

מבוא לרשתות מחשבים – נקודות עיקריות
מבוסס על חוברת הקורס של פרופ' רפאל רום

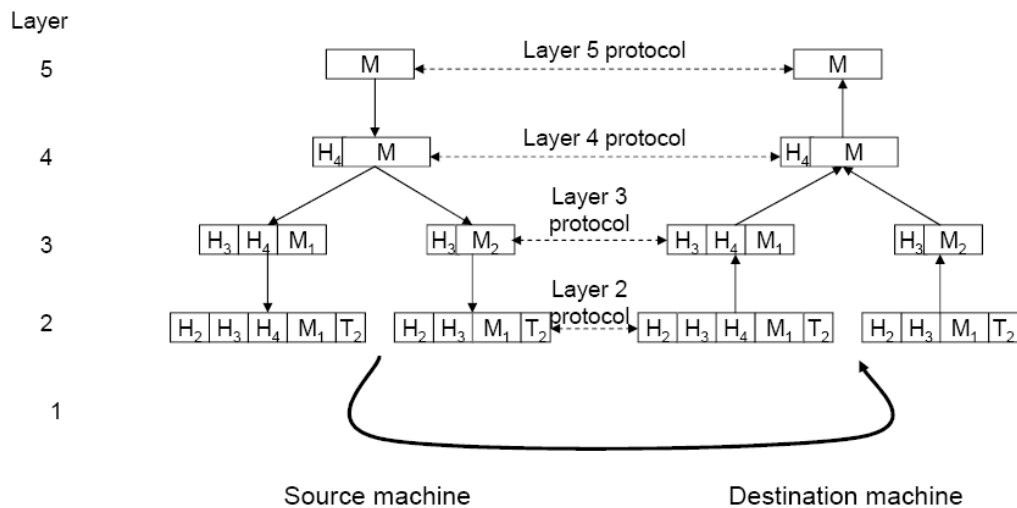
מודל השכבות

מספר	שם	תפקיד	פרוטוקולים נפוצים ויחידותיהם (PDU's)
1	שכבה פיזית Physical Layer	העברת סיביות, פתרון בעיות חשמליות ומכניות	טריוויאלי, מנצ'סטר, מנצ'סטר דיפרנציאלי יחידה: ביט
2	שכבה הקו *Data Link Layer	העברת מידע בדרך אמינה בין משדר למקלט, למרות רעשים והפרעות.	יחידה: Frame
3	שכבת הרשת Network Layer	מפעילה את ה-SUBNET (תת רשת התקשורת): ניתוב, בקרת זרימה והעברת חבילות בין רשתות	יחידה: Packet
4	שכבת התעבורה Transport Layer	העברת מידע בין קצוות הרשת בצורה אמינה. גם לקווי נל"ן וגם לקווי שידור	יחידה: TPDU
5	שכבת השיחה Session Layer	מאפשרת למכונות לקיים שיחה ביניהן. מבקרת את השיחה (דו-שיח, חד-שיח, אסימון וכ"ו)	יחידה: SPDU
6	שכבת ההצגה Presentation Layer	הצפנה, דחיסה והצגה של הנתונים	יחידה: PPDU
7	שכבת האפליקציה Application layer	מרכזת את המידע שמייחד את האפליקציה המשתמשת (מידע ברמה הגבוהה ביותר)	יחידה: APDU

* לעיתים שכבת ה-DLL מחולקת לשתי שכבות: DLC (Data Link Control) ו-MAC (Medium Access Control)

ארכיטקטורה של פרוטוקולים שכבתיים

זרימת מידע בערוץ וירטואלי השיך לשכבה 5



כל שכבה מוסיפה את ה-HEADER (ולעיתים גם TRAILER) של האינפורמציה הרלוונטית לה – ככל שיורדים בשכבות האינפורמציה גדלה.

השכבה הפיזית

זמני תקשורת:

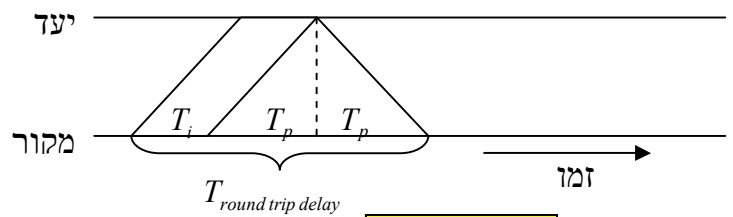
זמן התפשטות (השהייה): $T_p = \frac{D}{V}$

זמן שידור הודעה: $T_i = \frac{L_{bits}}{R_{bits/sec}}$

D מרחק בין מגברים (מטרים), V מהירות בתווך (מטרים לשנייה), R קצב שידור (ביטים לשנייה)

זמן מחזור (Round Trip Time):

תחילת משלוח (תחילת שידור חבילה בצד השולח) עד תום הגעת האישור (קבלת הביט האחרון של חבילת ה-ACK) – בד"כ מתעלמים מזמני עיבוד ההודעה ביעד ובזמן שידור חבילת האישור על קו היעד

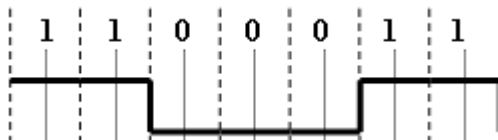


לכן במקרה זה: $T_{RTT} = T_i + 2T_p$

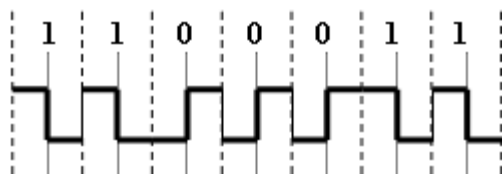
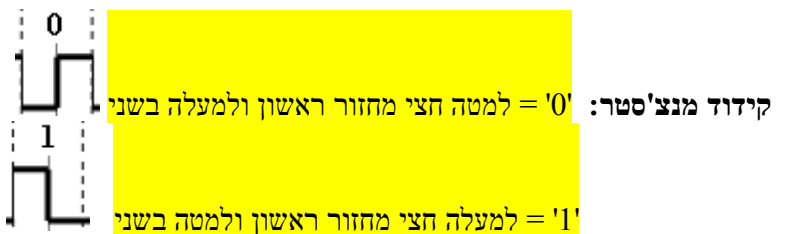
יחס זמן השהייה לזמן שידור: $a \equiv \frac{T_p}{T_i}$

תקשורת ספרתית-שיטות קידוד

Bit stuffing: למשל '0' אחרי כל 3 '1'ים: 111 → 1110

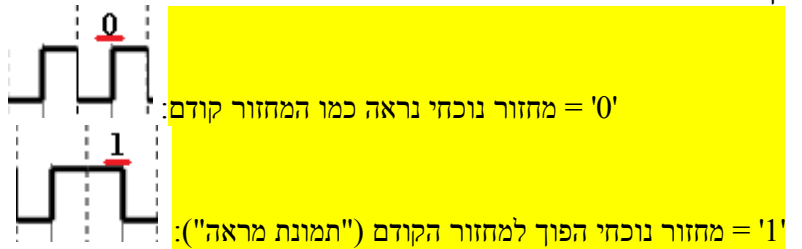


קידוד טריבינאלי: '0' מתח נמוך '1' מתח גבוה דוגמא:



דוגמא:

קידוד מנצ'סטר דיפרנציאלי:

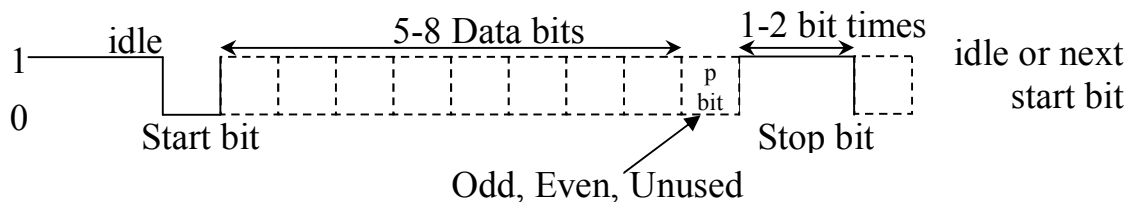


דוגמא:

לשים לב (מוסכמה?) מניחים שהמחזור לפני השידור היה: (למטה חצי מחזור ראשון ולמעלה בשני)

נכתב ע"י עדי פוקס
תקשורת אסינכרונית:

במצב idle משודר 1 קבוע.
התחלת שידור וסיומו על ידי start bit ו-stop bit.
אחרי המידע ביט זוגיות (parity).

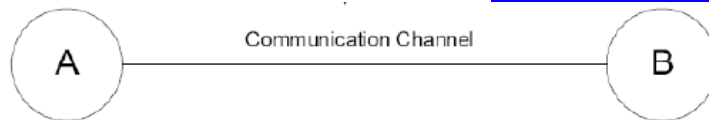


ע"מ שזה יעבוד טוב צריך שתדר המקלט יהיה גבוה מאוד מתדר המשדר ושתדר המשדר ידוע (כמובן שבמבחן זה לא קורה...)

תקורה (OVERHEAD)

היחס בין כמות הביטים שלא מעבירים מידע חדש (כמו ביט זוגיות, STOP ו-START) לבין כלל הביטים המשודרים

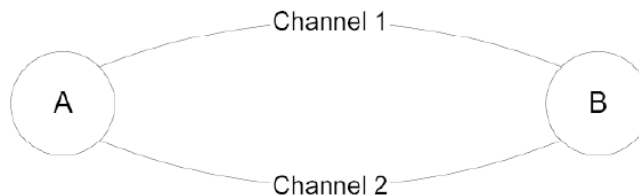
תרגיל בית 1 – מודל ערוץ תקשורת פשוט



A שולח ל-B חבילות, זמן שידור חבילה $T=1\text{sec}$, הסתברות לשגיאה p . אם קרתה שגיאה A שולח שוב את אותה הודעה.

מספר שידורים ממוצע לחבילה: $\bar{N} = \frac{1}{1-p}$ זמן שליחה ממוצע (חוק ליטל): $t_v = \bar{N} \cdot T = \left(\frac{1}{1-p}\right) \times 1[\text{sec}]$

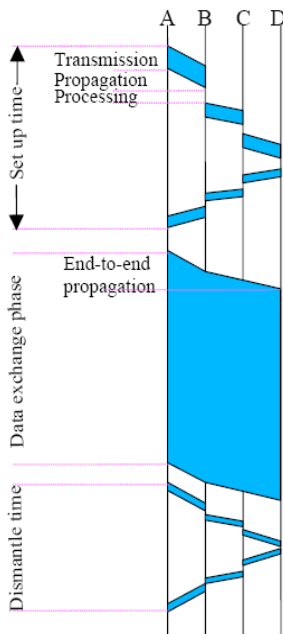
בעקבות תקציב נוסף שהוקצה למערכת התקשורת, נוסף עוד קו תקשורת בין A ו-B.



