

תירגול מס' 2 – שכבת קישור הנתונים



נושאי התיראול

□ טיפול שגיאול

- בדיקת זוגיות (Parity Checking).
- סכום הביקורת (Checksum).
- בדיקת יתירות מחזורית (CRC).

□ ALOHA

- שיטת Pure ALOHA.
- שיטת *slotted ALOHA*.

□ תרגילים לדוגמא.

טיפול שגיאות



קודים לתיקון / איזוי שגיאות

רעיון כללי:

- השיטה העיקרית לזיהוי שגיאות היא לשלוח עם המידע המקורי איזשהו מידע עודף המכיל נתונים כלשהם על המידע הנשלח (redundant information). מידע עודף זה יעזור לנו לזהות טעויות בקלט שהגיע.
- במקרה הקיצוני נשלח כל ביט פעמיים:
 - עבור הקלט 1011001 נשלח את ההודעה 11001111000011. אם המקבל יזהה איזשהו ביט שלא הגיע ב"זוג" הוא ידע שיש שגיאה.
- שאלה: אופציה שניה עבור הקלט היא להכפיל את הקלט ולשלוח את ההודעה 10110011011001 מה ההיתרונות / חסרונות של כל שיטה ?
- תשובה: עבור מקרה שבו נאבד 5 ביטים רצופים (10110), באופציה הראשונה נאבד את המידע ואופציה השניה לא!
 - חסרונות אופציה השניה הבדיקה היא יותר איטית !
 - אבל גם עבור האופציה הראשונה זהו כמובן פיתרון מסורבל ויקר בזמן שליחה וזיכרון ולא יזהה שגיאה שהתרחשה בשני ביטים רצופים! כמו כן הוא לא יכול לתקן שגיאה שהוא יזהה כי הוא לא ידע איזה ביט נכון ואיזה שגוי.
 - ניתן לכתוב כל ביט 3 פעמים ואז בזיהוי שלשה לא זהה נדע שיש טעות ואף נוכל לתקן אותה בסבירות טובה. גם שיטה זו יקרה מאוד.

One/Two Dimensional Parity Checking – *בדיקת זוגיות*

□ **במימד אחד:** הרעיון הוא להוסיף ביט אחד לכל בית (7 ביטים) באופן שישלים למספר זוגי של אחדות. שיטה זו מאפשרת לנו לזהות כל מספר אי זוגי של שגיאות בכל בית. (אם המקבל רואה שיש לו מס אי זוגי של אחדות אז הוא יודע שיש 1/3/5... שגיאות במידע).

Single Bit Parity:
Detect single bit errors

□ מס זוגי של שגיאות לא יזוהה
מכיוון שמס' האחדות יישאר זוגי



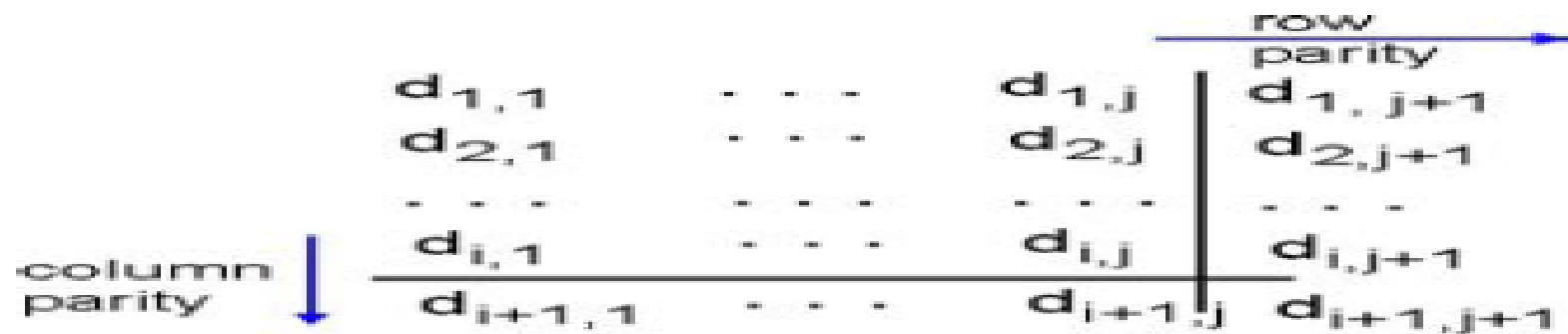
בדיקת זוגיות

□ בשני מימדים אנחנו מוסיפים בית נוסף על כל X בתים. בבית זה אנחנו רושמים ביטים המשלימים למספר זוגי של אחדות בסכום העמודות בביתים שלפניהם.

