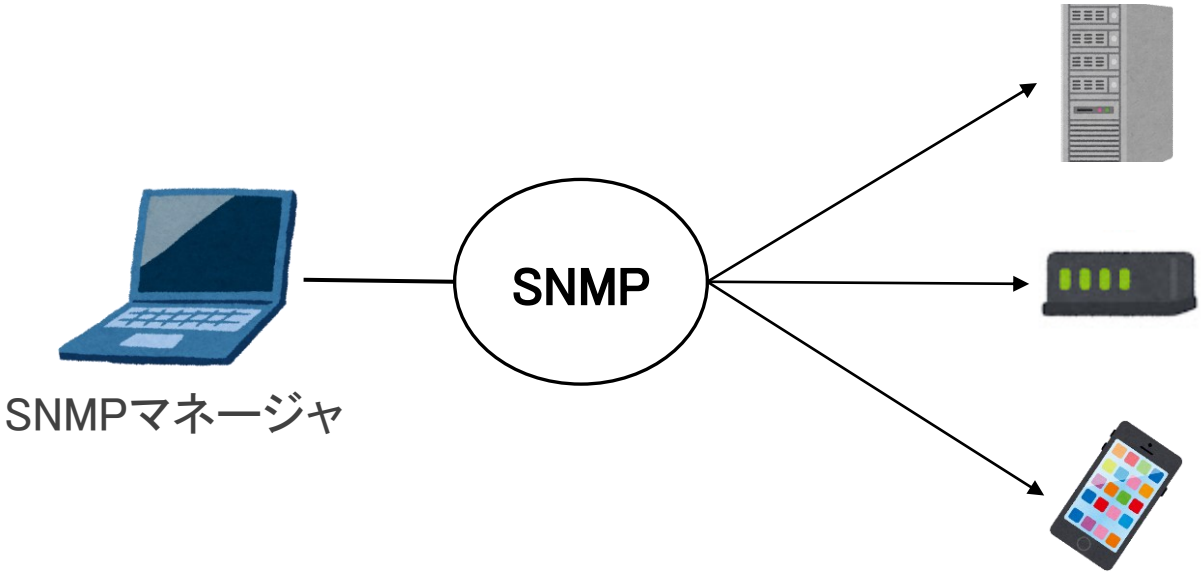
SNMP勉強資料

情報通信ネットワーク研究室
高橋優作

SNMP

- SNMP(Simple Network Management Protocol)
  - ネットワーク機器やハードウェア機器の管理をするためのプロトコル
  - **SNMPマネージャ**で情報の収集，監視を行う
  - **SNMPエージェント**がその対象となる

