מבוא לרשתות מחשבים (236334)

סיכום החומר בקורס "מבוא לרשתות מחשבים" (מדמ"ח) בטכניון

סיכם: בוריס צ'רקסקי – The Factor Squad

הבהרה: מסמך זה מסתמך במידה רבה על הקורס "מבוא לרשתות מחשבים" בטכניון, אך אינו חומר רשמי של הקורס, אלא סיכום אישי בלבד. המקורות לכתיבת המסמך הם ההרצאות והתרגולים, והזכויות שמורות לפקולטה למדעי המחשב בטכניון ולמוריה.

המסמך נכתב על ידי **בוריס צ'רקסקי**

תוכן עניינים

פרוטוקולים להצפת מידע ברשת

המטרה – להעביר הודעה בכל הרשת (Broadcast) תחת ההנחה שכל רכיב ברשת מכיר את שכניו המידיים.

Propagation of Information – PI

מימוש הפרוטוקול:

- קבל הודעה
- עבד את ההודעה
- הפץ את ההודעה לכל השכנים

תכונות הפרוטוקול:

- הפרוטוקול רץ ברמת הרשת ועל כן לא מטפל בשגיאות בקו (ששכבת הקו עושה)
- כל התחנות יקבלו את ההודעה בזמן סופי במידה והרשת קשירה
- כיוון שהפרוטוקול משתמש בדגל שבודק אם התחנה קיבלה כבר את ההודעה, ואין איפוס שלו, לא ניתן להריץ את הפרוטוקול יותר מפעם אחת
- ההודעה המופצת ברשת היא אותה ההודעה שנשלחה ע"י היוזם, כלומר אין שינוי של ההודעות בדרך.

Propagation of Information with Feedback – PIF

המטרה – להעביר מידע בכל הרשת ולתת חיווי ליוזם על התקדמות הפרוטוקול.

מימוש הפרוטוקול:

- קבל הודעה ושלך אותה לכל השכנים (נסמן את מספר ההודעות שנשלחו ב e_i)
- כאשר תתקבל הודעה מאחד השכנים, נקטין את e_i באחד.
  - אם $e_i = 0$ נשלח feedback למחשב שממנו התקבלה הודעת המקור

תכונות הפרוטוקול:

- הפרוטוקול רץ ברמת הרשת וכל כן לא מטפל בשגיאות בקו (ששכבת הקו עושה)
- כל התחנות יקבלו את ההודעה בזמן סופי במידה והרשת קשירה
- ההודעה המופצת ברשת היא אותה ההודעה שנשלחה ע"י היוזם, כלומר אין שינוי של ההודעות בדרך
- כשאר יוזם הפרוטוקול קיבל חיווי – הפרוטוקול הסתיים וניתן להסיק שכל ההודעות ברשת קיבלו את ההודעה

גילוי ותיקון שגיאות בשכבת הקו

אחד התפקידים החשובים של שכבת הקו הוא לזהות שגיאות שנפלו במידע שמועבר.

מרחק המינג

מספר הביטים השונים בין שתי מילים.

כדי לגלות d שגיאות, יש להבטיח שמרחק המינג בין כל שתי מילים יהיה לפחות $d+1$.

כדי לתקן d שגיאות, יש להבטיח שמרחק המינג בין כל שתי מילים בשפה יהיה לפחות $2d+1$.

קוד לתיקון שגיאה אחת

נתייחס למילים באורך m ביטים, כלומר מספר המילים החוקיות הוא 2^m.
נסמן את מספר הביטים שיש להוסיף לצורך הגנה (תיקון שגיאה) ב r.
נסמן את אורך המסגרת ב n וכן יתקיים $n=m+r$. בשפה יש 2^m מילים, וכל אחת מהן מיוצגת ע"י מחרוזת בת $n=m+r$ ביטים אך כיוון שיכולה ליפול שגיאה אחת במחרוזת, המילה בעצם יכולה להיות מיוצגת ע"י $1+n$ מחרוזות (המחרוזת המקורית + כל האפשרויות לסדר שגיאה אחת במילה).

לכן, כדי שנוכל לתקן x שגיאות נדרוש: $(1+n)2^m \le 2^n$, והתנאי על r שנקבל לגילוי שגיאה אחת הוא $(1+m+r) \le 2^r$.

קוד המינג לתיקון שגיאה אחת[1]

קוד המינג מקיים את התנאי שדרשנו לעליל, בו כל "ביט הגנה" מהווה ביט זוגיות עבור הביטים עליו הוא מגן.
בקוד המינג נמקם את סיביות הבקרה בביטים ה 2^i לכל i שמקיים $2^i \le n$. כעת כדי לקבוע אילו ביטים יגנו על כל ביט מידע, נפרק אותו להצגה הבינארית שלו, וכל ביט i שערכו 1 מסמן כי הביט ה 2^i בהודעה מגן עליו[2].

מבנה הודעה שנשלחת עבור $n=15, r=4, m=11$ וההודעה $D_1..D_{11}$ הוא: ($C_1..C_4$ הם הביטים לתיקון שגיאה)

C1	C2	D1	C3	D2	D3	D4	C4	D5	D6	D7	D8	D9	D10	D11

שימוש בקוד המינג לגילוי ותיקון רצף של שגיאות

נניח ואנו רוצים לשדר n הודעות אחת אחרי השניה, וכן ידוע כי יתכנו רצפים של שגיאות על הקו.
אם נשלח את ההודעות אחת אחרי השניה – עלול להווצר מצב בו היה רצף גדול של שגיאות בהודעה בודדת, ולא נוכל לזהות זאת. לעומת זאת, אם נסדר את ההודעות, אחת מעל השניה (תחת ההנחה שהן באורך זהה), ונשדר עמודה מהטבלה, נקבל כי גם במקרה של רצף של שגיאות, השגיאות יתפשטו על פני כל ההודעות, כלומר בכל הודעה יהיה מספר קטן של שגיאות. כיוון שקוד לזיהוי (ותיקון) שגיאות עובד ברמת ההודעה (ולא עמודה) – נוכל לזהות (ואולי אף לתקן) את השגיאה.

CRC – Cyclic Redundancy Check

כיוון שמעל $\mathbb{Z}_2$ פעולת חיבור וחיסור זהות – מתבצעות כ XOR, נגדיר התאמה בין מילים בינאריות לפולינומים מעל $\mathbb{Z}_2$. עבור המילה הבינארית $b=b_1..b_n$ נגדיר את הפולינום p באופן הבא: $x^i \in p \Leftrightarrow b_i = 1$.

אופן הפעולה:

- המשדר והמקלט מסכימים על פולינום יוצר $G(x)$ שדרגתו g
- המשדר רוצה לשדר את ההודעה $M(x)$
- המשדר ישדר את $T(x)$ כאשר $T(x) = x^g M(x) - \left[\left(x^g M(x) \right) \% G(x) \right]$

---

[1] נכונות הקוד נובעת מכך שלכל מספר יש פירוק בינארי יחיד.
[2] למשל עבור m = 11 נקבל r =4 ו n = 15. סיביות ההגנה יהיו 1,2,4,8. ולמשל עבור 13 (בבינארי 1101) הביטים שמגנים עליו הם ביטים מס 1,4 ו 8.