DATA	0 1 0 1 0 0 1	1
	1 1 0 1 0 0 1	0
	1 0 1 1 1 1 0	1
	0 0 0 1 1 1 0	1
	0 1 1 0 1 0 0	1
	1 0 1 1 1 1 1	0
Parity byte	1 1 1 1 0 1 1	0

Two Dimensional Bit Parity:

Detect and correct single bit errors



1	0	1	0	1	1
1	1	1	1	0	0
0	1	1	1	0	1
0	0	1	0	1	0

no errors

1	0	1	0	1	1
1	0	1	1	0	0
0	1	1	1	0	1
0	0	1	0	1	0

parity error

correctable single bit error

תיקון

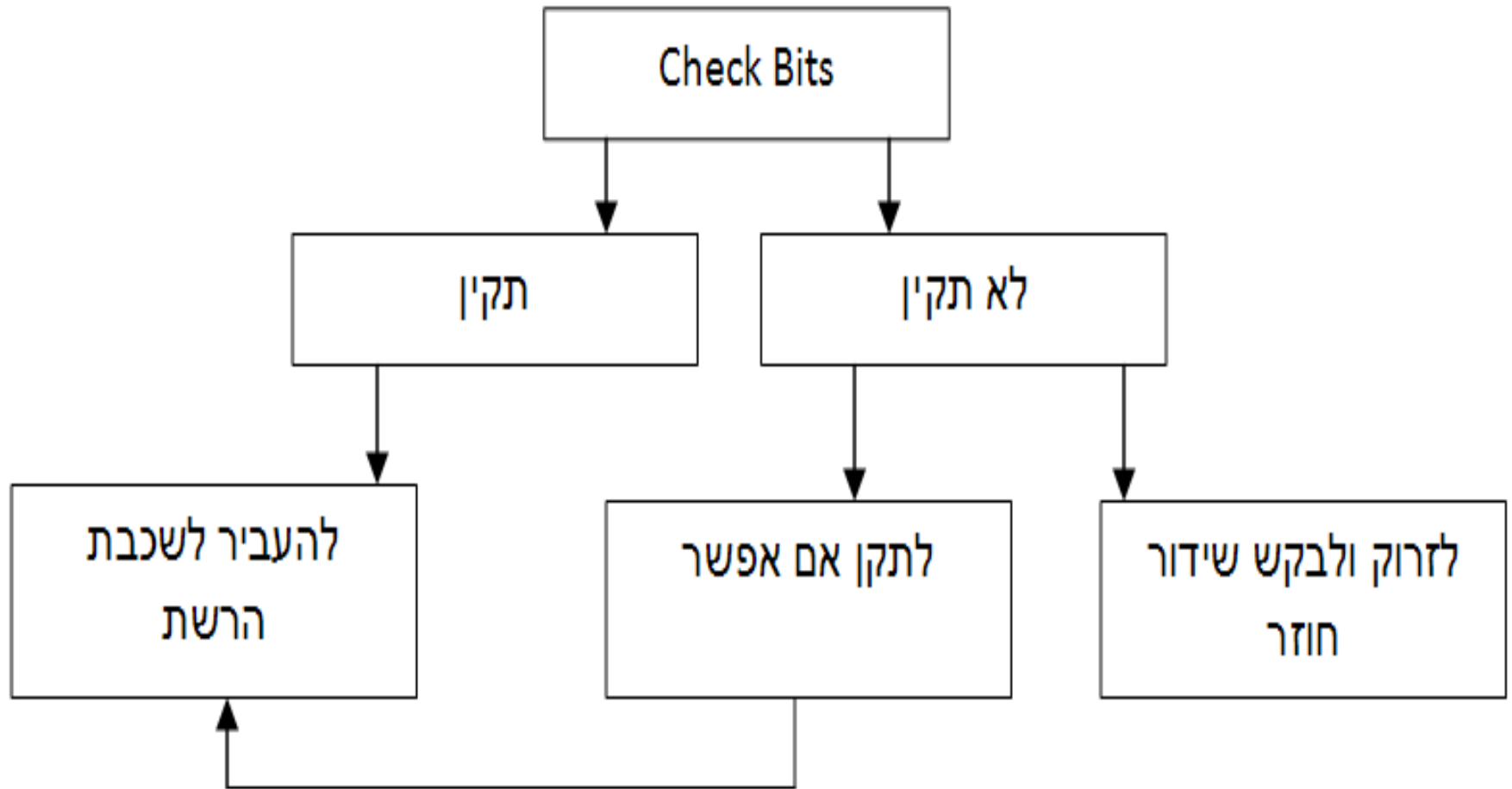
שגיאה אחת

גלוי

שגיאה אחת

שתי שגיאות (באותה שורה או עמודה)
שלוש שגיאות (תמיד נגלה אבל ייתכן ויחשב לשגיאה אחת)

סיכום בדיקת זוגיות



סיכום פיקורת - Checksum

- הרעיון: סוכמים את כל המילים שרוצים להעביר ומעבירים גם את התוצאה (התוצאה של חיבור כל המילים נקראת checksum). המקבל מבצע את אותו חישוב על המידע ומשווה את התוצאה שקיבל ל-checksum. אם קיימת שגיאה במידע או אפילו ב-checksum שנשלח, שגיאה זו תתגלה כי התוצאות יהיו שונות.
- קיימות המון דרכים למימוש:
המילים באורך 16bit.
מחברים אותם יחד בשיטת המשלים לאחד.
תזכורת: בשיטת המשלים לאחד x - נחשב המשלים של x כלומר כל ביט של x מתהפך
אם בחיבור היה carry אזי הוא מתווסף לתוצאה.

דואנא פסיכוס ביקורת

נניח שהבלוק הבא מכיל 16 סיביות (ביטים) הנשלחים ומשתמשים בסיכום ביקורת (checksum) של 8 סיביות.

10101001 00111001

נחבר את המספרים תוך שימוש בשיטת המשלים ל-1 (one's complement).

