

IP Addressing and Subnetting

עקרונות כתובות IP והחלוקה לתתי רשתות

IP Addressing and Subneting

כתב: אורגד לוצקי

Cisco Certified Academy Instructor

Microsoft Trainer



© Orgad Lootski

Kibbutz Ginegar 30053 ISRAEL

החברת שלפניכם עוסקת ביישום כתובות IP להתקני רשת, חלוקה לתתי רשתות והיררכית כתובות.

אני מנסה להסביר את רוב השלבים ואת הסיבות לביצוע הפעולות כפי שהן, אבל מראש אני מזהיר שמאמר זה אינו מיועד לתלמידים שאין להם שום מושג בחומר. כתבתי את המאמר כתזכורת בלבד, והוא לא מתיימר ללמד את כל החומר מהתחלה. מאמר זה יכול לעזור למי ששכח או לא בדיוק הבין את פעולת ה-Subnetting, ולמי שמעוניין לקבל דגשים על נקודות ספציפיות. אם את/ה לא יודע על מה אני מדבר, ואת/ה מנסה לעשות קיצורי דרך, זה לא המקום בשבילך. נחזור בקצרה על המשמעות של כתובת IP, קצת על האוקטות השונות וכיצד הן כתובות, מספרים בינאריים והמרות למספרים עשרוניים, הסבר על ה-Subnet Mask ועל פעולת ה-Subnetting.

ההתחלה מכילה הרבה פרטים טכניים שעשויים להיות לרובכם מוכרים. המקצוענים שביניכם ירצו אולי לעבור ישר לחלק העוסק ב-Subnetting, אבל אם תמצאו שם מושגים חדשים או לא מוכרים, יתכן שתמצאו בכל זאת לחזור קצת על ההקדמה ולרענן כמה ממושגי היסוד של הפרק.

חשוב לציין שהסבר זה הוא אמנם ארוך, אבל לא בגלל שהחומר מסובך במיוחד, אלא משום שאני נותן דוגמאות על גבי דוגמאות לכל אחת מהפעולות המתוארות. מי מכם שחש בקיאות מספקת בחומר יכול לדלג הלאה ולא לקרוא את כל הדוגמאות המובאות כאן.

החלק השלישי של המאמר מכיל את ביצוע והסבר ה-Subnetting בפועל, כך שמי שחש ידענות מופלגת יכול לדלג לחלק השלישי.

חלק ראשון

בחלק הראשון נעסוק ב- 3 המרכיבים החשובים של ה- IP addressing והם:

- IP Address
- Subnet Mask
- Default Gateway

לקראת סוף החלק נזכיר כיצד ניגשים להגדרות הפרוטוקול (אבל ממש בקצרה)

IP Address

כל מכשיר המשתמש ב- TCP/IP צריך כתובת IP. זה יכול להיות מחשב (כתובת אחת עבור כל כרטיס רשת), נתב או Router (כתובת אחת עבור כל "רגל" או Port של הנתב), מדפסות רשת (Network Interface Printer Device), או כל מכשיר (Device) אחר שצריך כתובת כזו.

כתובת IP בנוייה מ- 32 ביטים (או 4 בייטים) היכולים להיכתב בצורה עשרונית או בצורה בינארית. הכתובות הללו משמשות לזיהות את המחשב בתוך רשת המבוססת על פרוטוקול ה- TCP/IP.

בצורה עשרונית: 132.37.68.92

אותה כתובת בצורה בינארית: 10000100.00100101.01000100.01011100

כתובת IP מחולקת ל- 4 חלקים פיזיים שונים שכל אחד מהם נקרא אוקטטה – על שם העובדה שהוא מורכב מ- 8 ביטים (או בייט אחד). כל ביט כזה יכול לקבל את הערך 0 או את הערך 1. דוגמא של אוקטטה כזו יכול להיות כאמור 00110100.

מכיוון שמדובר באוקטטה, כלומר צירוף של 8 ביטים שונים, הרי שטווח של כל אוקטטה כזו יכול לנוע מ- 0 ועד 255, או בעצם 256 אפשרויות. מדוע? כיוון שכל ביט יכול לקבל 2 ערכים שונים (או 0 או 1), ויש לנו 8 ביטים שונים, לכן 2 בחזקת 8 שווה בדיוק 256.

בבינארי הערכים הללו משמשים כקצוות הטווח. כלומר, מצד הנמוך יש לנו 00000000 או 0 בעשרוני, ואז מתחילים לגדול לפי סדר הקומבינציות (00000001, 00000010, 00000100 וכו') עד שמגיעים לקצה השני, שזה 11111111 או 255 בעשרוני.

זהו המקום להזכיר כמה חוקי יסוד חשובים (לא יודעים למה? חזרו לספרי הלימוד בבקשה):

- אסור Network ID שמתחיל ב- 127.
- אסור Host ID שמסתיים עם המספר 255 (בבינארי). (11111111)
- אסור Host ID שמסתיים עם המספר 0 (00000000) בבינארי).

כלומר, כתובת כמו 127.56.9.12 – האם היא חוקית או לא? כמובן שלא.
וכתובת כמו 134.12.0.0 – האם אפשר לתת כזו כתובת למחשב? כמובן שלא.
וכתובת כמו 189.6.7.255 – האם אפשר לתת כזו כתובת למחשב? כמובן שלא.
וכתובת כמו 176.255.1.23 – האם אפשר לתת כזו כתובת למחשב? כן.
וכתובת כמו 14.255.255.12 – האם היא כתובת חוקית למחשב? כן.

כתובת ה- IP חייבת להיות ייחודית בתוך אותה רשת. כלומר, אסור 2 מחשבים עם אותה כתובת בתוך אותה רשת. אם תרצו, תוכלו לבנות 2 רשתות נפרדות אחת מהשנייה ולתת למחשבים באחת מהן כתובות זהות למחשבים ברשת השנייה. אבל ברגע ששתי הרשתות יחברו יחד, המחשבים בעלי הכתובות הזרות לא יוכלו לתקשר.

כתובת IP מחולקת ל- 2 חלקים לוגיים: כתובת רשת וכתובת מחשב, או באינגליזית Network ID ו- Host ID.

- ה- Host ID משמש לזיהוי המחשב הבודד בתוך הרשת, או לפי דוגמה מציאותית יותר, לזיהוי הבית ברחוב ע"י מתן מספר בית, או אזור החיוג ברשת הטלפון הציבורית.
- ה- Network ID מזהה את כתובת הרשת, או את מספר הרחוב, או למשל את מספר הטלפון הייחודי בתוך אותו אזור חיוג.

אם לשני מחשבים (Hosts) יש את אותה כתובת רשת (Network) אז הם נמצאים על אותה רשת, או בעצם הם נמצאים באותו רחוב. אם יש להם כתובות רשת שונות, אז מן הסתם הם נמצאים על רשתות שונות, או ברחובות שונים.

לשני מחשבים (Hosts) באותה רשת (Network) אסור שיהיה אותו מספר בית. למשל, אי אפשר לבנות 2 בתים בעלי המספר 100 ברחוב הרצל. אפשר בית מספר 100 ברחוב הרצל, ואפשר לחזור על המספר 100 גם ברחובות אחרים, העיקר שלא באותו רחוב – או בעצם באותה רשת. אסור שבאזור חיוג 03 יהיו 2 מנויי טלפון עם אותו מספר טלפון. אפשר להעניק מספר כמו 6165678 באזור חיוג 03, ואת אותו מספר גם באזור חיוג 04, אבל בטח שלא בתוך אותו אזור חיוג.

בדוגמא שנתתי בתחילת הפרק, חלק אחד מכתובת ה-IP הוא כתובת הרשת, והחלק השני הוא כתובת המחשב. אבל זו רק דוגמא אחת, והיא לא מחייבת, כמו שנראה בהמשך.

132	37	68	92
Network ID		Host ID	

הגדרה לא נכונה של כתובת IP עלולה לפגוע בתקשורת המחשב. כפילות הרי אסורה, ולכן אם נגדיר כתובת IP שכבר נמצאת בשימוש, אז שני המחשבים עלולים לאבד את יכולת התקשורת בפרוטוקול ה-TCP/IP.

Subnet Mask

ה-Subnet Mask הוא כתובת נוספת המלווה כל מחשב (חוץ מכתובת ה-IP) ומציין למחשב איזה חלק מתוך כתובת ה-IP הוא ה-Host ID, ואיזה חלק הוא ה-Host ID.

למשל, לפי הדוגמא הקודמת:

132	37	68	92
255	255	0	0

ה-Subnet Mask מופיע בדרך-כלל בצורה קבועה, אבל בהמשך נראה איך אפשר לשנות אותו.

שני הערכים הללו, ה-IP Address וה-Subnet Mask הם ערכים שחובה עלינו להגדיר אותם עבור כל מחשב ומחשב ברשת ה-TCP/IP. אם לא נעשה זאת, אותו מחשב לא יוכל לתקשר בפרוטוקול זה. הגדרה לא נכונה של Subnet Mask עלולה לגרום למחשב להתבלבל ולחשוב שהוא נמצא על רשת אחרת מזו שהוא נמצא בה באמת, או, לחלופין, לחשוב שכתובת מסויימת לא נמצאת על הרשת שלו אלא על רשת אחרת. עוד על ה-Subnet Mask בהמשך.

Default Gateway

איך יוכלו שני מחשבים לתקשר ביניהם אם הם נמצאים על רשתות שונות? באמצעות מתווך בין 2 הרשתות, מכשיר כמו נתב (Router) או בשמו האחר Default Gateway.

ה- Default Gateway היא בעצם כתובת IP של אותו מחשב או נתב שמשמש כיציאה החוצה מן הרשת המקומית אל רשת או רשתות אחרות. זוהי כתובת ה- IP של הנתב המשמש כיציאה המומלצת והרגילה של התקשורת היוצאת מהרשת שלנו אל רשתות מרוחקות אחרות. אם יש לנו כמה יציאות כאלה, אז ה- Default Gateway יהיה הכתובת של היציאה הקרובה ביותר אל המחשב שלנו, או הכתובת של היציאה האופטימלית מהרשת שלנו (אופטימלית מתייחס למהירות הנתב, קירבה פיזית, זמינות וכו').

אם לא נגדיר אותו, הרי שמחשב לא ידע מאיפה עליו להוציא את התשדורות שמיועדות לרשתות שהן שונות מהרשת שלו.

אם לא נגדיר אותו, אז מחשב יהיה מוגבל בתקשורת אך ורק עם המחשבים האחרים ברשת הפנימית שהוא נמצא עליה, וכל פניה למחשב שנמצא על רשת אחרת מזו שלו, תיענה בהודעה הבאה: "Destination host unreachable".

האמת היא (אבל אל תגלו לאף אחד) שאפשר לתקשר עם רשתות אחרות גם בלי ה- Default Gateway, אבל אז כל מחשב יצטרך לדעת בדיוק מאיפה הוא אמור לצאת החוצה מהרשת המקומית שלו. זה יכול להתבצע על-ידי קונפיגורציה של טבלת הראוטינג (Routing Table) על כל מחשב. אבל נכון לעכשיו, תזכרו שמחשב ללא Default Gateway לא יכול לתקשר עם מחשב שנמצא ברשת אחרת.

הגדרות ה- TCP/IP

כאמור כל המחשבים או המכשירים ברשת חייבים לקבל את הפרמטרים הללו. יש 2 דרכים לגדיר אותם: ידנית ואוטומטית. לא ניכנס ליתרונות וחסרונות של כל אחת מהשיטות.

הגדרה ידנית מתכוונת לכך שהאדמיניסטרטור הולך לכל אחד מהמחשבים או הנתבים וידנית מכניס את הגדרות ה- TCP/IP. כך הוא יודע שלכל מחשב או נתב תהיה כתובת IP קבועה מראש, שלא תשתנה כל שעה.

הגדרה אוטומטית מתכוונת לשימוש בשרתי DHCP שיחלק את הגדרות ה- TCP/IP באופן אוטומטי למחשבים השונים.

עם השימוש ב-DHCP יכול להיווצר מצב שבו המחשבים יקבלו כל יום כתובת IP אחרת, ולא את אותה כתובת שקיבלו בפעם הקודמת. זה בסדר גמור במקרה שמדובר במחשבים המשמשים כ-Clients. אבל כשמדובר במחשבים חשובים (מחשבים הממלאים תפקיד מרכזי ברשת, כמו WINS, DNS וכו') או במכשירים חשובים (כמו נתבים) זה מצב שאנחנו רוצים למנוע. לכן בד"כ עבור מחשבים פחות חשובים נשתמש בהגדרות אוטומטיות, אבל עבור המחשבים החשובים והנתבים (ואולי גם מדפסות וכו') נגדיר את הכתובות באופן ידני.

איך בודקים את הגדרות ה-TCP/IP של המחשב שלנו? תחת Windows NT מקישים Ipconfig (לדעת את הפקודות הללו טוב טוב!!!) בשורת הפקודה ומקבלים משהו כזה:

Windows NT IP Configuration

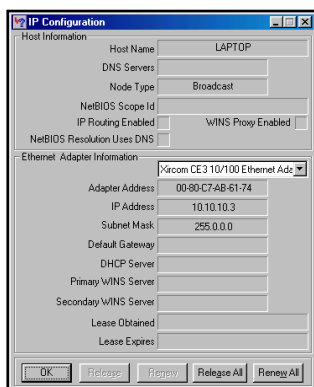
Ethernet adapter FE100AM1:

IP Address : 131.107.2.200

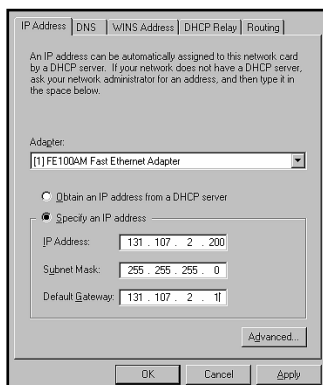
Subnet Mask : 255.255.255.0

Default Gateway : 131.107.2.1

תחת Windows 95/98 יש תוכנה שנקראת Winipcfg שמראה את אותם נתונים אבל בצורה גראפית:

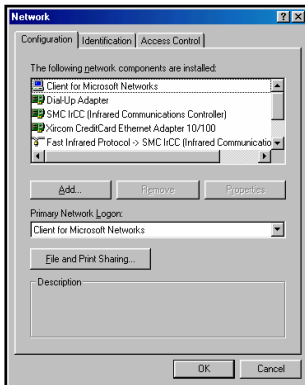


איך משנים את הגדרות ה-TCP/IP של המחשב שלנו? לחצן ימני על Network Neighborhood, מאפיינים, ואז – Protocols ומאפיינים של TCP/IP. מקבלים מסך כזה:

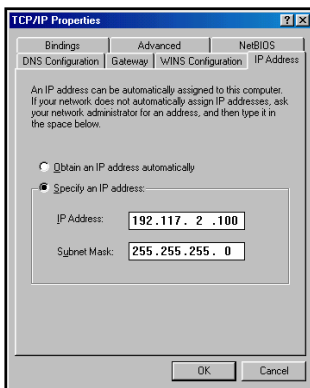


מכאן אפשר לשנות את כל הגדרות הפרוטוקול וההגדרות שונות, כמו כתובות DNS, WINS וכו'.

