L'Administration de Réseaux

L'éclatement géographique des matériels réseaux (Electroniques actives, postes de travail serveurs) nécessite de mettre en placedes outils pour accéder à distance aux paramètres de configuration et d'exécution de chaque matériel.

La majeure partie des fonctions d'administration doit demeurer transparente vis à vis des utilisateurs. Il en sera de même pour le suivi des flux de données transitant d'un réseau local à un autre sous-ensemble, ainsi que pour l'observation des paquets traversant les commutateurs et les routeurs.

Parmi les informations remontant jusqu'à l'administrateur, on note la configuration du réseau, les mesures de charges (canal, segment, coupleur, nombre de paquets, taille des paquets) et les temps de réponse en fonction du trafic.

Les tâches d'administration

Actions en temps réel

A partir de la connaissance de l'état du fonctionnement du réseau (surveillance et diagnostic des incidents, mesure de la charge réelle, maintenance, contrôle..) l'administrateur devra agir sur celui ci et assurer la sécurité.

Actions différées

Elles permettent de planifier, optimiser, quantifier et gérer les évolutions du réseau (statistiques, comptabilité, facturation, prévention, évaluation des charges…)

Actions préventives

Elles permettent d'avoir une vision à moyen et long terme, d'évaluer des solutions alternatives, de construire des "benchmarks", de choisir de nouvelles générations de produits, d'envisager les configurations, décider du plan d'extension, de vérifier la pertinence de la solution réseau pour un problème donné.

Quelques Problèmes

Collisions locales

Des collisions locales sur le réseau sont produites par des stations émettant simultanément. Des taux de collisions élevés laissent supposer que le problème se situe au niveau du câblage.

Collisions tardives

Les collisions tardives sont celles qui se produisent au-delà du cadre des collisions de 512 bits (les collisions normales se produisent dès les premiers octets d'une transmission). Ces collisions peuvent avoir deux origines. On peut avoir à faire avec une station défectueuse (carte, transceiver etc.) qui n'est plus conforme à la convention CSMA/CD. La station considérée n'"écoute" plus la ligne avant d'émettre.

Trames écourtées

Ce sont des trames qui sont plus courtes que le minimum requis de 64 octets. Ces trames écourtées sont souvent le produit de cartes ou de pilotes réseau défectueux.

Trames trop longues ("jabber")

Les trames dont la longueur dépasse les 1518 octets autorisés sont appelées "jabber" (trames "bavardes"). Leur origine est là encore à chercher dans des cartes ou des pilotes réseau défectueux.

"Negative frame check sequence" (FCS)

Il s'agit d'une trame dont les octets de contrôle, situés en fin de trame, ne correspondent pas à la somme des bits de la trame, calculée en guise de contrôle. Ceci indique une erreur de transmission, due à une carte réseau défectueuse, à un mauvais câblage ou à des rayonnements perturbateurs.

Trames fantômes

Une trame fantôme ressemble à une trame de données normale à laquelle on aurait amputée sa séquence initiale ("starting delimiter" 10 10 10 11).

Des courants d'équilibrage de tension ainsi que d'autres problèmes de câblage peuvent induire en erreur le répéteur et lui donner l'impression de recevoir un paquet de données. Les répéteurs sont sensibles aux tensions alternatives et aux composantes alternatives des courants parasites qui circulent sur les lignes. Ces interférences sont alors traitées et transmises comme données sur le réseau. Ces données ne correspondent bien entendu à rien.

L'administrateur doit être capable d'assurer la confidentialité des informations, c'est à dire de contrôler l'accès aux données sensibles.

L'administration ISO

L'administration d'un réseau englobe une large variété d'applications qui ont été regroupées par l'ISO en 5 domaines fonctionnels appelés SMFA (Specific Management Functional Area)

Le modèle fonctionnel

La gestion des erreurs

L'administrateur doit être prévenu dès qu'un problème survient : câble coupé, routeur hors service ...Il pourra alors isoler l'incident et remédier à la panne. Les techniques d'alerte sont sonores ou visuelles.

La gestion des performances

l'administrateur doit être capable d'identifier les éléments causant une dégradation des performances : segment de câble trop chargé, routeur mal adapté...