10101001

00111001

Sum 11100010 סיכום

Checksum 00011101 סיכום ביקורת

התבנית אשר תשלח: 10101001 00111001 00011101

דואנא פסיכוס ביקורת (המשק)

עתה שהקולט מקבל את התבנית שנשלחה ואין שגיאות שנוצרו בשידור נניח

10101001 00111001 00011101

כאשר הקולט מבצע סיכום של שלושת הקטעים שנשלחו, הוא אמור לקבל רצף של 1-ים, ואחרי חישוב המשלים ל-1 נקבל רצף של 0-ים אשר יראה כי לא נוצרו שגיאות בשידור.

10101001

00111001

00011101

Sum

11111111

Complement

00000000 means that the pattern is OK.

בדיקת יתירות מחזורית – CRC

- קוד נפוץ לגילוי שגיאות, המתוחכם בהרבה מסיבית הזוגיות הוא **CRC** (Cyclic Redundancy Code).
- לדוגמא **CRC** של 16 סיביות מאפשר לגלות:
 - כל השגיאות של 2 ביטים.
 - כל מספר אי-זוגי של שגיאות.
 - כל השגיאות ברצף של 16 ביטים או פחות.
- 99.99% של רצפים של 17 ביטים או יותר.
- פרוטוקול קצת מורכב אבל מאוד טכני בין השולח למקבל.

מחשבים:

□ **D(X)** – פולינום מדרגה n המייצג $n+1$ ביטים מהקלט. את הפולינום בונים ע"י הכפלה של כל ביט בערך 1 בחזקת מקדם הביט בקלט. דוגמה:

$$10011010 \rightarrow D(x) = X^7 + X^4 + X^3 + X^1$$

□ **G(X)** – מחלק פולינומיאלי מדרגה r .

■ ישנן תבניות קבועות של מחלקים כאלה וכל אחת מהן בעלת תכונות מסוימות שטובות לזיהוי טעויות מסוגים שונים.

■ לדוגמה – ה Ethernet משתמש ב $G(X)$ מדרגה 32.

■ דוגמאות נוספות:

CRC-8: $x^8 + x^2 + x^1 + 1$

CRC-10: $x^{10} + x^9 + x^5 + x^4 + x^1 + 1$

CRC-12: $x^{12} + x^{11} + x^3 + x^2 + 1$

CRC-16: $x^{16} + x^{15} + x^2 + 1$

CRC-CCITT: $x^{16} + x^{12} + x^5 + 1$

CRC-32: $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$

מיון:

- **$T(X)$** – ההודעה אותה שולח השולח. הודעה זו היא בגודל $r + (n+1)$ ביט.
- $T(X)$ מתחלק בדיוק בפולינום $G(X)$, ומכיל בתוכו למעשה את $D(X)$, ממנה יפענח המקבל את ההודעה המקורית.
תזכורת – $T(X)$ מתחלק ב $G(X)$ אם המעלה של $T(X) \leq$ מהמעלה של $G(X)$.
- **$R(X)$** – השארית המתקבלת מחלוקת $T(X)$ ב $G(X)$. (אם יש כזו).

דואנא פיאנא ק-CRC

1. יצירת ההודעה ע"י השולח:

א- $T(X) = D(X)$ (adds r zeros at the message end)

□ r הוא המעלה של $G(X)$.

ב- Divide $T(X)$ by $G(X)$ and find the remainder $R(X)$

□ המעלה של $G(X)$ היא 3 ולכן מוסיפים 3 אפסים מימין ל $D(X)$ ויוצרים את הרצף המחולק:

Example: $D = 110101$ $D(x) = X^5 + X^4 + X^2 + 1$ בחילוק עושים XOR בין הביטים. □
טבלת XOR: □

$G = 1001$ $G(x) = X^3 + 1$

$$\begin{array}{r}
 110011 \\
 1001 \overline{) 110101000} \leftarrow D(X) X^3 \\
 \underline{1001} \\
 1000 \\
 \underline{1001} \\
 0011 \\
 \underline{0000} \\
 0110 \\
 \underline{0000} \\
 1100 \\
 \underline{1001} \\
 1010 \\
 \underline{1001} \\
 011 \leftarrow R
 \end{array}$$

$T = 110101 | 011$

Xmitted message

p	q	$p+q$
0	0	0
0	1	1
1	0	1
1	1	0

דואנא פאנאס ק-CRC

ג- $T(X) = T(X) - R(X)$

- מחסירים את השארית ומקבלים את $T(X)$ מוכן לשליחה.
- $T(X)$ הזה מתחלק בדיוק ב $G(X)$ ומכיל את $D(X)$ שאליו משורשר מימין $R(X)$.

2. קבלת ההודעה:

- המקבל מכיר את $G(X)$ כי הוא קבוע ומוסכם מראש.
- הוא מחלק את $T(X)$ ב- $G(X)$ ומוצא את $R(X)$ - השארית.
- אם השארית היא 0 - משמע שאין שגיאות בהעברה.
- אם השארית היא לא 0 - משמע שיש שגיאות בהעברה.
- נחזור דוגמא הבא: ←

- ע"מ להוציא מההודעה שהתקבלה את ההודעה המקורית על השולח להסיר את r הביטים הימניים (מה שהיה $R(X)$), והוא נותר עם ה- $D(X)$ המקורי - ההודעה!