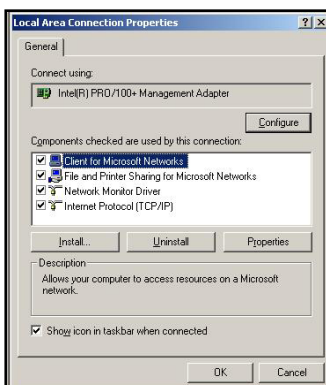
ב- Windows 95/98 ניגשים למסך ההגדרות באותו אופן, רק ששם הוא נראה קצת שונה. כמובן שאפשר לגשת לאותו מסך גם דרך אייקון ה- Network בלוח הבקרה:



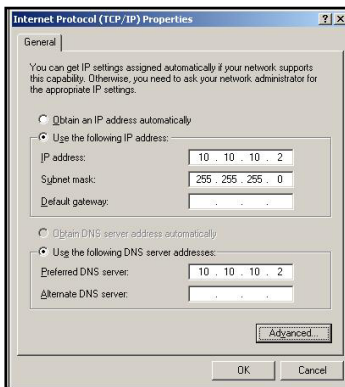
וכאשר נכנסים להגדרות הפרוטוקול מקבלים משהו כזה:



תחת Windows 2000 ניגשים להגדרות הפרוטוקולים וכרטיסי הרשת ע"י לחצן ימני על 'My Network Places', מאפיינים, ואז לחצן ימני על 'Local Area Connection' ומאפיינים. מקבלים משהו כזה:



ואז אם ניגשים למאפיינים של TCP/IP מקבלים חלון הגדרות כזה:



אגב, תוכנת ה- Ipconfig תחת Windows 2000 השתפרה בתחומים רבים, וכוללת בתוכה פקודות חדשות רבות. כדאי להתעניין בנושא.

זה אולי המקום לבקש ממכם לבדוק, לנסות, לזכור (והכי חשוב – להבין) את הפקודות הבאות:
.FTP, TFTP, TELNET, RCP, REXEC, RSH, LPD, LPR, LPQ

אבל הרבה הרבה יותר חשובות מבחינתכם הן הפקודות הבאות (לא רק הפקודה והפרמטרים שלה, אלא מה בדיוק היא מבצעת ומה ההבדל בינה לבין פקודה אחרת – חשוב ביותר! לא לוותר!!):
.PING, IPCONFIG, HOSTNAME, ROUTE, NETSTAT, NBTSTAT, ARP, TRACERT, FINGER

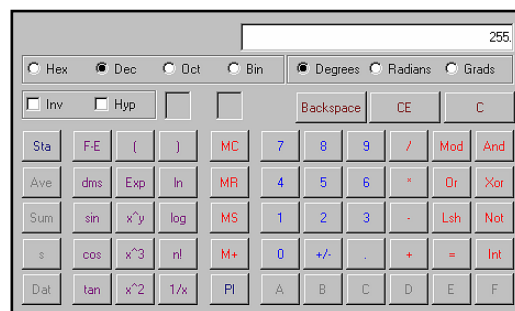
כאמור אנחנו כאמור לא נתעסק בהגדרות במאמר זה.

חלק שני

בחלק השני נדבר על המרה ממספרים בינאריים לעשרוניים ומעשרוני לבינארי, חלוקה ל-Classes השונים, אבחנה בין ה-Classes השונים ונדבר על כמות ה- Network ID וה- Host ID בכל רשת, ועל המשמעות של כתובות ציבוריות וכתובות פרטיות.

המרה מעשרוני לבינארי ולהיפך

נזכיר גם את העניין של ההמרה מעשרוני לבינארי ולהיפך. הדרך הקלה ביותר היא ע"י שימוש במחשבון של Windows או באמצעות קיצור הדרך או ע"י הקשת calc בפקודת ה- Run. בתוך המחשבון צריך לעבור לתצוגה מדעית בתפריט View, ואז אם נקיש מספר בעשרוני, נוכל ללכת לכפתור ה- Bin והפלא ופלא – המספר הפך לבינארי.



כנ"ל, אם נהיה במצב Bin ונקיש מספר בבינארי, נוכל לעבור למצב Dec וראו איזה פלא – המספר הפך לעשרוני.

לידיעתכם, במבחני מיקרוסופט המחשבון קיים, לעומת זאת בכל המבחנים של Cisco הוא אינו מופיע וחל איסור להשתמש בו (ו ב צ ד ק !!!) לפי דעתי (וניסיוני) התלמיד חייב להבין את הפעולה שהוא עושה. אם נשתמש במחשבון אז אמנם הפעולה תתבצע מהר, אבל התלמיד לא תמיד שם לב למה שהוא מקיש, ויתכן שבגלל טעות הבקלדה (כמו במילה הקודמת למשל...) המחשבון אמנם יתן תוצאה, אבל לא את התוצאה הנכונה.

עשו לכם מנהג של קבע, הפכו את העשרוני לבינארי ולהיפך באמצעות המוח ולא המחשבון. ככה גם תבינו את מה שאתם עושים, ולאחר תרגול מספיק תוכלו גם להבחין במגמות מסויימות בחשבונות שאתם עורכים.

מבינארי לעשרוני

אז איך הופכים מבינארי לעשרוני? פשוט מאוד. מציירים טבלה קטנה כמו זאת:

בטבלה יש 8 עמודות וכמה שורות שאתם רוצים. כל עמודה מציינת ביט אחד באוקטטה. בשורה העליונה כותבים את המספרים הבאים:

$2^7=128$	$2^6=64$	$2^5=32$	$2^4=16$	$2^3=8$	$2^2=4$	$2^1=2$	$2^0=1$

ומי שמסתכל טוב יכול לשים לב שאם מתבוננים מימין לשמאל, אז מיד רואים שעמודה אחת היא בעצם הכפולה של זו שלפניה, או בעצם לפי הנוסחה 2 בחזקת 0 , 2 בחזקת 1 , 2 בחזקת 2 , 2 בחזקת 3 וכו' עד 2 בחזקת 7 שזה שווה ל- 128 . אנחנו יכולים תיאורטית להמשיך עד שתכאב לנו היד, אבל במציאות אנחנו צריכים רק 8 ביטים (זוכרים? אוקטטה?), ולכן סופרים מ- 2 בחזקת 0 ועד 2 בחזקת 7 , סה"כ 8 ביטים.



יופי. עכשיו קיבלנו מספר בינארי מפוצץ כמו 00000101 . לפני שנבהלים נכתוב אותו לפי הסדר בשורה השניה בטבלה שלנו:

128	64	32	16	8	4	2	1
0	0	0	0	0	1	0	1

כדי להפוך לעשרוני פשוט נסתכל איפה כתובה הספרה 1 ואיפה כתובה הספרה 0 . איפה שכתוב 1 , נרשום את הערך שמעליה, ואיפה שכתוב 0 נתעלם מהערך שמעליה. בדוגמא שלנו:

128	64	32	16	8	4	2	1
0	0	0	0	0	1	0	1

מה כתבנו? 1 ו- 4 . מה הסכום של $1+4$? בדיוק 5 . לכן, התוצאה של 00000101 בעשרוני זה 5 .

דוגמא נוספת: 00110111 .

128	64	32	16	8	4	2	1
0	0	1	1	0	1	1	1

בחשבון מהיר: $1+2+4+16+32=55$ ולכן התוצאה היא 55 .

דוגמא נוספת: 11101110 .

128	64	32	16	8	4	2	1
1	1	1	0	1	1	1	0

בחשבון מהיר: $2+4+8+32+64+128=238$ ולכן התוצאה היא 238 .

בסדר? עכשיו כמה תרגילים. אתם לא חייבים לבצע אותם, אבל אני מאד ממליץ לכם. זה יעזור לכם בהמשך הפרק.



הפוך מבינארי לעשרוני את הערכים הבאים (בדוק את עצמך באמצעות מחשבון):

00001101

00011101

00001111

10110011

11110001

01101100

11111100

11000000

10110001

11111000

11000011

11111110

11110111

11111111

01010101

מעשרוני לבינארי

נעשה את אותו תרגיל עם הטבלה, אבל הפעם צריך להכניס ערכים עשרוניים ולקבל ערכים בינאריים.

נהפוך את המספר 131 לבינארי:

128	64	32	16	8	4	2	1

נבדוק: בעמודה הראשונה מצד שמאל, האם 128 נכנס ל-131?

כן.

כמה נשאר עודף?

3.

לכן תחת 128 כותבים 1 וזוכרים שנשארו לנו עוד 3 להכניס לטבלה.

128	64	32	16	8	4	2	1
1							

נמשיך ימינה. האם 64 נכנס ל-3?

לא.

האם 32 נכנס ל-3?

לא.

האם 16 נכנס ל-3?

לא.

האם 8 נכנס ל-3?

לא.

האם 4 נכנס ל-3?

לא.

בכל העמודות האלה נרשום 0, כיוון שהערכים שלהם לא נכנסים ל-3.

128	64	32	16	8	4	2	1
1	0	0	0	0	0		

נמשיך ימינה. האם 2 נכנס ל-3?

כן.

נכתוב 1 תחת העמודה.

כמה נשאר?

1.

האם 1 נכנס ב-1?

כן.

נכתוב גם תחת העמודה אחרונה 1.

128	64	32	16	8	4	2	1
1	0	0	0	0	0	1	1

מה קיבלנו? המספר 10000011, שזה בעשרוני בדיוק 131.



הפעם מקוצרת. הפוך 243 לבינארי.

128	64	32	16	8	4	2	1
1	1	1	1	0	0	1	1

משמאל לימין בטבלה: 128 נכנס ל- 243 ונשאר עודף 115. 64 נכנס ל- 115 ונשאר עודף 51. 32 נכנס ל- 51 ונשאר עודף 19. 16 נכנס ל- 19 ונשאר עודף 3. 8 לא נכנס ל- 3. 4 לא נכנס ל- 2. 3 נכנס ל- 3 ונשאר עודף 1. 1 נכנס ל- 1. לכן התוצאה היא 11110011.



הפוך מסריוני לבינארי את הסרכים הבאים (בצוק את עצמך באמצעות

מחשבוני):

- 20
- 65
- 92
- 192
- 193
- 220
- 224
- 239
- 240
- 247
- 248
- 250
- 252
- 254
- 255

חלוקה לפי Classes

כפי שאנחנו כבר יודעים יש חלוקה היררכית של כתובות האינטרנט החוקיות ל- 5 מחלקות או קבוצות: W.X.Y.Z מתייחס לארבעת האוקטטות של כתובת ה- IP. הקטגוריות נקראות Classes והן מציינות איזה חלק מתוך 4 האוקטטות משמש כ- Network ID, ואיזה חלק משמש כ- Host ID.

ניקח כתובת IP כמו W.X.Y.Z:

	W	X	Y	Z
Class A	Network	Host	Host	Host
Class B	Network	Network	Host	Host
Class C	Network	Network	Network	Host
Class D	-	-	-	-
Class E	-	-	-	-

אנו נתייחס רק ל- 3 החלוקות הראשונות, כיוון ש- Class D משמשת לביצוע Multicasting ו- Class E משמשת לכתובות ניסיוניות. לא יודעים למה? לחזור מהר לספר.

לשם תזכורת נחזור ונציין את טווחי הכתובות ומספרי המחשבים בכל אחד מה- Classes הנ"ל. נזכיר גם שמחייבים אותנו לקבוע את ההתחלה של האוקטטה הראשונה לפי סוג ה- Class בה היא נמצאת, למשל כתובת IP מ- Class A חייבת שהביט הראשון באוקטטה הראשונה (W) יהיה 0. לא יודעים? מהר לספרים!

Class	Leading Bit Pattern	Decimal Range of First Byte of Network Address	Maximum Networks	Maximum Nodes (Hosts) per Network
A	0	1-127	126	16777214
B	10	128-191	16384	65534
C	110	192-223	2097152	254



צהה אה ה- Class של הכתובות הבאות:

210.23.67.102

66.23.148.0

158.23.251.33

144.23.117.254

192.254.23.123

63.125.23.211

192.25.128.36

128.12.254.98

134.223.156.89

127.0.0.1

224.23.108.23

223.78.27.144

377.123.28.167

191.249.222.234

19.23.12.255

188.67.67.235

134.255.123.22

257.22.45.219

117.117.117.117

אבחנה בין ה- Classes השונים:

איך יודעים אם 2 מחשבים נמצאים על אותה רשת או לא, ע"י מבט בכתובות ה- IP שלהם?

במקרה של Class A נקח לדוגמא כתובת כזו: 17.0.0.0 – זו כתובת הרשת או ה- Network ID. כלומר, אם אתם זוכרים את הדוגמא של שם הרחוב ומספר הבית, אז זהו שם הרחוב. לשם של רחוב אין תוספות אחריו. כלומר אי אפשר לקרוא לרחוב בשם "הרצל 134". זה "הרצל" וזהו, בלי תוספות אחריו. איזו כתובת מחשב או Host ID יכולה להיות על הרשת הזו? כל כתובת שמתחילה בטווח שבין 17.0.0.1 ועד 17.255.255.254, כלומר סה"כ בערך כמעט 17 מיליון כתובות אפשריות לאותה רשת. מחשב עם הכתובת 17.132.12.56 נמצא על אותה רשת כמו המחשב 17.56.83.97. למה? כי כתובת הרשת של שניהם זהה. כלומר בית מספר ככה וככה נמצא ברחוב הרצל. בית אחר שמספרו ככה וככה נמצא גם הוא ברחוב הרצל, ולכן שני הבתים נמצאים באותו רחוב.

במקרה של Class B נקח לדוגמא כתובת כזו: 131.117.0.0 – זהו ה- Network ID. איזו כתובת מחשב או Host ID יכולה להיות על הרשת הזו? כל כתובת שמתחילה בטווח שבין 131.117.0.1 ועד 131.117.255.254, כלומר קצת יותר מ- 16 אלף מחשבים לאותה רשת. מחשב עם הכתובת 131.117.4.9 נמצא על אותה רשת כמו המחשב 131.117.35.72. למה? כי כתובת הרשת של שניהם זהה.

במקרה של Class C נקח לדוגמא כתובת כזו: 192.117.1.0 – זהו ה- Network ID. איזו כתובת מחשב או Host ID יכולה להיות על הרשת הזו? כל כתובת שמתחילה בטווח שבין 192.117.1.1 ועד 192.117.1.254, כלומר 254 מחשבים לאותה רשת. מחשב עם הכתובת 192.117.1.69 נמצא על אותה רשת כמו המחשב 192.117.1.17. למה? כי כתובת הרשת של שניהם זהה.



1. Circle the Network ID of this address: 1.102.45.177
2. Circle the Host ID of this address: 196.22.177.13
3. Circle the Network ID of this address: 133.156.55.102
4. Circle the Host ID of this address: 221.252.77.10
5. Circle the Network ID of this address: 123.12.45.77
6. Circle the Host ID of this address: 126.252.77.103
7. Circle the Network ID of this address: 13.1.255.102
8. Circle the Host ID of this address: 171.242.177.109
9. Circle the Network ID of this address: 193.156.155.192
10. Circle the Host ID of this address: 1.52.177.188
11. Circle the Network ID of this address: 7.77.45.77
12. Circle the Host ID of this address: 191.252.77.13
13. Circle the Network ID of this address: 191.15.155.2
14. Circle the Host ID of this address: 221.252.117.254
15. Circle the Network ID of this address: 203.10.233.1
16. Circle the Host ID of this address: 191.2.227.19
17. Circle the Network ID of this address: 23.156.1.92
18. Circle the Host ID of this address: 121.2.199.88



ציהק אק הקסווק בקוק מהכקוקבוק הקבא. אא בבל הכקוקבוק'ש בסווק.
ככו כן, סחן אק ה- ID Network ה- ID Host בבל אק מהכקוקבוק:

1. 245.12.33.102
2. 123.123.123.123
3. 199.23.107.255
4. 199.23.107.0
5. 156.266.12.103
6. 99.0.0.12
7. 153.0.0.0
8. 153.0.0.255
9. 191.23.255.255
10. 33.255.255.0
11. 12.0.0.0
12. 12.255.255.255
13. 12.0.0.255
14. 127.0.0.1
15. 127.23.109.122
16. 0.23.12.122

כמות Networks וכמות Hosts בכל רשת

איך יודעים רשתות יש בכל Class וכמה Hosts יכולים להיות בכל רשת (חוץ מע"י שינון בעל-פה של הטבלה...)?
פשוט מאוד. ברשת Class A אוקטטה אחת משמשת לשם הרשת (שם הרחוב) ו-3 אוקטטות משמשות לשמות המחשבים – Host ID (או מס' הבית בכל רחוב). אבל באוקטטה הראשונה לקחו לנו את הביט הראשון (חייב להיות 0), לכן נותרנו עם 7 ביטים למשחק עבור הרשת, ועוד 3 אוקטטות למשחק עבור ה-Host או 24 ביטים. כלומר 2 בחזקת 7 עבור הרשתות, ו-2 בחזקת 24 עבור מספר ה-Hosts בכל רשת.