La gestion de la configuration

Le responsable doit pouvoir visualiser , à tout instant, le plan du réseau et connaître les caractéristiques techniques des équipements connectés.

La gestion de l'exploitation du réseau

Il est du ressort de l'administrateur de réseau de fournir des statistiques sur l'exploitation du réseau : nombre de trames transmises, reçus , temps de connexion des utilisateurs, mode d'exploitation des ressources..

Il doit être dès lors possible d'élaborer une comptabilité permettant d'imputer les coûts par services.

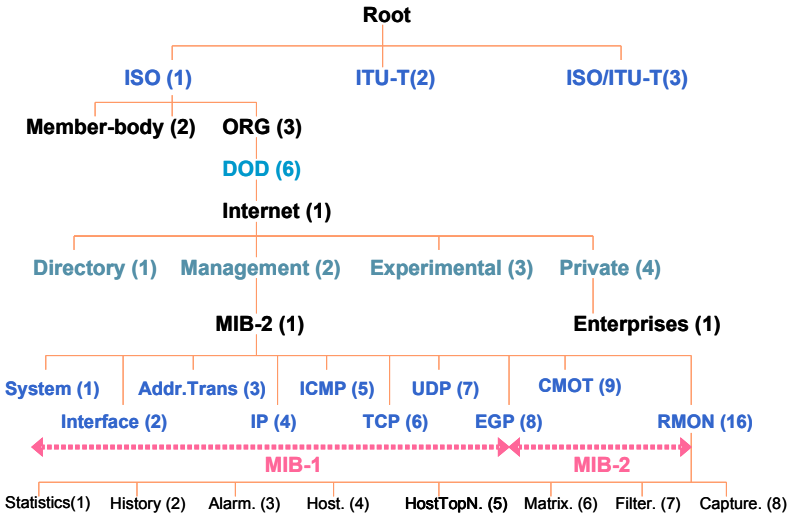
La gestion de la sécurité

L'administrateur doit être capable d'assurer la confidentialité des informations, c'est à dire de contrôler l'accès aux données sensibles.

Le modèle informationnel

L'ISO a défini un modèle qui permet de définir les objets administrés, leur nom, leur fonction, leurs relations, leurs attributs…

Cette structure normée est implantée dans une **base de données hiérarchique** appelée **MIB** (Management Information Base)



La MIB se contente de décrire les objets gérés dans les équipements. Cette description donne :

- le Nom
- le Type,
- le Format.

Elle n'en donne pas les valeurs.

Celles-ci évoluent dynamiquement et ne sont pas gardées dans la base. A chaque interrogation d'un élément de cette base, l'équipement va, en fait, les chercher directement dans ses registres internes.

Il existe plusieurs types de MIB.

MIB-I représente un ensemble de variables standardisées par l'Internet donnant la possibilité d'administrer de manière uniforme des machines fabriquées par des constructeurs différents. Ces variables concernent les divers paramètres de trafic sur les interfaces, la description du matériel ainsi que les paramètres du protocole TCP/IP.

MIB-II est un sur-ensemble de la MIB-I comportant 172 variables. Cet ensemble de variables est en théorie le plus petit dénominateur commun entre tous les types de matériels que l'on peut rencontrer sur un réseau (Concentrateurs, Commutateurs, Routeurs...) et la totalité des constructeurs.

MIB privée regroupe l'ensemble des variables "implémentées" par chaque fabriquant pour administration plus fine de son propre matériel. C'est la richesse de la MIB privée qui permet de faire une réelle différence de la qualité d'administration entre deux matériels similaires.

Champs des éléments de la MIB

Objet : chaîne de caractères utilisée comme synonyme de l'identificateur d'objet ;

Syntaxe : codage ASN1 ;

Définition : description textuelle du type de l'objet ;

Accès : lecture seule, écriture seule, lecture-écriture, non accessible ;

Statut : obligatoire (*mandatory*), optionnel ou obsolète.