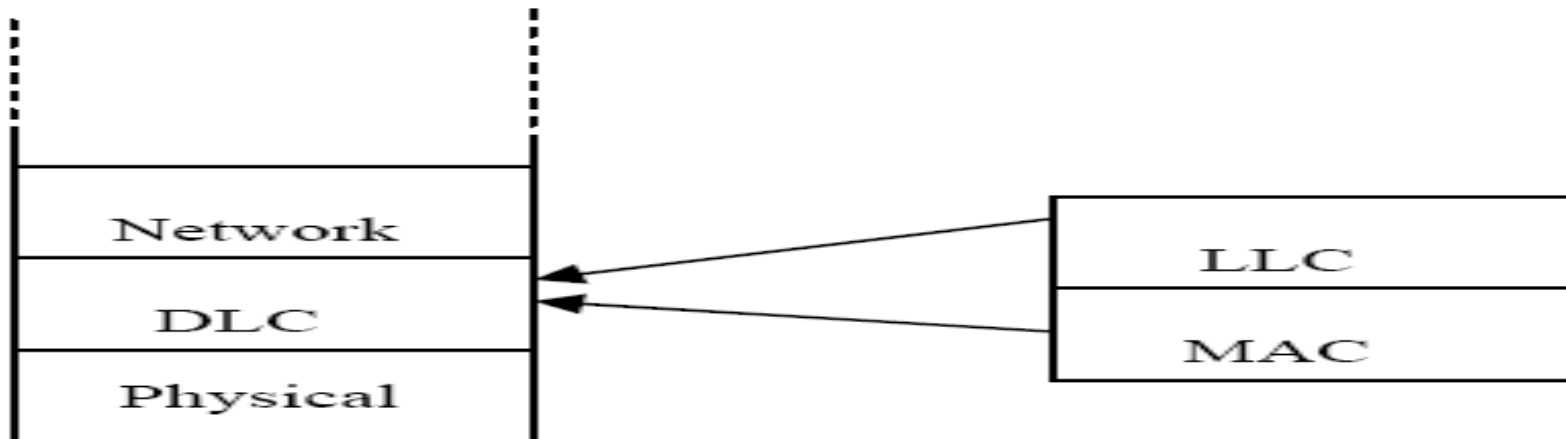
$$\begin{array}{r}
 110011 \\
 1001 \overline{) 110101011} \\
 \underline{1001} \\
 01000 \\
 \underline{1001} \\
 00011 \\
 \underline{0000} \\
 0110 \\
 \underline{0000} \\
 1101 \\
 \underline{1001} \\
 01001 \\
 \underline{1001} \\
 R = 0000
 \end{array}$$

ALOHA



מבוא

□ תחילה נמקם את הנושא במודל השכבות:



□ שכבת הDLC (*Data Link Control Layer*) - מורכבת משתי רמות:

- *LLC (Logical Link Control)* - האחראית למעבר תקין של חבילות מהמקור ליעד. מספור החבילות, הגעתם לפי הסדר וכו'.
- *MAC (Medium Access Control)* - בקרת הגישה לערוץ משותף. כאשר ערוץ התקשורת משותף יש צורך לנהל בקרת גישה לערוץ, וכאן נדון בשיטת *ALOHA*.

מודל הרשת

- מספר צרכנים אינסופי $N \rightarrow \infty$
- שתי הודעות המשודרות בפרקי זמן חופפים מתנגשות (collision) והתקלה מתגלה ע"י התחנות בסיום שידור הודעה.
- פרט להתנגשויות אין שגיאות נוספות בערוץ (אילו היו, הדבר היה נפתר ברמת ה-LLC).
- תהליך הופעת החבילות לשידור (הודעות חדשות + שידורים חוזרים) הוא פואסוני עם ממוצע G חבילות ליחידת זמן.
- כל החבילות באורך קבוע ונקבע יח' זמן כזמן שידור חבילה אחת.
- תזכורת: נוסחת ההתפלגות הפואסונית:

$$\Pr(k, \lambda, t) = \frac{(\lambda t)^k}{k!} e^{-\lambda t}$$

- מתארת הסתברות ל- k מופעים של תהליך הפואסוני אם פרמטר λ בפרק זמן t .

Pure ALOHA שיטת

- כל תחנה משדרת הודעה מתי שהיא רוצה, כלומר צומת משדר מיד כשיש לו חבילה לשידור.
 - במקרה של התנגשות יש לשדר את ההודעה מחדש.
 - בחירת מועד השידור הבא מתבצעת בצורה אקראית.
 - נחשב תעבורה (נצילות) עבור המודל הבא:
- זמן מסגרת – frame time – כמות הזמן הנדרשת להעברת מסגרת באורך קבוע.

$$t = \text{זמן מסגרת} = \frac{\text{אורך מסגרת (בסיים)}}{\text{קצב שידור}}$$

- מס' הודעות המידע הנשלחות בפרק זמן T או קצב ניסיונות השידור: G
- קצב הגעת הודעות חדשות וישנות הוא פואסוני:

$$P_k(t) = \frac{(G)^k}{k!} e^{-G}$$

כאשר k הוא מספר המסגרות.