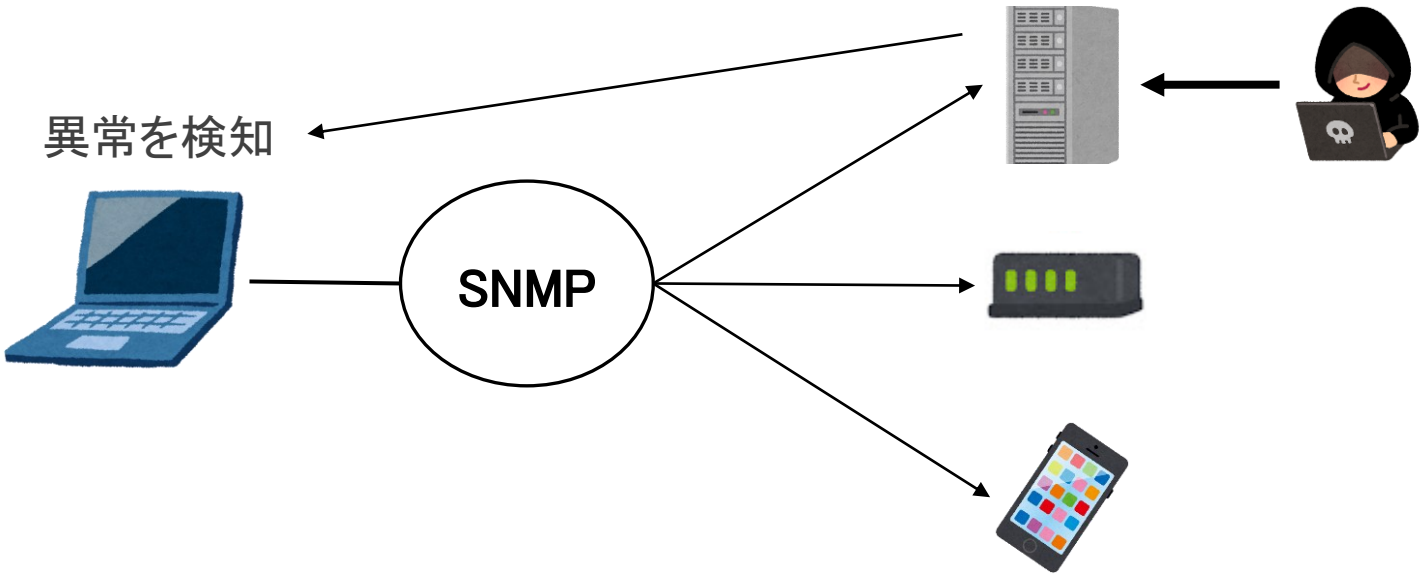


SNMP

- 主なバージョン
  - v1：平文で認証(RFC1157)
  - v2c：平文で認証．SNMPトラップにおける再送確認がある(RFC1901)
  - v3：暗号化可能(RFC2273)

SNMPの必要性

- ネットワークを監視下に置く
  - 1つ1つ手動で機器を確認せず監視を行える
- 速報性
  - 問題発生時，異常を即座に検知できる



SNMPの機能

- SNMPGet
  - マネージャからエージェントに対して情報を取得する
- SNMPSet
  - マネージャからエージェントに対して設定を行う
  - エージェントが持つ値を書き換える
- SNMPTrap
  - エージェントがマネージャにエージェント情報を通知する
  - push型プロトコル
- これら3つの機能はすべてUDP上で動作する(ポート番号は161, 162を使用)

OID(Object ID)

- OID
  - 個々のオブジェクトやリソースを一意に識別するための識別子
  - 階層的なツリー構造をもつ

```
andorssi@J:~$ snmpget -v2c -c public localhost 1.3.6.1.2.1.1.5.0
SNMPv2-MIB::sysName.0 = STRING: J
```

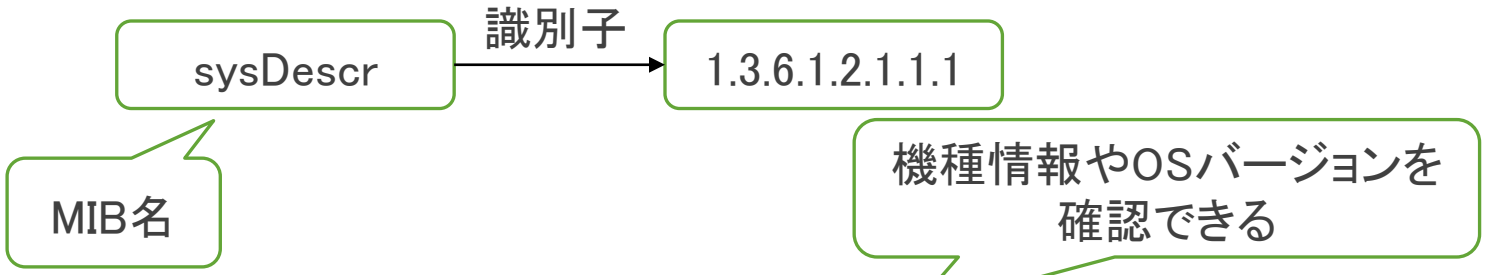
機器のホスト名を示すOID

- OIDの一例

OID	説明
1.3.6.1.2.1.2.2.1.2	インタフェース名(eth0など)
1.3.6.1.2.1.2.2.1.10	受信パケット総バイト数(32bit)
1.3.6.1.2.1.2.2.1.16	送信パケット総バイト数(32bit)