- נסמן את השגיאה ב $E(x)$ ויתקיים $x^i \in E(x) \Leftrightarrow$ יש שגיאה בביט ה –i בהודעה המקורית
- אם יש שגיאה[3], המקלט יקבל את ההודעה $T'(x) = T(x) + E(x)$
- המקלט יחשב את $T'(x) \% G(x)$
  - o אם השארית שונה מאפס – חלה שגיאה
  - o אם השארית שווה לאפס, המסגרת תקינה והמקלט יחשב את $M(x)$ באופן הבא: $T'(x)/x^g$ - הורדת g הביטים הנמוכים של ההודעה.

נשים לב כי שארית שווה אפס לא מבטיחה את תקינות ההודעה, עדיין ייתכנו שגיאות שלא גילינו, ובצורה מדוייקת יותר – שגיאה לא תתגלה כאשר הודעת השגיאה מתחלקת ללא שארית בפולינום היוצר, כלומר $\exists Z(x) \in \mathbb{Z}_2[x] : E(x) = G(x)Z(x) \Leftrightarrow E(x) \% G(x) = 0$.

Application
Transport
IP
MAC
Physical

יכולת גילוי השגיאות של CRC

- אם ב-$G(x)$ יש יותר מרכיב אחד – ניתן לגלות שגיאה בודדת
- אם ב - $x^0 \in G(x)$ וכן דרגתו של $G(x)$ גדולה מאחד – ניתן לגלות שתי שגיאות
- אם $G(x) \% (x+1) = 0$ ($G(x)$ מתחלק ללא שארית ב $x+1$) – ניתן לגלות מספר אי זוגי של שגיאות

מודל השכבות

התקשורת באינטרנט מתבצעת בעזרת 5 או 7 רמותשכל אחת מהן נותנת שירותים לרמות שמעל.

תפקידי השכבות והשירותים שהן נותנות:

1. Application– הרמה הגבוהה ביותר בה בעצם מתבצע מימוש האפליקציה שרוצה המשתמש.
2. שכבת התובלה - Transport– אחראית על חלוקת החבילות שמתקבלות משכבת האפליקציה לחבילות קטנות יותר, דאגה לכך שהחבילות יגיעו ליעדן בשלמותן, ללא טעויות, כפילויותושינויים בסדר.
3. IP– החבילה אחראית על ניתוב חבילות ברשת דרך מחשבי ביניים.
   - ההבטחה היחידה שנותנת שכבה זו לשכבת התובלה היא שאם חבילה תגיע למקום כלשהו – היא תגיע ליעד הנכון שלה.
4. שכבת בקרת הקו – אחראית לשליחת החבילה בין מחשבים סמוכים (ברשת אליה המחשב מחובר), דואגת לשלמות ההודעה והעברתה (גם במקרה של ערוץ רועש תוך כדי התמודדות עם התנגשויות)
   - ההבטחה של שכבה זו לשכבות מעל היא שהחבילות יגעו ליעדן תוך כדי שמירה על שלמותן, כלומר שהחבילותיגיעו ליעדן ללא שגיאות
5. השכבה הפיסית –physical– שכבה זו אחראית להעברת המידע ברמת החומרה, ומבטיחה שהעברת ההודעות היא FIFO, כלומר שאם חבילה X שנשלחה לפני חבילה Y אז חבילה X תגיע ליעדה לפני חבילה Y.

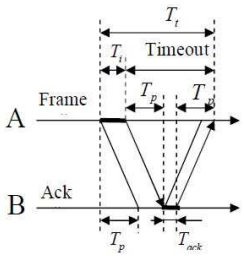
פרוטוקולי ARQ בשכבת הקו

סימונים ומושגים מקובלים

- T_i - זמן שידור המסגרת בתחנת המקור

[3] שגיאה = "התחלף הביט ה - i" – זה שקול לחיבור עם פולינום שבו יש 1 בביט ה –i.

- T_p - זמן התפשטות ההודעה על הקו
- T_a - הזמן שלוקח לשדר חבילת ackבצד המקבל (לא תמיד אפס!)
- RTT – Round Trip Time –הזמן שעובר מרגע ששולחים הודעה, עד שמגיע ack עליה ($RTT = 2T_p + T_a$)
- T_{out} - timeout –זמן ההמתנה המקסימאלי ל ack– הזמן שיעבור עד שננסה לשלוח את המסגרת מחדש, נמד מרגע סיום השידור (במצב האופטימאלי $T_{out} = 2T_p$)
- $T_t = T_i + T_{out}$ - זמן המחזור
- $\beta = \frac{RTT}{T_i} = \frac{2T_p + T_a}{T_i}$



הגדרת הניצולת בשכבת הקו

מספר החבילות שעולות לשכבת הרשת של התחנה המקבלת בזמן שידור מסגרת בודדת (T_i).

Stop and Wait

- התחנה השולחת מוסיפה מספר סידורי[4] לחבילה, שולחת אותה וממתינה לחיווי (ack).
- התחנה המקבלת, אם קיבלה את המסגרת שולחת ack עבור החבילה שקיבלה עם המספר הסידורי שקיבלה (RN)+1. נדגיש כי התחנה המקבלת שולחת את <u>מספר ההודעה שהיא מצפה לקבל כעת</u>, כך, במקרה של כשלון (למשל timeout או שמגיעה הודעה לא נכונה), התחנה המקבלת שולחת תמיד את אותו הדבר – החבילה הממוספרת שהיא מצפה לקבל כעת. ורק כאשר היא מקבלת חבילה זו, היא שולחת ACK, אחרי שהגדילה את ה-RN שלה ב-1

שתי התחנות מחזיקות שדהtimeout. במידהוה timeout פקע, החבילות ישודרו מחדש.

ניצולת $S = \frac{1-p}{1+\beta}$ כאשר p היא ההסתברות לשגיאה בקו ו $\beta = \frac{RTT}{T_i}$.

Go Back N(GBN)

- התחנה שולחתN חבילות שונות לפני שהתחנה תעבור להמתנה לחיווים.
- התחנה המקבלת יכולה לקבל <u>חבילה אחת בלבד</u> - זו עם המספר הסידורי הקטן ביותר, ולכן אם זו נאבדה, שאר התחנות שהגיעו יזרקו.
- בעת קבלת הודעה תקינה(כלומר, שהגיע חיווי מתאים), נקדם את החלון "חריץ" אחד קדימה
- אם לא התקבל חיווי עבור חבילה כלשהי לאחר timeout, נשלח את כל החלון מחדש
- Timeout, קיימות שתי גישות:
  - התחנה השולחת יכולה להחזיק timer גלובאלי, וברגע שזה פקע כל החלון ישלח מחדש

[4] תחת ההנחה שהשכבה הפיסית שומרת על FIFO, ניתן להסתפק במספרים סידוריים מודולו 2.

- התחנה השלוחת יכולה להחזיק timer לכל חבילה בחלון, וברגע שיפקע timer המסגרת הבודד לה שייך הtimer תשלח מחדש

גודל חלון אידאלי: $N = \beta + 1$.

ניצולת $S = \frac{1}{\gamma} = \frac{1-p}{1+\beta \cdot p}$, הניצולת מתרחקת מהאידאל 1-p ככל ש βp (תוחלת מספר המסגרות השגויות בזמן השווה ל round trip time) גדל.