Pure ALOHA שיטת

□ נחשב את ההסתברות להצלחת שידור בפרק זמן:

$$P_{\text{success}} = P\left(\begin{array}{c} \text{לא תשלח אף הודעה לערוץ} \\ \text{בפרק זמן } 2T \end{array}\right) = P_{\underbrace{k=0}_{\text{אף הודעה}}}(t=2T) = e^{-2G}$$

□ כל שידור במהלך תקופת הפגיעה יגרום ל- *collision*

□ S קצב היווצרות מסגרות בממוצע, השווה למעשה לנצילות של התעבורה:

$$\eta = S = Ge^{-2G}$$

□ המקסימום מתקבל עבור $G = 0.5$ ושווה $S = 0.5 * e^{-1} \approx 0.18$

עיסת slotted ALOHA

□ הפרוטוקול:

- כמו ALOHA רק שבנוסף קיים מנגנון לחירוף הזמן, כלומר הזמן מחולק למרווחי זמן שווים של T יח' זמן (Slots).
- צומת שיש לו חבילה (חדשה או שידור חוזר) לשידור משדר אותה בתחילת חריץ הזמן הקרוב ביותר, כלומר השידור מתבצע רק מתחילת slot.

□ חישוב תעבורה עבור מקרה זה

- תקופת הפגיעה מצטמצמת ל- T (התחנה המשדרת אינה רשאית לשלוח מידע כל הזמן. ונאלצת לחכות לתחילת מקטע הזמן הבא)

$$P_{\text{success}} = P_{N=0}(t=T) = e^{-G}$$

$$\eta = S = G \cdot e^{-G}$$

- המקסימום מתקבל עבור $G = 1$ ושווה $S = e^{-1} \approx 0.36$

סרף הנציאות

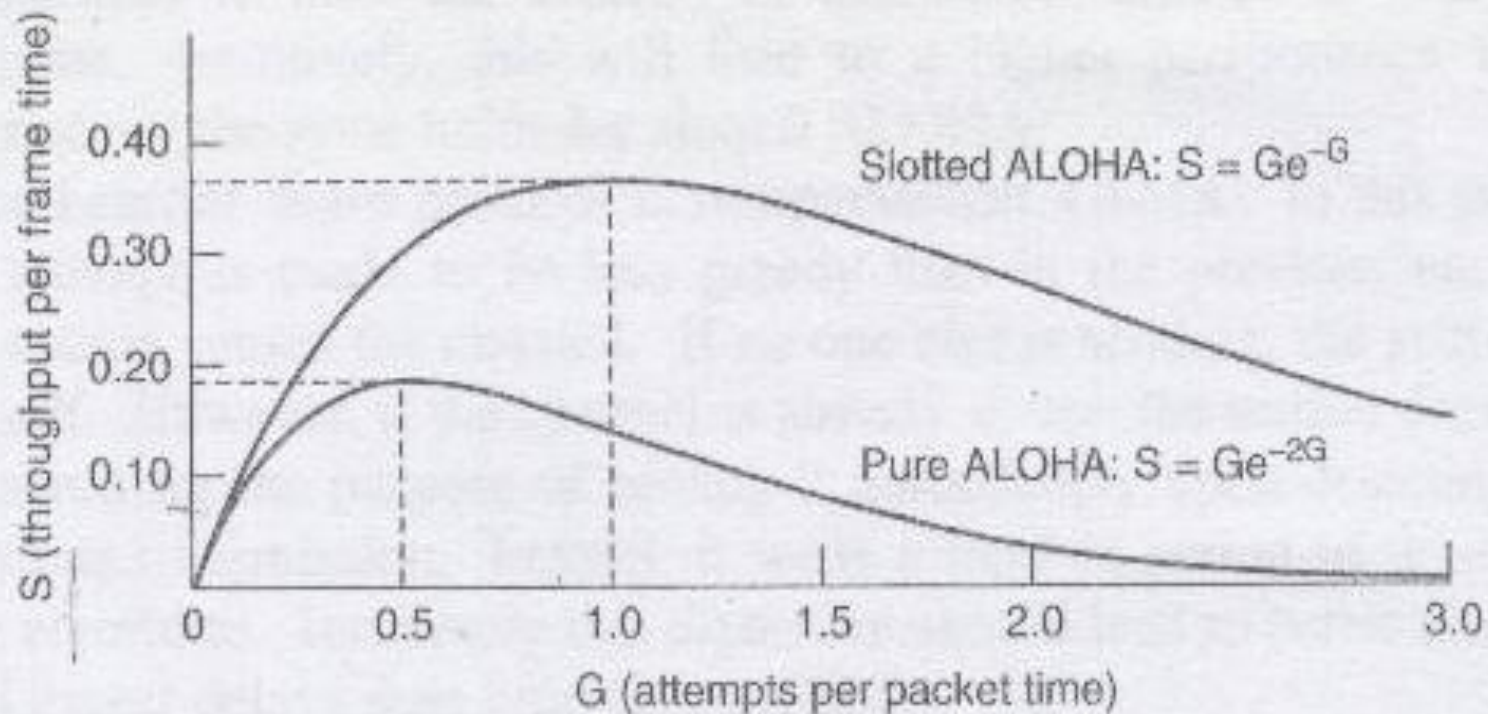


Fig. 4-3. Throughput versus offered traffic for ALOHA systems.

שיטת slotted ALOHA עבור מס' סופי של תחנות

□ N - תחנות.