A מחלק את החבילה לשני בלוקים: ערוץ 1 מהיר 0.5 עם אותה הסתברות שגיאה ערוץ 2 איטי יותר, זמן שליחה 0.8 אבל עם הסתברות שגיאה אפס

זמן שליחה ממוצע (לפי נוסחאת ההסתברות הכוללת):

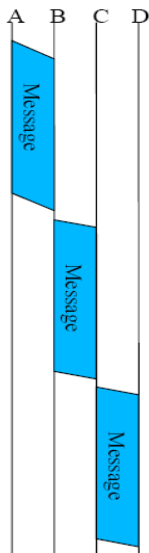
$A = P(\text{Ch. 1 succeeded on 1st baordcast})0.8\text{sec} + P(\text{Ch. 1 failed on 1st baordcast})(\text{mean Ch.1 sending time})$



1. CIRCUIT SWITCHING (מקובל במערכות טלפון):
 קיים שלב הקמת קשר (SETUP) כל קשר מקבל רוחב סרט ייחודי

יתרון: ללא השהיות וללא דחיסות לאחר שלב ההכנה

חסרון: רוחב סרט הולך לאיבוד

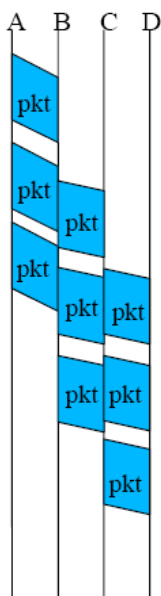


2. מיתוג הודעות (MESSAGE SWITCHING, STORE AND FORWARD):

ההודעות עוברות בשלמותן, אין מסלול מסויים בין המקור ליעד, גודל הודעה לא חסום.

יתרון: ללא שלב הכנה

חסרון: הרבה זכרון בצמתים



3. מיתוג חבילות, מיתוג מנות

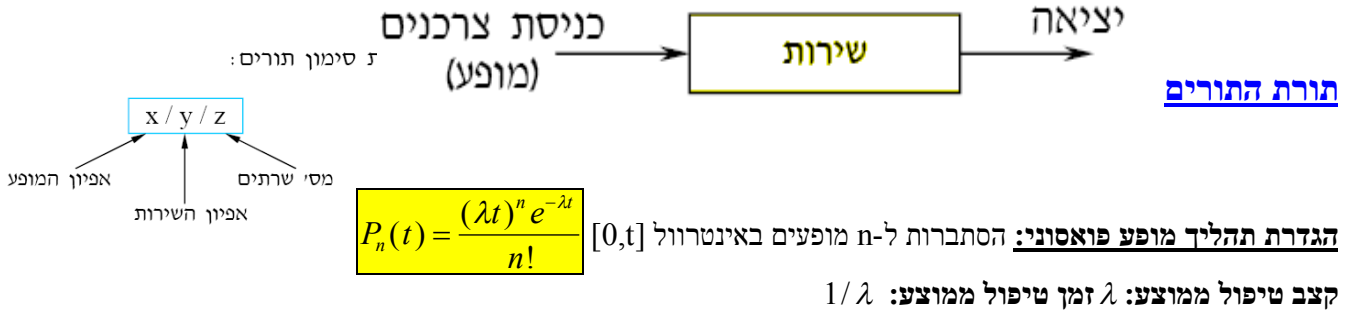
כל חבילה עוברת בנפרד (אולי במסלול שונה)

יתרונות:

ניצול טוב של רוחב הסרט
 פחות השהייה
 פחות זכרון

חסרונות:

נוצרת דחיסות
 ההודעות מגיעות בסדר שונה משנשלחו וצריך לסדרם



תכונות:

1. הסתברות למופע יחיד באינטרוול אינפיניטסימלי dt : $P_1(dt) = (\lambda dt)e^{-\lambda dt} = (\lambda dt)(1 - \lambda dt + \dots) = \lambda dt + O(dt) \approx \lambda dt$
 2. הסתברות שאין מופעים באינטרוול אינפיניטסימלי dt : $P_0(dt) = e^{-\lambda dt} = (1 - \lambda dt + \dots) = 1 - \lambda dt + O(dt) \approx 1 - \lambda dt$
 3. הסתברות שיש שניים או יותר מופעים באינטרוול אינפיניטסימלי dt : $P_k(dt) = \frac{\lambda^k (dt)^k e^{-\lambda dt}}{k!} = O(dt) \approx 0$
- הגדרת $O(dt)$:** $\lim_{dt \rightarrow 0} \frac{O(dt)}{dt} = 0$
4. מופעים באינטרוולים בלתי חופפים הינם בלתי תלויים
 5. סכום שני תהליכי מופע פואסוניים בלתי תלויים בעלי פרמטרים λ_1, λ_2 הוא מופע פואסוני בעל פרמטר $(\lambda_1 + \lambda_2)$
 6. פילוג הזמן בין שני מופעים חופפים הוא אקספוננציאלי עם פרמטר λ
- הגדרת זמן שירות אקספוננציאלי:** $F(t) = P(T \leq t) = 1 - e^{-\mu t}$
- קצב שירות ממוצע:** μ זמן טיפול ממוצע: $1/\mu$ עבור $\mu \rightarrow \infty$ זמן שירות מידי, עבור $\mu \rightarrow 0$ זמן שירות אינסופי

תכונות:

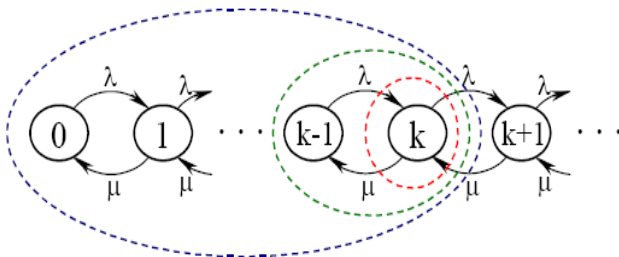
1. הסתברות לזמן טיפול אינפיניטסימלי dt : $P(T \leq t) = 1 - e^{-\mu t} \approx \mu dt$
2. חוסר זכרון: $F(t) = P(T \leq t_0 + t | T > t_0) = P(T \leq t)$

מצב יציב (זה בעיקר מה שמעניין אותנו) - $d/dt=0$ - כל תופעות המעבר דעכו

סימון: הסתברות ל-n צרכנים: $P_n = \lim_{t \rightarrow \infty} P_n(t)$

משפט LITTLE (עבור מצב יציב בלבד!): $\bar{N} = \lambda \cdot \bar{T}$

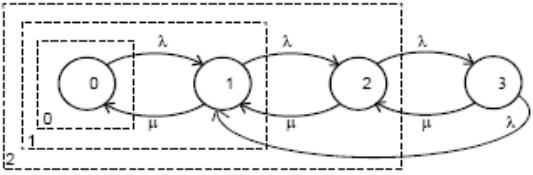
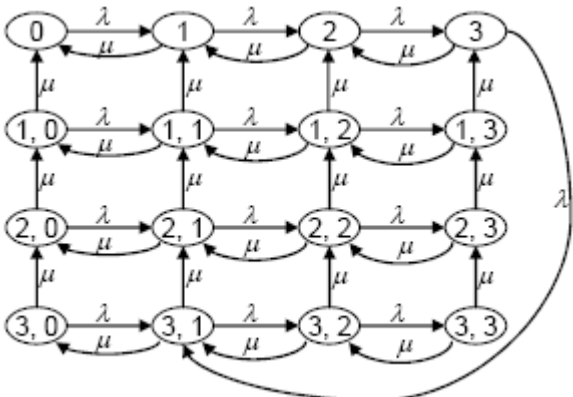
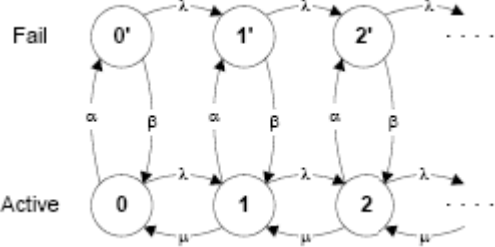
\bar{N} מספר צרכנים ממוצע בתור בנקודת זמן, \bar{T} זמן השהייה ממוצעת במערכת הסבר: צרכן מגיע לתור ורואה \bar{N} צרכנים בתור, לאחר זמן \bar{T} הוא מקבל שירות, עוזב את התור ומשאיר אחריו \bar{N} צרכנים.

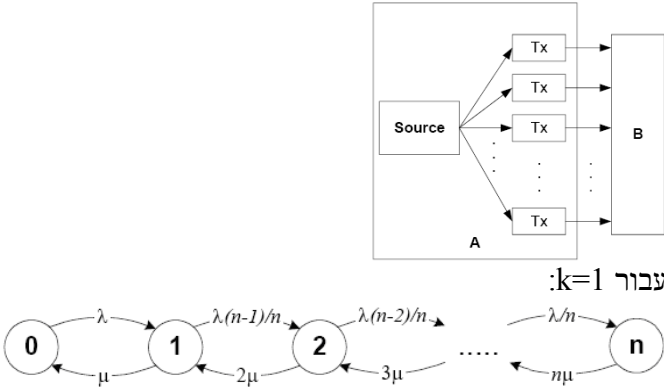


- דיאגרמות זרימה** דרך לתיאור משוואות באמצעות חתכים
- עבור כל חתך יש לשמור על האילוף: **מה שנכנס = מה שיוצא**
- זרימה = **הסתברות המצב * קצב מעבר**
- לדוגמא החתך הכחול יוצר את האילוף: $\lambda P_k = \mu P_{k+1}$
- החתך האדום יוצר את האילוף: $\lambda P_k + \mu P_k = \mu P_{k+1} + \lambda P_{k-1}$

תנאי יציבות	\bar{T} השהייה ממוצעת במערכת \bar{N} מספר לקוחות ממוצע במערכת	P_0, P_n	דיגראמת מצבים	סוג תור
$\frac{\lambda}{\mu} < 1$, $P_0 > 0$	$\bar{N} = \sum_{i=0}^{\infty} i P_i = 1 \Rightarrow$ $\bar{N} = \frac{\rho}{1-\rho} = \frac{\lambda}{\mu-\lambda}$ $\bar{T}_{Little} = \frac{\bar{N}}{\lambda} = \frac{1}{\mu-\lambda}$ <p>השהייה ממוצעת בתור:</p> $\bar{W} = \bar{T} - \frac{1}{\mu} = \frac{\rho}{\mu-\lambda}$ <p>מספר לקוחות ממוצע בתור:</p> $\bar{N}_Q = \lambda \bar{W} = \frac{\lambda \rho}{\mu-\lambda} = \frac{\rho^2}{1-\rho}$	$\lambda P_{n-1} = \mu P_n \Rightarrow$ $P_n = P_0 \cdot \rho^n \quad n = 1, 2, \dots$ $\sum_{i=0}^{\infty} P_i = 1 \Rightarrow \sum_{i=0}^{\infty} P_0 \cdot \rho^i = 1 \Rightarrow$ $P_0 = (1-\rho)$		M/M/1 שרת יחיד, תור יחיד אינסופי, תהליך מופע פואסוני, זמן טיפול אקספוננציאלי.
המערכת תמיד יציבה, תור סופי	$\bar{N} = \rho \underbrace{\frac{(m\rho^{m+1} - (m+1)\rho^m + 1)}{(1-\rho)(1-\rho^{m+1})}}_{\cong S}$ $\cong \rho S$ $*\bar{N}(\rho=1) = \frac{m}{2}$ $\bar{T}_{Little} = \frac{\bar{N}}{\lambda(1-P_m)} = \frac{\rho}{\lambda(1-P_m)} S$ $= \frac{S}{\mu(1-P_m)}$ <p>נשים לב: פה מוגדרת הסתברות לאובדן חבילה (מצב P_m)</p> <p>* קצב אובדן חבילות: $R = P_m \cdot \lambda$</p>	$P_n = P_0 \cdot \rho^n, 0 \leq n \leq m$ $P_0 = \begin{cases} \frac{1-\rho}{1-\rho^{m+1}} & ; \rho \neq 1 \\ \frac{1}{m+1} & ; \rho = 1 \end{cases}$		M/M/1/m שרת יחיד, תור יחיד בגודל m, תהליך מופע פואסוני, זמן טיפול אקספוננציאלי.