רשת Class A:

Network ID. Host ID. Host ID. Host ID

00000000.00000000.00000000.00000000

7 Bits for Network ID and 8+8+8 Bits for Host ID

אין רשת שמתחילה ב-0 ולכן האפשרות הראשונה שלנו היא:

00000001.00000000.00000000.00000000

נהפוך את האוקטטות לעשרוני ונראה מה יצא:

1.0.0.0

מה הקומבינציה האחרונה שאפשר לעשות באוקטטה הראשונה?

01111111.00000000.00000000.00000000

נהפוך את האוקטטות לעשרוני ונראה מה יצא:

127.0.0.0

עצור! מה המשמעות של כתובת כזו? נכון מאוד! זוהי כתובת השמורה עבור ה-Loopback Address, ולכן אין אנו יכולים להשתמש בה. לכן, נחזור לאוקטטה הראשונה ונשנה אותה לאחד פחות מ-127, כלומר 126:

01111110.00000000.00000000.00000000

נהפוך את האוקטטות לעשרוני ונראה מה יצא:

126.0.0.0

כלומר טווח כתובות הרשת השונות ב- Class A הוא בין 1 ל- 126.

כמה אפשרויות יש לנו באוקטטה הראשונה? או שנספור על האצבעות (אבל יש לנו רק 20, כולל הרגלים...) או שנלך בדרך המתמטית: לקחו לנו את הביט הראשון ונותרו לנו רק 7 ביטים לשחק איתם. 2 בחזקת 7 שווה 128 אפשרויות שונות אבל מכיוון שהרשת הראשונה (הכל 0) והרשת האחרונה (127) נופלות, יש לנו 126 אפשרויות שונות (מ- 1 ועד 126).

עכשיו מה לגבי מספר ה- Hosts האפשריים בכל אחת מ- 126 הרשתות השונות? נהפוך שוב לבינארי ונראה. הקצה התחתון של הקומבינציות האפשריות הוא:

00000001.00000000.00000000.00000000

נהפוך את האוקטטות לעשרוני ונראה מה יצא:

1.0.0.0

אבל אי אפשר לתת כתובת Host שנגמר ב- 0, לכן נחזור ונשנה את הכתובת לאחד יותר מ- 0, כלומר 1.

00000001.00000000.00000000.00000001

נהפוך את האוקטטות לעשרוני ונראה מה יצא:

1.0.0.1

נתחיל להעלות את הביטים אחד אחד (00000010, 00000011, 00000100, 00000101 וכ'ו) ונחפש את הקומבינציה הכי גבוהה:

01111110.11111111.11111111.11111111

אגב, למה באוקטטה הראשונה כתבתי 01111110 ולא 01111111? כיוון שזוהי הכתובת השמורה ה- Loopback Address, ולכן אין לנו יכולים להשתמש בה. לא לשכוח!

נהפוך לעשרוני ונראה מה יצא:

126.255.255.255

אבל כמו שציינו בהתחלה, אסור Host ID שנגמר ב- 255, ולכן נוריד מהאוקטטה האחרונה ביט אחד, ונהפוך אותה ל- 254:

01111110.11111111.11111111.11111110

נהפוך לעשרוני ונראה מה יצא:

126.255.255.254

כלומר, רשת מסוג Class A יכולה להכיל מחשבים בין הטווחים 0.0.1 עד ל- 255.255.254. כמה אפשרויות אלה? מדובר ב- 3 אוקטטות, כל אחת 8 ביטים. לכן:

$$2^8 * 2^8 * 2^8 = 2^{24} = 16777216$$

אבל שוב, אי אפשר הכל 0 והכל 1, לכן מורידים 2:

$$16777216 - 2 = 16777214$$

רשת Class B:

Network ID. Network ID. Host ID. Host ID

10000000.00000000.00000000.00000000

6+8 Bits for Network ID and 8+8 Bits for Host ID

האפשרות הראשונה שלנו היא:

10000000.00000000.00000000.00000000

נהפוך את האוקטטות לעשרוני ונראה מה יצא:

128.0.0.0

מה הקומבינציה האחרונה שאפשר לעשות באוקטטה הראשונה?

10111111.11111111.00000000.00000000

נהפוך את האוקטטות לעשרוני ונראה מה יצא:

191.255.0.0

כלומר טווח כתובות הרשת השונות ב- Class B הוא בין 128.0 ל- 191.255. כמה אפשרויות יש לנו באוקטטה הראשונה? לקחו לנו שני ביטים מובילים ולכן נותרו לנו רק 6 ביטים לשחק איתם. 2 בחזקת 6 שווה 64 אפשרויות שונות. באוקטטה השנייה יש לנו 8 ביטים מלאים, ולכן 2 בחזקת 8 שווה 256. 64 כפול 256 שווה 16384 אפשרויות שונות לרשתות ב- Class B. אגב, למה כאן לא הורדתי 2, כמו ברשת Class A? התבוננו באוקטטה הראשונה וראו שחייבו אותנו ששני הביטים הראשונים יהיו 10, ולכן בעצם לעולם לא יהיה מצב שבו כל הביטים יהיו 0 או 1.

עכשיו מה לגבי מספר ה- Hosts האפשריים בכל אחת מ- 16384 הרשתות השונות? נהפוך שוב לבינארי ונראה. הקצה התחתון של הקומבינציות האפשריות הוא:

10000000.00000000.00000000.00000000

נהפוך את האוקטטות לעשרוני ונראה מה יצא:

128.0.0.0

אבל אי אפשר לתת כתובת Host שנגמר ב- 0, לכן נחזור ונשנה את הכתובת לאחד יותר מ- 0, כלומר 1.

10000000.00000000.00000000.00000001

נהפוך את האוקטטות לעשרוני ונראה מה יצא:

128.0.0.1

נתחיל להעלות את הביטים אחד אחד (00000010, 00000011, 00000100, 00000101 וכו') ונחפש את הקומבינציה הכי גבוהה:

10111111.11111111.11111111.11111111

נהפוך לעשרוני ונראה מה יצא:

191.255.255.255

אבל כמו שציינו בהתחלה, אסור Host ID שנגמר ב- 255, ולכן נוריד מהאוקטטה האחרונה ביט אחד, ונהפוך אותה ל- 254:

01111110.11111111.11111111.11111110

נהפוך לעשרוני ונראה מה יצא:

126.255.255.254

כלומר, רשת מסוג Class B יכולה להכיל מחשבים בין הטווחים 0.1 עד ל- 255.254 כמה אפשרויות אלה? מדובר ב- 2 אוקטטות, כל אחת 8 ביטים. לכן:

$$2^8 * 2^8 = 2^{16} = 65536$$

אבל שוב, אי אפשר הכל 0 והכל 1, לכן מורידים 2:

$$65536 - 2 = 65534$$

רשת Class C:

Network ID. Network ID. Network ID. Host ID

11000000.00000000.00000000.00000000

5+8+8 Bits for Network ID and 8 Bits for Host ID

האפשרות הראשונה שלנו היא:

11000000.00000000.00000000.00000000

נהפוך את האוקטטות לעשרוני ונראה מה יצא:

192.0.0.0

מה הקומבינציה האחרונה שאפשר לעשות באוקטטה הראשונה?

11011111.11111111.11111111.00000000

נהפוך את האוקטטות לעשרוני ונראה מה יצא:

223.255.255.0

כלומר טווח כתובות הרשת השונות ב- Class C הוא בין 192.0.0 ל- 223.255.255

כמה אפשרויות יש לנו באוקטטה הראשונה? לקחו לנו שלושה ביטים מובילים ולכן נותרו לנו רק 5 ביטים לשחק איתם. 2 בחזקת 5 שווה 32 אפשרויות שונות. באוקטטה השנייה יש לנו 8 ביטים מלאים, ולכן 2 בחזקת 8 שווה 256. באוקטטה השלישית יש לנו 8 ביטים מלאים, לכן 2 בחזקת 8 שווה גם כאן ל- 256. 32 כפול 256 כפול 256 שווה 2097152 אפשרויות שונות לרשתות ב- Class C.

עכשיו מה לגבי מספר ה- Hosts האפשריים בכל אחת מ- 2097152 הרשתות השונות? נהפוך שוב לבינארי ונראה. הקצה התחתון של הקומבינציות האפשריות הוא:

11000000.00000000.00000000.00000000

נהפוך את האוקטטות לעשרוני ונראה מה יצא:

192.0.0.0

אבל אי אפשר לתת כתובת Host שנגמר ב- 0, לכן נחזור ונשנה את הכתובת לאחד יותר מ- 0, כלומר 1.

11000000.00000000.00000000.00000001

נהפוך את האוקטטות לעשרוני ונראה מה יצא:

192.0.0.1

נתחיל להעלות את הביטים אחד אחד (00000010, 00000011, 00000100, 00000101 וכו') ונחפש את הקומבינציה הכי גבוהה:

11011111.11111111.11111111.11111111

נהפוך לעשרוני ונראה מה יצא:

191.255.255.255

אבל כמו שציינו בהתחלה, אסור Host ID שנגמר ב- 255, ולכן נוריד מהאוקטטה האחרונה ביט אחד, ונהפוך אותה ל- 254:

11011111.11111111.11111111.11111110

נהפוך לעשרוני ונראה מה יצא:

223.255.255.254

כלומר, רשת מסוג Class C יכולה להכיל מחשבים בין הטווחים 1 עד ל- 254 כמה אפשרויות אלה? מדובר באוקטטה אחת בת 8 ביטים. לכן:

$$2^8 = 256$$

אבל שוב, אי אפשר הכל 0 והכל 1, לכן מורידים 2:

$$256-2=254$$

כדי לסכם את כל המספרים הללו:

Class	Leading Bit Pattern	Number of Bits left in the first octet	Number of Bits in the other Network ID octets	Total number of Bits used for the Network ID	Total number of Networks (2 in the power of the number of bits used for the Network ID)
A	0	7	-	7	126
B	10	6	8	6+8=14	16384
C	110	5	8+8=16	5+16=21	2097152

Class	Total number of Bits for the Host ID	2 in the power of Host ID Bits	Total number of Hosts (minus all 0 and all 1)
A	8+8+8=24	16777216	16777214
B	8+8=16	65536	65534
C	8	256	254

כתובות ציבוריות וכתובות פרטיות

כאשר אנחנו מחלקים כתובות IP (לא משנה באיזו שיטה – ידנית או אוטומטית) אנחנו יכולים לבחור בין 2 סוגי כתובות: כתובות ציבוריות (Public Addresses) וכתובות פרטיות (Private Addresses).

כתובת ציבורית

כתובת ציבורית היא כתובת שניתנת לנו ע"י האחראים על חלוקת הכתובות באינטרנט. ברוב המקרים אנחנו צריכים לשלם כסף עבור הכתובות הללו. מחשבים שמקבלים כתובות ציבוריות יכולים להשתמש בכתובות הללו להתחבר לאינטרנט, והכתובות הללו הן שלנו לנצח (או לפחות עד שנפסיק לשלם עבורם).

כתובת פרטית

כתובת פרטית היא כתובת שלא רשומה בשום מקום (הכוונה שהיא לא ניתנת לנו ע"י איזה שהוא אירגון, ואנחנו לא צריכים לשלם עבורה). כתובות פרטיות הן כתובות שאנחנו מציאים ומשתמשים בהם לצרכינו הפרטיים. אפשר להמציא כל כתובת, אבל מומלץ ללכת לפי קווים מנחים שהוגדרו במסמך ה-RFC 1918 (לא יודע מה זה RFC? לך ללמוד), כמו למשל כתובת מסוג 10.0.0.0 וכו'. כאשר אנחנו משתמשים בכתובות פרטיות אנחנו לא יכולים להתחבר באמצעותם לאינטרנט. אף נתב ציבורי לא יעביר אלינו תשדורות, ואם ננסה להתחבר בכל זאת, אנחנו עלולים לפגוע בתקשורת של מי שמשתמש באותן כתובות מכוח העובדה שהוא כן קנה אותן בכסף. אם בכל זאת נרצה לחבר רשת כזו אל האינטרנט נצטרך להשתמש בשירותים שונים כמו NAT (Network Address Translation) או Proxy מסוגים שונים.

כאשר אנחנו רוצים לבנות רשת באמצעות כתובות ציבוריות אנחנו צריכים לקנות מספק האינטרנט (ISP) שלנו כתובת של רשת, כמו למשל 137.57.0.0 ועליה להתחיל להגדיר את כל ה-Hosts אחד אחד. בכתובות פרטיות דבר ראשון שנעשה זה לקבוע לעצמנו את כתובת הרשת (כאמור מומלץ מאוד לבדוק קודם ב-RFC 1918 כדי לראות אילו כתובות יכולות לשמש אותנו), למשל 10.0.0.0, ואז להתחיל לחלק כתובות Hosts מתוך הרשת הזו לכל אחד מהמחשבים ברשת שלנו.

חלק שלישי

בחלק זה נתחיל להסביר כיצד פועל ה-Subnet Mask, פעולת ה-AND, הסבר קצר על סגמנטים, הסבר ארוך על פעולת ה-Subnetting, Subnetting של אוקטטה שלמה ושל חלק מאוקטטה, וחלוקה של טווחי רשתות.

איך פועל ה-Subnet Mask?

אנחנו יודעים שאם יש לנו כתובת IP שבה האוקטטה הראשונה נמצאת בטווח מסויים, אפשר במבט אחד לקבוע חד משמעית מאיזה Class אותה כתובת. אבל שיוך הכתובת ל-Class מסויים לא נותנת לנו שום דבר כשמדובר ברשת שמחולקת ל-Subnets, כפי שעוד מעט נראה.

למעשה, שיוך הכתובת ל-Class מסויים הוא שיוך קוסמטי בלבד. מה שבאמת קובע הוא כתובת הרשת, או ה- Network ID של אותה כתובת. אני צריך לקבוע באותה כתובת אילו מהאוקטטות יהיו שייכות לכתובת הרשת, ואילו מהאוקטטות יהיו שייכות ל-Host ID. כאן נכנס לתמונה ה-Subnet Mask.

ה-Subnet Mask מאפשר לי לקבוע באופן שרירותי אילו מהאוקטטות יהיו שייכות ל- Network ID. בגלל שאני יודע ש-Class A משתמש רק באוקטטה הראשונה לכתובת הרשת, ניתן לו Subnet Mask קבוע של 255.0.0.0. אם אני יודע ש-Class B משתמש ב-2 האוקטטות הראשונות לכתובת הרשת, ניתן לו Subnet Mask קבוע של 255.255.0.0.