□ G_i - מס' חבילות (חדשות וחוזרות) בממוצע המופיעות בתחנה i במשך זמן חריץ (T יח' זמן).

□ S_i - קצב היווצרות מסגרות בממוצע או מס' חבילות בממוצע שתחנה i מצליחה לשדר במשך זמן חריץ (T יח' זמן).

□ הנחת המודל: במצב יציב, תהליך הופעת חבילות (חדשות + חוזרות) בחריץ הוא ברנולי עם פרמטר G_i

$$S_i = \left(\begin{array}{c} \text{הסתברות כי} \\ \text{תחנה } i \text{ תשדר} \end{array} \right) \cdot \left(\begin{array}{c} \text{התחנות} \\ \text{פרט } i - \text{ל לא ישדרו} \end{array} \right) = G_i \prod_{\substack{j=1 \\ j \neq i}}^N (1 - G_j)$$

$$G = \sum_{i=1}^N G_i \quad \left[\frac{\text{חבילות}}{\text{slot}} \right] \quad \text{קצב שידור ברשת}$$

$$S = \sum_{i=1}^N S_i \quad \left[\frac{\text{חבילות}}{\text{slot}} \right] \quad \text{תעבורת הרשת}$$

שיטת *slotted ALOHA* עבור מס' סופי של תחנות

□ נבחן מקרה פרטי (מעניין)

בו כל התחנות זהות:

$$G_i = \frac{G}{N}$$

$$S_i = \frac{S}{N}$$

$$S_i = G_i \prod_{j=1}^N (1 - G_j) = \frac{G}{N} \left(1 - \frac{G}{N}\right)^{N-1}$$

$$S = N \cdot S_i = G \left(1 - \frac{G}{N}\right)^{N-1}$$

□ עבור $N \rightarrow \infty$ נקבל וזה בדיוק במקרה של **slotted ALOHA** עבור ∞ תחנות,

והמקסימום מתקבל עבור $G=1$.

$$\left(1 + \frac{1}{n}\right)^{\frac{n}{1}} \rightarrow e$$

$$\left(1 - \frac{1}{n}\right)^{\frac{n}{1}} \rightarrow e^{-1}$$

$$\Rightarrow G \left(1 - \frac{G}{N}\right)^{(N-1)} = G \left[\left(1 - \frac{G}{N}\right)^{\frac{N}{G}} \right]^{\frac{G}{N}(N-1)} \rightarrow G e^{-G}$$

$$\frac{G}{N}(N-1) \rightarrow G$$

$$\left[\left(1 - \frac{G}{N}\right)^{\frac{N}{G}} \right] \rightarrow e^{-1}$$

שיטת Slotted Aloha עם חפיפות שאורכן שונה מאורך החריץ

■ נתונה רשת Slotted Aloha עם מספר אינסופי של תחנות. אורך כל חריץ הוא שנייה אחת. אורך כל הודעה היא 0.9 שניות בהסתברות P ו-1.1 שניות בהסתברות $(1-P)$.

■ תוך שימוש בסימון $\Pr(k, \lambda, t)$ כדי לציין את ההסתברות ל- k מופעים, ב- t יחידות זמן כאשר התפלגות המופעים היא פואסונית עם פרמטר λ , כפונקציה של מספר מופעים ממוצע בחריץ (G) בלבד, נענה על השאלה: מה הניצולת של הערוץ עבור $P=1$ (כל ההודעות הן באורך 0.9)?

$$\Pr(k, \lambda, t) = \frac{e^{-\lambda t} (\lambda t)^k}{k!}$$

ניצולת: מספר חבילות המועברות לשכבת הרשת בפרק זמן השווה לזמן שידור מסגרת אחת

■ תזכורת:

■ תשובה:

■ כל ההודעות הן באורך 0.9 לכן יכולות להתנגש רק עם הודעות מאותו חריץ. בחריץ הנוכחי ישדרו תחנות שקיבלו החלטה לצאת לשידור בחריץ הקודם. נסמן ב- k את מס' המסגרות המשודרות בחריץ בהצלחה.

$$k = 1 \cdot \Pr(\text{successful slot}) = \Pr(1, G, 1) = Ge^{-G}$$

■ הסבר:

$$\Pr(k = 1, \lambda = G, t = 1) = \frac{e^{-G \cdot 1} (G \cdot 1)^1}{1!} = Ge^{-G}$$

שיטת Slotted Aloha עם חפיפות שאורכן שונה מאורך החריץ

□ נחשב את הניצולת ע"פ ההגדרה:

$$\begin{array}{l} 1 \text{ slot} - Ge^{-G} [\text{frames}] \\ S - S [\text{frames}] \end{array} \rightarrow \begin{array}{l} 1[\text{sec}] - Ge^{-G} [\text{frames}] \\ 0.9[\text{sec}] - S [\text{frames}] \end{array} \rightarrow S = 0.9 \cdot Ge^{-G}$$

תרגילים לדוגמא



תרגיל 1

■ נתון דגל 01111110, משתמשים בשיטת ה-bit stuffing (אחרי 5 אחדות רצופים מכניסים 0) עבור זרם הביטים הבא:

■ ממש את השיטה על זרם הביטים הבאים:

0110111110011111101011111111101111010

■ הנח שזרם הביטים הבא מגיע:

011111101111101100111110011111011111011000111111010111110

■ הוצא את ה-bit stuffing והראה היכן הדגלים.