OID Object Identifier

L'OID est une suite d'entiers (chemin jusqu'à l'objet dans l'arbre)

org	1.3	
dod	1.3.6	
internet	1.3.6.1	
directory	1.3.6.1.1	
mgmt	1.3.6.1.2	
mib-2	1.3.6.1.2.1	Les MIB standards sont sous l'OID 1.3.6.1.2.1 (mib, mib-2)
experimental	1.3.6.1.3	
private	1.3.6.1.4	Les MIB privées sont sous l'OID 1.3.6.1.4.1 (enterprises)
enterprises	1.3.6.1.4.1	

Object Descriptor = nom de l'objet (unique). Ex : SysName (Nom de l'Administrateur) a pour OID 1.3.6.1.2.1.1.5 et l'instance du nom de l'administrateur a comme identifiant 1.3.6.1.2.1.1.5.0

L'exemple suivant utilise l'outil snmputil pour afficher la valeur de l'objet SysName. Vous noterez que l'OID doit commencer par un point pour indiquer la racine de l'arbre.

```
C:\>snmputil getnext 192.168.0.3 public .1.3.6.1.2.1.1.5
Variable = system.sysName.0
Value    = String CHAMBONPORTABLE
```

Les MIB standards sont sous l'OID 1.3.6.1.2.1 (mib, mib-2)

Les MIB privées sont sous l'OID 1.3.6.1.4.1 (enterprises)

Les fichiers MIB :

Il existe de nombreux fichiers MIB ("des MIBs") en fonction des domaines à administrer. Chaque fichier MIB correspond à un "point d'entrée" dans la structure globale (MIB) de l'équipement (si ce domaine est géré)

- *public MIB file* : spécification en fin de RFC

RFC 1066 = MIB-1 (TCP/UDP/IP)
RFC 1213 = MIB-2 (TCP/UDP/IP)
RFC 1286 = bridge MIB
RFC 1643 = Ethernet-Like MIB
RFC 1748, 1749 = Token-Ring MIB
RFC 1512 = FDDI MIB
RFC 1604 = Frame Relay MIB
RFC 1757 = RMON *Remote MONitoring MIB* (-> Ethernet)
RFC 1513 = RMON Token-Ring (groupe suppl. pour TR)
RFC 2021 = RMON-2
RFC 1612 = DNS Resolver MIB
RFC 1611 = DNS Server MIB
…

- *private MIB file* : fabricant de l'équipement

Extrait du fichier MIB-2 : RFC 1213

```
RFC1213-MIB DEFINITIONS ::= BEGIN

        IMPORTS
                mgmt, NetworkAddress, IpAddress, Counter, Gauge,
                        TimeTicks
                    FROM RFC1155-SMI
                OBJECT-TYPE
                        FROM RFC-1212;

        --  This MIB module uses the extended OBJECT-TYPE macro as defined in [14];
          --  MIB-II (same prefix as MIB-I)
         mib-2      OBJECT IDENTIFIER ::= { mgmt 1 }
    l'objet  mib-2  est l'objet n°1 dans l'objet mgmt. L'OID de mgmt étant 1.3.6.1.2, l'OID de mib-2 sera  1.3.6.1.2.1
          -- textual conventions
          DisplayString ::=
```

```
        OCTET STRING
    -- This data type is used to model textual information taken from the NVT ASCII character set.  By
convention, objects with this syntax are declared as having SIZE (0..255)

    PhysAddress ::=
        OCTET STRING
    -- This data type is used to model media addresses.  For many types of media, this will be in a binary
representation.
    -- For example, an ethernet address would be represented as a string of 6 octets.
    -- groups in MIB-II
    system      OBJECT IDENTIFIER ::= { mib-2 1 }
    interfaces  OBJECT IDENTIFIER ::= { mib-2 2 } Le groupe « interfaces » est le deuxième objet de mib-2
    at          OBJECT IDENTIFIER ::= { mib-2 3 }
    ip          OBJECT IDENTIFIER ::= { mib-2 4 }
    icmp        OBJECT IDENTIFIER ::= { mib-2 5 }
    tcp         OBJECT IDENTIFIER ::= { mib-2 6 }
    udp         OBJECT IDENTIFIER ::= { mib-2 7 }
    egp         OBJECT IDENTIFIER ::= { mib-2 8 }
    -- historical (some say hysterical)
    cmot        OBJECT IDENTIFIER ::= { mib-2 9 }
    transmission OBJECT IDENTIFIER ::= { mib-2 10 }
    snmp        OBJECT IDENTIFIER ::= { mib-2 11 }
```