	W	X	Y	Z	Default Subnet Mask
Class A	Network	Host	Host	Host	255.0.0.0
Class B	Network	Network	Host	Host	255.255.0.0
Class C	Network	Network	Network	Host	255.255.255.0

לאחר שהאדמיניסטרטור קובע את ה-Subnet Mask של הרשת ומיישם אותו על כל המחשבים באותה רשת, פרוטוקול ה-IP בוחן את כתובת ה-IP דרך המסיכה – ה-Subnet Mask כדי להחליט לגבי כתובת הרשת שלו. המילה Mask (מסיכה) מקבל את המשמעות של "עדשה" (כמו במשקפיים), כיוון שהתוכנה מתבוננת בכתובת ה-IP דרך העדשות של ה-Subnet Mask כדי לקבוע מה היא כתובת הרשת שלה.

ל- Subnet Mask יש פורמט קבוע. הוא יכול להיות משהו כמו 255.0.0.0, או משהו כמו 255.255.255.0. הפורמט הזה קובע למחשב איזו אוקטטה שייכת לכתובת הרשת באמצעות פעולה מתמטית שנקראת AND.

הפעולה הזו היא בעצם פעולת כפל פשוט. להזכירכם (לכל המוגבלים מתמטית...) בפעולת כפל רגילה כפולה של כל מספר ב- 0 שווה תמיד 0. המחשב לוקח את כתובת ה- IP הנתונה (הוא הרי רואה אותה בפורמט בינארי) וכותב מתחתיה את ה- Subnet Mask שהגדרנו לו.

שוב, לאלה מאיתנו שקשה להם לקלוט מהר, הטבלה הבאה תעזור להמחיש זאת ביתר קלות. ניקח סתם מספר בבינארי ונבצע עליו את פעולת ה- AND.

1	1	1	0	1	1	1	0
☐	☐	☐	☐	☐	☐	☐	☐
1	1	1	1	1	1	1	1

פעולת ה- AND היא כאמור פעולת כפל. אז ניקח ביט אחד ונכפיל אותו בביט שמתחתיו, ובשורה שלמטה נכתוב את התוצאה.

1	1	1	0	1	1	1	0
---	---	---	---	---	---	---	---

גם למוגבלים שבינינו יש בוודאי את היכולת לראות שכאשר כופלים ערך בינארי כלשהו (השורה העליונה בטבלה) בערך בינארי אחר, שכולו 1 (או בעשרוני – 255), תמיד תתקבל התוצאה הזוהה לשורה שמעליה. ואם נכפיל את הערך הבינארי (בשורה העליונה בטבלה) בערך בינארי אחר שכולו 0 (או בעשרוני – 0), אז תמיד תתקבל תוצאה הזוהה ל-0.



אם ניקח כתובת ב- Class A כמו 64.0.0.10, אנחנו יודעים שרק האוקטטה הראשונה מייצגת את ה- Network ID. נהפוך 64.0.0.10 לבינארי ונכתוב את הכתובת בטבלה הבאה:

כלומר (כל הכתובת):

$$01000000.00000000.00000000.00001010 = 64.0.0.10$$

מתחתיה נכתוב את ה- Subnet Mask בבינארי:

$$11111111.00000000.00000000.00000000 = 255.0.0.0$$

עכשיו נבצע פעולת AND או כפל בין שניהם, ביט למעלה כפול ביט למטה, ונכתוב את התוצאה מתחת:

$$01000000.00000000.00000000.00000000 = 64.0.0.0$$

רואים שהתוצאה של האוקטטה הראשונה זהה לכתובת הרשת המקורית – 64.0.0.0. עכשיו, נניח שהמחשב בעל הכתובת הזו מנסה לעשות Ping למחשב אחר בעל כתובת 64.0.0.45. לפני שהוא מבצע משלוח חומר כלשהו, המחשב שלי חייב לבצע החלטה: האם כתובת היעד נמצאת אצלי ברשת המקומית, או שמא אני צריך לשלוח את המידע החוצה לרשת המרוחקת דרך ה- Default Gateway? לכן הוא מיד מבצע פעולת AND של כתובת היעד עם ה- Subnet Mask המקורי שלי.

נכתוב זאת שוב:

$$01000000.00000000.00000000.00101101 = 64.0.0.45$$

מתחתיה נכתוב את ה- Subnet Mask בבינארי:

$$11111111.00000000.00000000.00000000 = 255.0.0.0$$

עכשיו נבצע פעולת AND או כפל בין שניהם, ביט למעלה כפול ביט למטה, ונכתוב את התוצאה מתחת:

$$01000000.00000000.00000000.00000000 = 64.0.0.0$$

מה יצא? יצא שכתובת הרשת המקורית שלי זהה לכתובת הרשת של היעד. לכן מה המסקנה? היעד נמצא ברשת שלי, ואז המחשב שלי יבצע את כל הפעולות הנדרשות כדי להגיע אליו (ARP וכו').



נניח שאני מנסה לעשות Ping למחשב בעל הכתובת 64.32.67.88. שוב פעם, המחשב שלי מבצע פעולת AND עם כתובת היעד ומשווה אותה לתוצאת ה- AND של הכתובת המקורית שלי. אם התוצאה שווה, זה באותה רשת. אם לא – שלח אותו לכל הרוחות לכיוון ה- Default Gateway!

$$01000000.00100000.01000011.01011000 = 64.32.67.88$$

מתחתיה נכתוב את ה- Subnet Mask בבינארי:

$$11111111.00000000.00000000.00000000 = 255.0.0.0$$

עכשיו נבצע פעולת AND או כפל בין שניהם, ביט למעלה כפול ביט למטה, ונכתוב את התוצאה מתחת:

$$01000000.00000000.00000000.00000000 = 64.0.0.0$$

מה יצא? יצא שכתובת הרשת המקורית שלי זהה לכתובת הרשת של היעד. לכן מה המסקנה? היעד נמצא ברשת שלי, ואז המחשב שלי יבצע את כל הפעולות הנדרשות כדי להגיע אליו (ARP וכו'). אין למחשב שלי שום ברירה במקרה הזה. הוא חייב לשלוח את התשדורת ברשת המקומית, וגם אם הוא מאוד ירצה, הוא לא יכול לנתב את התשדורת דרך שום ראوتر או כל מכשיר אחר, פרט למשלוח רגיל בתוך הרשת המקומית.

עוד דוגמא? בבקשה:



מהמחשב שלי למחשב 65.32.67.88.

$$01000001.00100000.01000011.01011000 = 65.32.67.88$$

מתחתיה נכתוב את ה- Subnet Mask בבינארי:

$$11111111.00000000.00000000.00000000 = 255.0.0.0$$

עכשיו נבצע פעולת AND או כפל בין שניהם, ביט למעלה כפול ביט למטה, ונכתוב את התוצאה מתחת:

$$01000001.00000000.00000000.00000000 = 65.0.0.0$$

אופס! יצאה כתובת רשת אחרת. לכן הוא שולח את המידע אל ה- Default Gateway. אין למחשב שום ברירה במקרה הזה. ברגע שהוא מזהה שהכתובת המרוחקת נמצאת על רשת אחרת, הוא אוטומטית מפנה את התשדורת אל הנתב שאמור להציע נתיב אל אותה רשת מרוחקת, או במילים אחרות, ה- Default Gateway.



כתוב את ה- Default Subnet Mask של הכתובות הבאות לפי סדרן-

:Classes

1. 210.23.67.102

2. 66.23.148.0

3. 158.23.251.33

4. 144.23.117.254

5. 192.254.23.123

6. 144.207.78.1

7. 63.125.23.211

8. 192.25.128.36

9. 128.12.254.98

10. 134.223.156.89

11. 127.0.0.1

12. 224.23.108.23

13. 223.78.27.144

14. 77.123.28.167

15. 191.249.222.234

Subnetting

נניח שלארגון יש הרבה מחשבים, או שהוא מחולק לכמה סניפים המרוחקים זה מזה גיאוגרפית, אז הגיוני שאותו ארגון ישאף לחלק את הרשת הענקית שלו לכמה סגמנטים או רשתות קטנות יותר המחוברות ביניהן באמצעות נתבים (Routers).

מה זה סגמנט?

סגמנט הוא חלק של רשת. מקטע של רשת שמופרד ממקטע אחר ע"י נתבים (או Routers). אפשר לאפיין סגמנט כמקטע פיזי של רשת שבתוכו אפשר לקבל מידע ממחשב אחר באמצעות Broadcasts. כלומר, שני מחשבים שנמצאים בסגמנט אחד יכולים לחליף ביניהם מידע באמצעות ברודקסטינג. להזכירכם, נתבים לא מעבירים ברודקסטים, ולכן אם יש לנו רשת שמחולקת ל- 2 חלקים ע"י נתב באמצע, יש לנו 2 סגמנטים. אם הרשת היתה מופרדת ע"י מכשירים אחרים כדוגמת Switch או Bridge אז לא היו לנו 2 סגמנטים, כיוון שהמכשירים המוזכרים כאן מסוגלים לבצע העברה של ברודקסטים מצד אחד לצד השני, ולכן כן נוכל לקבל מידע ממחשב אחר שנמצא מעבר ל- Bridge ע"י ברודקסטינג.

זה חשוב כיוון שאם נחזור להתחלת המאמר, בו דיברנו על כתובות IP, קבענו חוק שאומר שכל המחשבים או ה- Devices שנמצאים על רשת אחת חייבים שתהיה להם אותה כתובת רשת, או Network ID. לכן, לכל המחשבים בסגמנט אחד חייבת להיות אותה כתובת רשת. כמובן שתהיה להם Host ID שונה (כדי לספק את הדרישה של כתובת IP ייחודית לכל מחשב), אבל לכולם תהיה אותה כתובת רשת. חשוב גם לזכור שאם יש לנו 2 סגמנטים, לכל אחד תהיה כתובת רשת אחרת, כדי שהמחשבים באותו סגמנט ידעו שהם לא נמצאים על אותו סגמנט כמו המחשבים האחרים. אם היתה לנו רשת שמחולקת ל- 10 סגמנטים, זאת אומרת שהינו צריכים כתובת רשת אחת עבור כל סגמנט, כלומר 10 כתובות רשת שונות. כל המחשבים בסגמנט אחד יהיו בעלי אותה כתובת רשת ואותו Subnet Mask.

היתרונות לחלוקה כזו הם בין היתר:

- הקטנה משמעותית של עומס הרשת: כולנו מעדיפים פחות תעבורה, בין אם מדובר בפקק תנועה ובין אם מדובר בעומס על ספק האינטרנט שלנו. אילמלא הרשת הייתה מחולקת באמצעות נתבים, הרי שכולנו היינו מתערבבים ברשתות של האחרים, והעומס היה הופך להיות בלתי נסבל. באמצעות הנתבים רוב התעבורה תישאר ברשת המקומית ולא תפריע לרשתות השכנות. רק ה-Packets שמיועדים לרשתות אחרות יורשו לצאת החוצה דרך הנתבים, ואילו תשדורות מקומיות כמו Broadcasts יישארו בתוך הרשת המקומית – המקום הטבעי שלהם.
- שיפור הביצועים של הרשת: הוא בעצם המשמעות של הקטנת העומס ברשת.
- ניהול פשוט יותר: קל יותר לבודד בעיות ברשת כאשר הרשתות הן גופים קטנים יחסית. כך אפשר למשל לנתק חיבור של נתב ולראות האם הבעיה ממשיכה וכו'.
- חיבור זול יותר של רשתות: כאשר אני מחלק רשת גדולה אחת לכמה תת-רשתות קטנות יותר אני יכול לבחור באמצעות איזה מדיום אני מחבר ביניהם. בנוסף, החיבור ביניהם (WAN) לא צריך לפעול כל הזמן, אלא רק כאשר יש לו דרישה.
- חיבור של כמה טופולוגיות רשת: כאשר אני משלב למשל Ethernet עם Token Ring אני צריך לעשות זאת באמצעות נתבים או מכשירים אחרים, והרי כדי להשתמש בנתב אני צריך בעצם לבנות 2 רשתות נפרדות.

לכן כאשר ארגון כל כך גדול מקבל רק כתובת רשת אחת (Network ID) מה-InterNIC, אז לאותו ארגון יש בעיה. תארו לעצמכם שכל ארגון שיש לו 5 סניפים עם 300 מחשבים בכל סניף יבקש וגם יקבל 5 כתובות רשת נפרדות של Class B! מאגר הכתובות הפנוי בעולם היה נגמר תוך בערך שבוע.

בעיה נוספת שיכולה להיווצר קשורה בנתבים (Routers) שמחברים את העולם: אם כל נתב באינטרנט היה צריך לדעת על הקיום של כל רשת קיימת, אז טבלאות הראוטינג (Routing Tables) היו הופכות להיות כל כך גדולות שרק העדכון שלהם היה לוקח שעות, שלא נדבר על העומס ברשת והעיכוב בהעברת הנתונים בעולם.



לחברת Zibi Ba-Libi יש 100 סניפים, כל אחד מהם עם 10 מחשבים. החברה יכולה לבחור האם לקנות כתובת רשת אחת כללית, ולבנות רשת אחת גדולה שתקיף את כל 100 הסניפים שלה, או שמא לקנות כמה כתובות של רשתות ולפצל את הרשת שלה לכמה רשתות קטנות יותר.

בכל מקרה, אם מדובר ברשת אחת גדולה – איזה Class צריכה החברה לקנות? Class B כמובן, כיוון שמדובר ב-100 סניפים ובכל אחד כ-10 מחשבים, כלומר 1000 מחשבים. אבל Class B מספיק לכמה מחשבים? חזור לטבלה בתחילת הפרק וראה שמדובר בכ-65000 מחשבים – מעל ומעבר למה שהחברה צריכה במציאות.

ואם מדובר בכמה רשתות קטנות יותר – איזה Class צריכה החברה לקנות? Class C כמובן, אבל נוצר מצב שהחברה תצטרך לקנות 100 רשתות שלמות של Class C, ולבזבז משהו כמו 5000 כתובות IP חוקיות למהדרין, רק כדי שיהיו לה 100 סניפים עם סה"כ 1000 מחשבים.

אני אמליץ לחברה לרכוש כתובת רשת אחת ב-Class B - למשל 130.57.0.0, ונרצה לחלק את אותה רשת לכמה תת-רשתות. אבל איך נעשה את זה? הרי לכל המחשבים ברשת של החברה תהיה אותה כתובת רשת של 130.57. איך נוכל לגרום למחשבים לדעת שעכשיו הם לא נמצאים באותה רשת?

חלוקה פיזית של הרשת למקטעים לא תעזור. למה? כיוון שאם נפריד את אותה רשת ל-100 חלקים ונחבר אותם באמצעות נתבים, אם מחשב אחד ירצה לתקשר עם מחשב אחר במקטע אחר של הרשת, הוא צריך לדעת שהוא אמור לשלוח את המידע דרך Default Gateway כלשהו, אחרת הוא יחפש אותו מקומית אצלו במקטע, וכמובן שהוא לא ימצא. אני חייב למצוא דרך לומר לאותו מחשב "תראה, אתה שייך לרשת כזאת, והוא שייך לרשת אחרת, לכן אנא שלח את המידע שלך דרך הנתב החוצה ואל תנסה לחפש אותו אצלך ברשת".

לכן, המציאו את מושג ה-Subnetting.