MIB

- MIB(Management Information Base)
  - ネットワーク機器やシステム管理情報をツリー構造で格納するデータベース
  - マネージャやエージェントはMIBを参照して情報の取得を行う
- OIDはMIBで定義された各項目を識別するための番号




```
andorssi@J:~$ snmpwalk -v2c -c public localhost sysDescr
SNMPv2-MIB::sysDescr.0 = STRING: Linux J 6.6.87.2-microsoft-standard-WSL2 #1 SMP PREEMPT_DYNAMIC Thu Jun  5 18:30:46 UTC
 2025 x86_64
andorssi@J:~$ snmpwalk -v2c -c public localhost 1.3.6.1.2.1.1.1
SNMPv2-MIB::sysDescr.0 = STRING: Linux J 6.6.87.2-microsoft-standard-WSL2 #1 SMP PREEMPT_DYNAMIC Thu Jun  5 18:30:46 UTC
 2025 x86_64
```

SNMPをつかえるようにする

- インストール(Linux)
  - $ sudo apt install snmp snmpd
- 設定ファイルの編集
  - $ sudo vi /etc/snmp/snmpd.conf
  - こんな感じのがつらつらと書かれている

```
###############################################################################
#
# snmpd.conf
# An example configuration file for configuring the Net-SNMP agent ('snmpd')
# See snmpd.conf(5) man page for details
#
###############################################################################
# SECTION: System Information Setup
#

# syslocation: The [typically physical] location of the system.
#   Note that setting this value here means that when trying to
#   perform an snmp SET operation to the sysLocation.0 variable will make
#   the agent return the "notWritable" error code.  IE, including
#   this token in the snmpd.conf file will disable write access to
#   the variable.
#   arguments:  location_string
rocommunity public
syslocation "My Location"
sysContact  "My Contact"

# sysservices: The proper value for the sysServices object.
#   arguments:  sysservices_number
sysServices    72
```

```
# agentaddress: The IP address and port number that the agent will listen on.
#   By default the agent listens to any and all traffic from any
#   interface on the default SNMP port (161).  This allows you to
#   specify which address, interface, transport type and port(s) that you
#   want the agent to listen on.  Multiple definitions of this token
#   are concatenated together (using ':'s).
#   arguments: [transport:]port[@interface/address],...

# agentaddress  127.0.0.1,[::1]
agentaddress udp:161,udp6:[::1]:161
```

SNMPを使えるようにする

- rocommunityを設定することで，監視するアドレスの対象を決められる

```
###############################################################################
# SECTION: Access Control Setup
#
#   This section defines who is allowed to talk to your running
#   snmp agent.

# Views
#   arguments viewname included [oid]

#  system + hrSystem groups only
# view   systemonly  included   .1.3.6.1.2.1.1
# view   systemonly  included   .1.3.6.1.2.1.25.1

view all included .1
access notConfigGroup "" any noauth exact all none none

# rocommunity: a SNMPv1/SNMPv2c read-only access community name
#   arguments:  community [default|hostname|network/bits] [oid | -V view]

# Read-only access to everyone to the systemonly view
#rocommunity  public default -V systemonly
#rocommunity6 public default -V systemonly
rocommunity public localhost -V all
rocommunity public 192.168.[redacted] -V all
```