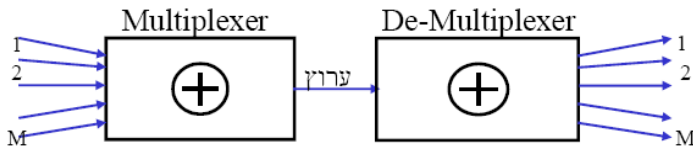
תנאי יציבות	\bar{N}, \bar{T}	P_0, P_n	דיגרמת מצבים	סוג תור
$\frac{\lambda}{\mu} < 1,$ $P_0 > 0$	$\bar{N} = \sum_{i=0}^{\infty} iP_i = \sum_{i=1}^{\infty} i \frac{\rho^i}{i} P_0 =$ $\sum_{i=1}^{\infty} \rho^i P_0 = P_0 \frac{\rho}{1-\rho} \Rightarrow$ $\bar{N} = \frac{\rho}{1-\rho} [1 - \ln(1-\rho)]$ <p>זמן השהייה ממוצע: נחשב קודם זמן הגעה הממוצע:</p> $\bar{\lambda} = \sum_{i=0}^{\infty} \lambda_i P_i = \lambda P_0 + \sum_{i=1}^{\infty} i \lambda \frac{\rho^i}{i} P_0 =$ $\lambda P_0 \left(1 + \sum_{i=1}^{\infty} \rho^i \right) = \frac{\lambda}{1-\rho} P_0$ <p>לפי משפט ליטל: $T = \frac{\bar{N}}{\bar{\lambda}} = \frac{1}{\lambda - \mu}$</p>	$= \frac{\lambda}{\mu} \cdot \frac{\lambda}{2\mu} \cdot \frac{\lambda}{3\mu} \dots \frac{(k-1)\lambda}{k} P_0 = \frac{\rho^k}{k} P_0, 1 \leq k$ <p>מנרמול: $P_0 + \sum_{k=1}^{\infty} \frac{\rho^k}{k} P_0 = 1$ נקבל משימוש בנוסחה:</p> $\sum_{k=1}^{\infty} \frac{\rho^k}{k} = -\ln(1-\rho)$ $P_0 = \frac{1}{1 + \sum_{k=1}^{\infty} \frac{\rho^k}{k}} = \begin{cases} \frac{1}{1 - \ln(1-\rho)}, & \rho < 1 \\ 0, & \rho \geq 1 \end{cases}$		<p>תור M/M/1 עם קצבים משתנים כשהמערכת ריקה קצב ההגעה: λ עבור מערכת עם n צרכנים קצב ההגעה הוא: $n\lambda$ וקצב הטיפול $n\mu$</p> <p>מהתוצאה על זמן ההשהייה נובע כי אפקטיבית אין המתנה בתור!</p>
המערכת תמיד יציבה, תור סופי	$\bar{N} = \rho \left(1 - \frac{\rho^m}{S \cdot m!} \right)$ $\bar{T} = \frac{1}{\mu} \left(1 - \frac{\rho^m}{S \cdot m!} \right)$	$P_k = \frac{\rho^k}{k!} P_0, 0 \leq k \leq m$ $P_0 = \frac{1}{\sum_{k=0}^m \frac{\rho^k}{k!}} \equiv \frac{1}{S}$		<p>M/M/m/m m שרתים, תור יחיד בגודל m, תהליך מופע פואסוני, זמן טיפול אקספוננציאלי</p>
		<p>תרגול 4 – חישוב מזעזע...</p>		<p>שירות ראשון חריג זמן השירות ללקוח שמגיע למערכת ריקה הוא μ' זמן שירות ללקוח שמגיע למערכת לא ריקה הוא μ</p>
<p>לשים לב: אין תור = זמן טיפול $1/\mu$ עבור תור סופי המערכת תמיד יציבה</p>				

תנאי יציבות	\bar{N}, \bar{T}	P_0, P_n	דיגראמת מצבים	סוג תור
$\frac{\lambda}{\mu} < 1,$ $P_0 > 0$	$\bar{N} = 0 \cdot P_0 + 1 \cdot P_1 + 2 \cdot P_2 + 3 \cdot P_3 = P_1 + 2P_2 + 3P_3$ $= \frac{\rho + 3\rho^2 + 6\rho^3}{1 + 2\rho + 3\rho^2 + 3\rho^3}$ $\bar{T} = P_0 \cdot \mu^{-1} + P_1 \cdot 2\mu^{-1} + P_2 \cdot 3\mu^{-1} + P_3 \cdot \mu^{-1}$	<p>מספר ההודעות הממוצע בתור (הנוכחי!): לפי דיאגראמת המצבים העליונה: 0: $\lambda P_0 = \mu P_1 \Rightarrow P_1 = \rho P_0$ 1: $\lambda(P_1 - P_3) = \mu P_2 \Rightarrow P_2 = \rho(P_1 - P_3) = \rho^2 P_0 - \rho P_3$ 2: $\lambda(P_2 - P_3) = \mu P_3$</p> $P_2 = \frac{\rho^2 + \rho^3}{1 + \rho + \rho^2} P_0$ $P_0 + P_1 + P_2 + P_3 = 1$ <p style="text-align: center;">⋮</p> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: auto;"> $P_0 = \frac{1 + \rho + \rho^2}{1 + 2\rho + 3\rho^2 + 3\rho^3}$ </div>	<p>מספר לקוחות בתור הנוכחי</p>  <p>סכימה עבור 2 תורים</p>  <p>הסבר: מצב (i,j) מסמל i לקוחות בתור 1 וj לקוחות בתור 2</p>	<p>(ת"ב 3 שאלה 1) מערכת עם אינסוף תורים בגודל 3 אינסוף שרתים, לכל אחד תור בגודל 3. כשתור i מתמלא עוברים לתור i+1 ולא חוזרים אליו יותר</p>
$P_{active} \cdot \mu > \lambda$ $\lambda < \frac{\mu\beta}{\alpha + \beta}$ $\beta > 0$		$\sum_{i=0}^{\infty} P_i \beta = \sum_{i=0}^{\infty} P_i \alpha, \quad P_{Fail} + P_{Active} = \sum_{i=0}^{\infty} P_i + \sum_{i=0}^{\infty} P_i = 1$ $P_{Active} \cdot \alpha = (1 - P_{Active}) \beta \Rightarrow P_{Active} = \frac{\beta}{\alpha + \beta}$ $P_{Busy} = \sum_{i=1}^{\infty} P_i$ $(P_0 + P_1) \cdot \lambda = P_1 \cdot \mu$ <p style="text-align: center;">⋮</p> $\lambda \sum_{i=0}^{\infty} (P_i + P_i) = \mu \sum_{i=1}^{\infty} P_i \Rightarrow \lambda \cdot 1 = \mu \cdot P_{Busy} \Rightarrow P_{Busy} = \frac{\lambda}{\mu}$		<p>(ת"ב 3 שאלה 2) שרת יחיד שמתקלקל מופע הקלקולים הוא פואסוני עם פרמטר α, מופע התיקונים פואסוני עם פרמטר β</p>

תנאי יציבות	\bar{N}, \bar{T}	P_0, P_n	דיגראמת מצבים	סוג תור
מספר מצבים סופי, תמיד יציבה		<p>עבור $k=1$:</p> $P_i = \frac{n-i+1}{i\mu} \lambda P_{i-1} = \left[\prod_{j=1}^i \left(\frac{n-j+1}{jn} \right) \rho \right] P_0$ $= \left(\frac{\rho}{n} \right)^i P_0 \prod_{j=1}^i \frac{n-j+1}{j} = \frac{n!}{(n-i)! i!} \left(\frac{\rho}{n} \right)^i P_0$ $= \binom{n}{i} \left(\frac{\rho}{n} \right)^i P_0, \forall 0 \leq i \leq n \quad \sum_{i=0}^n P_i = 1 \Rightarrow$ $\sum_{i=0}^n \binom{n}{i} \left(\frac{\rho}{n} \right)^i P_0 = \left(\frac{\rho}{n} + 1 \right)^n P_0 = 1 \Rightarrow P_0 = \left(\frac{\rho}{n} + 1 \right)^{-n}$ <p>הסתברות שהודעה תישלח: $\frac{n}{n+\rho}$</p> <p>לכן התעבורה היא: $\lambda \frac{n}{n+\rho}$</p> <p>עבור $k=n$: כמו תור M/M/m/m</p>	 <p>עבור $k=1$: כמו תור M/M/m/m</p> <p>עבור $k=n$: כמו תור M/M/m/m</p>	<p>(ת"ב 4 שאלה 1) n תורים בגודל 1 מתוכם נבחרים באקראי k במערכת מחוברים n תאי Tx בגודל 1, כשרוצים לשלוח בוחרים k תאים ושולחים עותקים</p>

DEMULTIPLEXING-ו MULTIPLEXING

משמש להעביר מספר זרמי מידע על ערוץ יחיד.



סוגי מרבבים:

מרבב קבוע:

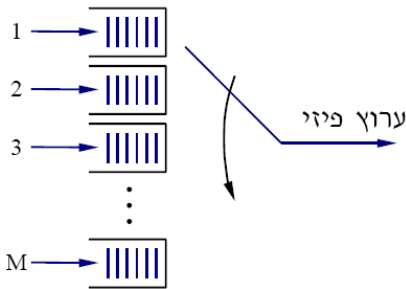
לכל אחד ממקורות המידע זמן שידור קבוע מראש גם אם אין לו מה לשדר. (TDMA).

יתרונות:

- זמן שידור מובטח לכל מקור.
- אין צורך בזיהוי המקור בגוף ההודעות.
- ניתן "לשלוף" בקלות רבה מידע של מקור מסוים.