נקודות שצריך לקחת בחשבון לפני שמתחילים לבצע את פעולת ה-Subnetting:

לפני שמתחילים את החלוקה ל-Subnets צריך להביא בחשבון מספר דברים:

- כמה כתובות רשת (Network ID) אני צריך? אחת עבור כל Subnet, ואחת עבור כל קשר WAN.
- כמה כתובות מחשב (Host ID) אני צריך על כל Subnet? אחת עבור כל כרטיס רשת או Interface ברשת, ואחת עבור כל רגל של נתב.
- מה יהיה הגידול הצפוי ברשת שלי? אם אני יודע שבתוך שנתיים אני צריך לפתוח עוד 5 סניפים, הרי שמראש אני אבנה רשת בחלוקה כזו שתאפשר לי גידול עתידי צפוי.

לאחר שהכנתי את הנתונים הללו צריך ליצור:

- Subnet Mask אחד עבור כל הרשת.
- Subnet ID ייחודי עבור כל תת-רשת או Subnet.
- טווח כתובות Hosts אפשרי לכל Subnet.

יצירת Subnets היא למעשה הפעולה של יצירת תת-רשתות קטנות יותר מתוך רשת-אם אחת גדולה. ארגון עם כתובת רשת אחת יכול שיהיו לו כמה Subnet Addresses עבור כל תת-רשת, אבל שיהיה לו גם Network ID אחד ראשי עבור כל הרשת כולה. כל Subnet הוא עדיין חלק מתוך הרשת הגדולה, אבל יש לו למעשה גם מאפיין או שם נוסף שמפריד אותו משאר תת-הרשתות.

אפשר להמחיש זאת באמצעות דוגמה פשוטה: נניח שיש משפחה עם כמה ילדים. לכל הילדים יהיה אותו שם משפחה, ולפי השם הזה יגיע אליהם הדואר. אבל לכל ילד יש מאפיין שהוא שלו בלבד, ואפשר לפי המאפיין הזה לדעת על מי מדברים. למשל, ג'יפה המכוערת עם המוחננה, או זלדה הג'ינג'ית עם הפרונקעל. פרט למאפיין הזה יש לכל ילד גם שם פרטי שאפשר לפנות אליו לפיו. הפתרון הזה מאפשר לארגון להשאיר את כתובת הרשת (Network ID) הראשי שלו, אבל יחד עם זאת גם לפצל אותו לכמה תת-רשתות.

בדוגמה שלנו, הדוור שהולך כל בוקר ברגל מתיבת דואר אחת לשניה לא צריך לדעת על קיום כל בני המשפחה, ולא צריך להטריד אותו בכל השמות, הכינויים והתחביבים של בני המשפחה. הוא לא צריך לזכור את כל טבלת הראוטינג, אלא רק את שם המשפחה, כלומר את שם הרשת הראשית. הוא מכניס את הדואר לתוך התיבה לפי שם משפחה. אבל כשמגיע אחד מבני המשפחה לתיבה ומרוקן אותה, הדואר הפנימי בין בני המשפחה ממויין לפי תיאור נוסף, פנימי לגבי אותה משפחה. בנוסף, בגלל שפעולת ה-Subnetting מאפשרת לכמה תת-רשתות להיות מאוחדות יחד, טבלת הראוטינג של אותו ארגון תהיה קטנה יותר ויעילה יותר. כמו-כן, כאמור גם הביצועים של הרשתות הקטנות ישתפרו מכיוון שהן לא מעבירות ביניהן Broadcasting וכו'. אנחנו יוצרים Subnetting ע"י מתן Subnet ID לכל מחשב באותה רשת, ומשלבים כתובת זו בתוך הכתובת הכללית של אותו מחשב.

אמרנו כבר שאי אפשר לשנות את החלק של ה- Network ID בכתובת ה- IP, כיוון שכל המחשבים באותה רשת חייבים שתהיה להם אותה כתובת רשת כדי שיוכלו לתקשר זה עם זה. לפי הדוגמא של חברת Zibi-Ba-Libi כל הרשת שייכת לכתובת 130.57. עכשיו אני רוצה לקחת חלק מכתובת ה- Host, לגנוב ממנו חלק, ולהשתמש בו ככתובת תת-הרשת, או ה- Subnet ID של אותה תת-רשת.

Subnetting של אוקטטה שלמה:

בכתובת Class B רגילה, כמו זו של החברה בדוגמא, 2 האוקטטות הראשונות משמשות לכתובת הרשת, ו-2 האוקטטות האחרונות משמשות כ- Host ID. איזה Default Subnet Mask היה לכתובת כזו?

מה שנעשה עכשיו זה לקחת את האוקטטה השלישית, ולומר לכולם "זו תהיה כתובת תת-הרשת (Subnet ID) שלכם".

איך עושים זאת? פשוט מאוד: במקום לתת Subnet Mask רגיל של 255.255.0.0, נגדיר עכשיו Subnet Mask חדש לכל הרשת ונקרא לו 255.255.255.0.

לכל המחשבים בסניף אחד ניתן כתובת כזו: 130.57.1.0, והטווח של כל רשת כזו יהיה מ- 130.57.1.1 עד 130.57.1.254. בסניף שני ניתן כתובת כזו: 130.57.2.0, והטווח יהיה מ-1 ועד 254, וכו' וכו'.

כמה רשתות יהיו לי? 254 רשתות שונות, כשלכל אחת יש עד 254 מחשבים. כל זה במקום רשת אחת ענקית עם כ- 65000 מחשבים.

ברור שיש כאן בזבוז של כתובות, כיוון שאם נכפיל 254 ב- 254 יצא לנו 64516 מחשבים בסה"כ, ואילו אם היינו משאירים את המצב כמו שהוא, היינו יכולים להכניס 65534 מחשבים לאותה רשת. אבל זהו מחיר נמוך לשלם עבור היכולת לחלק את הרשת שלי ל- 254 תת-רשתות קטנות.

כדי להסביר בצורה יותר מוחשית, ניקח את המצב כמו שהיה, ונראה כיצד הוא השתנה לאחר היישום של ה-Subnetting:

לפני:

Network ID		Host ID	
130	57	X	Y

ה-Subnet Mask המקורי היה:

255	255	0	0
-----	-----	---	---

יש לנו רק רשת אחת גדולה עם 2 בחזקת 16 אפשרויות ל-Host ID שונות (65534 כתובות שונות).

אחרי:

Network ID		Subnet ID	Host ID
130	57	X	Y

יש לנו רשת אחת גדולה שמחולקת ל-2 בחזקת 8 תת-רשתות (254 ולא 256 כצפוי, כיוון שאי אפשר הכל 0 ואי אפשר הכל 1 – לזכור את זה טוב טוב!!!), כשכל תת-רשת כזו יכולה להכיל 2 בחזקת 8 כתובות שונות (גם כאן, רק 254 ולא 256).

איך עשינו את הטריק? פשוט מאוד. לקחנו את ה-Subnet Mask ושינינו אותו. כברירת מחדל ה-Subnet Mask המקורי היה 255.255.0.0, וכעת הפכנו אותו ל-255.255.255.0, וכאילו "גנבנו" אוקטטה אחת מה-Host ID ונתנו אותה במתנה לחלק של ה-Network ID.

ה-Subnet Mask החדש הוא:

255	255	255	0
-----	-----	-----	---

ננסה להסביר את המושג בצורה הבינארית שלו: נסו להיזכר איך יודע המחשב האם כתובת מסויימת נמצאת על הרשת "שלו" או שמא היא נמצאת על רשת אחרת, ויש להפנות את התעבורה אליה דרך ה-Default Gateway.

דיברנו על פעולת ה-AND. נחזור עליה כאן:
ניקח כתובת כלשהי, למשל 130.57.13.5, ונראה האם היא נמצאת באותה רשת כמו 130.57.60.12.

$$10000010.00111001.00001101.00000101 = 130.57.13.5$$

מתחתיה נכתוב את ה-Subnet Mask בבינארי:

$$11111111.11111111.00000000.00000000 = 255.255.0.0$$

עכשיו נבצע פעולת AND או כפל בין שניהם, ביט למעלה כפול ביט למטה, ונכתוב את התוצאה מתחת:

$$10000010.00111001.00000000.00000000 = 130.57.0.0$$

רואים שהתוצאה זהה לכתובת הרשת המקורית – 130.57.0.0.
עכשיו, נניח שהמחשב בעל הכתובת הזו מנסה לעשות Ping למחשב אחר בעל כתובת 130.57.60.12. לפני שהוא מבצע משלוח חומר כלשהו, המחשב שלי חייב לבצע החלטה: האם כתובת היעד נמצאת אצלי ברשת המקומית, או שמא אני צריך לשלוח את המידע החוצה לרשת המרוחקת דרך ה-Default Gateway? לכן הוא מיד מבצע פעולת AND של כתובת היעד עם ה-Subnet Mask המקורי שלי. נכתוב זאת שוב:

$$10000010.00111001.00111100.00001100 = 130.57.60.12$$

מתחתיה נכתוב את ה-Subnet Mask בבינארי:

$$11111111.11111111.00000000.00000000 = 255.255.0.0$$

עכשיו נבצע פעולת AND או כפל בין שניהם, ביט למעלה כפול ביט למטה, ונכתוב את התוצאה מתחת:

$$10000010.00111001.00000000.00000000 = 130.57.0.0$$

מה יצא? יצא שכתובת הרשת המקורית שלי זהה לכתובת הרשת של היעד. לכן מה המסקנה? היעד נמצא ברשת שלי, ואז המחשב שלי יבצע את כל הפעולות הנדרשות כדי להגיע אליו (ARP וכו'). כלומר ברשת מסוג זה, עם ה-Subnet Mask המקורי של 255.255.0.0, שני המחשבים נמצאים באותה רשת (זה גם ברור לפי מבט חטוף בכתובת הרשת של שניהם – 130.57).

עכשיו בואו נשתמש ב- Subnet Mask החדש – 255.255.255.0 ונראה כיצד זה משפיע על חלוקת הרשת. ניקח את אותן דוגמאות ממקודם, רק הפעם עם ה- Subnet Mask החדש.

ניקח את אותה כתובת - 130.57.13.5, ונראה האם היא נמצאת באותה רשת כמו 130.57.60.12.

$$10000010.00111001.00001101.00000101 = 130.57.13.5$$

מתחתיה נכתוב את ה- Subnet Mask בבינארי:

$$11111111.11111111.11111111.00000000 = 255.255.255.0$$

עכשיו נבצע פעולת AND או כפל בין שניהם, ביט למעלה כפול ביט למטה, ונכתוב את התוצאה מתחת:

$$01000000.00111001.00001101.00000000 = 130.57.13.0$$

רואים שהתוצאה זהה לכתובת הרשת המקורית – 130.57.13.0. שימו לב: הפעם הרשת מקבלת שם חדש. לא רק 2 אוקטטות כמו קודם, אלא 3 אוקטטות.

עכשיו נכתוב את כתובת היעד:

$$10000010.00111001.00111100.00001100 = 130.57.60.12$$

מתחתיה נכתוב את ה- Subnet Mask בבינארי:

$$11111111.11111111.11111111.00000000 = 255.255.255.0$$

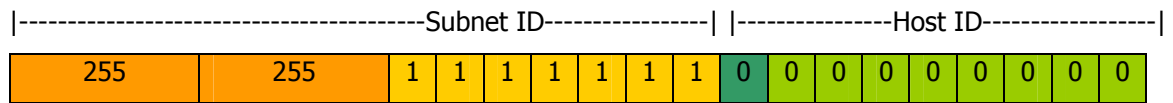
עכשיו נבצע פעולת AND או כפל בין שניהם, ביט למעלה כפול ביט למטה, ונכתוב את התוצאה מתחת:

$$10000010.00111001.00111100.00000000 = 130.57.60.0$$

מה יצא? יצא שכתובת הרשת השניה שונה לגמרי מכתובת הרשת הראשונה, לכן, מן הסתם, לא מדובר באותן רשתות. כלומר ע"י שינוי ה- Subnet Mask בצורה פשוטה ביותר הצלחנו לציין לשני המחשבים שהם לא נמצאים על אותה רשת כמו מקודם, למרות שלא נגענו בכתובת ה- IP עצמה.

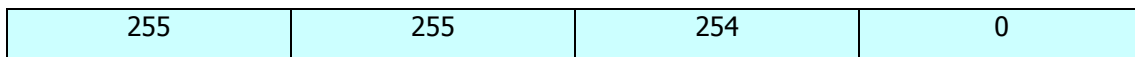
השגנו את מה שביקשנו. הפרדנו רשת אחת גדולה של Class B להרבה תת-רשתות קטנות. כמה תת-רשתות? חזור לתחילת הדוגמא וראה.

אם "נגנוב" את הביט הכי ימני מתוך ה- Subnet ID ונתן אותו במתנה ל- Host ID, מה זה יתן לנו?



עכשיו יש לנו רק 7 ביטים ל- Subnet ID, ו- 9 ביטים ל- Host ID.

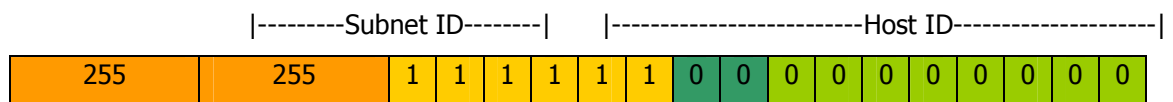
ניקח את המספר הבינארי 11111110 ונהפוך לעשרוני. כמה יצא? 254. לכן, ה- Subnet Mask החדש יהיה 255.255.254.0



כמה תת-רשתות יכולות להיות לנו אם נשתמש בכזה Subnet Mask? בואו נראה: 2 בחזקת מספר הביטים שנשארו, כלומר 2 בחזקת 7 שווה 128. האם יש לנו 128 תת-רשתות שונות? לא, כיוון שאסור לנו להשתמש בכתובת שהיא הכל 0, וכתובת שהיא הכל 1. לכן 128 פחות שתי האפשרויות הללו שווה 126, ואכן יש לנו 126 תת-רשתות אפשריות.

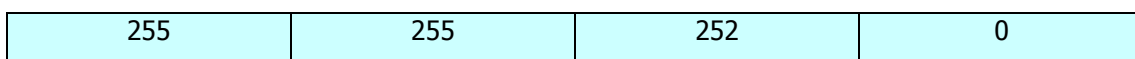
בואו נבדוק כמה כתובות אפשריות של Host ID יש לנו: עכשיו יש לנו 9 ביטים לשחק איתם. לכן, 2 בחזקת 9 שווה 512. האם יש לנו 512 כתובות שונות? לא, כיוון שאסור לנו להשתמש בכתובת שהיא הכל 0, וכתובת שהיא הכל 1. לכן 512 פחות שתי האפשרויות הללו שווה 510, ואכן יש לנו 510 Host ID שונים בכל תת-רשת אפשרית.

נניח לרגע שגם 510 מחשבים ברשת לא מספיקים לי. אני רוצה יותר. אז בואו ניקח עוד ביט אחד וניתן אותו מתנה ל- Host ID:



עכשיו יש לנו רק 6 ביטים ל- Subnet ID, ו- 10 ביטים ל- Host ID.