■ נתון דגל 01111110, משתמשים בשיטת ה-bit stuffing הנח שבמקום שאחרי 5 אחדות רצופים מכניסים 0, משתמשים בשיטה הבא: אחרי 011111 מכניסים 0. הנח שזרם הביטים הבא מגיע:

01101111101111110111110101111110.

■ הוצא את ה-bit stuffing והראה היכן הדגלים.

פתרון תרגיל 1

פתרון סעיף א':

דגל: 01111110, bit stuffing (אחרי 5 אחדות רצופים מכניסים 0)

מקור: 0110111110011111101011111111101111010

תוצאה:

01111110

0110111110001111101010 11111011111001111010

דגלים מסומנים באדום

תוספת אפסים מסומנת בצהוב

פתרון סעיף ב':

מקור: 011111101111101100111110011111011111011000111111010111110

נסמן את הדגלים:

011111101111101100111110011111011111011000111111010111110

דגלים מסומנים באדום

תוצאה:

11111110011111011111111111100

פתרון סעיף ג':

דגל: 01111110, bit stuffing (אחרי 011111 מכניסים 0)

מקור: 011011111011111101111110101111110

נסמן את הדגלים:

01101111101111110111110101111110

דגלים מסומנים באדום

תוצאה:

1 11111

תרגיל 2

□ נתון מידע באורך של D סיביות, כאשר מחלקים את המידע ל- i שורות בעלות אורך של j סיביות (הנחו כי D ניתן לחלוקה ב- i וב- j ללא שארית). מוצע לקודד את המידע ע"י סיבית זוגיות שמחושבת עבור כל שורה וסיבית זוגיות שמחושבת עבור כל עמודה.

א- הוכחו שכל שגיאה אחת יכולה להתגלות ואף להיות מתוקנת ע"י הקוד המוצע.

ב- מצאו דוגמא של שתי שגיאות שניתנות לגילוי. האם כל קומבינציה של שתי שגיאות ניתנות לתיקון?

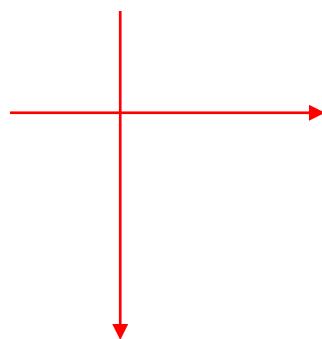
ג- מהי התקורה של הקוד המוצע, וכיצד ניתן להקטין את התקורה הזו למינימום. הצעו ביטוי מתמטי המבטיח תקורה מינימלית.

פתרון תרגיל 2

□ פתרון סעיף א':

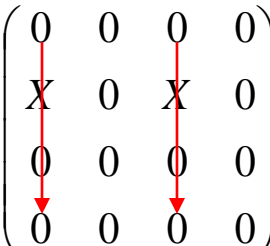
כל שגיאה בודדת שמתרחשת במטריצת הנתונים גורמת לשגיאה בביט הזוגיות של שורה אחת ועמודה אחת. החיתוך בין השורה לעמודה מציין את הביט השגוי וכך ניתן לתקנו.

לדוגמא :


$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & X & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

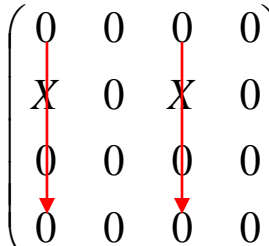
פתרון תרגיל 2

□ פתרון סעיף ב':

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ X & 0 & X & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$


דוגמא להצלחת האלגוריתם רק בגילוי שגיאה:

לא כל קומבינציה ניתנת לתיקון , למשל :

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ X & 0 & X & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$


במקרה שמתואר באיור ניתן רק לגלות את המיקום של העמודות בהם קיימת השגיאה, אך לא ניתן לקבל את המיקום המדויק של השגיאה.

פתרון תרגיל 2

פתרון סעיף ג':

- התקורה של הקוד המוצע הינה: $i+j+1$, כאשר i סיביות מגינות על העמודות ו- j סיביות מגינות על השורות, סיבית נוספת משותפת להגנה של העמודה והשורה האחרונות במטריצה. על מנת לקבל תקורה מינימלית צריך לבצע: $\min \{i + j + 1\}$ תחת ההנחה ש- $i \cdot j = D$.
- נבצע זאת ע"י הצבה גזירה והשוואה ל- 0 ונקבל:

$$\min \{i + j + 1\}_{i \cdot j = D} \Rightarrow \min'_i \left\{ i + \frac{D}{i} + 1 \right\} = \left\{ 1 - \frac{D}{i^2} \right\} = 0 \Rightarrow i^2 = D \Rightarrow i = \sqrt{D}$$
$$\Rightarrow \min \{i + j + 1\} = \min \left\{ \sqrt{D} + \sqrt{D} + 1 \right\} = 2\sqrt{D} + 1$$

- נוסחה זו נכונה בהנחה שניתן להוציא שורש שלם ל- D . במקרה ולא ניתן, כלומר מתקבל מספר שאינו שלם, נחפש את המספר השלם הראשון שהכי קרוב ל- \sqrt{D} שגדול או קטן מ- \sqrt{D} שמתחלק ב- D ללא שארית. המטרה לקבל מטריצה שתשאף להיות ריבועית.