חסרונות:

- בזבוז זמן (רוחב סרט) כשאין למקור מסוים מה לשדר.
- כל מקור מוגבל כל הזמן בקצב השידור האפקטיבי שלו.
- כדי להוסיף מקורות מידע, צריך לקצר זמן יחסי (להקטין קצב אפקטיבי) של כל מקור (ע"י קיצור פיזי של זמן השידור של כל מקור או ע"י הגדלת קצב השידור הכולל).



מרבב סטטיסטי:

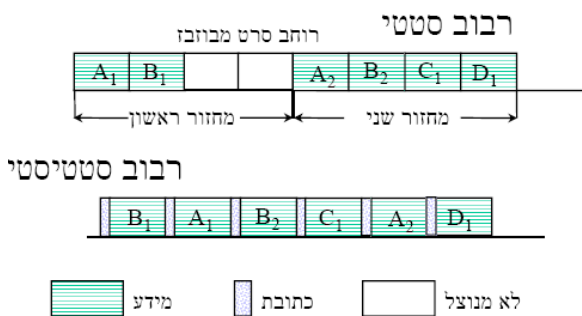
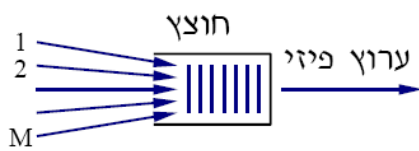
משדר את ההודעות לפי סדר הגעתן.

יתרונות:

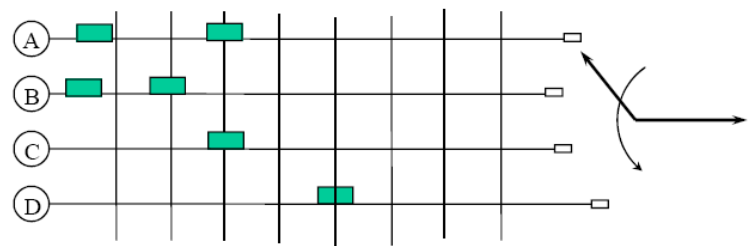
- ניתן לנצל בצורה טובה יותר את רוחב הסרט הכולל.
- מקורות המידע אינם מוגבלים כל הזמן בקצב השידור האפקטיבי (עדיין יש מגבלה של קצב השידור הכולל).
- ניתן להוסיף בקלות רבה מקורות מידע נוספים, ללא שינויים מהותיים במבנה המרבב.

חסרונות:

- אין הבטחת זמן שידור לכל מקור.
- המרבב צריך לזהות כל מקור בגוף ההודעות.
- כדי לשלוף הודעות של מקור מסוים, צריך להסתכל על כל ההודעות כל הזמן.



רובב סטטי וסטטיסטי - דוגמא



פרוטוקולים קוויים: STOP & WAIT

שיטת פינג פונג (שליחה -> קבלת ACK -> שליחה וחוזר חלילה..)

שיטה לקווי Half Duplex – קווים בהם אפשר להעביר מידע רק בכיוון אחד.

הנחות:

- ACK מכיל את מספר המסגרת לה מצפים.
- מסגרות לא עוקפות (FIFO)
- לשם מספור המסגרות מספיקה סיבית 1 (זוגי, אי זוגי)

הגדרות:

- זמן עיבוד = זמן מרגע הגעת המסגרת לתחנת יעד עד לשליחת ה ACK של תחנת היעד על קבלת המסגרת.

- t_I זמן שידור מסגרת האינפורמציה

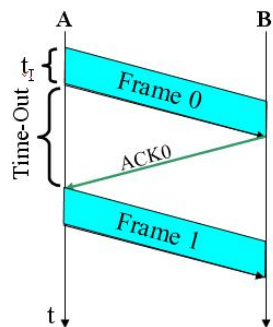
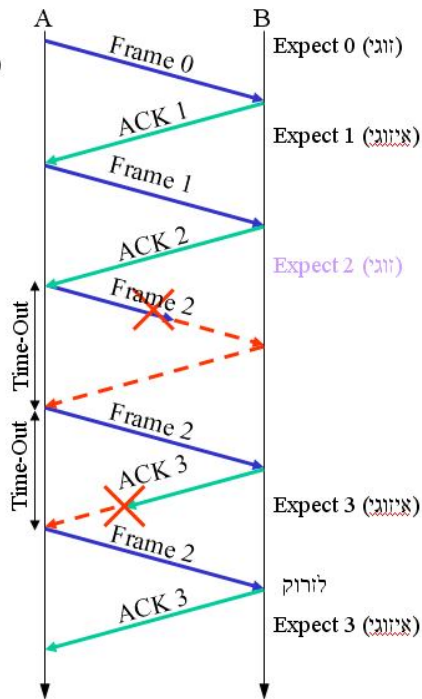
- t_P זמן התפשטות בתוך

- t_{out} (timeout) = הזמן המקסימלי מתחילת שידור מסגרת עד לקבלת ACK,

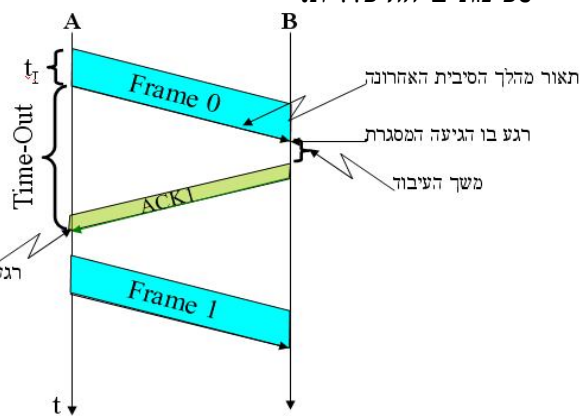
אם לא קיבלנו ACK עד לזמן timeout נשלח את אותה מסגרת.

- t_T זמן המחזור

סכימת שיחה כללית:



ובמערכת אידאלית זמני עיבוד ושידור ACK אפסיים:



חישובים במערכת S&W אידאלית:

$$S = \frac{t_I}{t_T} \triangleq \frac{1}{a}$$

נצילות

$$t_T \triangleq t_I + t_{OUT}$$

זמן מחזור:

עבור S&W עם ערוץ רועש = הסתברות P לכשלון בשליחת מסגרת או ACK:

P [מסגרת תשודר פעם אחת] = $(1-p)$

P [מסגרת תשודר פעמיים] = $p(1-p)$

P [מסגרת תשודר k פעמים] = $p^{k-1}(1-p)$

$$\bar{k} = \sum_1^{\infty} k \cdot p^{k-1} (1-p) = \frac{1}{1-p}$$

מספר שידורים ממוצע:

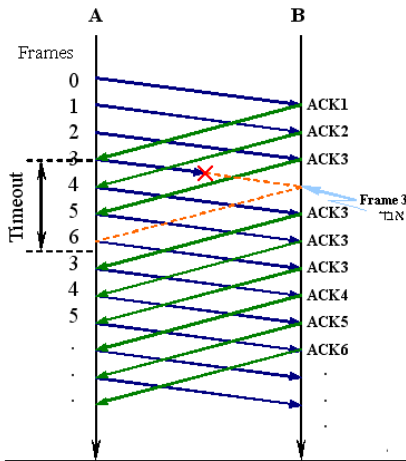
$$t_V = \bar{k} \cdot t_T = \frac{t_T}{1-p}$$

זמן שידור ממוצע להודעה:

$$S = \frac{t_I}{t_V} = \frac{(1-p)t_I}{t_T} = \frac{(1-p)t_I}{2t_I + 2t_P} = \frac{1-p}{2+2a} = \frac{\text{זמן שידור אופטימלי}}{\text{זמן לשידור בפועל}}$$

פרוטוקולים קוויים: Go Back N

שיפור של S&W עבור הזמן בו הערוץ בטל = שליחת מספר רב של מסגרות מתוך הנחה כי לא צריך לחכות לACK



מהלך הפרוטוקול

- A משדר מסגרות לפי הסדר בזו אחר זו.
- כאשר מתקבל NAK (או Time out) חוזר למסגרת האחרונה עליה לא קיבל חייווי וממשיך משם לפי הסדר.
- כל חיווי מאשר את כל המסגרות הקודמות
- ACK_N נשלח = כל המסגרות מ-0 עד n-1 התקבלו נכון
- אם זמן ה-Time Out ארוך מספיק אפשר להתגבר על אבדן חיווי ("ע"י קבלת חייווי ACK של מסגרת מאוחרת יותר)
- שימוש ב NAK מאפשר שדור חוזר בזמן מקוצר (אך צריך להיזהר בתזמונים)

תעבורה

מספר שידורים ממוצע הוא כמו ב S&W: $\bar{k} = \frac{1}{1-p}$

מסגרת שנשלחת פעם אחת דורשת זמן של t_I

מסגרת שנשלחת פעמיים דורשת זמן של $t_T + t_I$

מסגרת שנשלחת K פעמים דורשת זמן של $(k-1) \cdot t_T + t_I$

זמן ממוצע לשליחת מסגרת: $t_V = E[(k-1) \cdot t_T + t_I] = (\bar{k}-1)t_T + t_I$

$$t_V = \left(\frac{1}{1-p} - 1 \right) \cdot t_T + t_I = t_I \cdot \left(1 + \frac{ap}{1-p} \right)$$

והנצילות היא: $S = \frac{t_I}{t_V} = \frac{1-p}{1-p+ap}$

GBN - סיכום

- כאשר משדרים מסגרת מחדש מניחים שכל המסגרות שאחריה לא הגיעו.
 - ⇐ גורם לחסכון חוצצים אצל B.
 - ⇐ מפשט את יניחול החשבונות אצל A.
 - B מקבל מסגרות לפי הסדר בלבד (זורק מסגרות להן איננו מצפה, גם לא משובשות).
 - חיווי מסגרת מעיד על קבלת כל המסגרות שלפניה.
 - B רשאי לשמור מסגרות לא לפי הסדר אך אינו יכול לאשרן.
- הפרוטוקול חייב לעבוד כשורה גם אם באופן אקראי מסגרות או חיוויים הולכים לאיבוד**
- אובדן חבילות / חיוויים ← יתכן מצב קיפאון אם B לא שולח ACK.
 - B ← שולח מדי פעם הודעת ACK על המסגרת האחרונה לה הוא מצפה.
 - A זורק את חבילות המקור רק לאחר קבלת חיווי.
 - הגודל N מתאר את המספר המקסימלי של מסגרות שמותר ל-A לשלוח לפני קבלת חיווי (גודל החלון).

בעיית חלון המסגרות

חלון: מספר המסגרות שמותר ל-A לשלוח לפני שמקבל אישור על הראשונה. מספרים סידוריים עולים ועקב סופיות אורכי המסגרות לא ניתן לייצר מספרים עולים לאינסוף.

דוגמא: שדה המספור הוא בן 3 סיביות. (כל המספרים הם מודולו 8). ברור שגודל החלון לכל היותר 8, אחרת B לא ידע אם למשל Frame 0 מכיל מידע של המסגרת הראשונה או של המסגרת התשיעית. האם חלון = 8 בסדר? **תשובה: לא**

נשלחו מסגרות מספר 0 עד 7 והתקבלו חיוויים. נשלחו מסגרות 8 - 15 (שקול ל 0 - 7 לפי המודולו) וכולן אבדו.