זמן שידור ממוצע של מסגרת עד שהיא מגיעה ליעדה $\frac{1+\beta p}{1-p}$ כשאר γ הוא מספר החבילות הממוצע הנשלח

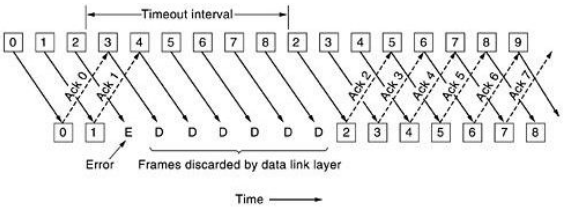
בפרק זמן γ_i - הזמן מרגע שחבילה i הגיעה לתחילת החלון, ועד שהחבילה ה i נשלחה בהצלחה.

Selective Repeat

- בהמשך ל GBN– התחנה המקבלת שומרת חלון בגודל M ותקבל עד M מסגרות שנשלחו אליה, גם אסנאבדה חבילה עם מספר סידורי קטן יותר במהלך הדרך
- התחנה המקבלת שולחת חיווים בנוגע לחבילה לה היא מחכה (זו עם המספר הסידורי הקטן ביותר), וכן מוסיפה מידע אודות החבילות שהיא שומרת אצלה – החבילות שאין צורך לשדר שוב (עד M חבילות)

גודל החלון:

- אם ידוע כי כל ההודעות מגיעות ליעדן, גודל החלון האופטימאלי הוא $\beta + 1$
- אם ידוע כי כל השידורים החוזרים של הודעה משובשת מגיעים ליעד, גודל החלון האופטימאלי הוא $2\beta + 1$
- אם ידוע כי כל השידורים החוזרים של הודעה המשובשת i פעמים מגיעים ליעד, גודל החלון האופטימאלי הוא $(i+1)\beta + 1$



תורת התורים

נתייחס לקו תקשורת כאל "תור" אליו נכנסות חבילות, שמחכות לקבל שירות מהקו (שידור עליו). ניתן לאפיין את התנהגות הקו (זמן ההמתנה הממוצע, ממוצע מספר החבילות הממתינות וכו') לפי קצב השידור בו, קצב הגעת החבילות לקו ומספר\סוג השרתים.

כמו כן נגדיר את "זמן השירות" של חבילה להיות הזמן שלוקח למשדר לשדר אותה. זמן זה תלוי באורך ההודעה וקצב השידור של המשדר.

תהליך פואסוני – תהליך בו הלקוחות מגיעים בקצב אקראי לחלוטין.
בתהליך פואסוני ההסתברות שבזמן t יגיעו n חבילות כאשר מגיעות כ λ חבילות בממוצע ליחדת זמן היא

$P_n(t) = \frac{(\lambda t)^n}{n!} e^{-\lambda t}$, ולכן ממוצע מספר ההופעות של חבילות בתהליך פואסוני בזמן t הוא $E(t) = \lambda t$.

תור M/M/1

תור M/M/K/N משמעותו – תור עם לקוחות שמגיעים פואסונית, זמן שירות מפולג פואסונית, עם K שרתים ו N לקוחות לכל היותר.

התור הנ"ל מאופיין ע"י:

- קצב הגעת החבילות לתור מפולג פואסונית עם פרמטר λ
- קצב השירות מפולג פואסונית עם פרמטר μ , כלומר זמן השירות מפולג אקספוננציאלית עם פרמטר $\frac{1}{\mu}$
- התור יכול להכיל אינסוף חבילות וכן הטיפול בחבילות הוא FIFO
- קיים שרת יחיד

דיאגרמת מצבים

ניתן לתאר תהליך של הגעה וטיפול במשימות בתור בעזרת דיאגרמת מצבים. מצב i מתאר מצב בו יש i משימות בתור.

- המעבר בין מצב i למצב i+1 מתבצע ע"י λp_i - ההסתברות שיהיו i לקוחות כפול קצב הגעת הלקוחות
- המעבר בין מצב i למצב i-1 מתבצע ע"י μp_i - ההסתברות שיהיו i לקוחות כפול קצב שירות הלקוחות

עבור כל חתך בדיאגרמת המצבים מתקיים – סכום ההסתברויות על הקשתות היוצאות = סכום ההסתברויות שעל הקשתות הנכנסות, וכן $\sum_{i\in|digram|} p_i = 1$, ומכאן ניתן לחשב את כל ההסתברויות של הדיאגרמה.

חתך בדיאגרמה הוא כל קו סגור שניתן לצייר על הדיאגרמה.

המצב היציב

ההסתברות שמספר הלקוחות שיקבלו שירות הוא n ללא תלות בזמן, כלומר שהתור "לא מתפוצץ", במצב היציבמתקיים $\lim_{t\to\infty} P_n(t) = P_n$ כלומר, $\frac{dP_n(t)}{dt} = 0$. התנאי למצב יציב הוא: $\rho = \frac{\lambda}{\mu} < 1$ (ניצולת השרת : מספר החבילות המגיעות מנורמל בזמן השירות שלהן), ועבורו מתקיים $P_n = \underbrace{(1-\rho)}_{P_0}\rho^n$.

תוחלת מספר החבילות בתור היא $E(n) = \sum_{n=0}^{\infty} nP_n$ וכאשר $\rho \to 1$ "התור מתפוצץ" (התוחלת שואפת לאינסוף).

- במצב היציב בתור M/M/1 נקבל $E(n) = \frac{\rho}{1-\rho}$

משפט ליטל (Little)

לכל מערכת תורים שהיא work conservative[5] (ובפרט M/M/1) מתקיים: $E(n) = E(\lambda)\cdot E(t)$.

- $E(n)$ - תוחלת מספר החבילות בתור, כולל זאת שמקבלת שירות
- $E(t)$ - זמן השהייההממוצע עד לקבלת שירות (הזמן להמתנה + הזמן לשירות)

[5] תור הוא work conservative אמ"מ השרת לא עובד רק כאשר אין ממתינים (כלומר כל עוד יש חבילות ממתינות, השרת ייתן להן שירות)

- $E(\lambda)$ - תוחלת זמני ההגעה לתור = $\sum_{i=1}^{\infty}\lambda_i \cdot p_i$ (בהנתן ויש λ שונים)

עבור תור M/M/1 נקבל ע"פ משפט ליטל כי $E(t)=\frac{1}{\mu-\lambda}$.

זמן <u>ההמתנה</u>(ללא שירות) בתור M/M/1 עבור μ קבוע הוא $T_Q=E(t)-\frac{1}{\mu}$.

שרת אחד לעומת n שרתים

בהנתן תור בודד עם פרמטרים λ ו μ , וכן מערכת בעלת n תורים עם פרמטרים $\frac{\lambda}{n}$ ו $\frac{\mu}{n}$ הגיוני לחשוב כי המודל השני עדיף, אך במידה והגעת הלקוחות היא לא אחידה בין השרתים, נקבל כי תוחלת זמן ההמתנה עלולה לגדול, וזאת כיוון שהמערכת לא תהיה work conservative– יהיו שרתים מובטלים על אף שיש לקוחות ממתינים (זאת כי הלקוחות ממתינים בתור אחר מהתור של השרת המובטל).

אם זאת, הוספת שרתים תקטין בממוצע את העומס על כל אחד מהשרתים (על אף שזמן ההמתנה יגדל).