ניקח את המספר הבינארי 11111100 ונהפוך לעשרוני. כמה יצא? 252. לכן, ה- Subnet Mask החדש יהיה 255.255.252.0

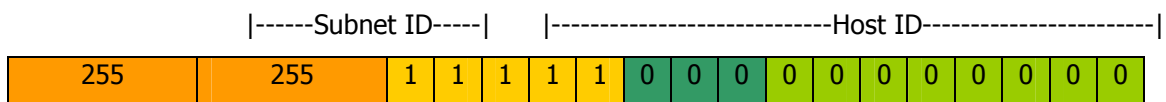


כמה תת-רשתות יכולות להיות לנו אם נשתמש בכזה Subnet Mask? בואו נראה: 2 בחזקת מספר הביטים שנשארו, כלומר 2 בחזקת 6 שווה 64. לא לשכוח שאסור לנו להשתמש בכתובת שהיא הכל 0, וכתובת שהיא הכל 1. לכן 64 פחות שתי האפשרויות הללו שווה 62, ואכן יש לנו 62 תת-רשתות אפשריות.

בואו נבדוק כמה כתובות אפשריות של Host ID יש לנו: עכשיו יש לנו 10 ביטים לשחק איתם. לכן, 2 בחזקת 10 שווה 1024 פחות 2 שווה 1022, ואכן יש לנו 1022 Host ID שונים בכל תת-רשת אפשרית.

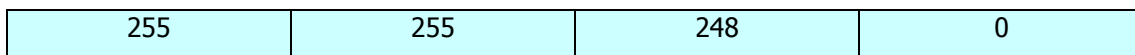
שימו לב איך אנחנו מתחילים לצמצם את מספר התת-רשתות, אבל יחד עם זאת להעלות את מספר המחשבים בכל תת-רשת כזאת. נמשיך.

נניח לרגע שגם 1022 מחשבים ברשת לא מספיקים לי. אני רוצה יותר. אז בואו ניקח עוד ביט אחד וניתן אותו מתנה ל- Host ID:



עכשיו יש לנו רק 5 ביטים ל- Subnet ID, ו- 11 ביטים ל- Host ID.

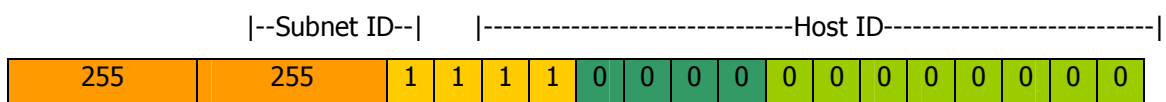
ניקח את המספר הבינארי 11111000 ונהפוך לעשרוני. כמה יצא? 248 (בדוק!). לכן, ה- Subnet Mask החדש יהיה 255.255.248.0



כמה תת-רשתות יכולות להיות לנו אם נשתמש בכזה Subnet Mask? בואו נראה: 2 בחזקת מספר הביטים שנשארו, כלומר 2 בחזקת 5 שווה 32. האם יש לנו 32 תת-רשתות שונות? לא, כיוון שאסור לנו להשתמש בכתובת שהיא הכל 0, וכתובת שהיא הכל 1. לכן 32 פחות 2 שווה 30, ואכן יש לנו 30 תת-רשתות אפשריות.

מעניין, בואו נבדוק כמה כתובות אפשריות של Host ID יש לנו: עכשיו יש לנו 11 ביטים לשחק איתם. לכן, 2 בחזקת 11 שווה 2048. האם יש לנו 2048 כתובות שונות? לא, כיוון שאסור לנו להשתמש בכתובת שהיא הכל 0, וכתובת שהיא הכל 1. לכן 2048 פחות 2 שווה 2046, ואכן יש לנו 2046 Host ID שונים בכל תת-רשת אפשרית.

נניח לרגע שגם 2046 מחשבים ברשת לא מספיקים לי. אני רוצה יותר. אז בואו ניקח עוד ביט אחד וניתן אותו מתנה ל- Host ID:



עכשיו יש לנו רק 4 ביטים ל- Subnet ID, ו- 12 ביטים ל- Host ID.

ניקח את המספר הבינארי 11110000 ונהפוך לעשרוני. כמה יצא? 240 (בדוק!). לכן, ה- Subnet Mask החדש יהיה 255.255.240.0

255	255	240	0
-----	-----	-----	---

כמה תת-רשתות יכולות להיות לנו אם נשתמש בכזה Subnet Mask? בואו נראה: 2 בחזקת מספר הביטים שנשארו, כלומר 2 בחזקת 4 שווה 16, פחות 2 שווה 14, ואכן יש לנו 14 תת-רשתות אפשריות.

בואו נבדוק כמה כתובות אפשריות של Host ID יש לנו: עכשיו יש לנו 12 ביטים לשחק איתם. לכן, 2 בחזקת 12 שווה 4096. 4096 פחות 2 שווה 4094, ואכן יש לנו 4094 Host ID שונים בכל תת-רשת אפשרית.

עוד?

נניח לרגע שגם 4096 מחשבים ברשת לא מספיקים לי. אני רוצה יותר. אז בואו ניקח עוד ביט אחד וניתן אותו מתנה ל- Host ID:

---Sub---			-----Host ID-----															
255	255	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

עכשיו יש לנו רק 3 ביטים ל- Subnet ID, ו-13 ביטים ל- Host ID.

ניקח את המספר הבינארי 11100000 ונהפוך לעשרוני. כמה יצא? 224 (בדוק!). לכן, ה- Subnet Mask החדש יהיה 255.255.224.0

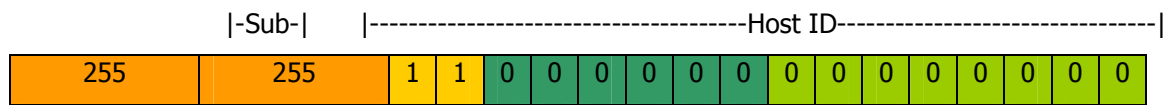
255	255	224	0
-----	-----	-----	---

כמה תת-רשתות יכולות להיות לנו אם נשתמש בכזה Subnet Mask? בואו נראה: 2 בחזקת מספר הביטים שנשארו, כלומר 2 בחזקת 3 שווה 8. 8 פחות 2 שווה 6, ואכן יש לנו 6 תת-רשתות אפשריות.

כמה כתובות אפשריות של Host ID יש לנו: עכשיו יש לנו 13 ביטים לשחק איתם. לכן, 2 בחזקת 13 שווה 8192. 8192 פחות 2 שווה 8190, ואכן יש לנו 8190 Host ID שונים בכל תת-רשת אפשרית.

תראו מה קורה: עכשיו יש לי רק 6 תת-רשתות, אבל 8190 מחשבים בכל תת-רשת. נמשיך עוד:

נניח לרגע שגם 8190 מחשבים ברשת לא מספיקים לי. אני רוצה יותר. אז בואו ניקח עוד ביט אחד וניתן אותו מתנה ל- Host ID:



עכשיו יש לנו רק 2 ביטים ל- Subnet ID, ו- 14 ביטים ל- Host ID.

ניקח את המספר הבינארי 11000000 ונהפוך לעשרוני. כמה יצא? 192 (בדוק!). לכן, ה- Subnet Mask החדש יהיה 255.255.192.0

255	255	192	0
-----	-----	-----	---

כמה תת-רשתות יכולות להיות לנו אם נשתמש בכזה Subnet Mask? בואו נראה: 2 בחזקת מספר הביטים שנשארו, כלומר 2 בחזקת 2 שווה 4. 4 פחות 2 שווה 4, ואכן יש לנו 2 תת-רשתות אפשריות.

כמה כתובות אפשריות של Host ID יש לנו? עכשיו יש 14 ביטים לשחק איתם. לכן, 2 בחזקת 14 שווה 16384. 16384 פחות 2 שווה 16382, ואכן יש לנו 16382 Host ID שונים בכל תת-רשת אפשרית.

הגענו למצב שבו הרשת הגדולה מפוצלת ל- 2 תת-רשתות, כשכל אחת יכולה להכיל עד 16382 מחשבים שונים. להזכירכם, כל זה במקום להשתמש בכתובת הרשת המקורית עם ה- Subnet Mask המקורי, שבה היתה לנו רק רשת אחת, אבל עם כ- 65000 מחשבים שונים. יש כאן אובדן גדול מאוד של כתובות פנויות (כ- 32000 לעומת כ- 65000), אבל שוב, זהו מחיר זול לשלם עבור היכולת להפריד את הרשת הגדולה הזאת לתת-רשתות.

האם אני יכול "לגנוב" עוד ביט אחד לטובת ה- Host ID? בואו נראה: אם ניקח עוד אחד נישאר עם ביט אחד עבור ה- Subnet ID, וזה אומר 2 בחזקת 1, כלומר 2. אבל אי אפשר הכל 0 והכל 1 ולכן זה 2 פחות 2, או בעצם 0 (אפס), ולכן אי אפשר להמשיך בפעולת ה"גניבה".

טבלה מסכמת של כתובת מ- Class B:

מס' הביטים המכוסים	Bit Pattern	כמה Subnets יש לנו?	מהו ה- Subnet Mask?	כמה Host ID לכל תת-רשת?
2	11000000	2	255.255.192.0	16382
3	11100000	6	255.255.224.0	8190
4	11110000	14	255.255.240.0	4094
5	11111000	30	255.255.248.0	2046
6	11111100	62	255.255.252.0	1022
7	11111110	126	255.255.254.0	510
8	11111111	254	255.255.255.0	254

- מספר הביטים המכוסים הוא המספר של הביטים שנשאר לטובת ה- Subnet ID.
- ה- Bit Pattern הוא המבנה של האוקטטה השלישית, זו שעליה "רבים" ה- Subnet ID וה- Host ID.

כתובת מ- Class A:

בואו נבצע את אותה פעולה, אבל הפעם עם כתובת אחרת, מ- Class A:

ניקח כתובת כמו 30.0.0.0 לפני (מתחת - ה- Subnet Mask המקורי):



Network ID	Host ID		
30	X	Y	Z

255	0	0	0
-----	---	---	---

יש לנו רק רשת אחת גדולה עם 2 בחזקת 24 אפשרויות ל- Host ID שונות (3 אוקטטות שווה 24 ביטים), או 16777216 אפשרויות, אבל אסור לשכוח להוריד 2, לכך בעצם 16777214 אפשרויות שונות. עכשיו "נגנוב" אוקטטה אחת לטובת ה- Subnet ID:

Network ID	Subnet ID	Host ID	
30	X	Y	Z

255	255	0	0
-----	-----	---	---

מעניין. בואו נבדוק כמה כתובות אפשריות של Host ID יש לנו: עכשיו יש לנו 17 ביטים לשחק איתם. לכן, 2 בחזקת 17 שווה 131072. האם יש לנו 131072 כתובות שונות? לא, כיוון שאסור לנו להשתמש בכתובת שהיא הכל 0, וכתובת שהיא הכל 1. לכן 131072 פחות שתי האפשרויות הללו שווה 131070, ואכן יש לנו Host ID 131070 שונים בכל תת-רשת אפשרית.

נמשיך באופן הזה לאורך כל האוקטטה השניה. נוסיף כל פעם ביט אחד עבור ה- Host ID, עד שנגיע ל- 2 הביטים הכי שמאליים, ול- Subnet Mask של 255.192.0.0. אם עדין אתם לא מבינים איך זה נעשה, קחו את הדוגמאות הקודמות והמשיכו באותה דרך לגבי הדוגמא הנוכחית.

טבלה מסכמת של כתובת מ- Class A:

מס' הביטים המכוסים	Bit Pattern	כמה Subnets יש לנו?	מהו ה- Subnet Mask?	כמה Host ID לכל תת-רשת?
2	11000000	2	255.192.0.0	4194302
3	11100000	6	255.224.0.0	2097150
4	11110000	14	255.240.0.0	1048574
5	11111000	30	255.248.0.0	524286
6	11111100	62	255.252.0.0	262142
7	11111110	126	255.254.0.0	131070
8	11111111	254	255.255.0.0	65534

- מספר הביטים המכוסים הוא המספר של הביטים שנשאר לטובת ה- Subnet ID.
- ה- Bit Pattern הוא המבנה של האוקטטה השניה, זו שעליה "רבים" ה- Subnet ID וה- Host ID.
- אתם לא צריכים לזכור בדיוק את הכמויות של ה- Host ID בכל תת-רשת. רק בקירוב.

כתובת מ- Class C:

בחלוקה ל- Subnets של רשת מ- Class C המצב משתנה במקצת. יש כאן כמה בעיות שצריך לתת את הדעת עליהן, ולכן דין חלוקת Class C שונה מחלוקת Classes אחרים, למרות שהרעיון נשאר זהה.

ניקח כתובת כמו 192.80.55.0
לפני (מתחת - ה- Subnet Mask המקורי):



Network ID			Host ID
192	80	55	X

255	255	255	0
-----	-----	-----	---

יש לנו רק רשת אחת עם 2 בחזקת 8 אפשרויות ל- Host ID שונות (אוקטטה אחת שווה 8 ביטים), או 256 אפשרויות, אבל אסור לשכוח להוריד 2, לכן בעצם 254 אפשרויות שונות.

עכשיו "נגנוב" אוקטטה אחת לטובת ה- Subnet ID:

Network ID			Subnet ID
192	80	55	X

255	255	255	255
-----	-----	-----	-----

מה קרה? לא נשארו לנו כלל כל ביטים לצורך ה- Host ID. כל הביטים של ה- Host ID משמשים עכשיו ל- Subnet ID, וברור שמצב כזה הוא לא מצב תקין. לכן אי אפשר לקיים רשת מ- Class C עם Subnet Mask של 255.255.255.255, ותזכרו את זה.

מה שאנחנו יכולים לעשות זה לקחת פחות מ- 8 ביטים בשביל ה- Subnet ID, ולהשאיר כמה ביטים עבור ה- Host ID.

אם "נגנוב" את הביט הכי ימני מתוך ה- Subnet ID ונתן אותו במתנה ל- Host ID, מה זה יתן לנו? בואו נראה:

-----Subnet ID-----										Hst-	
255	255	255	1	1	1	1	1	1	1	0	0

עכשיו יש לנו רק 7 ביטים ל- Subnet ID, וביט אחד ל- Host ID. ניקח את המספר הבינארי 11111110 ונהפוך לעשרוני. כמה יצא? 254 (בדוק!). לכן, ה- Subnet Mask החדש יהיה 255.255.254.0

255	255	255	254
-----	-----	-----	-----

כמה תת-רשתות יכולות להיות לנו אם נשתמש בכזה Subnet Mask? בואו נראה: 2 בחזקת מספר הביטים שנשארו, כלומר 2 בחזקת 7 שווה 128. 128 פחות 2 שווה 126, ואכן יש לנו 126 תת-רשתות אפשריות.

עכשיו צריך לבדוק כמה כתובות אפשריות של Host ID יש לנו: עכשיו יש לנו ביט אחד לשחק איתו. לכן, 2 בחזקת 1 שווה 2. האם יש לנו 2 כתובות שונות? לא. למה לא? כיוון שאסור לנו לשכוח שאסור להשתמש בכתובת שהיא הכל 0, וכתובת שהיא הכל 1. לכן, 2 פחות 2 שווה 0 (אפס), ולכן אי אפשר לחלק רשת מ- Class C ל- 128 תת-רשתות שונות. תזכרו את זה.

חשוב לזכור:

כלומר עד עכשיו ראינו שלכתובת מ- Class C אי אפשר לתת Subnet Mask של 255.255.255.255, וגם לא 255.255.255.254, לכן בעצם אי אפשר לחלק כתובת רשת מ- Class C ל- 254 או 126 תת-רשתות כמו בשאר ה- Classes. חשוב לזכור את הנקודה הזו.