Définition du groupe interfaces

```
-- the Interfaces group

        -- Implementation of the Interfaces group is mandatory for all systems.
```

Le premier objet de ce groupe est un objet simple. Son nom est ifnumber, la variable associée sera du type entier, obligatoire et uniquement accessible en lecture

Son OID sera 1.3.6.1.2.1.2.1

```
ifNumber OBJECT-TYPE
    SYNTAX  INTEGER
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
            "The number of network interfaces (regardless of
            their current state) present on this system."
    ::= { interfaces 1 }
```

```
C:\>snmputil getnext 192.168.0.3 public .1.3.6.1.2.1.2.1
Variable = interfaces.ifNumber.0
Value    = Integer32 3
```

La machine de test est équipée de 3 interfaces.

On pourra écrire également : snmputil get 192.168.0.3 public interfaces.ifNumber.0

le .0 correspond à l'instance de la variable.

Le deuxième objet est un objet composé (tableau)

```
-- the Interfaces table

-- The Interfaces table contains information on the entity's
-- interfaces.  Each interface is thought of as being
-- attached to a `subnetwork'.  Note that this term should
-- not be confused with `subnet' which refers to an
-- addressing partitioning scheme used in the Internet suite
-- of protocols.
```

Ce tableau s'appelle ifTable,

```
ifTable OBJECT-TYPE
    SYNTAX  SEQUENCE OF IfEntry
    ACCESS  not-accessible
    STATUS  mandatory
    DESCRIPTION
            "A list of interface entries.  The number of
            entries is given by the value of ifNumber."
    ::= { interfaces 2 }
```

L'écran suivant illustre le coté inaccessible de l'objet IfTable. Le logiciel d'interrogation a été programmé pour donner la valeur de la première variable du tableau.

```
C:\>snmputil getnext 192.168.0.3 public .1.3.6.1.2.1.2.2
Variable = interfaces.ifTable.ifEntry.ifIndex.1
Value    = Integer32 1


C:\>snmputil getnext 192.168.0.3 public .1.3.6.1.2.1.2.2.1
Variable = interfaces.ifTable.ifEntry.ifIndex.1
Value    = Integer32 1
```

Iftable définit une table alors que ifEntry décrit la structure d'un enregistrement du tableau. IfTable est une structure de données définie dans ifEntry dont le numéro de séquence est précisé dans ifIndex.

```
ifEntry OBJECT-TYPE
    SYNTAX  IfEntry
    ACCESS  not-accessible
    STATUS  mandatory
    DESCRIPTION
            "An interface entry containing objects at the
            subnetwork layer and below for a particular
            interface."
    INDEX   { ifIndex }
    ::= { ifTable 1 }
```

Les informations contenues dans le tableau sont pour chaque interface, son numéro (ifIndex), une description (ifDescr), un type (ifType) etc... jusqu'à ifspecific. Soit au total 22 informations par interface.

```
IfEntry ::=
    SEQUENCE {
        ifIndex
            INTEGER,
        ifDescr
            DisplayString,
        ifType
            INTEGER,
        ifMtu
            INTEGER,
        ifSpeed
            Gauge,
        ifPhysAddress
            PhysAddress,
        ifAdminStatus
            INTEGER,
        ifOperStatus
            INTEGER,
        ifLastChange
            TimeTicks,
        ifInOctets
            Counter,
        ifInUcastPkts
            Counter,
        ifInNUcastPkts
            Counter,
        ifInDiscards
            Counter,
        ifInErrors
            Counter,
        ifInUnknownProtos
            Counter,
        ifOutOctets
            Counter,
        ifOutUcastPkts
            Counter,
        ifOutNUcastPkts
            Counter,
        ifOutDiscards
            Counter,
        ifOutErrors
            Counter,
        ifOutQLen
            Gauge,
        ifSpecific
            OBJECT IDENTIFIER
    }
```