חלוקת ערוץ שידור משותף

קיימות 2 גישות לחלוקת ערוץ משותף:

1. חלוקה סטאטית – הערוץ יחולק ל n תת-ערוצים, וכל תחנה תקבל תת-ערוץ. לגישה זו יש 2 תת גישות:
   1.1. FDM– חלוקת ערוץ לפי תדרים. ערוץ ברוחב C יחולק ל $\frac{C}{N}$ ערוצים ברוחב $\frac{C}{N}$
   1.2. TDM– חלוקת ערוץ לפי זמנים. ערוץ ברוחב C, יחולק לN פרקי זמן וכל תחנה תשדר $\frac{1}{N}$ זמן, אך על <u>כל</u> רוחב הערוץ
2. תחרות על הערוץ – תחנה שרוצה לשדר תקבל את <u>כל</u> הערוץ.

פרוטוקולים בשכבת בקרת הקו – פרוטוקולי MAC

ALOHA

פרוטוקול שתקף בערוצים בהם כל התחנות יכולות להאזין על הערוץולשמוע התנגשויות.
הפרוטוקול:

- שלח מסגרת
- האזן על הקו, אם הייתה התנגשות – המתן זמן אקראי ושדר שוב (וכן הלאה...)

ניצולת $S=G\cdot e^{-2G}$ - כאשר G הוא ממוצע מספר המסגרות שמשודרות על הקו (התפלגות פואסונית עם ממוצע G, או לחלופין, קצב הגעת הודעות G). פונק' זו מקבלת מקס' של 18% עבור G=0.5.

אם מספר התחנות סופי נקבל כי הניצולת היא $S=G\left(1-\frac{G}{N}\right)^{N-1}$

Slotted ALOHA

"ציר הזמן" מחולק לחריצים בגודל זמן שליחת מסגרת ולתחנות מותר להתחיל לשדר רק בתחילת חריץ.
כדי שלא תהיה התנגשות במהלך חריץ i-1 צריכה להיות בדיוק תחנה אחת שתחכה לשדר.

ניצולת $S = G \cdot e^{-G}$ עם מקס' של 36% עבור G=1.

CSMA

כמו ALOHA– דורש שכל התחנות יכולות להאזין על הקו ולזהות התנגשויות. הפרוטוקול:

האזן על הקטן, אם הערוץ שקט – שדר, אחרת:

- Persistent – 1– התחנה ממתינה עד שיהיה שקט ומשדרת מיד. אם הייתה התנגשות, התחנה מחכה זמן אקראי ומנסה לשדר מחדש
- Non – Persistent– התחנה ממתינה זמן אקראי לפני שתנסה להאזין מחדש

לפי Binary Exponential Backoff Algorithm נקבל כי לאחר i התנגשויות התחנה מגרילה יוניפורמית זמן המתנה בתחום $\left[1, 2^i - 1\right]$. לפי תקן10<Etherneti וכן אחרי 16 התנגשויות החבילה תיזרק.

CSMA/CD - Collision Detection

כמו CSMA פרט לכך שאם התחנה מתחילה לשדר ומזהה התנגשות – היא מפסיקה את השידור שלה (פחות זמן מבוזבז במקרה של התנגשות)

ניצולת $S = \frac{1}{1 + \frac{2\tau(1-A)}{A \cdot T}}$ כאשר τ הוא זמן ההתפשטות המקסימאלי בערוץ.

- $A = k \cdot p(1-p)^{k-1}$ ההסתברות להתחלה של שידור מוצלח בחריץ כלשהו על ערוץ שמשתמשות בו k תחנות וההסתברות לשידור בחריץ כלשהו של תחנה כלשהי היא p.

נשים לב שבפרוטקול זה שילוב של הקטנת המרחק בין התחנותוהקטנת גודל החבילותעלול להביא לכך שתחנות לא יזהו התנגשויות. זאת כיוון שהקטנת המסגרות מתחת לחסם תחתון של 2τ (τ = זמן התפשטות) יביא לכך שתחנות שיאזינו על הקו ישמעו שקט – יסיימו לשדר, וימשיכו לשמוע שקט (כאשר בפועל בזמן השידור שלהן, גם תחנה אחרת שידרה) – לא יהיה זיהוי של ההתנגשות.
הקטנה המרחק משפיע ישירות על זמן השידור ולכן ההשפעה דומה.

פרוטוקולים להזמנת ערוץ

Reservation ALOHA

פרוטוקול ש"מרחיב" את Slotted ALOHA באופן הבא:

- תחנה שרוצה לשדר מסגרת באורך X סלוטים תנסה לשלוח על הערוץ לפי פרוטוקול Slotted ALOHA הודעת בקרה שמשמעותה "אני עומדת לשדר X סלוטים"
  - אם הודעת הבקרה נשלחה בהצלחה (הרי שכל התחנות שמעו אותה לפי הנחת הבסיס על הערוץ שעליו עובד ALOHA), התחנה יכולה לשדר ב X הסלוטים הבאים ללא הפרעה

- אם הייתה התנגשות בהודעת הבקרה – התחנה תחכה מספר רנדומאלי של סלוטים שלא מוזמנים לפני שתנסה להזמין את הערוץ שוב

נעדיף להשתמש ב Reservation ALOHAעל CSMA/CD בערוצים בהם יש "תחנה מנהלת" או בערוצים בהם התחנות לא שומעות אחת את השנייה (ולכן CSMA/CD לא יעבוד).

Bit Map

כל תחנה משדרת בתורה 1 אם יש לה מה לשדר, ו0 אחרת – שידור המפה. קיים חריץ מיוחד לשידור המפה, ולאחריו יבוא חריצי המידע.
לאחר שידור המפה, כל התחנות שצריכות לשדר, משדרות לפי הסדר – אין התנגשויות כי כל התחנות יודעות מתי תורן לשדר.

ניצולת $S = \frac{d}{d+1}$ כאשר d הוא היחס בין זמן שידור data לשידור הדגל במפה⇐השיטה יעילה כאשר d גדול.

שידור קול ברשת cellular לפי תקן GSM

קיימת תחנת בסיס, וכן תחום התדרים מחולק ל 248 רצועות. 124 רצועות לשידור לתחנת הבסיס (ערוץ עולה), ו124 רצועות לשידור מתחנת הבסיס (ערוץ יורד). כל רצועה מחולקת ל 8 ערוצי קול – 1 לכל שיחה. על כל רצועה משודרות מסגרות המכילות 8 חריצי זמן⇐קצב השידור הוא $270.8\frac{Kb}{\sec}$.

לכל שיחה מוקצה חריץ קבוע בערוץ העולה ובערוץ היורד – כל שיחה מקבלת ערוץ דו כיווני.

Reservation ALOHA ברשתות GSM

אחת ל 120ms משודרת הודעת בקרה על כל אחתמרצועות התדרים. מסגרת זו משמשתלהחלפת מידע בינהתחנות המשדרות לתחנת הבסיס (הזמנת ערוץ לפי reservation ALOHA למשל).