אם אי אפשר לגנוב ביט אחד מה- Subnet ID, בואו ננסה לגנוב 2 ביטים:

אז בואו ניקח עוד ביט אחד וניתן אותו מתנה ל- Host ID:

-----Subnet ID-----										--Host ID--	
255	255	255	1	1	1	1	1	1	1	0	0

עכשיו יש לנו רק 6 ביטים ל- Subnet ID, ו- 2 ביטים ל- Host ID. ניקח את המספר הבינארי 11111100 ונהפוך לעשרוני. כמה יצא? 252 (בדוק!). לכן, ה- Subnet Mask החדש יהיה 255.255.252.0

255	255	255	252
-----	-----	-----	-----

כמה תת-רשתות יכולות להיות לנו אם נשתמש בכזה Subnet Mask? בואו נראה: 2 בחזקת מספר הביטים שנשארו, כלומר 2 בחזקת 6 שווה 64. 64 פחות 2 שווה 62, ואכן יש לנו 62 תת-רשתות אפשריות.

עכשיו צריך לבדוק כמה כתובות אפשריות של Host ID יש: עכשיו יש לנו 2 ביטים לשחק איתם. לכן, 2 בחזקת 2 שווה 4. 4 פחות 2 שווה 2, ואכן יש לנו 2 Host ID שונים בכל תת-רשת אפשרית. 62 תת-רשתות, כשלכל אחת 2 מחשבים. קצת דבילי, לא? אבל אולי דווקא זה מה שהלקוח צריך? נמשיך. נניח שלא מספיקים לו 2 מחשבים בכל תת-רשת. הוא צריך יותר. אז בואו ניקח עוד ביט אחד וניתן אותו מתנה ל- Host ID:

-----Subnet ID-----							-----Host ID-----			
255	255	255	1	1	1	1	1	0	0	0

עכשיו יש לנו רק 5 ביטים ל- Subnet ID, ו- 3 ביטים ל- Host ID. ניקח את המספר הבינארי 11111000 ונהפוך לעשרוני. כמה יצא? 248 (בדוק!). לכן, ה- Subnet Mask החדש יהיה 255.255.248.0

255	255	255	248
-----	-----	-----	-----

כמה תת-רשתות יכולות להיות לנו אם נשתמש בכזה Subnet Mask? בואו נראה: 2 בחזקת מספר הביטים שנשארו, כלומר 2 בחזקת 5 שווה 32. 32 פחות 2 שווה 30, ואכן יש לנו 30 תת-רשתות אפשריות.

עכשיו צריך לבדוק כמה כתובות אפשריות של Host ID יש לנו: עכשיו יש לנו 3 ביטים לשחק איתם. לכן, 2 בחזקת 3 שווה 8. 8 פחות 2 שווה 6, ואכן יש לנו 6 Host ID שונים בכל תת-רשת אפשרית.

לא מספיקים 6 מחשבים בכל תת-רשת. רוצה עוד. נמשיך. ניקח עוד ביט אחד וניתן אותו מתנה ל- Host ID:

-----Subnet ID-----							-----Host ID-----			
255	255	255	1	1	1	1	0	0	0	0

עכשיו יש לנו רק 4 ביטים ל- Subnet ID, ו- 4 ביטים ל- Host ID.

ניקח את המספר הבינארי 11110000 ונהפוך לעשרוני. כמה יצא? 240 (בדוק!). לכן, ה- Subnet Mask החדש יהיה 255.255.240.0

255	255	255	240
-----	-----	-----	-----

כמה תת-רשתות יכולות להיות לנו אם נשתמש בכזה Subnet Mask? בואו נראה: 2 בחזקת מספר הביטים שנשארו, כלומר 2 בחזקת 4 שווה 16. 16 פחות 2 שווה 14, ואכן יש לנו 14 תת-רשתות אפשריות.

עכשיו צריך לבדוק כמה כתובות אפשריות של Host ID יש לנו: עכשיו יש לנו 4 ביטים לשחק איתם. לכן, 2 בחזקת 4 שווה 16. 16 פחות 2 שווה 14 ואכן יש לנו 14 Host ID שונים בכל תת-רשת אפשרית.

נוצר מצב בו יש לנו עד 14 תת-רשתות כשבכל אחת מהם יש מקסימום 14 מחשבים. זה יכול להיות נוח לחברה קטנה עם כמה מחלקות קטנות.

אם זה לא מספיק, ניקח עוד ביט אחד וניתן אותו מתנה ל- Host ID:

-----Subnet ID-----						-----Host ID-----				
255	255	255	1	1	1	0	0	0	0	0

עכשיו יש לנו רק 3 ביטים ל- Subnet ID, ו- 5 ביטים ל- Host ID.

ניקח את המספר הבינארי 11100000 ונהפוך לעשרוני. כמה יצא? 224 (בדוק!). לכן, ה- Subnet Mask החדש יהיה 255.255.224.0

255	255	255	224
-----	-----	-----	-----

כמה תת-רשתות יכולות להיות לנו אם נשתמש בכזה Subnet Mask? בואו נראה: 2 בחזקת מספר הביטים שנשארו, כלומר 2 בחזקת 3 שווה 8. 8 פחות 2 שווה 6, ואכן יש לנו 6 תת-רשתות אפשריות.

עכשיו צריך לבדוק כמה כתובות אפשריות של Host ID יש לנו: עכשיו יש לנו 5 ביטים לשחק איתם. לכן, 2 בחזקת 5 שווה 32. 32 פחות 2 שווה 30 ואכן יש לנו 30 Host ID שונים בכל תת-רשת אפשרית.

6 תת-רשתות עם עד 30 מחשבים בכל תת-רשת. לא מספיק? רוצים יותר מחשבים? ניקח עוד ביט אחד וניתן אותו מתנה ל- Host ID:

-----Sub-----					-----Host ID-----					
255	255	255	1	1	0	0	0	0	0	0

עכשיו יש לנו רק 2 ביטים ל- Subnet ID, ו- 6 ביטים ל- Host ID.

ניקח את המספר הבינארי 11000000 ונהפוך לעשרוני. כמה יצא? 192 (בדוק!). לכן, ה- Subnet Mask החדש יהיה 255.255.192.0

255	255	255	192
-----	-----	-----	-----

כמה תת-רשתות יכולות להיות לנו אם נשתמש בכזה Subnet Mask? בואו נראה: 2 בחזקת מספר הביטים שנשארו, כלומר 2 בחזקת 2 שווה 4. 4 פחות 2 שווה 2, ואכן יש לנו 2 תת-רשתות אפשריות.

עכשיו צריך לבדוק כמה כתובות אפשריות של Host ID יש לנו: עכשיו יש לנו 6 ביטים לשחק איתם. לכן, 2 בחזקת 6 שווה 64. 64 פחות 2 שווה 62 ואכן יש לנו 62 Host ID שונים בכל תת-רשת אפשרית.

יש לי כעת 2 תת-רשתות, כשבכל אחת יש מקסימום 62 מחשבים. סה"כ 124 מחשבים המחולקים ל- 2 קבוצות, במקום הרשת המקורית (להזכירכם, מדובר ב-Class C) שבה היו עד 254 מחשבים בקבוצה אחת. יש כאן ביזבוז של כתובות IP, אבל יתכן וזה מה שהלקוח רוצה: שני סניפים בני לא יותר מ- 62 מחשבים בכל סניף, והוא מוכן לשלם על התענוג הזה כאילו הוא משתמש ב- 254 כתובות, ולא רק 124.

האם לדעתכם אפשר לגנוב עוד ביט אחד? לא. לא יודעים למה? סימן שיש לנו בעיה חמורה, ואני מציע אחת משתי דרכים: לכו לישון, לשטוף פנים או לנגב קצת חומס, או לחלופין לכו וחפשו עבודה בנגרות ©.

טבלה מסכמת לכתובת מ-Class C:

מס' הביטים המכוסים	Bit Pattern	כמה Subnets יש לנו?	מהו ה- Subnet Mask?	כמה Host ID לכל תת-רשת?
2	11000000	2	255.255.255.192	62
3	11100000	6	255.255.255.224	30
4	11110000	14	255.255.255.240	14
5	11111000	30	255.255.255.248	6
6	11111100	62	255.255.255.252	2
7	11111110	126	255.255.255.254	-
8	11111111	254	255.255.255.255	-

- מספר הביטים המכוסים הוא המספר של הביטים שנשארו לטובת ה- Subnet ID.
- ה- Bit Pattern הוא המבנה של האוקטטה הרביעית, זו שעליה "רבים" ה- Subnet ID וה- Host ID.

5. ניקח את המספר הבינארי 11000000 ונהפוך לעשרוני. לכן, ה- Subnet Mask החדש יהיה 255.255.192.0

255	255	192	0
-----	-----	-----	---

6. כמה תת-רשתות יכולות להיות לנו אם נשתמש בכזה Subnet Mask? בואו נראה: 2 בחזקת מספר הביטים שנלקחו, כלומר 2 בחזקת 2 שווה 4. 4 פחות 2 שווה 4, ואכן יש לנו 2 תת-רשתות אפשריות (זוכרים למה פחות 2? כי אי-אפשר הכל 0 ואי-אפשר הכל 1).
7. כמה כתובות אפשריות של Host ID יש לנו? יש כאן 14 ביטים לשחק איתם. לכן, 2 בחזקת 14 שווה 16384. 16384 פחות 2 שווה 16382, ואכן יש לנו 16382 Host ID שונים בכל תת-רשת אפשרית. לא מאמינים? לכו לטבלה, או חפשו עבודה אחרת.
8. מה הכתובת של כל אחת מהרשתות שנוצרו (ה- Network ID)? נתבונן ב- 2 הביטים שגנבנו וננסה להוציא מהם את כל הקומבינציות האפשריות:

- 00
- 01
- 10
- 11

אני רואה 4 אפשרויות שונות (קל לחשב: 2 בחזקת 2 שווה 4) אבל אסור לשכוח את הכל 0 והכל 1, לכן:

- 01
- 10

כלומר בדיוק 2 אפשרויות שונות. נשלב את האפשרויות האלו ב- Network ID המקורי שלי:

191	107	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-----	-----	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

והאפשרות השניה:

191	107	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-----	-----	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

הופך לעשרוני, ומוצא את כתובות הרשתות שלי:

- 191.107.64.0
- 191.107.128.0

9. מה טווחי ה-Hosts בכל אחת מהרשתות הללו? נתבונן ברשת הראשונה:

הקצה האחד של הטווח מתחיל כאשר ה-Host ID הוא הכי נמוך, כלומר 0:

191	107	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-----	-----	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

אבל כמו שאמרנו אי-אפשר Host ID שהוא הכל 0, לכן נוסף 1:

191	107	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
-----	-----	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

הקצה השני של הטווח נגמר כאשר ה-Host ID הוא הכי גבוה, כלומר 1:

191	107	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
-----	-----	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

אבל כמו שאמרנו אי-אפשר Host ID שהוא הכל 1, לכן נחסיר 1:

191	107	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
-----	-----	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

מה התרגום של שני הקצוות לעשרוני? זהו טווח הרשת הראשונה.

191.107.64.1

עד

191.107.127.254

בטווח זה אפשר להקצות כתובות IP למחשבים שאמורים להיות חלק מהרשת הראשונה.

מה לגבי הרשת השניה? הקצה האחד של הטווח מתחיל כאשר ה-Host ID הוא הכי נמוך, כלומר 0:

191	107	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-----	-----	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

אבל כמו שאמרנו אי-אפשר Host ID שהוא הכל 0, לכן נוסף 1:

191	107	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
-----	-----	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

הקצה השני של הטווח נגמר כאשר ה-Host ID הוא הכי גבוה, כלומר 1:

191	107	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
-----	-----	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

אבל כמו שאמרנו אי-אפשר Host ID שהוא הכל 1, לכן נחסיר 1:

191	107	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
-----	-----	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

מה התרגום של שני הקצוות לעשרוני? זהו טווח הרשת הראשונה.

191.107.128.1

עד

191.107.191.254

בטווח זה אפשר להקצות כתובות IP למחשבים שאמורים להיות חלק מהרשת השניה.

נניח שאני רוצה לחלק את אותה רשת מקורית, אבל הפעם ל-4 תת-רשתות. שוב פעם, עם טבלה הכל היה מסתדר תוך שניה. אבל אם אין לי את הטבלה מול העיניים אני חייב ללכת בדרך הידנית:



1. אני רוצה לחלק רשת גדולה ל-4 תת-רשתות. כתובת הרשת היא 191.107.0.0. ה-Subnet Mask המקורי שלה הוא 255.255.0.0.

Network ID		Host ID	
191	107	X	Y

255	255	0	0
-----	-----	---	---

2. הופך 4 לבינארי:

128	64	32	16	8	4	2	1
0	0	0	0	0	1	0	0

3. בכמה ביטים השתמשתי כדי לייצג 4 בבינארי? ב-3 ביטים.
 4. לכן, כעת נגנוב 3 ביטים מה-Host ID של ה-Subnet Mask המקורי לטובת ה-Subnet ID.

255	255	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-----	-----	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

5. ניקח את המספר הבינארי 11100000 ונהפוך לעשרוני. לכן, ה-Subnet Mask החדש יהיה 255.255.224.0

255	255	224	0
-----	-----	-----	---

6. כמה תת-רשתות יכולות להיות לנו אם נשתמש בכזה Subnet Mask? 2 בחזקת מספר הביטים שנלקחו, כלומר 2 בחזקת 3 שווה 8. 8 פחות 2 שווה 6, ואכן יש לנו 6 תת-רשתות אפשריות.

מה התרגום של שני הקצוות לעשרוני? זהו טווח הרשת השניה.

191.107.64.1

עד

191.107.95.254

מה לגבי הרשת השלישית?

הקצה האחד של הטווח מתחיל כאשר ה- Host ID הוא הכי נמוך, כלומר 0. אבל כמו שאמרנו אי-אפשר Host ID שהוא הכל 0, לכן נוסף 1:

191	107	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
-----	-----	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

הקצה השני של הטווח נגמר כאשר ה- Host ID הוא הכי גבוה, כלומר 1. אבל כמו שאמרנו אי-אפשר Host ID שהוא הכל 1, לכן נחסיר 1:

191	107	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
-----	-----	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

מה התרגום של שני הקצוות לעשרוני? זהו טווח הרשת השלישית.

191.107.96.1

עד

191.107.127.254

מה לגבי הרשת הרביעית?

הקצה האחד של הטווח מתחיל כאשר ה- Host ID הוא הכי נמוך, כלומר 0. אבל כמו שאמרנו אי-אפשר Host ID שהוא הכל 0, לכן נוסף 1:

191	107	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
-----	-----	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

הקצה השני של הטווח נגמר כאשר ה- Host ID הוא הכי גבוה, כלומר 1. אבל כמו שאמרנו אי-אפשר Host ID שהוא הכל 1, לכן נחסיר 1:

191	107	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
-----	-----	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

מה התרגום של שני הקצוות לעשרוני? זהו טווח הרשת הרביעית.

191.107.128.1
עד
191.107.159.254

מה לגבי הרשת החמישית?

הקצה האחד של הטווח מתחיל כאשר ה- Host ID הוא הכי נמוך, כלומר 0. אבל כמו שאמרנו אי-אפשר Host ID שהוא הכל 0, לכן נוסף 1:

191	107	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
-----	-----	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

הקצה השני של הטווח נגמר כאשר ה- Host ID הוא הכי גבוה, כלומר 1. אבל כמו שאמרנו אי-אפשר Host ID שהוא הכל 1, לכן נחסיר 1:

191	107	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
-----	-----	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

מה התרגום של שני הקצוות לעשרוני? זהו טווח הרשת החמישית.