デフォルトだとsystemonlyになっている

SNMPを使えるようにする

- ラズパイ側の設定

yutak@raspberrypi: /

ファイル(F)　編集(E)　タブ(T)　ヘルプ(H)

```
# sysservices: The proper value for the sysServices object.
#   arguments:  sysservices_number
sysServices    72



###############################################################################
# SECTION: Agent Operating Mode
#
#   This section defines how the agent will operate when it
#   is running.

# master: Should the agent operate as a master agent or not.
#   Currently, the only supported master agent type for this token
#   is "agentx".
#
#   arguments: (on|yes|agentx|all|off|no)

master  agentx

# agentaddress: The IP address and port number that the agent will listen on.
#   By default the agent listens to any and all traffic from any
#   interface on the default SNMP port (161).  This allows you to
#   specify which address, interface, transport type and port(s) that you
#   want the agent to listen on.  Multiple definitions of this token
#   are concatenated together (using ':'s).
#   arguments: [transport:]port[@interface/address],...

# agentaddress  127.0.0.1,[::1]
agentaddress udp:161,udp6:[::]:161
```

```
###############################################################################
# SECTION: Access Control Setup
#
#   This section defines who is allowed to talk to your running
#   snmp agent.

# Views
#   arguments viewname included [oid]

#  system + hrSystem groups only
# view   systemonly  included   .1.3.6.1.2.1.1
# view   systemonly  included   .1.3.6.1.2.1.25.1

view all included .1
access notConfigGroup "" any noauth exact all none none

# rocommunity: a SNMPv1/SNMPv2c read-only access community name
#   arguments:  community [default|hostname|network/bits] [oid | -V view]

# Read-only access to everyone to the systemonly view
rocommunity  public localhost -V all
# rocommunity6 public default -V systemonly
rocommunity public 192.168.11.45 -V all

# SNMPv3 doesn't use communities, but users with (optionally) an
# authentication and encryption string. This user needs to be created
# with what they can view with rouser/rwuser lines in this file.
#
# createUser username (MD5|SHA|SHA-512|SHA-384|SHA-256|SHA-224) authpassphrase [DES|AES] [privpassphrase]
# e.g.
# createuser authPrivUser SHA-512 myauthphrase AES myprivphrase
#
# This should be put into /var/lib/snmp/snmpd.conf
#
# rouser: a SNMPv3 read-only access username
#    arguments: username [noauth|auth|priv [OID | -V VIEW [CONTEXT]]]
rouser authPrivUser authpriv -V systemonly

# include a all *.conf files in a directory
includeDir /etc/snmp/snmpd.conf.d
```

snmpd.confの役割①

- SNMPエージェントの設定ファイル
- com2sec
  - セキュリティ定義を行う
  - NAME: 任意の名前
  - SOURCE: 応答する送信元IPアドレスやホスト名(もしくはdefault)
  - COMMUNITY: コミュニティ名

```
# com2sec NAME SOURCE COMMUNITY
com2sec local localhost private
```

snmpd.confの役割②

- group
  - バージョン定義を行う
  - NAME: 任意の名前
  - MODEL: SNMPバージョン
  - SECURITY: セキュリティ名(com2secのNAMEを書く)

```
# group NAME MODEL SECURITY
group Mygroup v2c local
```

- view
  - 取得可能な情報の範囲定義を行う
  - NAME: 任意の名前
  - TYPE: included(指定OID範囲) か excluded(指定OID範囲外)
  - SUBTREE: MIBのオブジェクトID
  - MASK: 16進数の数値を入れる(デフォはff)

```
# group NAME TYPE SEBTREE [MASK]
view view_all included .1
```

snmpd.confの役割③

- access
  - アクセス設定を行う
  - NAME: groupで設定したNAME
  - CONTEXT: SNMPバージョンがv1、v2cの場合は""(空文字)
  - MODEL: any/v1/v2c/usmのいずれか(SNMPバージョン)
  - LEVEL: noauth(v1かv2c)/auth/priv のいずれか
  - PREFX: exact/prefix のいずれか
  - READ: 読み取りに関して事前に定義していたviewのNAME
  - WRITE: 書き込みに関して事前に定義していたviewのNAME/無いなら「none」を設定
  - NOTIFY: 通知に関して事前に定義していたviewのNAME/無いなら「none」を設定

```
# access NAME CONTEXT MODEL LEVEL PREFX READ WRITE NOTIFY
access Mygroup “” any noauth exact view_all none none
```