- תחנת הבסיס מסמנת לכל סלוט האם הוא תפוס (וכן ע"י איזו תחנה), אם לא – כל התחנות רשאיות להתחרות עליו
- תחנות מתחרות על סלוטים פנויים כדי לשלוח לתחנת הבסיס בקשות להזמנת ערוץ

DOCSIS

מיועד לרשתות כבלים בעלות טופולוגית עץ – יש "תחנת קצה"

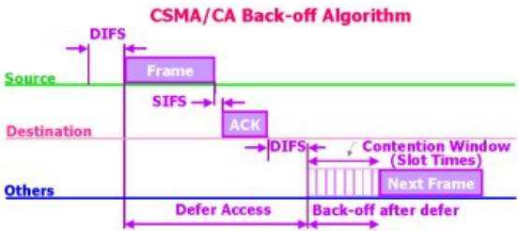
- שימוש בשני ערוצים, ערוץ עולה בשביל התחנות שמתחרות על שליחה, וערוץ יורד בשביל ה-CMTS
- יש שימוש ב Reservation ALOHA.
- בתחנת הקצה יושבת לוגיקה בשם CMTS שאחראית על הקצאת הסלוטים
- כל סלוט בערוץ העולה יכול להיות בעל אחד משני סטאטוסים – פנוי לכל התחנות, או מוזמן עבור תחנה X
- כל תחנה רשאית לשדר בקשות להזמנת ערוץ מהCMTS על הערוץ העולה
- כיוון שהתחנות לא יכולות להאזין לזו, הן אינן יודעות על התנגשויות ועל כן חייבותלחכות לחיווי מהCMTS האם הזמנתן התקבלה או לא
  - אם ההזמנה התקבלה – הCMTS מודיע לתחנה כמה סלוטים היא קיבלה (לא בהכרח כמה שהיא ביקשה)
  - בקשה יכולה להתקבל גם אם הCMTS לא החליט כמה זמן להקציב לתחנה, ותשלח הודעת חיווי מיוחדת

- אם חיווי לא התקבל, התחנה תנסה להזמין את הערוץ שוב לפי Binary Exponential Backoff
- אם תחנה קיבלה אישור לשדר, היא יכולה לבקש בעזרת (piggybacking) עוד סלוטים לשידור

אופני תזמון ב DOCSIS

האופן בו תחנת ה CMTS מחליטה על הקצאת הסלוטים לתחנות. קיימות מספר גישות:

1. FCFS – First Come First Served– כשמו כן הוא
   - קל למימוש
   - לא נותן תמיכה במתן עדיפויות לתחנות
2. RR - Round Robin (נקרא גם unsolicited Grant Service)– זמן השידור מחולק יוניפורמית בין כל התחנות
   - תזמון שמבטיח הוגנותבין כל התחנות
   - לא נותן תמיכה במתן עדיפויות לתחנות
3. WRR – Weight Round Robin– כמו RR רק מאפשר מתן מספר סלוטים שונה לכל התחנות (אבל עדיין כל התחנות מקבלות זמן שידור)
   - תזמון שמבטיח הוגניון בין כל התחנות
   - מאפשר מתן עדיפויות
4. EDF – Earliest Deadline First– ה CMTS קובע את הזמן המקס' שבוחבילה צריכה להיות משודרת, וזאת עם ה deadline הקרוב ביותר תשודר ראשונה



CSMA/CA

פרוטוקול DCF (Distributed Coordination function) של תקן ה wireless ,

הבעיות במימוש CSMA/CD ברשתות wireless הוא הסתברות גבוהה מאוד לשגיאה, וכן חוסר היכולת להקשיב ולשדר בו זמנית (תקן 802.11). הפרוטוקול פועל כמו CSMA/CD עם השינויים הבאים:

- בעת האזנה לערוץ – אם הוא תפוס, נחכה מיד זמן אקראי לפי Binary Exponential Backoff
- שימוש ב Stop & Wait ברמת הMAC– אחרי שידור מסגרת נחכה ל ack ואם זה לא הגיע, נשדר את אותה המסגרת שוב
- Fragmentation ברמת ה MAC– פירוק הודעה להודעות קצרות יותר (הודעות קצרות פחות פגיעות)
- בפעם הראשונה בה תחנה רוצה לשדר היא מחכה זמן השווה ל DIFS. אם הערוץ שקט – תשדר, אחרת תעבור לתהליך להאזין על הקו שוב לצורך קבלת ack
- התחנה המקבלת מחכה זמן SIFS ומשדרת ack
- DIFS > SIFS ולכן מובטח שהתחנה המקבלת תוכל לשדר ack ללא הפרעה

Physical carrier sense– כשכל התחנות שומעות זו את זו.

Hidden terminal effect– כאשר 2 התחנות נמצאות בטווח שידור של access point, אך לא בטווח שידור זו של זו (שידור של שתיהן ביחד לaccess point יכשל למרות שהן לא ידעו זו על זו).

Exposed terminal effect– כאשר יש 2 זוגות של תחנות שלא נמצאות בטווח זו של זו, אך הפקוטוקול לא יאפשר שידור שלהן במקביל.

Virtual Carrier Sense:

- תחנה A שרוצה לשדר משדרת RTS (Request to send) ל B. כל תחנה בטווח של A שתשמע הודעה זו לא תתחיל לשדר כדי לתת לA את האפשרות לשדר
- אם B מסכימה לקבל את ההודעה של A היא שולחת הודעת CTS (Clear to Send). כל תחנה בתחום של B תשמע את ההודעה הזאת ולא תתחיל לשדר כדי לתת ל B את האפשרות לקבל את ההודעה
- אם A קבילה CTS היא תשדר ותחכה ל ack (אם אחד כזה לא הגיע זה ייתכן רק בגלל שגיאה, כי הפרוטוקול מכסה התנגשויות) – הפרוטוקול יתחיל מחדש

בעצם כל ההודעות בטווח של A ישמעו את RTS, כל ההודעות בטווח של B ישמעו את CTS ולכן התשדורת בין A ל B תהיה בטוחה מהתנגשויות.

נשים לב כי CSMA/CA לא מבטיח שלא יהיו התנגשויות. סוגי ההתנגשויות הן:

	RTS	CTS	DATA	ACK
RTS	אפשרי	אפשרי	אפשרי	לא אפשרי
CTS	אפשרי	אפשרי	אפשרי	לא אפשרי
DATA	אפשרי	אפשרי	אפשרי	לא אפשרי
ACK	לא אפשרי	לא אפשרי	לא אפשרי	לא אפשרי

Point Coordination Function של תקן ה wireless

הפרוטוקולעובד עם תחנה מרכזית ששולחת הודעות בקרה (beacon) על הערוץ בפרקי זמן קבועים ובכך שולטת על הרשת (כמו CMTS ב DOCSIS).

SONET/SDH

טופולוגית טבעת לשידור voice– הערוץ מחולק לחלקים, כל חלק מוקצה לתשדורת דו כיוונית בין זוג תחנות כלשהן. בשביל תשדורת סינכורנית משתמשים בערוץ ללא החלוקה הווירטואלית הנ"ל.

Resilient Packet Ring (RPR)

תקן לשידור מידע סינכרוני על רשתות טבעתיות.
לכל מסגרת מצמידים שדה מקור ויעד ברמת הMAC. כשמסגרת מגיעה לתחנה, היא בודקת האם המסגרת מיועדת לה אם כן – מעבירה אותה לחוצץ מקומי ומפסיקה את שידורה , אחרת, מעבירה אותה בהמשך הטבעת (אלא אם כתובת המקור של המסגרת זו התחנה עצמה, במקרה של Broadcasts או שתחנת היעד לא נמצא).

רוחב פס– הקצב בו ניתן לשדר מסגרות חדשות על הקו = זמן השידור×מספר התחנות שמשדרות.