המקבל מאשר את מסגרת 7 (עם ACK0) אך השולח מבין זאת כאישור למסגרת 15. טענה: חלון = 7 בסדר

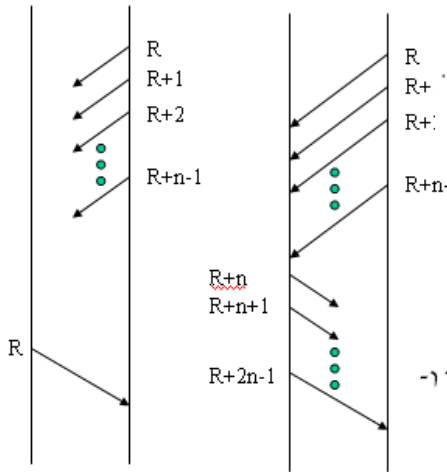
פרוטוקולים קוויים: SR – Selective Repeat

שיפור של GBN: המקבל מאשר כל מסגרת בנפרד ומבקש שליחה חוזרת של מסגרות שחסרות לו

פרוטוקול Selective Repeat (SR) חלון $m=2n$

ניתן לקבל מסגרות שלא לפי הסדר!

משדר:



- משדרת חבילות ברצף לפי הסדר בזו אחר זו.

- כאשר מקבלת $Nack(i)$ משדרת שוב את הודעה מס' i (או לאחר $time-out(i)$).

- $ack(i)$ מאשר קבלת חבילות עד i כולל.

- התחנה המשדרת חייבת לשמור עותק של כל חבילה עד קבלת אישור הקליטה.

- התחנה הקולטת מעבירה לרמת ה- Network חבילות לפי הסדר. הודעות שהגיעו לא

- לפי הסדר נשמרות בחוצצים עד הגעת קודמותיהן.

מקלט:

- קולטת חבילות גם לא לפי הסדר.

- משדרת $Nack(i)$ במידה וחל שיבוש בחבילה i או התקבלו חבילות במספור גבוה יותר ו-

- i לא התקבלה.

- משדרת $ack(i)$ אם קיבלה חבילות עד i כולל.

נצילות במצב אידאלי (בהנחה שאין שידורים חוזרים מיותרים): $\eta = 1 - p$ יותר טובה מ-GBN!

שיפורים:

PIGGYBACK "שק קמח"

– הודעת מידע או בקרה נושאת אישור של הפרוטוקול לכוון ההפוך

– האישור מעוכב זמן מוגבל כדי לנצל הזדמנות

– איחוד אישורים

Transmitter Initiated

– משדר יכול לבקש סטטוס על קבלת אינפורמציה במקלט

– מקלט מאשר על מסגרות מידע או בתגובה לסטטוס

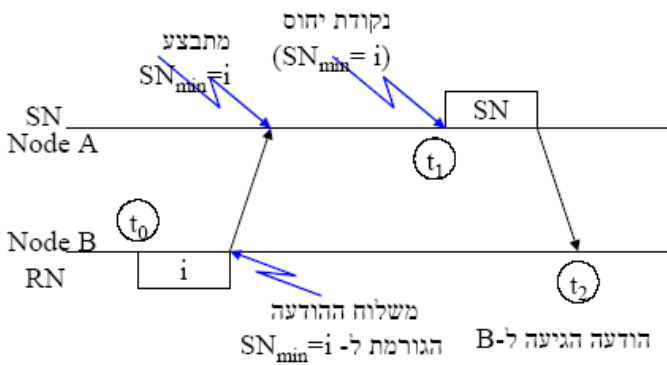
הגדרות:
 משדר:

SN_{min} - מספר ההודעה הקטן ביותר שטרם אושר.
 SN_{max} - מספר ההודעה הקטן ביותר שטרם נשלח.

כל ההודעות הנשלחות נושאות: $SN \in [SN_{min}, SN_{max})$
 ונחשבות "בדרך" או "בטיפול".

מקלט:

RN - מספר ההודעה הבאה שמצפים לה.



הצורך באתחול דרוש בכל פעם שקו נופל ועולה מחדש. יש לדאוג לסילוק המסגרות הישנות מן הקו!

בעיה:

כיצד להגדיר את רגע נפילת הקו?
 כיצד להגדיר רגע זה בשני צידי הקו?

פתרון: נגדיר תקופת זמן שבה הצדדים מסכימים כי דרוש אתחול.

הנחה: קיים מנגנון העוקב ומזהה נפילת קווים. (מבוסס למשל על אי התקדמות SN או RN)

הערות:

- אין מנגנון חסין למעקב. המנגנון עשוי להכריז על קו שבור למרות שהקו פועל כהלכה.
- מנגנון טוב צודק ברוב המקרים.
- מהפרוטוקול חייב לעבוד כשורה גם אם הכרזת הנפילה הייתה מוטעית.

תכונות פרוטוקול אתחול (פרוטוקול המקיים תכונות אלה נקרא "פרוטוקול מבטיח סנכרון"):

1. עקיבה (Follow Up) - להבטיח שאם צד אחד מכריז על נפילה (מוצדקת או מוטעית) יעשה כך גם הצד השני.
2. ניקוי הערוץ (Clear) - קיום זמן מסוים בין נפילה לעליה שבו הערוץ ריק ממסגרות ישנות.
3. אתחול המשתנים (Reset)
4. מניעת קיפאון (Deadlock Freedom) - לדאוג לכך שאם הקו "בסדר" שני הצדדים יתחילו להעביר מסגרות.

משפט: פרוטוקול GBN עם מספרים סופיים העובד בצרוף לפרוטוקול אתחול המבטיח סנכרון, מבטיח תקשורת אמינה

בדיקות תקשורת אמינה לפרוטוקול:

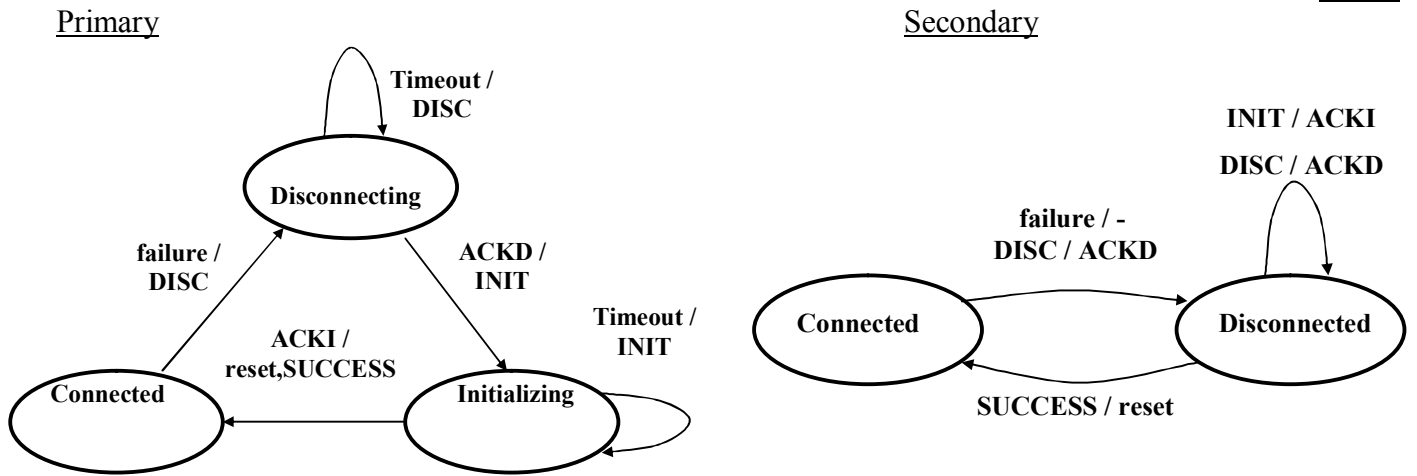
1. לא יהיו שכפולי חבילות:
2. לא ילכו לאיבוד חבילות
3. לא יהיה DEADLOCK

עם התקנת הקו מחליטים:

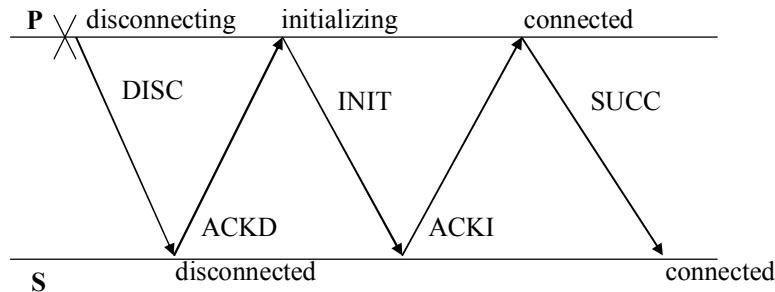
צד אחד יהא צד ראשוני (Primary) צד שני יהא צד משני (Secondary)
חשוב: תכונת Primary, Secondary תופסת רק לשלב האתחול. בשלב העבודה יש שני פרוטוקולים DLC, כמתואר לעיל.

המצב "איתחול" (Initialization) הוא מצב בו נמצאים, עד אשר עוברים למצב "מחובר" (Connected).

דוגמא:



הסבר לדיגרמה:



דוגמא לאתחול:

פרוטוקול אתחול - הערות

- S נכנס ל - Up לפני P. יש פרוטוקולים המוודאים כי P נכנס ל - Up ראשון.
- ניתן לשלוח מסגרות מבחן לפני המעבר ל - Up כדי למנוע מעברים מיותרים (קרוטס).
- יש פרוטוקולים מאוזנים שאינם מבוססים על הקצאה ראשוני - משני. איך המערכת עובדת:
- פרוטוקול אתחול עובד כל הזמן.
- הקו עולה ויורד ופרוטוקול האתחול מסנכרן בין הצדדים.
- כפי שנראה בהמשך, פרוטוקול HDLC משתמש במסגרות SNRM (מתאים בתור INIT), DISC, UA לצורך אתחול. האתחול מבוסס על חסם עליון של זמן התפשטות ותגובה, ואינו עובד כשורה אם זמנים אלה לא ניתנים לחסימה. אחת הבעיות היא ש - UA משמש גם כ - ACKD וגם כ - ACKI.

Connected=Up*

פרוטוקולי אתחול: HDLC

פרוטוקול סטנדרטי ממשפחת N - Go - Back .

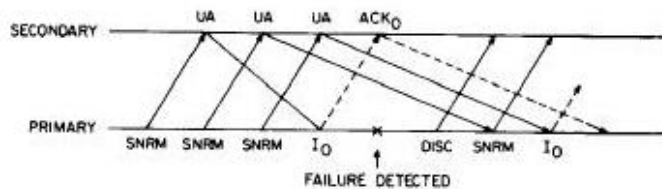
אחד הפרוטוקולים היותר נפוצים.