תרגיל 3

□ אנו רוצים לשדר את ההודעה 1100100100101011
באמצעות CRC8 ($G=100000111$).

א- מהי מילת הקוד שתשודר ?

ב- הניחו כי הביטים השלישי, הרביעי והחמישי בספירה משמאל
התהפכו. מה תהיה תוצאת בדיקת ה-CRC בצד המקבל ?

ג- איך יודע המקבל כי הייתה שגיאה ?

פתרון תרגיל 3

פתרון סעיף א':

נתון: CRC8,

הודעה 1100100100101011,

$$G = 100000111$$

$r=8$ נחשב את R כאשר

$$R = \frac{D \cdot 2^r}{G}$$

• מילת הקוד לכן תהיה:

$$\underbrace{1100100100101011}_D \underbrace{10000001}_R$$

$$\begin{array}{r}
 1100101101011011 \\
 100000111 \overline{) 110010010010101100000000} \\
 \underline{100000111} \\
 0100101010 \\
 \underline{100000111} \\
 000101101101 \\
 \underline{100000111} \\
 00110101001 \\
 \underline{100000111} \\
 0101011101 \\
 \underline{100000111} \\
 00101101000 \\
 \underline{100000111} \\
 00110111100 \\
 \underline{100000111} \\
 0101110110 \\
 \underline{100000111} \\
 00111000100 \\
 \underline{100000111} \\
 0110000110 \\
 \underline{100000111} \\
 R = 010000001
 \end{array}$$

פתרון תרגיל 3

פתרון סעיף ב':

□ המקבל קיבל את המידע הבא:

$$\underbrace{\overbrace{111100010010101}^{\text{שגיאות}} 1}_{D} \underbrace{10000001}_R$$

□ המקבל יודע את האורך של R ומכאן כמובן גם את האורך של D ,
ויחשב את אותו חישוב

$$R = \frac{D \cdot 2^r}{G}$$

□ אם תיהיה תוצאה שונה עבור R יידע שעשה טעות, G כאמור ידוע
ל-2 הצדדים.

$$\begin{array}{r}
 1111001111110000 \\
 100000111 \overline{) 111100010010101100000000} \\
 \underline{100000111} \\
 0111001010 \\
 \hline
 100000111 \\
 \underline{100000111} \\
 0110011011 \\
 \underline{100000111} \\
 0100111000 \\
 \underline{100000111} \\
 000111111101 \\
 \underline{100000111} \\
 0111110101 \\
 \underline{100000111} \\
 0111100100 \\
 \underline{100000111} \\
 0111000110 \\
 \underline{100000111} \\
 0110000010 \\
 \underline{100000111} \\
 0100001010 \\
 \underline{100000111} \\
 000001101000 \\
 R = 11010000
 \end{array}$$

פתרון תרגיל 3

פתרון סעיף ג':

□ כאמור בגלל השגיאה קיבלנו תוצאה שונה עבור R , המקבל משווה את R -השקיבל מהשולח לעומת R -השחישב בעצמו ובגלל שהם לא שווים יידע שהיתה שגיאה בהודעה.

תרגיל 4

א- איזה בעיה בא לפתור אלגוריתם Pure ALOHA ?

ב- מה ההבדל בין אלגוריתם מסוג Channel Partitioning לאלגוריתם מסוג Random Access ?

ג- הסברו כיצד פועל אלגוריתם Pure ALOHA. מה החסרון והיתרון העיקריים ?

פתרון תרע"ד 4

פתרון סעיף א':

- בא לפתור את בעיית ה-MULTI ACCESS CHANNEL בו יש ערוץ שידור יחיד אבל משדרים רבים.
- אם יותר מתחנה אחת משדרות בו זמנית יש היתנגשות והרס של השדר.
- הבעיה היא לתת אלגוריתם מבוזר אשר יתאם את תזמון השדרים של התחנות ככה שיהיה fairness ניצולת (יעילות או efficiency) מקסימלית של הערוץ.

פתרון תרע"ף 4

פתרון סעיף ב':

- **ב- CP** (Channel Partitioning) מחלקים את הערוץ ל- N תתי ערוצים אשר מאפשר שידור בו זמני בלי היתנגשויות. דוגמאות הם TDMA ו-FDMA.
- **ב- RA** (Random Access) כל תחנה משדרת בערך מתי שבא לה ואם מזהה היתנגשות משדרת שוב בהסתברות מסויימת. דוגמאות הם slotted\pure ALOHA ו-CSMA\CD.

פתרון תרגיל 4

פתרון סעיף ג':

- כשיש לתחנה מה לשדר היא משדרת מיידית. שיהיו התנגשויות, כאשר המסגרות המתנגשות יהרסו. קיים משוב מהערוץ, ולכן התחנה המשדרת יודעת לזהות התנגשות על ידי האזנה לערוץ. במידה והמסגרת נהרסה, המשדר מחכה פרק זמן אקראי, ושולח את המסגרת שוב.
- היתרון של Pure ALOHA שהוא מבוזר לחלוטין ולא דורש שום סינכרון מראש.
- החסרון שהיעילות מאד נמוכה בשל ההסתברות הגדולה להתנגשות.