Buffer Insertion Ring (BIR)

- לכל תחנה שני חוצצים –buffer_a ו buffer_b
- Buffer_a משומש לטעינת מסגרות המיועדות לשידור מהתחנה
- Buffer_b מהווה חלק מהטבעת ומעביר מידע שעובר בה
- ברגע ש buffer_b ריק ניתן לשדר את המסגרות שנמצאת ב buffer_a על הטבעת
- מסגרת שמגיע בזמן השידור מ buffer_a לתחנה, תכנס ל buffer_b ותשודר בסוף השידורמ buffer_a

- גודל ה buffer‌ים הוא כגודל המסגרת הגדולה ביותר שניתן לשדר

המרחק הממוצע שמסגרת תעבור הוא $\frac{N}{2}$ (כלומרבממוצע שתי תחנות משדרות מ buffer_a) והניצולת היא 2 עבור התפלגות יעדים אחידה, עם רוחב פס של 4T הזמין לתחנות.

הניצולת בפרוטוקול זה מוגדרת להיות "קצב השידור של מידע חדש(מ buffer_a) לתוך הטבעת חלקי קצב השידור הכולל", כלומר $\frac{N}{d}$.

מנגון העדיפויות ב Buffer Insertion Ring ימומש בעזרת מספר buffer'ים – אחד לכל עדיפות.

BIR ו RPR דו כיווני

ב BIR דו כיווני בין כל 2 תחנות בטבעת יש 2 קווי תקשורת – אחד לכל כיוון.

כעת המרחק הממוצע שמסגרת תעבור קטן ל $\frac{N}{4}$ ולכן הניצולת היא 4 על כל אחת מ2 הטבעות, ורוחב הפס הזמין לכל התחנות הוא 4T על כל קו, ולכן 8T סה"כ.

ב RPRדו כיווני יש התמודדות בפני נפילות כיוון שבמקרה של נפילה, 2 גישות עקריות:

- Steering Mode– ברגע שיש נפילה שולחים את ההודעות דרך הקו שלא נפל
- Wrapping Mode–ניתן לעשות "פניית פרסה" ולשדר מסגרת לתחנה הקודמת במסלול, במקום לתחנה הבאה שנפלה. זאת מבוצע ע"י העברת מסגרות מטבעת אחת לשנייה. נשים לב כי Wrapping Mode עובד לפרק זמן כלשהו

Connecting LANs using Brigs

אלגוריתם הלמידה – כל גשר מאזין על הרשתות עליהן הוא מחובר, ומעדכן את טבלאות הניתוב שלו בהתאם לכתובת המקור של המסגרת שנשלחות עליו. אם הגשר לא מכיר את התחנה, הוא ישדר את החבילה על כל הרשתות להן הוא מחובר.

פרוטוקול לבניית עץ פורש

1. כל גשר מתחיל "מתעורר" ושולח הודעת בקרה BDPU שהוא השורש
2. כל גשר בוחר את ה root portלהיות הפורט שממנו התקבל ה BDPU
3. כל גשר שקיבל BDPU מה root port משדר על כל ה LAN-ים שהוא ה designated bridge שלהם BDPU שמכיל את המידע הבא:
   - הזהות של הגשר השולח והפורט שלו שאמור להיות ה designated port באותו הLAN
   - הזהות של הגשר השולח שחושב שהוא השורש
   - המחיר של המסלול שמוביל לשורש דרך הגשר השולח

פרוטוקול זה לא בהכרח מוצא עץ פורש מינימאלי![6]

[6] אחת הסיבות העיקריות היא שהשורש נבחר להיות זה עם המספר הסידורי הנמוך ביותר, שלא בהרכח מניב עץ מינימאלי

העברת הודעות באמצעות גשרים

כל גשר שומר לעצמו טבלאות ניתוב מהצורה של (מחשב – פורט), כלומר, על איזה פורט יש לשדר את המידע כדי להגיע למחשב המבוקש.
טבלאות אלה מתוחזקות תוך כדי התפשטות המידע ברשת – ברגע שגשר מקבל הודעה שהוא צריך להעביר, אם אין כניסה מתאימה בטבלאות שלו, הוא עושה Broadcast על כל הרשתות אליהן הוא מחובר(פרט לזאת שממנה הגיעה ההודעה). כמו כן, הגשר מוציא בחבילת ה MAC את ה source MAC address (נסמנהA) ומוסיף לטבלה שלו כניסה עבור הפורט שעליו התקבלה הודעה (נסמנו portX), והכתובת שממנה הגיעה הודעה (כי כעת הוא יודע מאיזה פורט הוא יקבל הודעות מהמחשב הנ"ל), כלומר תתווסף כניסה לטבלה מהצורה (portX–A).

התמודדות עם שינויי טופולוגיה

השורש (שכן ידוע מי הוא בגלל שהעץ הפורש כבר קיים) שולח מדי פעם הודעת Hello כל designated bridgeשולחהודעה זו על ה LAN שלו, אם הודעה זו לא מתקבלת תוךפרק זמן כלשהו, מתחילים את פרוטוקול בניית העץ מחדש.

TCP/IP

לא ניתן לחבר את כל העולם בגשרים בלבד מפאת גודלה של רשת האינטרנט (האלגוריתם למציאת עץ פורש למשל לא יתכנסלעולם על מיליארדי המחשבים שיש בעולם).
לעומת זאת, ניתן תאורטית ניתן לחבר את כל העולם ברשת של ראוטרים (במקרה זה כל מחשב יהווה subnet, ולכן צריך מספר עצום של כתובות IP), אך שימוש בגשרים וראוטרים ביחד מביא לניצול טוב יותר של מרחב כתובות הIP וכן להקטנה בגודל טבלאות הניתוב של הראוטרים (שכן שומר רק LAN-ים, שבהם יכולים להיות מאות מחשבים).

נשים לב כי TCP פועל בצורה "דומה" לבן כלאיים של Selective Repeat ו GBN בשכבת התובלה, אם זאת הוא מתבסס על הבטחות <u>חלשות יותר</u> של שכבת הIP שלא מבטיחות FIFO (כמו השכבה הפיסית)

מבנה האינטרנט

האינטרנט מורכב מהרבה רשתות מקומיות (LAN) שמחוברות ביניהן ע"י ראוטרים.
קבוצה כלשהי של רשתות מקומיות + ראוטרים משוייכים לרשות אדמיניסטרטיבית כלשהי שאחראית על הרצת פרוטוקולי ניתובבתוך ה"חלק הרשת" שלה, וכל פרוטוקולי ניתובבינה לבין שאר הישויות האדמיניסטרטיביות.
לכל ראוטר יש טבלת ניתוב שעל פיה הוא מחליט לאיזו רשת לנתב את החבילה שהגיעה אליו.

נשים לב שבשליחת הודעות באינטרנט, כתובות ה IP בחבילה נשארות זהות, ומשתנות כתובות הMAC במעבר על extended – LANs.