פרוטוקול HDLC משתמש במסגרות SNRM (מתאים בתור INIT), UA, DISC, לצורך אתחול. האתחול מבוסס על חסם עליון של זמן התפשטות ותגובה,

ואינו עובד כשורה אם זמנים אלה לא ניתנים לחסימה.

אחת הבעיות היא ש - UA משמש

גם כ - ACKD וגם כ - ACKI .



מבנה מסגרת:

F	Addr	Ctrl	Data	CHK	F
---	------	------	------	-----	---

F - דגל המסגור בד"כ 01111110. מותרים גם אחרים.

Addr - כתובת. זהות התהליך בצומת המקבל.

Ctrl - בקרת המסגרת (ראה השקף הבא).

Data - הנתונים המועברים במסגרת זו.

CHK Checksum - לצורך בקרת שגיאות בד"כ CRC-16 או CRC-32.

HDLC Control Field

Information Frame

1	2	3	4	5	6	7	8
0	Seq # (SN)		P/F	Next (RN)			

Supervisory Frame

1	2	3	4	5	6	7	8
1	0	Type	P/F	Next (RN)			

Supervisory Frame types

Type	Short Name	Name	Meaning
0	RCVRRDY	Receiver Ready	ACK
1	RJCT	Reject	NACK
2	RNR	Receiver not Ready	Stop
3	SRJ	Selective Reject	

Unnumbered Frame

1	2	3	4	5	6	7	8
1	1	Type	P/F	Modifier			

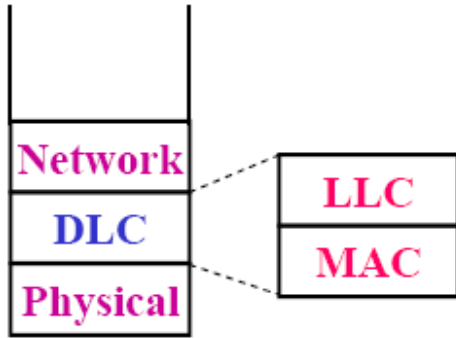
Unnumbered Frame types

Disc - Disconnect ; SNRM – Set Normal Response Mode (INIT)

FRMR - Frame Reject

UA - Unnumbered ACK (ACKD and ACKI)

MAC: בקרת כניסה לערוצים משותפים



עבור ערוצים משותפים, שכבת הקו (LINK) מחולקת ל-2: **LLC** – Link Level Control, בקרת העברת מידע קווי (תמיד קיימת, ראינו פרוטוקולים שלה: S&W,GBN,SR) **MAC** – Medium access control מבקרת את השליחה לערוץ המשותף לכמה תחנות המשרות באופן בלתי תלוי

בעיות מרכזיות:

נצילות: כמות התעבורה והשהייה.

יציבות: האם הרשת שומרת על תכונותיה בעומס משתנה \ גבוה.

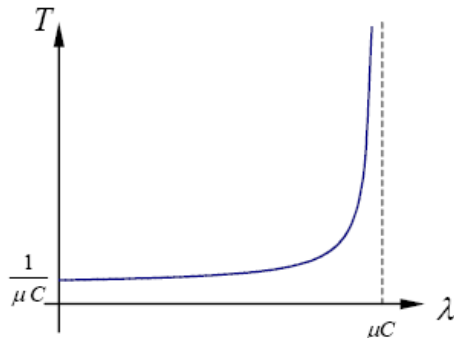
חסם על מדדים: השהייה וזמן גישה. (שימושי זמן אמת).

ערוץ כללי קלאסי (לא משותף):

מופע פואסוני עם פרמטר λ

אורך הודעה מפולג אקספוננציאלי עם תוחלת $1/\mu$

קיבול הערוץ: C ביט לשניה



משך שידור הודעה: $1/(\mu C)$ זמן ממוצע לשליחת הודעה: $\bar{T} = 1/(\mu C - \lambda)$

חלוקת ערוץ סטטית:

FDMA – Frequency Division Multiple Access - (חלוקת תדר) תחום התדרים מתחלק בין הלקוחות השונים.

עבור N לקוחות: קיבול אפקטיבי של הערוץ C/N

קצב הגעה אפקטיבי לכל תת ערוץ λ/N

$$\bar{T}_{FDMA} = \frac{1}{\left(\mu \frac{C}{N} - \frac{\lambda}{N}\right)} = N \cdot \bar{T}$$

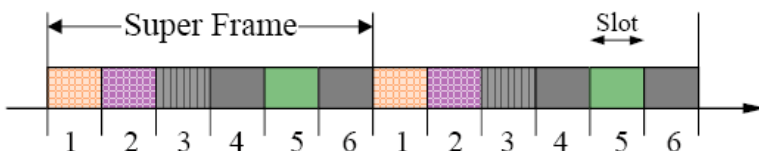
לכן זמן ההמתנה הממוצע הוא:

TDMA – Time Division Multiple Access

(חלוקת תדר) תחום הזמן מתחלק בין הלקוחות השונים.

לכן זמן ההמתנה הממוצע הוא: $\bar{T}_{TDMA} = N \cdot \bar{T}$

TDMA Schedule



הבעיה היא ששיטות חלוקת ערוץ סטטיות טובות בעיקר אם אנחנו יודעים את מספר הצרכנים וכולם מעבירים מידע בערך במידה שווה.

לכן לעיתים ננסה לחלק את כל הערוץ לכל אחד מהצרכנים.

MAC: שיטת ALOHA

- כל צרכן משדר מתי שהוא רוצה לאחר הגעת הודעתו.
- מאחר שכל התחנות שומעות אחת אתה שניה קיימות התנגשויות (Collisions) הניתנות לגילוי **לאחר השידור**,
- הודעה שהתנגשה תשודר מחדש.
- זמן ההשהיה לשידור מחדש הוא אקראי כדי לא ליצור סדרה אינסופית של התנגשויות.
- כאשר יש התנגשויות, כל ההודעות המתנגשות תשודרנה מחדש (אפילו אם החפיפה היא של ביט אחד).

בעיה: קצב נמוך = נצילות ערוץ נמוכה. קצב גבוה = הרבה התנגשויות = נצילות ערוץ נמוכה. **צריכים לקבוע נקודת עבודה!** בערוץ כללי מגדירים:

תעבורה (throughput) כמות המידע הממוצעת העוברת דרך הערוץ ביחידת זמן (נמדד בביטים לשניה) בערוץ משותף מגדירים **גודל אחר:**

$$S = \frac{\text{כמות מידע שהועברה}}{\text{כמות מידע אידיאלית}} = \frac{\text{תעבורה}}{\text{קצב שידור}} \quad (\text{נצילות } S)$$

$S = \text{percentage of time that the channel successfully transmits information}$

ניתוח תעבורת ALOHA – הנחות:

- אורך כל ההודעות זהה ושווה ל- T שניות, זמן שידור.
- המופע של הודעות חדשות הוא פואסוני.
- מספר צרכנים אינסופי.

• g קצב הופעת (תזמון) הודעות למשלוח בערוץ.

- + גם חדשות וגם ישנות.
- + לא כל ההודעות מצליחות
- + הגודל G נקרא עומס מוצע (Offered Load).
- + הנחת קירוב - העומס המוצע מפולג פואסוני אף הוא.
- P_{SUC} הסתברות הצלחה של הודעה

• $G = g \cdot T$ מספר ההודעות הממוצע הנשלח בערוץ במשך זמן השווה לזמן שידורה של הודעה אחת (נשלח = מוצלח או לא).

• $g \cdot T$ הוא מספר ההודעות הממוצע שמשדרים במשך זמן שידורה של הודעה אחת אך רק P_{SUC} מתוכן מצליחות.

• תעבורה:

$$S = g \cdot P_{SUC} + (1 - P_{SUC}) \cdot g = g \cdot T \cdot P_{SUC}$$

$S = 0 \cdot g(1 - P_{SUC}) + T \cdot g \cdot P_{SUC} = g \cdot T \cdot P_{SUC}$

ניתוח תעבורת ALOHA – סימונים:

S: תעבורה. η : נצילות – תעבורה לקצב שידור. T: זמן שידור הודעה. g: עומס מוצע – קצב ממוצע של הופעת הודעות למשלוח, חדשות וישנות.

חישובים:

אורך החבילות אחיד $\left[\begin{matrix} \text{יח' זמן} \\ \text{חבילה} \end{matrix} \right]$, קצב הגעת הודעות חדשות וישנות הוא פואסוני $P_k(t) = \frac{(gt)^k}{k!} e^{-gt}$

אזי ממוצע מספר ההודעות הנשלחות (לא בהכרח מצליחות) בפרק זמן T: $G \hat{=} gT$ משך תקופת הפגיעה: $2T$.

הסתברות להצלחת שידור בפרק זמן T: $P_{success} = P\left(\begin{matrix} \text{לא תשלח אף הודעה לערוץ} \\ \text{בפרק זמן } 2T \end{matrix} \right) = P_{\text{אף הודעה}}(k=0, t=2T) = e^{-2gT}$

והתעבורה היא: $S = Ge^{-2gT}$

מקסימום ב $G = 1/2$ לכן: $S_{max} \approx 0.18$ תעבורה מקסימלית די נמוכה

MAC: שיטת SLOTTED ALOHA

- הזמן מחולק למרווחי זמן שווים של T יח' זמן (Slots)
- שידור מתבצע רק מתחילת slot

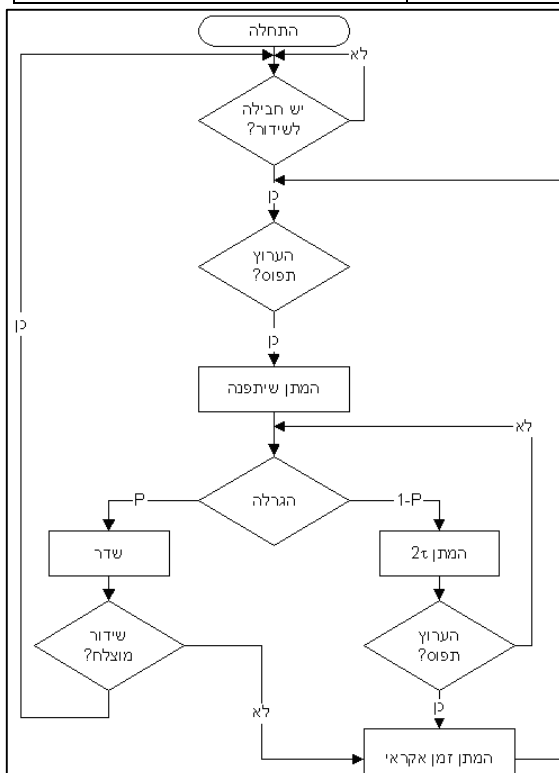
תקופת הפגיעה מצטמצמת ל-T, לכן: $P_{success} = P_{N=0}(t=T) = e^{-gT}$

והתעבורה היא: $S = Ge^{-gT}$ מקסימום ב $G = 1$ לכן: $S_{max} \approx 0.36$

MAC: CSMA (הישת גל נושא)

שכלול של ALOHA נועד לצמצם את זמן הפגיעה. התחנות יכולות לזהות אם הערוץ פנוי. לפני כל נסיון הן בודקות ויש שלוש וריאציות לפעולה:

פרוטוקול	פנוי	תפוס	משמעות
1-persistent	משדר	מחכה שיתפנה ומשדר מיד (אם התנגש, מחכה זמן אקראי לשידור החוזר)	מעודד התנגשויות בערוצים עם עומס גבוה.
Non-persistent		ממתין זמן אקראי ובודק שוב	מגדיל זמן בטלה בערוצים עם עומס נמוך.
P-persistent		ממתין שיתפנה ואז בהסתברות P: משדר מיד. בהסתברות (1-P): ממתין 2τ (הסבר ב-CD) ואז בודק שוב. אם פנוי מגריל שוב P; אם תפוס ממתין זמן אקראי ומתחיל מהתחלה.	נקודת אמצע בין שתי האפשרויות.