191.107.160.1
עד
191.107.191.254

מה לגבי הרשת השישית?

הקצה האחד של הטווח מתחיל כאשר ה- Host ID הוא הכי נמוך, כלומר 0. אבל כמו שאמרנו אי-אפשר Host ID שהוא הכל 0, לכן נוסף 1:

191	107	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
-----	-----	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

הקצה השני של הטווח נגמר כאשר ה- Host ID הוא הכי גבוה, כלומר 1. אבל כמו שאמרנו אי-אפשר Host ID שהוא הכל 1, לכן נחסיר 1:

191	107	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0
-----	-----	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

מה התרגום של שני הקצוות לעשרוני? זהו טווח הרשת השישית.

191.107.192.1
עד
191.107.223.254

הדרך המהירה לחישובי תת-רשתות וטווחים

נסכם את הממצאים שלנו בטבלה קטנה:

טווח	כתובת תת-רשת	1	Subnet Mask	כתובת רשת מקורית
191.107.32.1 עד 191.107.63.254	191.107.32.0	1	255.255.224.0	191.107.0.0
191.107.64.1 עד 191.107.95.254	191.107.64.0	2		
191.107.96.1 עד 191.107.127.254	191.107.96.0	3		
191.107.128.1 עד 191.107.159.254	191.107.128.0	4		
191.107.160.1 עד 191.107.191.254	191.107.160.0	5		
191.107.192.1 עד 191.107.223.254	191.107.192.0	6		

תגידו, כמה ביטים גנבנו לצורך ה- Subnet ID? לא זוכרים? לכו לתחילת הדוגמא. **3 ביטים**.

אם נכניס את הביטים האלה לטבלה בינארית, נראה תופעה מעניינת:

128	64	32	16	8	4	2	1
1	1	1	0	0	0	0	0

מהו הערך של **הביט הימני** ביותר (מאלה שגנבנו)? 32.

128	64	32	16	8	4	2	1
0	0	1	0	0	0	0	0

חיזרו לטבלה למעלה וראו את כתובות תת-הרשת שנוצרו:

191.107.32.0
191.107.64.0
191.107.96.0
191.107.128.0
191.107.160.0
191.107.192.0

שימו לב באיזה קפיצות הרשתות עולות. יש כאן מגמה ברורה:

191.107.32.0
191.107.64.0
191.107.96.0
191.107.128.0
191.107.160.0
191.107.192.0

**חוק: כתובת הרשת הראשונה מתחילה בדיוק
באותו מספר, כמו הערך של הביט הימני ביותר
מבין הביטים ה"גנובים".**

**כך גם הדילוגים בין הכתובות עולים באותו
ערך.**

**ועד לאיזה ערך מדלגים? עד לערך של ה-
Subnet Mask, פחות הערך של אותו ביט ימני
גנוב.**

191.107.32.0
191.107.64.0
191.107.96.0
191.107.128.0
191.107.160.0
191.107.192.0

ה- 255.255.224.0 Subnet Mask

למי שנפל האסימון, כל הכבוד. למי שגילה את זה לבד, ברכות. ולמי שלא יודע על מה אני מדבר, בבקשה לחזור לתחילת הדוגמאות החלק 2 ולקרוא הכל מהתחלה. תודה שטסתם ארקיע. אגב, אופציית הנגרות עדין פתוחה בפניכם...

ומה לגבי הטווחים של ה- Host ID בכל תת-רשת? שימו לב, גם כאן יש היגיון:

רשת ראשונה 191.107.32.0

191.107.32.1

עד

191.107.63.254

רשת שניה 191.107.64.0

191.107.64.1

עד

191.107.95.254

רשת שלישית 191.107.96.0

191.107.96.1

עד

191.107.127.254

וכו...

חוק: הטווח מתחיל בכתובת הרשת פלוס 1, ונגמר בכתובת הרשת הבאה בתור מינוס 2!!!

רשת ראשונה 191.107.32.0

191.107.32.1

עד

191.107.63.254

רשת שניה 191.107.64.0

191.107.64.1

עד

191.107.95.254

רשת שלישית 191.107.96.0

191.107.96.1

עד

191.107.127.254

רשת רביעית 191.107.128.0

למה פלוס 1? כי אם זה לא היה כך היינו מקבלים כתובת של מחשב כמו 191.107.64.0, וכבר סגרנו על זה שאסור כתובת Host שמסתיימת ב-0 (הכל 0).
למה פחות 2? כי אם זה היה כתובת הרשת הבאה פחות 1, הינו מקבלים כתובת של מחשב כמו 191.107.127.255, והרי סיכמנו כבר בהתחלה שאסור בהחלט שתהיה כתובת של Host שמסתיימת ב-255 (הכל 1).

בדרך זו אפשר למצוא בשניות את כל הכתובות וטווחי הרשתות השונות.



נניח שאני רוצה לחלק את אותה רשת מקורית, אבל הפעם ל- 50 תת-רשתות. שוב פעם, עם טבלה הכל היה מסתדר תוך שניה. אבל אם אין לי את הטבלה מול העיניים אני חייב ללכת בדרך הידנית:

1. אני רוצה לחלק רשת גדולה ל- 50 תת-רשתות. כתובת הרשת היא 191.107.0.0. ה- Subnet Mask המקורי שלה הוא 255.255.0.0.

Network ID		Host ID	
191	107	X	Y

255	255	0	0
-----	-----	---	---

1. הופך 50 לבינארי:

128	64	32	16	8	4	2	1
0	0	1	1	0	0	1	0

2. בכמה ביטים השתמשתי כדי לייצג 50 בבינארי? ב- 6 ביטים.

3. לכן, כעת נגנוב 6 ביטים מה- Host ID של ה- Subnet Mask המקורי לטובת ה- Subnet ID.

255	255	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
-----	-----	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

4. ניקח את המספר הבינארי 11111100 ונהפוך לעשרוני. לכן, ה- Subnet Mask החדש יהיה 255.255.252.0

255	255	252	0
-----	-----	-----	---

5. כמה תת-רשתות יכולות להיות לנו אם נשתמש בכוה Subnet Mask? 2 בחזקת מספר הביטים שנלקחו, כלומר 2 בחזקת 6 שווה 64. 64 פחות 2 שווה 62, ואכן יש לנו 62 תת-רשתות אפשריות.

6. כמה כתובות אפשריות של Host ID יש לנו? יש כאן 10 ביטים לשחק איתם. לכן, 2 בחזקת 10 שווה 1024. 1024 פחות 2 שווה 1022, ואכן יש לנו 1022 Host ID שונים בכל תת-רשת אפשרית.

7. מה הכתובת של כל אחת מהרשתות שנוצרו (ה- Network ID)? נתבונן ב- 6 הביטים שגנבנו וננסה להוציא מהם את כל הקומבינציות האפשריות. עכשיו זה יהיה מטופש להתחיל לכתוב את כל האפשרויות השונות. לכן, נקצר כמו שלמדנו קודם.

2 בחזקת 6 שווה 64. פחות 2 שווה 62, כלומר סה"כ 62 תת-רשתות שונות. רצינו 50, אבל קיבלנו 62. ניקח את האפשרות הראשונה:

191	107	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
-----	-----	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

מה הערך של הביט הימני ביותר שגנבנו?

128	64	32	16	8	4	2	1
0	0	0	0	0	1	0	0

לכן, לפי מה שלמדנו קודם, הכתובות של הרשתות יהיו (קפיצות של 4):

191.107.4.0

191.107.8.0

191.107.12.0

191.107.16.0

...אין לי כוח לכתוב את כולם...

הרשתות האחרונות:

191.107.240.0

191.107.244.0

191.107.248.0

ה- Subnet Mask

255.255.252.0

לכן אני נזהר לא להגיע בקפיצות שלי אל ה- Subnet Mask. סה"כ 62 תת-רשתות שונות (למי שמטיל דופי, נא לכתוב את כולם ולעמוד בפניה!).

מה הטווחים של כל תת-רשת? נלך לפי מה שלמדנו קודם. הטווח מתחיל בכתובת הרשת פלוס 1, ונגמר בכתובת הרשת הבאה בתור מינוס 2:

רשת ראשונה 191.107.4.0

191.107.4.1

עד

191.107.7.254

רשת שניה 191.107.8.0

191.107.8.1

עד

191.107.11.254

רשת שלישית 191.107.12.0

191.107.12.1

עד

191.107.15.254

רשת רביעית 191.107.16.0

וכו...

רשת אחרונה 191.107.248.0

191.107.248.1

עד

191.107.251.254

אני רוצה להראות לכם דוגמא נוספת, אבל הפעם עם רשת מ-Class C, כיוון שכאן החישובים אמנם עובדים בדיוק באותה דרך, אבל למראית עין יש כאן קושי מסויים.



1. אני רוצה לחלק רשת גדולה ל-20 תת-רשתות. כתובת הרשת היא 199.80.75.0. ה-Subnet Mask המקורי שלה הוא 255.255.255.0.

Network ID			Host ID
199	80	75	Y
255	255	255	0

2. הופך 20 לבינארי:

128	64	32	16	8	4	2	1
0	0	0	1	0	1	0	0

3. בכמה ביטים השתמשתי כדי לייצג 20 בבינארי? ב-5 ביטים.

4. לכן, כעת נגנוב 5 ביטים מה-Host ID של ה-Subnet Mask המקורי לטובת ה-Subnet ID.

255	255	255	1	1	1	1	1	0	0	0
-----	-----	-----	---	---	---	---	---	---	---	---

5. ניקח את המספר הבינארי 11111000 ונהפוך לעשרוני. לכן, ה-Subnet Mask החדש יהיה 255.255.248.0

255	255	255	248
-----	-----	-----	-----

6. כמה תת-רשתות יכולות להיות לנו אם נשתמש בכזה Subnet Mask? 2 בחזקת מספר הביטים שנלקחו, כלומר 2 בחזקת 5 שווה 32. 32 פחות 2 שווה 30, ואכן יש לנו 30 תת-רשתות אפשריות.

7. כמה כתובות אפשריות של Host ID יש לנו? יש כאן 3 ביטים לשחק איתם. לכן, 2 בחזקת 3

שווה 8. 8 פחות 2 שווה 6, ואכן יש לנו 6 Host ID שונים בכל תת-רשת אפשרית.

8. מה הכתובת של כל אחת מהרשתות שנוצרו (ה- Network ID)? נתבונן ב-5 הביטים שגנבנו וננסה להוציא מהם את כל הקומבינציות האפשריות. עכשיו זה יהיה מטופש להתחיל לכתוב את כל האפשרויות השונות. לכן, נקצר כמו שלמדנו קודם.

2 בחזקת 5 שווה 32. 32 פחות 2 שווה 30, כלומר סה"כ 30 תת-רשתות שונות. רצינו 20, אבל קיבלנו 30. ניקח את האפשרות הראשונה:

255	255	255	0	0	0	0	1	0	0	0
-----	-----	-----	---	---	---	---	---	---	---	---

מה הערך של הביט הימני ביותר שגנבנו?

128	64	32	16	8	4	2	1
0	0	0	0	1	0	0	0

לכן, לפי מה שלמדנו קודם, הכתובות של הרשתות יהיו (קפיצות של 8):

199.80.75.8

199.80.75.16

199.80.75.24

199.80.75.32

...אין לי כוח לכתוב את כולם...

הרשתות האחרונות:

199.80.75.224

199.80.75.232

199.80.75.240

ה- Subnet Mask

255.255.248.0

לכן אני נזהר לא להגיע בקפיצות שלי אל ה- Subnet Mask. סה"כ 30 תת-רשתות שונות.

מה הטווחים של כל תת-רשת? נלך לפי מה שלמדנו קודם. הטווח מתחיל בכתובת הרשת פלוס 1, ונגמר בכתובת הרשת הבאה בתור מינוס 2. כאן צריכים להיזהר, כי קשה לנו להבחין במבט ראשון את ההתחלה והסוף של כל רשת. אין לנו כאן אוקטטה אחרונה עם 0 או 255 שנוכל לזהות כבעייתית, ולכן חייבים להיצמד לחוק הפלוס 1 מינוס 2 שלמדנו קודם:

רשת ראשונה **199.80.75.8**

199.80.75.9

עד

199.80.75.14

רשת שניה **199.80.75.16**

199.80.75.17

עד

199.80.75.22

רשת שלישית **199.80.75.24**

199.80.75.25

עד

199.80.75.30

רשת רביעית 199.80.75.32

וכו...

רשת אחרונה 199.80.75.240

199.80.75.241

עד

199.80.75.246

סה"כ 30 תת-רשתות, כשכל תת-רשת מכילה 6 Host ID, כמו שמצאנו באמצעות הטבלה. שוב, חשוב מאוד לשים לב ל- Subnetting ברשתות של Class C, כיוון שכאמור קשה מאוד לזהות במבט ראשון התחלה וסיום של תת-רשת. לזכור את זה טוב טוב.

לסיכום:

דברים שחשוב לזכור:

לזכור את הפרמטרים שחייבים להגדיר כדי שמחשב יוכל לעבוד ב-TCP/IP.

לזכור את ההבדלים בין כתובות ציבוריות לכתובות פרטיות.

לזכור את ההבדלים והיתרונות של חלוקה אוטומטית וחלוקה ידנית.

אין כזה דבר כתובת שמורכבת מהכל 0 או הכל 1, לכן תמיד נזכור להוריד 2 מהתוצאה הסופית.

אין כתובת רשת שמתחילה ב-127.

לדעת לזהות בעיות אופייניות בחלוקת כתובות IP.

לזכור את מבנה האוקטטה הראשונה בכל אחד מה-Classes השונים.

לזכור איך בודקים כמה Network ID יש לנו בכל Class.

לזכור איך יודעים כמה Host ID יש לנו בכל Class.

לזכור את ה-Default Subnet Mask עבור כל אחד מה-Classes.

לדעת לפתור שאלות ששאלות איזה Class צריך אם אני צריך ככה וככה מחשבים.

לדעת לזהות לפי ה-Classes איזו כתובת מתאימה לאיזו רשת, והאם יש בעיות כתוצאה מחוסר התאמה.

לדעת מדוע צריך את פעולת ה-Subnetting. למה זה טוב.

לדעת מהי ההגדרה של סגמנט.

לדעת מהם השיקולים במתן כתובות רשת – אחת עבור כל Subnet ואחת עבור כל קשר WAN.

לדעת למי צריך לתת כתובות IP – אחד עבור כל Device ברשת, ואחד עבור כל רגל של נתב.

לדעת לחשב כמות Subnets רצוייה, לפי צרכים עכשוויים, ולפי דרישות עתידיות.

לזכור את הקפיצות במספר ה-Subnets – 2, 6, 14, 30, 62, 126, 254.

לזכור את הקפיצות ב-Subnet Masks – 192, 224, 240, 248, 252, 254, 255.

לדעת (או לזכור) איך לחשב את מספר ה-Hosts בכל Subnet.

להבין את הדרך המקוצרת לחישוב ה-Subnet ID לאחר החלוקה (לפי הביט הימני ביותר).

לדעת איך לחשב את טווחי כל אחד מה-Subnets (הקצה התחתון +1 עד הקצה העליון – 2).

להבין את הבעיות האפשריות כתוצאה מחלוקה לא נכונה של כתובות IP בתוך הטווחים הללו.

לזהות חריגות מהטווחים (למשל המחשב נמצא בטווח התקין, אבל ה-Default Gateway שלו לא).

קושיות, שאלות, סייגים, טענות, מענות, מין, סעד וכלכלה-

orgad@ginegar.net

להתראות בשמחות- אורגד