トラヒック量取得シェルスクリプト

```
#!/bin/bash
OID_IN="IF-MIB::ifInOctets.2"
OID_OUT="IF-MIB::ifOutOctets.2"
OUTPUT_FILE="traffic.csv"

# ヘッダー行を最初に書く（既に存在していればスキップ）
if [ ! -f "$OUTPUT_FILE" ]; then
   echo "timestamp,in_bps,out_bps" > "$OUTPUT_FILE"
fi

while true; do
   in1=$(snmpget -v2c -c public -Oqv localhost $OID_IN)
   out1=$(snmpget -v2c -c public -Oqv localhost $OID_OUT)
   sleep 10
   in2=$(snmpget -v2c -c public -Oqv localhost $OID_IN)
   out2=$(snmpget -v2c -c public -Oqv localhost $OID_OUT)

   in_bps=$(( (in2 - in1) * 8 / 10 ))
   out_bps=$(( (out2 - out1) * 8 / 10 ))

   timestamp=$(date +"%Y-%m-%d %H:%M:%S")

   echo "[$timestamp] 受信: ${in_bps} bps, 送信: ${out_bps} bps"
   echo "$timestamp,$in_bps,$out_bps" >> "$OUTPUT_FILE"
done
```

ifInOctetsはトラヒック量を取得可能
.2はeth0を参照(.1はlo)

public，localhostは指定した名前

10秒間のパケット数の差をとることでトラヒック量を測定

トラヒック量取得シェルスクリプト

- snmpgetコマンド
  - snmpget <バージョン> -c <コミュニティ名> <エージェントIP> <OID>
  - OIDはMIB名を指定してもOK
- ラズパイからPCのトラヒック量を取得する
  - snmpget -v2c -c public -Oqv 192.168.XX.XX IF-MIB::ifOutOctets.4
  - if::Descrで対象IPのインタフェース一覧を確認できる．LAN接続は".4"なので，4番を指定

192.168.XX.XX

```
yutak@raspberrypi:~ $ snmpwalk -v2c -c public 192.168.XX.XX IF-MIB::ifDescr
IF-MIB::ifDescr.1 = STRING: Software Loopback Interface 1.
IF-MIB::ifDescr.2 = STRING: WAN Miniport (IPv6).
IF-MIB::ifDescr.3 = STRING: Microsoft 6to4 Adapter.
IF-MIB::ifDescr.4 = STRING: Intel(R) Wi-Fi 6 AX200 160MHz.
IF-MIB::ifDescr.5 = STRING: WAN Miniport (IKEv2).
IF-MIB::ifDescr.6 = STRING: Norton VPN Wintun Adapter.
IF-MIB::ifDescr.7 = STRING: WAN Miniport (IP).
…
```

トラヒック量取得シェルスクリプト

- 先ほどのshファイルを実行した結果
  - 送受信ともにトラヒックが発生していることが分かる

```
yutak@raspberrypi:~/snmp_traffic $ ./snmp03.sh
[2025-10-07 17:08:01] 受信: 30608 bps, 送信: 23226 bps
[2025-10-07 17:08:12] 受信: 3696833 bps, 送信: 211741 bps
[2025-10-07 17:08:22] 受信: 1682124 bps, 送信: 126476 bps
[2025-10-07 17:08:32] 受信: 1239252 bps, 送信: 108968 bps
[2025-10-07 17:08:43] 受信: 832152 bps, 送信: 124113 bps
[2025-10-07 17:08:53] 受信: 1417117 bps, 送信: 143003 bps
[2025-10-07 17:09:03] 受信: 2143231 bps, 送信: 107391 bps
[2025-10-07 17:09:13] 受信: 2100907 bps, 送信: 73284 bps
[2025-10-07 17:09:24] 受信: 3504570 bps, 送信: 116736 bps
[2025-10-07 17:09:34] 受信: 2466808 bps, 送信: 78729 bps
^C
```