מחלקות של כתובות IP

כתובות הIP מחולקות למחלקות לפי גודל (מספר המחשבים) ברשת הפנימית (subnet):

שם	מרחב כתובות	מספר המחשבים	כמות	מבנה הכתובות			
מחלקה A	כתובות < 128.0.0.0	2^{24}	2^{8}	Subnet [0..8]	Host [8.31]		
מחלקה B	כתובות < 192.0.0.0	2^{16}	2^{16}	Subnet [0..15]	Host [16..31]		
מחלקה C	כתובות < 224.0.0.0	2^{8}	2^{24}	Subnet [0..23]	Host [24..31]		
מחלקה D	כתובות < 240.0.0.0			1110	Multi cast group – used for Bcast		

כתובות IPמיוחדות

- Directed Broadcast Address– שידור חבילה לכל רשת X (אפשרית רק אם המחשב המשדר נמצא ברשת X בעצמו, אחרת החבילה תזרק). מאופיינת ע"י network address = X ו host address = 11…111.
- Limited Broadcast Address– שידור חבילה לכל ה"רשת שלי" מאופיינת ע"י network ו host address 111...11
- This Host– אם גם ה host address וגם ה network address הם 00..00 אז החבילה נשלחה "ממני". כתובת זו יכולה להיות רק ב source address

תהליך שליחת החבילה ברשת

נניח כי A מחובר לראוטר R שמחובר לראוטר B, וכן A רוצה לשלוח הודעה ל B. תהליך השליחה הוא:
שכבת הIP של A מורידה את החבילה לשכבת הקו, שכבה זו מוצאת את כתובת ה MAC של R וכותבת אותה בשדה ה dest, ושולחת את החבילה לR. R מקבל את החבילה, שכבת הקו מוודאת שאכן החבילה יועדה לה ומעבירה אותה לרמת ה IP. רמת הIP רואה שכתובת היעד היא כתובת ה IP של B, לכן זו מורידה את החבילה לרמת הקו שמוצאת את כתובת הMAC של B ושולחת לה את החבילה, מכאן ההמשך ברור.

נשים לב:

1. שכאשר חבילת IP עוברת בראוטר, הראוטר מעדכן את שדה TTL(בדר"כ החסרה ב-1)ושדה ה checksum צריך להיות מחושב מחדש בהתאם.
2. מחשב יקבל הודעות limited broadcast בשכבת הרשת ובשכבת הקו מהרשת בא הוא נמצא, גם עם חלו טעויות בטבלאות הניתוב למיניהן

Address Resolution Protocol (ARP)

פרוטוקול לתרגום כתובות IP לכתובות MAC, יושב ישירות מעל ה Ethernet (מעל ה MAC ומתחת ל IP). אופן פעולת הפרוטוקול:

- אם תחנה A רוצה לדעת את כתובת הMAC של B, A תשלח מסגרת Ethernet המכילה בקשת ARP request– כתובת ה IP המבוקשת נשלחת ב Payload של ההודעה. הודעה זו נשלחת מכתובת A לכתובת MAC Broadcast
- כל התחנות ב LAN תקבל את ההודעה הזאת, אך רק שכבת הIP של B תזהה שזוהי כתובת הIP שלה, ותענה לA עם חבילת ARP replay בה כתובת ה MAC שלה

נשים לב כי תחנת המקור שולחת את הודעת ה ARP, ראוטרים בדרך לא יפיצו אותה. במידה והIP של היעד נמצא ב subnet של ראוטר מסויים, או שדרך ראוטר זה עובר הניתוב לרשת של היעד, ראוטר זה יענה, ויחזיר את כתובת ה MAC שלו (כיוון שהוא ידע כבר לנתב ליעד).

Dynamic Host Configuration Protocol (DHCP)

פרוטוקול ברמת האפליקציהשמאפשר להתחבר לאינטרנט "אוטומטית". הפרוטוקול מספק למחשב המבקש את הדברים הבאים:

- כתובת IP (שאותה שרת ה DHCP יכול "להשכיר" או להקצות לצמיתות")
- כתובת הIP של ה default router[7]

[7] נותנים את כתובת ה IP ולא כתובת ה MAC של הראוטר כדי שניתן יהיה להחליף כרטיס רשת לראוטר בלי להודיע לכל הרשת למשל

- כתובת ה IP של שרת ה DNS.

אופן פעולת הפרוטוקול:

- מחשב A שרוצה להתחבר לאינטרנט שולח הודעת DHCP discover בחבילת IP מ this host (כתובת 0) ב limited broadcast, שכבת ה MAC של מחשב A שולחת חבילת MACעם כתובת ה MAC של A ושולחת אותה לכתובת MAC Broadcast
- שרת ה DHCP שולח ל A הודעת DHCP offer בחבילת IP שנשלחת מה IP של השרת ב limited Broadcast, שכבת הרשת של השרת שולחת כעת את חבילה עם כתובת הMAC של השרת ב Mac Broadcast לכל הרשת. תפקידהשל הודעה זו הוא להציע למחשב A כתובת IP.
- מחשב A מקבל את הודעות ה DHCP offer מכל שרתי ה DHCP שבסביבתו (שכן ההודעה נשלחת ב broadcast ויכולה להקלט ע"י מספר שרתי DHCP). מחשב A בוחר DHCP אחד, ושולח לו הודעת DHCP request ב broadcast כמו בשלב הראשון (כי אין לו עדיין כתובת IP, למרות שאחת כבר הוצעה לו)
- שרת ה DHCP שמקבל את הודעת ה DHCP request מקצה את כתובת הIP ושאר הדברים הנדרשים למחשב A ושולח לו בחזרה הודעת DHCP ack
- בסוף תהליך זה A יודע את כתובת ה IP של השרת וכן הוא קיבל את כל המידע לו הוא זקוק כדי להתחבר לרשת

DHCP זה פרוטוקול שרץ מעל UDP ועל כן עליו לדאוג בעצמו להתאוששות מתקלות, כלומר הלקוח (A במקרה שלנו) צריך לדאוג לזה.

אלגוריתם קידום חבילות באינטרנט (forwarding algorithm)

זהו אלגוריתם שעל פיו מנותבות החבילות באינטרנט. כאשר הגיעה החבילה לראוטר זהו הפרוטוקול שנכנס לפעולה:

1. הוצא מחבילת הIP את כתובת הIP של היעד.
2. מתוך כתובת היעד חשב את כתובת ה subnet, כלומר כתובת ה LAN לה שייכת כתובת היעד
   2.1. אם ה subnet נמצא בטבלת הניתוב – שלח לאותו subnet.
   2.2. אם ה subnet לא נמצא בטבלת הניתוב – שלח ל default entry של הראוטר.
   2.3. אם לא קיימת default entry בראוטר – זרוק את החבילה.

Distance Vector Routing

פרוטוקול לבניית טבלאות הניתוב באינטרנט.עקרון פעולתו של הפרוטוקול הוא שאחת לפרק זמן כלשהו שולח כל router לשכניו את ווקטור המרחקים שלו – כלומר"כמה עולה" לנתב חבילות דרכו לכל הרשתות אליהן הוא מחובר. השכנים שמקבלים את הווקטור הנ"לצריכים לנסות לעדכן את הטבלאות שלהן בהתאם ל"מחירים הזולים ביותר".

בעיה שנוצרת מאלגוריתם זה, הוא ששכנים מודיעים זה לזה על הטבלאות שלהם, דבר שיכול ליצור בעיה במידה ושכן א' חושב שהוא מעביר חבילות לצד ג' דרך שכן ב', ואילו שכן ב' חושב שהוא מעביר הודעות לצד ג' דרך שכן א'. כדי להתמודד עם הבעיה הזאת משתמשים ב split horizon– לא שולחים לשכן עדכונים על מסלולים שעוברים דרכו. קל להבין שבעיה זו לא מתכנסת, כי עבור מעגל בגודל 3 נקבל שוב את אותה הבעיה, אך זהו פתרון מקומי אפשרי.

בעיה נוספת שקיימת היא העובדה ששולחים רק מרחקים בוקטור, ולא שולחים את הראוטר דרכו מושג המסלול עם המחיר הנקוב.

UDP

פרוטוקול שרץ בשכבת התובלה. פרוטוקול זה קל למימוש בגלל פשטותו וההבטחות המעטות שהוא נותן לשכבה שמעליו (שכבת האפליקציה).