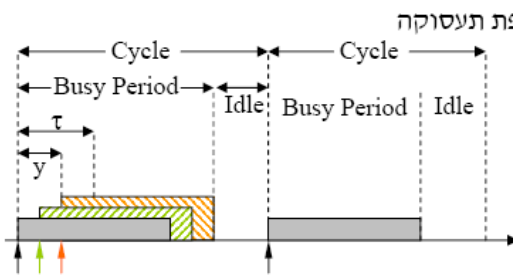
מודל P-Persistent

ביצועים CSMA

הנחות:

- כל ההודעות שוות באורכן: T שניות.
- זמן התפשטות בין כל שני צרכנים: τ שניות.
- העומס המוצע על הערוץ: g.
- המקרה הגרוע כל זוג צמתים מקור - יעד נמצאים במרחק τ .

התבוננות בציר הזמן:



סדרה של מחזורים (Cycles) כאשר כל מחזור מורכב מתקופת תעסוקה (Busy) ותקופת בטלה (Idle).

אורך התקופות אינו קבוע!

- B אורך ממוצע של תקופת תעסוקה.
- I אורך ממוצע של תקופת בטלה.
- $C = B + I$ אורך ממוצע של זמן המחזור.

- קיימים שני סוגים של תקופת תעסוקה: מוצלחת, לא מוצלחת.
- לא יותר מהודעה מוצלחת אחת במחזור.

תעבורה:

$$S = \frac{U}{C} = \frac{U}{B+I}$$

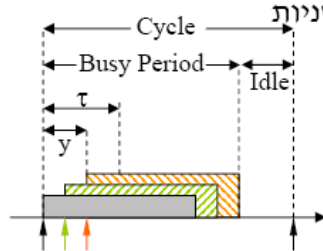
אם במשך מחזור, הערוץ "מבלה" בממוצע U שניות בהעברת אינפורמציה "טובה" אזי

חישוב U:

הזמן "הטוב": בתקופה מוצלחת T. בתקופה לא מוצלחת 0.

$$U = T \cdot P_{SUC} + 0(1 - P_{SUC}) = T \cdot P_{SUC}$$

- P_{SUC} היא ההסתברות למחזור מוצלח. רק ההודעה הראשונה במחזור יכולה להיות מוצלחת.
- תתכן התנגשות רק עם הופעת הודעה נוספת במשך τ שניות מהופעת ההודעה הראשונה.
- \Leftarrow תקופה מוצלחת כאשר אין מופע במשך τ .



$$U = T \cdot e^{-g\tau} \quad P_{SUC} = P_0(\tau) = e^{-g\tau}$$

מכאן:

$$F_I(x) = P[\tilde{I} \leq x] = 1 - P[\tilde{I} > x] = 1 - P[x \text{ זמן מופע במשך } \tau] = 1 - e^{-gx} \quad \text{חישוב I:}$$

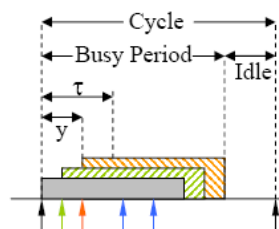
$$f_I(x) = g \cdot e^{-gx}$$

$$I = \int_0^{\infty} x f_I(x) dx = \frac{1}{g}$$

תקופת התעסוקה B

הגדרה: Y מועד הופעת ההודעה האחרונה במחזור. בתקופה מוצלחת $Y = 0$.

אורך תקופת התעסוקה $T + Y + \tau$ (זמן ניקוי הערוץ).



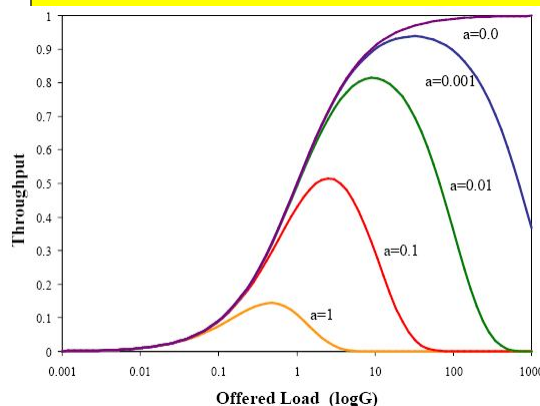
$$B = \bar{Y} + T + \tau$$

$$F_Y(y) = P[Y \leq y] = P[(y, \tau) \text{ זמן מופע במשך } \tau] = e^{-g(\tau-y)} \quad 0 < y < \tau$$

$$S = \frac{gT e^{-g\tau}}{g(T + 2\tau) + e^{-g\tau}} = \frac{G e^{-aG}}{G(1 + 2a) + e^{-aG}}$$

$$\bar{Y} = E[Y] = \tau - \frac{1 - e^{-g\tau}}{g} \quad \text{הצבה:}$$

$$a = \frac{\tau}{T} \quad G = gT \quad \text{כאשר}$$

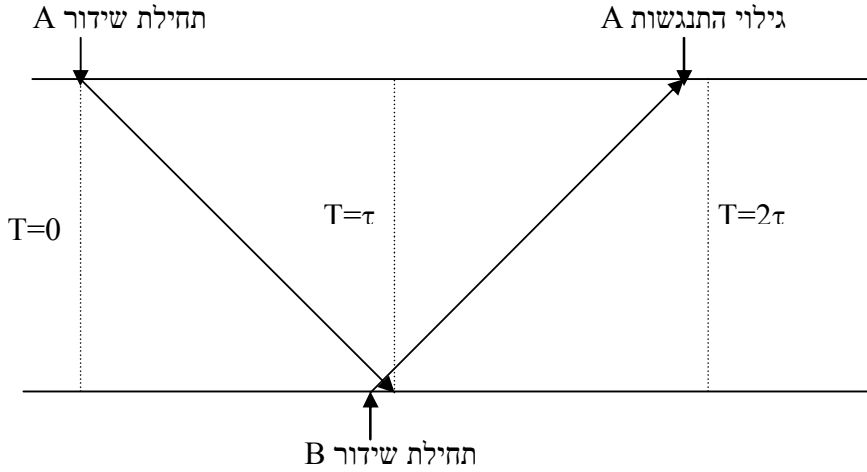


CD:MAC (זיהוי התנגשויות)

שיפור ביצועי CSMA ע"י קיצור המחזורים הבלתי מוצלחים. צומת מקשיב לערוץ בזמן שהוא משדר, קולט אם השידור שלו נקי, ואם לא - מפסיק לשדר. ניתן לבצע בקלות בטופולוגיות חוטיות. (מאד קשה לביצוע ברשת אלחוטית, כי השידור חוסם את הקליטה).

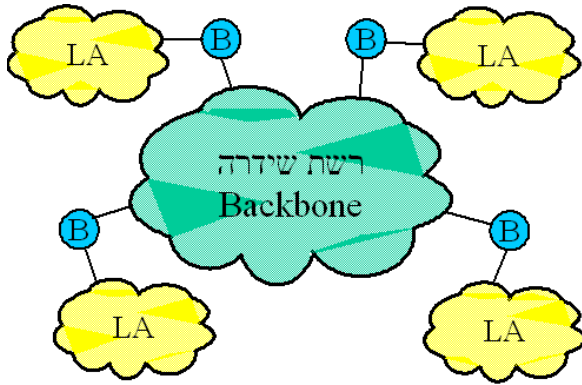
פרק הזמן המקסימלי לזיהוי התנגשות

פרק הזמן הוא 2τ (זו גם בעקיפין הסיבה להמתנה של 2τ ב-p-persistent CSMA)



דוגמא: 1-persistent CSMA/CD: Ethernet

חיבור רשתות: גשר



- נועד לחבר שתי רשתות מקומיות ברמת MAC.
- מבצע פעולה לוגית: מעביר הודעה מ - LAN1 אל LAN2.
- מיישם את אלגוריתם הגישה בכל רשת.
- גורם השהייה נוספת (עקב פעולה לוגית).
- בודק Checksum - לא כדאי להעביר הודעות פגומות

בעיות שגשר פותר:

- יותר מדי מחשבים ברשת אחת.
- מגבלת מרחק בין משתמשים או שדה כתובת.
- גודל גיאוגרפי: שתי רשתות קצרות עדיפות על פני רשת ארוכה.
- מגדיל את הקיבול.
- אמינות - רשת אחת נופלת השניה עובדת.
- בטחון \ סודיות - לא כל ההודעות עוברות בכל מקום (פריצה אינה חושפת הכל).

גשר שקוף

גשר שהכנסתו אינה דורשת שינוי תוכנה בצרכנים.

פעולות טיפוסיות של גשר שקוף בין רשתות A, B:

1. קולט הודעה על A.
2. בודק Checksum.
3. האם היעד נמצא על A ?
 כן: זרוק הודעה.
 לא: העבר הודעה לרשת B.

פעולות הכרחיות שנדרשות מכל גשר:

- לקלוט כל הודעה על הרשת < חומרה מהירה ויעילה.
- ידיעה אם הצמתים נמצאים ברשת A או ברשת B.
- תרגום מפרוטוקול A לפרוטוקול B (אם יש צורך).

בעיית ניתוב המסלול:

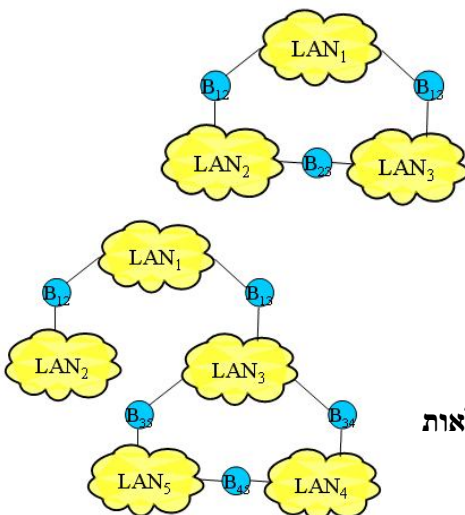
בדוגמא הזו אם LAN₁ שולח הודעה לLAN₂ היא תעבור דרך גשר B12 אבל גם במסלול LAN₁ → B13 → LAN₃ → B23 → LAN₂

הודעות כפולות ובזבוז משאבים!

