1.) לממש אלגוריתמיים EDF וBOUNDED DELAY. כל אלגוריתם תזמון מקבל כקלט גודל של התור (בחבילות) ושם של הקובץ כולל בתוכו החבילות מגיעות. כל שורה בתוך קובץ קלט כוללת חבילות מוגדרות על ידי שלשות (AMOUNT_SLACK_VALUE).

(5,3,6) (2,4,1) (6,2,4) // during this time slot arriving 5 packets whose slack is 3 and the value is 6, etc.

מספר השורות בתוך הקובץ מגדיר כמות הסלוטים בהרצה. לכל הרצה על קובץ קלט הפלט צריך להיות:
Expected output:
Total arrived packets %d, total dropped packets %d, total transmitted packets %d, total transmitted value %d.

בקובץ MAKEFILE הטרגט לEDF הוא **edf** ול-BOUNDED DELAY הוא **bd**.

input_file example:

(2,3,6) (3,4,1)
(3,4,5)

Running example: edf 4 input_file

Total arrived packets 8, total dropped packets 4, total transmitted packets 4, total transmitted value 17.

The detailed run of EDF on the sample_input can be found here.

בפגישת זום נתתי כמה דוגמאות של EDF וBOUNDED DELAY הדוגמאות אפשר למצוא כאן.
בחלק הזה אתם תממשו 2 אלגוריתמיים EDF וBOUNDED DELAY, תוסיפו קובץ README מסביר איך לקמפל ולהריץ, הקלט שישתמשתם (sample_input) והתוצאות ל-2 אלגוריתמים עבור קובץ sample_input שהשתמשתם בהרצאות וקבצי תוצאות על אותו קלט (edf_sample ו bd_sample).

2.) בחלק התיאורטי אתם תצטרכו לענות על השאלות הבאות:
א. אם לכל חבילות נכנסות יש אותו ערך אבל ערכים של סלאקים שונים איזה אלגוריתם עדיף יותר, EDF או BOUNDED DELAY לפונקציה מטרה המגודרת. נמק למה.
ב. אם לכל חבילות נכנסות יש אותו ערך של סלאק אבל ערכים של חבילות שונים איזה אלגוריתם עדיף יותר, EDF או BOUNDED DELAY, לפונקציה מטרה מוגדרת. נמק למה.
ג.במקרה הכללי שחבילות נכנסות גם ערכים וגם סלאקים יכולים להיות מתחומים של ערכים מסויים מתי EDF עדיף יותר מBOUNDED DELAY והפוך. נמק למה.
ד.תציעו אלגוריתמים נוספים לניהול תור שמתאימים להיות ממומשים במודל ופונקציה מטרה מוגדרים. תוסיפו יתרונות וחסרונות לכל שלושת האלגוריתמים.

כל התשובות לסעיפים א'-ד' תוסיפו ל YOURID.PDF_2.

בחינה בעל פה אתם צריכים להבין בעיית בקרת הגודש ולהסביר תשובות לסעיפים א'-ד'.

התשובות לכל השאלות מגישים כקובץ ZIP YOURID.ZIP כולל בתוכו 2 ספריות, ספריה לכל שאלה. פגישת הזום עם הסברים:

https://zoom.us/rec/play/u5B8cuv7rDs3SNOVsQSDA_N7W420L6is0SqY_6YIyEvgB3YBNQCkM-BGYes-GLxfAhWCCgqILDIm03Ev?continueMode=true&_x_zm_rtaid=ViSvpBVbSQyp9-luCiV3KA.1595253575090.d4a68ef823cc93f87abdc1461e33c720&_x_zm_rhtaid=617