- פרוטוקול זה מאפשר לגלות שגיאות, וכן מאפשר להריץ כמה אפליקציות בו זמנית.
- פרוטוקול זה לא מאפשר תיקון שגיאות, או שליחה מחדש של חבילות.

פרוטוקול UDP משתמש בפורטים (Ports) כדי לדעת לאיזו אפליקציה לשלוח את החבילה שקיבל.

TCP

פרוטוקול נוסף שרץ בשכבת התובלה, פרוטוקול זה חזק יותר מפרוטוקול UDP, ומבטיח יותר לשכבת האפליקציה.

ההבדל העיקרי בין TCP ל UDP הוא שפרוטוקול TCP מבטיח end to end reliability, כלומר הוא מבטיח שחבילות שיצאו ממחשב השולח יגיעו למחשב היעד בשלמותן ללא שגיאות (פונקציונליות דומה לזו של פרוטוקולי ARQ בשכבת הקו).

TCPvs UDP

נשים לב אם זאת שפרוטוקול TCP מסובך בהרבה מפרוטוקולי ARQ כיוון זה יושב מעל שכבת הIP שלא מבטיחה FIFO בשליחת הודעות, וכן הפרוטוקול עלול לעבודעם מספר גדול של אפליקציות בו זמנית. מסיבה זו גם התקורה (הזמן המבוזבז, מספר החבילות השנלחות) היא גדולה מאוד יחסית ל UDP, אם זאת, נרצה להשתמש בפרוטוקול זה כאשר האמינות והיציבות נחוצה לשכבת האפליקציה (כל שירותי ה HTTP וכו, ובכללי, כל מה שהוא לא Real Time).

לעומת זאת, UDP הינו פרוטוקל קל למימוש, ומהיר, התקורה בשימוש בו נמוכה, אך אין שום אמינות. באפליקציות בהן חשוב ה Real Time, ועיקוב בהגעת החבילות אינו מתקבל על הדעת נעדיף להשתמש ב UDP (שירותי ווידאו\אודיו וכו').

הקמת קשר TCP

מחשב A שרוצה ליצור תקשורת TCP עם מחשב B יעשה זאת באופן הבא:

1. מחשב A ישלח הודעת בקרה Syn למחשב B.
2. מחשב B ישלח הודעת בקרה Syn ack למחשב A.
3. מחשב A ישלח לB הודעת ack שמודיעה שהוא קיבל את Syn ack וההודעות הבאות יהיו הודעות data.
4. מחשב A ישלח את כל המידע שרצה לשלוח לB.
5. מחשב A ישלח הודעת Fin למחשב B שמודיעה על סוף התקשורת.

שלבים 3-1 נקראים TCP handshake (או triple handshake) והן מכילות מספרים סידוריים כדי לזהות תקלות.

End to end reliability in TCP

הדרך בה TCP מבטיח את ה end to end reliability היא 3 duplicate Ack's ו timeout.
השכבה ממומשת כשילוב של GBN ו Selective Repeat, ועל כן שולחת מספר כלשהו של חבילות, ומצפה לקבל Ack-ים על כולן. אם הדבר לא קורה (חבילה נאבדה או התעכבהבגלל הניתוב), אחרי שנקבל 3 ack-ים על חבילה

כלשהי, נניח שהיא נאבדה ונשלח אותה מחדש. כמו כן במידה ולא הגיעו 3 ack-ים, יפקע אחרי זמן מה ה timeout והחבילה תשלח מחדש.

בקרת גודש – Congestion Control

התאמת קצב שליחת ההודעות לעומס ברשת וקצב קבלת המידע של מחשב היעד, זאת כדי שהראוטרים בדרך לא יזרקו את החבילות בגלל עומס רב עליה.

בקרת הגודש מתבצעת בעזרת גודל חלון השליחה של הפרוטוקול – כלומר כמות המידע שנשלחת ליחידת זמן. הבקרה על גודל החלון מתבצעת באמצעות 2 משתנים:

- slow start threshold–Sstresh
- congestion window–Cwnd

השולח נמצא באחד משני מצבים:

- Slow start בו גודל החלון הוא 1, ועל כל ack החלון גדל ב 1 עד שגודלו מגיע ל sstresh ואז הוא נכנס למצב congestion avoidance
- Congestion avoidance בו גדל cwnd ב $\frac{1}{cwnd}$ עם קבלת על ack.

שליטה על גודל החלון מתבצעת במקרה של גילוי איבודי חבילות - הפרוטוקול מניח שהדבר התרחש בגלל העומס ברשת, ועל כן מוריד את גודל החלון באופן הבא:

- אם החבילה נאבדה בעקבות timeout גודל החלון יורד ל – 1, ה sstresh משתנה ל $\frac{cwnd}{2}$, והשולח נכנס למצב slow start
- אם החבילה נאבדה בעקבות duplicate ack אז גודל החלון יורד ל $\frac{cwnd}{2}$, ה sstrech יורד משתנה ל $\frac{cwnd}{2}$ והשולח נכנס למצב של congestion avoidance

דברים שימושיים לפתרון תרגילים

- סכום סדרה חשבונית $S_n = \frac{n(a_1 + a_n)}{2}$
- תוחלת של מ"מ מפולג גאומטרית $E(X_{Geo}) = \frac{1}{p}$ כאשר p היא ההסתברות ל"הצלחה"
- התפלגות מעריכית (זוהי התפלגות חסרת זכרון): $f(X_{Geo}) = \lambda e^{-\lambda x}$ כאשר λ הוא קצב ההתפלגות. תוחלת של מ"מ מפולג מעריכית היא $E(X_{\exp}) = \frac{1}{\lambda}$
- $\sum_{i=o}^{\infty} \frac{x^i}{i!} = e^x$
- ההסתברות לנפילת שגיאה במסגרת הוא ביחס ישר לגודל המסגרת
- התפלגות פואסונית בדידה $P(k,\lambda,t) = \frac{(\lambda t)^k}{k!} e^{-\lambda t}$, כשאר k הוא מספר המופעים, t הזמן ו λ - קצב המופע
- סכום של סדרה הנדסית, סופית: $S_n = \frac{a_1(q^n - 1)}{q - 1}$, אינסופית: $S_{n\to\infty} = \frac{a_1}{1-q}$.
- מתקיים $\sum_{i=1}^{\infty} ip^i = p\sum_{i=1}^{\infty} ip^{i-1} = \sum_{i=1}^{\infty} \left(p^i\right)' = \left(\sum_{i=1}^{\infty} p^i\right)' = \left(S_n - sum\,of\ \ geo\ progression\right)'$

מקורות

- חוברת ההרצאות והתרגולים של הקורס "מבוא לרשתות מחשבים 236334 – הטכניון"
- סיכום קורס מבוא לרשתות מחשבים של אור גלעד
- התפלגות מעריכית - http://en.wikipedia.org/wiki/Exponential_distribution
- TCP - http://en.wikipedia.org/wiki/Transmission_Control_Protocol
- תורת התורים(M/M/K/N) http://en.wikipedia.org/wiki/Queueing_model
- GBN - http://en.wikipedia.org/wiki/Go_back_N

תודות

- קיריל דברובולסקי וקיריל ליסובצב (או בעצם קיריל×[דוברובולסקי + ליסובצב]) על הערות ושיפורים.
- דוד ארינזון על שיפורים, הצעות, והגייה (אני משתמש באופיס בטא, אין פה הגייה בעברית).