Chaque élément de l'enregistrement est ensuite déclaré:

```
ifIndex OBJECT-TYPE
    SYNTAX  INTEGER
```

```
        ACCESS  read-only
        STATUS  mandatory


        DESCRIPTION
                "A unique value for each interface.  Its value
                ranges between 1 and the value of ifNumber.  The
                value for each interface must remain constant at
                least from one re-initialization of the entity's
                network management system to the next re-
                initialization."
        ::= { ifEntry 1 }

    ifDescr OBJECT-TYPE
        SYNTAX  DisplayString (SIZE (0..255))
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "A textual string containing information about the
                interface.  This string should include the name of
                the manufacturer, the product name and the version
                of the hardware interface."
        ::= { ifEntry 2 }

    ifType OBJECT-TYPE
        SYNTAX  INTEGER {
                    other(1),          -- none of the following
                    regular1822(2),
                    hdh1822(3),
                    ddn-x25(4),
                    rfc877-x25(5),
                    ethernet-csmacd(6),
                    iso88023-csmacd(7),
                    iso88024-tokenBus(8),
                    iso88025-tokenRing(9),
                    iso88026-man(10),
                    starLan(11),
                    proteon-10Mbit(12),
                    proteon-80Mbit(13),
                    hyperchannel(14),
                    fddi(15),
                    lapb(16),
                    sdlc(17),
                    ds1(18),           -- T-1
                    e1(19),            -- european equiv. of T-1
                    basicISDN(20),
                    primaryISDN(21),   -- proprietary serial
                    propPointToPointSerial(22),
                    ppp(23),
                    softwareLoopback(24),
                    eon(25),            -- CLNP over IP [11]
                    ethernet-3Mbit(26),
                    nsip(27),           -- XNS over IP
                    slip(28),           -- generic SLIP
                    ultra(29),          -- ULTRA technologies
                    ds3(30),            -- T-3
                    sip(31),            -- SMDS
                    frame-relay(32)
                }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The type of interface, distinguished according to
                the physical/link protocol(s) immediately `below'
                the network layer in the protocol stack."
        ::= { ifEntry 3 }

    ifMtu OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The size of the largest datagram which can be
                sent/received on the interface, specified in
                octets.  For interfaces that are used for
                transmitting network datagrams, this is the size
                of the largest network datagram that can be sent
                on the interface."
        ::= { ifEntry 4 }

    ifSpeed OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
```

```
    STATUS  mandatory
    DESCRIPTION
            "An estimate of the interface's current bandwidth
            in bits per second.  For interfaces which do not
            vary in bandwidth or for those where no accurate
            estimation can be made, this object should contain
            the nominal bandwidth."
    ::= { ifEntry 5 }

ifPhysAddress OBJECT-TYPE
    SYNTAX  PhysAddress
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
            "The interface's address at the protocol layer
            immediately `below' the network layer in the
            protocol stack.  For interfaces which do not have
            such an address (e.g., a serial line), this object
            should contain an octet string of zero length."
    ::= { ifEntry 6 }

ifAdminStatus OBJECT-TYPE
    SYNTAX  INTEGER {
                up(1),       -- ready to pass packets
                down(2),
                testing(3)   -- in some test mode
            }
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
            "The desired state of the interface.  The
            testing(3) state indicates that no operational
            packets can be passed."
    ::= { ifEntry 7 }

ifOperStatus OBJECT-TYPE
    SYNTAX  INTEGER {
                up(1),       -- ready to pass packets
                down(2),
                testing(3)   -- in some test mode
            }
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
            "The current operational state of the interface.
            The testing(3) state indicates that no operational
            packets can be passed."
    ::= { ifEntry 8 }

ifLastChange OBJECT-TYPE
    SYNTAX  TimeTicks
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
            "The value of sysUpTime at the time the interface
            entered its current operational state.  If the
            current state was entered prior to the last re-
            initialization of the local network management
            subsystem, then this object contains a zero
            value."
    ::= { ifEntry 9 }

ifInOctets OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
            "The total number of octets received on the
            interface, including framing characters."
    ::= { ifEntry 10 }

ifInUcastPkts OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
            "The number of subnetwork-unicast packets
            delivered to a higher-layer protocol."
    ::= { ifEntry 11 }

ifInNUcastPkts OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
```

```
    STATUS  mandatory
    DESCRIPTION
            "The number of non-unicast (i.e., subnetwork-
            broadcast or subnetwork-multicast) packets
            delivered to a higher-layer protocol."
    ::= { ifEntry 12 }

ifInDiscards OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
            "The number of inbound packets which were chosen
            to be discarded even though no errors had been
            detected to prevent their being deliverable to a
            higher-layer protocol.  One possible reason for
            discarding such a packet could be to free up
            buffer space."
    ::= { ifEntry 13 }

ifInErrors OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
            "The number of inbound packets that contained
            errors preventing them from being deliverable to a
            higher-layer protocol."
    ::= { ifEntry 14 }

ifInUnknownProtos OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
            "The number of packets received via the interface
            which were discarded because of an unknown or
            unsupported protocol."
    ::= { ifEntry 15 }

ifOutOctets OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
            "The total number of octets transmitted out of the
            interface, including framing characters."
    ::= { ifEntry 16 }

ifOutUcastPkts OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
            "The total number of packets that higher-level
            protocols requested be transmitted to a
            subnetwork-unicast address, including those that
            were discarded or not sent."
    ::= { ifEntry 17 }

ifOutNUcastPkts OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
            "The total number of packets that higher-level
            protocols requested be transmitted to a non-
            unicast (i.e., a subnetwork-broadcast or
            subnetwork-multicast) address, including those
            that were discarded or not sent."
    ::= { ifEntry 18 }

ifOutDiscards OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
            "The number of outbound packets which were chosen
            to be discarded even though no errors had been
            detected to prevent their being transmitted.  One
            possible reason for discarding such a packet could
            be to free up buffer space."
    ::= { ifEntry 19 }
```

```
ifOutErrors OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
            "The number of outbound packets that could not be
            transmitted because of errors."
    ::= { ifEntry 20 }

ifOutQLen OBJECT-TYPE
    SYNTAX  Gauge
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
            "The length of the output packet queue (in
            packets)."
    ::= { ifEntry 21 }

ifSpecific OBJECT-TYPE
    SYNTAX  OBJECT IDENTIFIER
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
            "A reference to MIB definitions specific to the
            particular media being used to realize the
            interface.  For example, if the interface is
            realized by an ethernet, then the value of this
            object refers to a document defining objects
            specific to ethernet.  If this information is not
            present, its value should be set to the OBJECT
            IDENTIFIER { 0 0 }, which is a syntatically valid
            object identifier, and any conformant
            implementation of ASN.1 and BER must be able to
            generate and recognize this value."
    ::= { ifEntry 22 }
```

