**מדריך למשתמש - SIMPLE NETWORK MANAGEMENT PROTOCOL**

SNMP הוא פרוטוקול יישום המשתמש ב-UDP (הפרוטוקול הבסיסי) לשידור מאובטח. הוא משמש גם לניטור הרשת, זיהוי תקלות רשת והגדרת התקנים מרוחקים.

*רכיבי snmp :*   
השתמשנו בשני רכיבי SNMP במודל המוצע שלנו. הם מנהל SNMP וסוכן SNMP

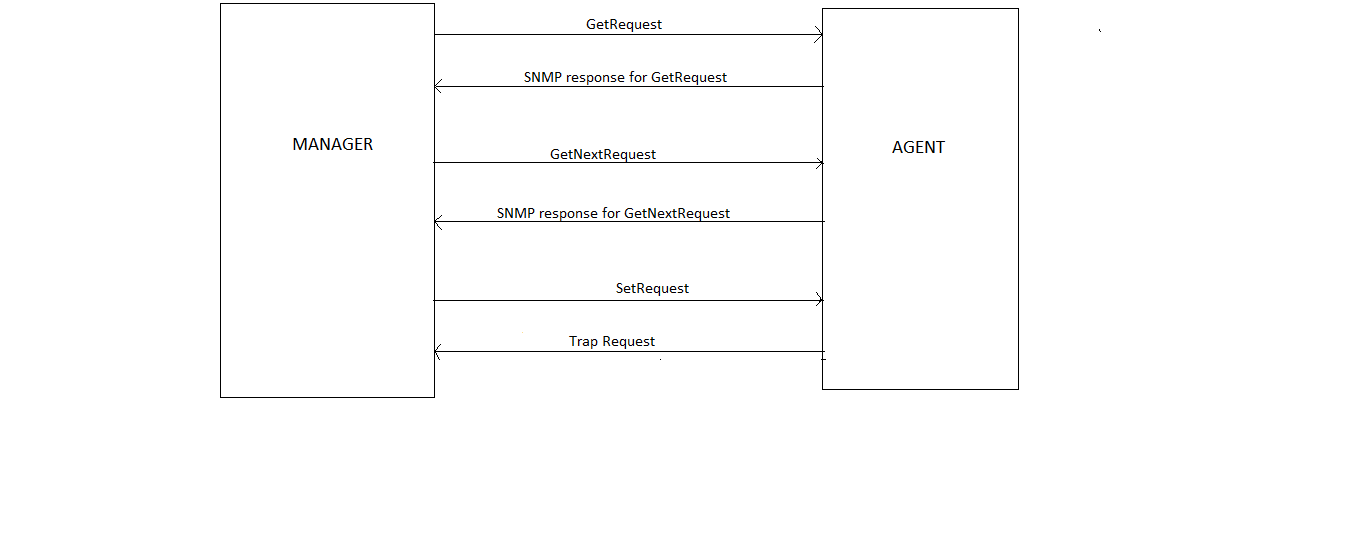
* SNMP manager – משמש כתחנת ניהול הרשת NMS . זוהי מערכת מרכזית שמשמשת לניטור הרשת (monitoring) .
* SNMP Agent – מנהל את המכשירים (devices) מכשירים מנוהלים יכולים להיות switch / router/pc ועוד.

הודעות SNMP : במודל snmp גרסא V1 ע"פ ההרחבה שבחרנו , ישנם הודעות ספציפיות שבהן הוא משתמש ,  
 השתמשנו בתוכנית בהודעות הבאות :

* **GetRequest –** הודעה זו משמשת לקבלת נתונים מסוכן SNMPAgent .   
  ראשית, snmpManager שולח הודעת getRequest כדי לבקש נתונים מסוכן SNMP ,   
  בעקבות כך סוכן SNMP מגיב עם הערך המבוקש באמצעות הודעת Response .
* **GetNextRequest** – הודעה זו משמשת לזיהוי נתונים זמינים בסוכן SNMP .   
  snmpManager יכול לבקש נתונים ברציפות עד שהנתונים יהיו זמינים . ע"י הודעה זו snmpManager יוכל לדעת את כל הנתונים הזמינים על הsnmpAgent .
* **SetRequest** – משמש את מנהל הSNMP כדי להגדיר ערך value של OID בסוכן .
* **Response** – Response היא ההודעה שנשלחת למנהל snmpManager מהסוכן SNMPAgent .   
  אם זה בחזרה להודעת getRequest היא תכיל את הנתונים המבוקשים בהודעה .   
  אם זה בחזרה להודעת setRequest היא תכיל את הערך החדש שהוגדר , כאישור לכך שאכן הערך הוגדר .

**MIB : Management Information Base**  
זהו קובץ טקסט המפרט את הנתונים והערכים המשמשים לחלק מסוים של התקני SNMP .   
בתוך הMIB ישנם OID שונים שכל אחד מהם מכיל ערך ספציפי עבור ההתקן .

**אז איך התוכנית שלנו עובדת ?**השתמשנו בכמה מחלקות , SNMP Manager ו-SNMP Agent ו-MIBManagerבצד הסוכן SNMPAgent .   
 בעזרת מחלקות אלו נאחסן את הOIDs של ההתקנים , והערכים התואמים לOID.