עוד בעיה בשליחה: עבור שליחה מLAN₁ לLAN₂ יהיה LIVELOCK! אם נסתכל על רשת כזו ההודעה תמשיך להסתובב ברשתות 3,4,5 לנצח

SPANNING TREE

כדי שהכנסתו של הגשר תהיה שקופה יש לדאוג שהטופולוגיה תהיה חסרת לולאות (פותר בעיית Livelock אך לא מונע בזבוז משאבים)



גשר עם ניתוב מקור - Source Routing (הנחה: המקור יודע את הטופולוגיה! יכול לדעת את כל המסלול לכל יעד!)

בשדה הכותרת יש סיבית "מקומיות": 0 - היעד על אותה רשת. 1 - היעד על רשת אחרת.
בתוך המסגרת מתואר מסלול מלא מראש המקור ליעד:

- לכל רשת יש מספר זיהוי משלה (12 סיביות).
- לכל גשר יש מספר יחיד ברשת (4 סיביות).

תאור המסלול: $B_1 \rightarrow N_1 \rightarrow B_2 \rightarrow N_2 \rightarrow \dots \rightarrow N_k \rightarrow 0$

גשר מסתכל רק על מסגרות עם מספר סיביות מקומיות "1". כל גשר בוחר רק את המסגרות שלו (Bi).

מנגנון לאיתור מסלולים:

מקור שאינו יודע מסלול ליעד שולח מסגרת גישוש (Discovery) המציפה את הרשת.
כל גשר שדרכו עוברת מסגרת גישוש רושם את שמו במסגרת.

מסגרת גישוש המגיעה ליעד מוחזרת ישירות לפי המסלול ההפוך במסגרת הגישוש.

לא נוצרות לולאות כי הגשר בודק אם שמו כבר מופיע, ואם כן, זורק את מסגרת הגישוש.

הערה: הצפה גורמת למס' אקספוננציאלי של חבילות [לפי מס' מסלולים] (בגשר שקוף הצפה גורמת למס' ליניארי של הודעות).
גשרים משתמשים ב-TIMER בכדי להתגבר על התיישנות מידע ושנויי טופולוגיה.

אלגוריתם בניית העץ הפורש:

תיאור כללי

כל תחנה שולחת HELLO לתחנות הבנות שלה בכל פרק זמן. ההודעות מכילות את זהות השורש הידועה לה ומרחקה ממנו.
במידה ויש צורך בעדכון, התחנות המקבלות את ההודעה מעדכנות את נתונין.
אם תחנה מצפה לקבל HELLO מרשת האב ולא מקבלת אותו (פוקע T_{out}) היא מחפשת אב חדש.

מבנה החבילות

כל חבילה בפרוטוקול נקראת BPDU (Bridge Protocol Data Unit) או HELLO ומכילה:

- SID - זיהוי הגשר השולח
- LID - זיהוי המנהיג
- DIS - המרחק של השולח מהמנהיג

הנחה: לכל גשר ולכל port מספר זיהוי ייחודי.

איתחול לפני שליחת הודעה ראשונה, תקשיב התחנה לפרק זמן של 1 שניה כדי לבדוק האם קיים מנהיג ברשת. אחרת:

```
leaderID = stationID
DFL = 0 (Distance from Leader)
lanList = {All LANs attached}
parentLAN = ∅
timer[All LANs] = 1 sec
```

עם פקיעת timer[i]

if (i == parentLAN) initialize	אבדה הדרך לשורש
else if (i ∉ lanList) lanList = lanList ∪ i	קבל אחריות על רשת יתומה
if (i ∈ lanList) send HELLO	שמור על אחריות של רשת בת

עם קבלת HELLO על רשת האב – lan[i]

```

if (leaderID > LID) or (
    (leaderID==LID)&&(DFL≠Dis+1) )
    leaderID = LID
    lanList = {all connected} \ {i}
    DFL = Dis + 1
    timer[i] = Fall down time
else if ( (leaderID == LID) &&
    (DFL == Dis + 1) )
    timer[i] = Fall down time
Else
    Initialize

```

עדכן מסלול חדש לאב.
על העדכון להגיע לכל הרשתות
המחוברות.

הכל בסדר, אפס טיימר.

זו לא רשת האב. הפש אב חדש.

עם קבלת HELLO על רשת lan[i] שאינה רשת האב

```

if (leaderID > LID) ||
    (leaderID==LID && DFL>Dis+1) ||
    (leaderID==LID && DFL==Dis+1 &&
    && parentID > SID)
    leaderID = LID
    lanList = {all connected} \ {i}
    parentLAN = i
    DFL = Dis + 1
    timer[i] = Fall down time
    timer[≠i] = 1
else if (leaderID == LID) &&
    ( (DFL > Dis) ||
    (DFL == Dis && Station_ID > SID) )
    LAN_List = LAN_List - i
    Timer [LANi] = Fall down time
Else
    LAN_List = LAN_List + i
    timer [LANi] = 1 sec

```

נמצא אב חדש!

designated נמצא bridge חדש.

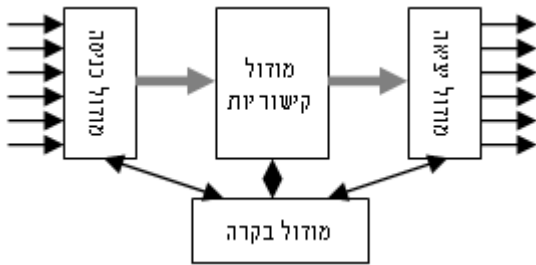
זו רשת בת.

התוצאה

גשר יעביר אך ורק הודעות מרשת השייכת ל-LAN_list או לאב אל רשת השייכת ל-LAN_list או לאב, כל עוד המקור והיעד אינם באותה רשת.

מתגים – SWITCHES

פונקציות המתג:



- מיתוג: העברת הודעה מהדק כניסה להדק יציאה
- ניתוב: בחירת הדק היציאה
- התאמת קצבים בין כניסה ליציאה
- תזמון

- בצירור: מבנה כללי של מתג. חבילות מגיעות אל המתג מכל אחת מהכניסות. המתג שולח כל חבילה ליציאה הנכונה על פי ה-header של אותה חבילה.
- חבילה שמגיעה נכנסת למודול הכניסה.
- ברגע שהמתג מוכן לטפל בחבילה היא מועברת למודול הקישוריות ששולח אותה ליציאה הנכונה במודול היציאה.
- מודול הבקרה מתזמן את שלושת הרכיבים.

מטריצת התעבורה

$$\Lambda = \begin{pmatrix} \lambda_{11} & \dots & \dots & \dots & \lambda_{1N} \\ \vdots & \ddots & & \ddots & \vdots \\ \vdots & & \lambda_{ij} & & \vdots \\ \vdots & \ddots & & \ddots & \vdots \\ \lambda_{N1} & \dots & \dots & \dots & \lambda_{NN} \end{pmatrix}$$

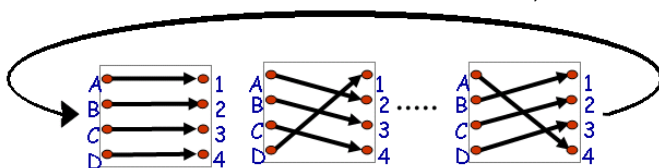
קצב הגעת חבילות מכניסה i ליציאה j מתפלג פואסונית עם מקדם λ_{ij} .

מטריצת תעבורה $\Lambda = [\lambda_{ij}]$ היא קבילה אם"ם:

1. לכל i מתקיים $\sum_{j=1}^N \lambda_{ij} < 1$, כלומר אין כניסה שמועמסת יותר מדי.
2. לכל j מתקיים $\sum_{i=1}^N \lambda_{ij} < 1$, כלומר אין יציאה שמועמסת יותר מדי.

תעבורה אחידה $\forall i, j: \lambda_{ij} = \lambda$. כדי שמטריצת תעבורה תהיה קבילה בתעבורה אחידה: $\lambda < 1/N$.

אלגוריתמי תזמון



Uniform Cyclic Scheduling .1

כל זוג כניסה/יציאה מחובר כל N חריצי זמן.

Uniform Random Scheduling .2

בכל חריץ זמן הגרל פרמוטציה כלשהי מבין $N!$ האפשרויות. ההסתברות לחיבור כניסה i ליציאה j בחריץ כלשהו היא $1/N$.

Birkhoff-von Neumann בעיה בעזרת אלגוריתם

איך ניתן לתזמן את המטריצה הבאה?

$$\begin{pmatrix} \frac{1}{2}-\varepsilon & \frac{1}{4}-\varepsilon & \frac{1}{4}-\varepsilon \\ \frac{1}{4}-\varepsilon & \frac{3}{8}-\varepsilon & \frac{3}{8}-\varepsilon \\ \frac{1}{4}-\varepsilon & \frac{3}{8}-\varepsilon & \frac{3}{8}-\varepsilon \end{pmatrix}$$

ראשית, נמצא מטריצה dually stochastic (בה סכום כל שורה וכל עמודה הוא 1) אשר מכסה את המטריצה שלנו, ונכפול במחלק הגדול ביותר, כלומר 8:

$$\tilde{\Lambda}^1 = \begin{pmatrix} 4 & 2 & 2 \\ 2 & 3 & 3 \\ 2 & 3 & 3 \end{pmatrix}$$

כעת בכל צעד נמצא פרמוטציית ניתוב P_k , אותה נכפול בתעבורה הקטנה ביותר שהיא מעבירה α_k , ואת זה נחסר ממטריצת התעבורה הנוכחית: $\tilde{\Lambda}^{k+1} = \tilde{\Lambda}^k - \alpha_k P_k$.

$$\tilde{\Lambda}^1 = \begin{pmatrix} 4 & 2 & 2 \\ 2 & 3 & 3 \\ 2 & 3 & 3 \end{pmatrix} \quad P_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \alpha_1 = 3$$

$$\tilde{\Lambda}^2 = \begin{pmatrix} 1 & 2 & 2 \\ 2 & 0 & 3 \\ 2 & 3 & 0 \end{pmatrix} \quad P_2 = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \quad \alpha_2 = 2$$

$$\tilde{\Lambda}^3 = \begin{pmatrix} 1 & 2 & 0 \\ 0 & 0 & 3 \\ 2 & 1 & 0 \end{pmatrix} \quad P_3 = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} \quad \alpha_3 = 2$$

$$\tilde{\Lambda}^4 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \quad P_4 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \quad \alpha_4 = 1$$

כעת נחלק את ה- α בסכומם ונקבל את הסתברויות הפרמוטציות:

$$P_1 \text{ w.p. } 3/8$$

$$P_2 \text{ w.p. } 1/4$$

$$P_3 \text{ w.p. } 1/4$$

$$P_4 \text{ w.p. } 1/8$$

קיבלנו כי כל תא במטריצת התעבורה מקבל ניתוב על פי הדרוש.