Affichage des valeurs de la Table

```
C:\>enummibtable -t .1.3.6.1.2.1.2.2 -n 3
ifEntry 2       3
=====================================================================

1       2       2
65539   128     128
65540   688428  688428
MS TCP Loopback interface<00>   1       1
Intel<R> PRO/Wireless 2200BG Network Connection<00>     16      16
Intel<R> PRO/100 VE Network Connection<00>      0       0
24      0       0
6       105     105
6       688327  688327
1520    402585  402585
1500    154     154
1500    0       0
10000000        60      60
54000000        0       0
100000000       0       0
        0       0
<00><0E>5<05>ð<DF>      0       0
<08><00>P<C8><A3><A4>   0       0
1       0       0
1       127.0.0.1       127.0.0.1
1       192.168.0.3     192.168.0.3
1       192.168.0.4     192.168.0.4
1       1       1
1       65540   65540
0       65539   65539
0       255.0.0.0       255.0.0.0
0       255.255.255.0   255.255.255.0
7516403 255.255.255.0   255.255.255.0
39327   1       1
816553037       1       1
52042   1       1
62      65535   65535
636036  65535   65535
0       65535   65535
220     0.0.0.0 0.0.0.0
261     127.0.0.0       127.0.0.0
0       192.168.0.0     192.168.0.0
0       192.168.0.3     192.168.0.3
0       192.168.0.4     192.168.0.4
0       192.168.0.255   192.168.0.255
0       224.0.0.0       224.0.0.0
0       255.255.255.255 255.255.255.255
0       65539   65539
1       1       1
1       65540   65540
7516403 1       1
11976   1       1
20097560        65540   65540
52042   65540   65540
75      65540   65540
350161  25      25
0       1       1
13      20      20
190     20      20
0       25      25
0       20      20
0       20      20
0       1       1
0       -1      -1
0       -1      -1
0       -1      -1
0       -1      -1
0       -1      -1
.0.0    -1      -1
.0.0    -1      -1
.0.0    -1      -1
End of table.
```