על מנת להמחיש את הפעולות של SNMP אחסנו באופן ידני כמה ערכי OID בMIBManager .   
אנחנו שולחים רצף הודעות מהSNMPManager לSNMPAgent ובחזרה .   
  
בתרשים שלהלן נדגים איך התוכנית שלנו עובדת :   


SNMPManager יאזין לפורט מספר 161 תמיד .   
SNMPAgent יאזין לשני הפורטים 161 או 162 , זה תלוי במה שנחליט בקובץ הקונפיגורציה .  **תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי**

כל שליחה של הודעה SNMPGet / SNMPGetNext / SNMPSet מהמנהל לסוכן , עוקבת אחרי מבנה הpacket הזה :

* ASN1 header for overall SNMP Message
* SNMP Message Header with including version and Community String
* SNMP PDU Message Structure with indication of Request Type
* Variable Binding List with ObjectID and values(can hold N number of OIDs in one Request.)

עבור כל שליחת SNMP Get Request ממנהל SNMP לסוכן SNMP נקבל הודעת תגובה עם OID וערכים משויכים.

עם קבלת הודעות SNMP Get Request על סוכן SNMP ,הוא יבדוק שcommunity string תקפה, אחרת הוא ימחק את ההודעה. כמו כן, הוא יבדוק אם ל-OID שהתקבל יש כניסה תקינה במסד הנתונים של MIB. אם זה ערך לא ידוע, מנהל SNMP ישלח הודעת תגובה עם ערך שגיאת שם לא קיים.

אם כל ה-OIDs הם ערכים נאותים, הוא ישלח הודעת תגובה עם הערכים המתאימים.

עם קבלת הודעת בקשת ה- SNMP Set, הוא יבדוק אם ל-OID יש ערכים מתאימים, אם הוא מתאים, SNMP ישנה את הערכים בהתאם במסד הנתונים של מנהל MIB.

סוכן SNMP ישמור גם רשימת OIDs ברשימה, כדי שיוכל להגיש בקשה ל-SNMP Get Next Request ויכיר את הערכים העתידיים שלו .

**הוראות הפעלה של התוכנית :**

אם רוצים לעבוד על אותו מחשב :   
קודם כל נדאג שהערכים בקובץ הקונפיגורציה conf/snmp.properties יהיו אלו :

agentport=162

managerport=161

islocalhosttestingenabled=true

אם רוצים לעבוד על מחשבים שונים :

**agentport**=**161  
managerport**=**161  
islocalhosttestingenabled**=**false   
agentaddr**=**10.0.0.9**

agentport - מציין את הפורט לו אנו מאזינים ב-SNMPagent אנו יכולים לשנות בהתאם לצרכים.

Manager Port - מציין את הפורט שבו מנהל SNMPmanager מאזין

Islocalhosttestingenabled - - אם זה על מצב true, זה מציין שגם הסוכן וגם המנהל יפעלו על אותה מחשב. אם לא , על מחשבים שונים .

נפעיל את SNMPAgent ראשון . כעת הוא יאזין לפורט שהחלטנו בקובץ הקונפיגורציה . נקבל את הפלט הבא :

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי  
נפעיל את הSNMPManager ונקבל את הפלט הבא :

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי



כאן נוכל להחליט איזה פעולה אנחנו רוצים לבצע ע"י בחירה ב-1-2-3-4 .

אם נבחר באופציה מס 1 – נבחר בgetRequest , נקבל את הפלט הבא :   


כאן נכניס את כמות הoid שלהם אנחנו רוצים לעשות את פקודת הgetRequest . נניח אם נכתוב 1 נקבל אופציה להכניס OID אחד, אם נכתוב 2 נקבל אופציה להכניס שני OID .   
תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי  
  
כעת נכניס את הoid מהרשימה שלנו :

1.3.4.234.1.8.6   
  
1.3.4.234.1.8.7

1.3.4.234.1.8.8

לדוגמא הכנסנו את הראשון :



נקבל פלט בmanager :

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

ובמקביל נקבל פלט בagent :





הפלט שלעיל מכיל את הvalue של OID שהכנסנו .

אם נבחר באופציה מס 2 – נבחר בgetNextRequest , נקבל את הפלט הבא :

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

כעת נצטרך להכניס כמות OID שעליהם נרצה להפעיל את הפקודה , נבחר ב1 -   
לאחר מכן נצטרך להכניס את הOID שאנחנו רוצים קבל את הערך של הoid הבא אחריו .

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

הפלט שנקבל בmanager -

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

במקביל נקבל פלט בagent של הvalue הבא אחריו –

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי



אם נבחר באופציה מס 3 – נבחר setRequest , נקבל את הפלט הבא :

שוב נצטרך לבחור כמות OID ושוב נצטרך להכניס את הoid .   
השוני כאן הוא שצריך להכניס value חדש שיהיה מקושר לOID.

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

כעת נכניס ערך חדש לvalue - > לדוגמא נכניס800MB

נקבל פלט בmanager :

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי



הפלט שנקבל בAgent :

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי



אם נבחר באופציה מס 4 – נבחר Exit ,   
אופציה זו תוציא אותנו מהloop של הSNMPManager .