

SNMP ou l'administration distante.

1.6.3 SNMP-GETNEXT

Cette commande **retourne** le prochain **objet** qui **suit** celui spécifié dans la commande. Elle effectue un parcours ordonné de l'arbre. On pourra définir 2 usages pour cette commande :

- Permettre **d'obtenir** des **objets** sans connaître leur OID précisément.
- Dans le cas où une variable est constituée d'un **tableau** de valeurs, il est possible de **scruter** les valeurs de cette variable en laissant la commande GetNext récupérer les index suivants. Il est donc possible d'avoir des trous dans la table (index non attribué) et de les gérer automatiquement par cette commande.

Syntaxe de la commande SNMPGETNEXT à respecter :
snmpgetnext [arguments communs] objectID [objectID]...

Exemple : `snmpgetnext 127.0.0.1 public`

Utilisation : Obtenir la liste des interfaces réseau d'un agent (lo, eth0, ...), obtenir les tables de routage d'un agent

1.6.4 SNMP-SET

Elle est utilisée pour **créer ou modifier** des objets gérés.

La trame doit spécifier les objets à affecter et les valeurs qui leurs seront affectées.

Si dans la trame un des objets ne peut être affecté ou une des valeurs est mauvaise, alors aucun des objets à affecter ne le sera et la trame de réponse indiquera une erreur et sur quel objet elle se situe.

On peut aussi réaliser des suppressions et des inversions de lignes (ou objets) dans une table. La méthode pour ajouter ou supprimer des "lignes"(objets) d'une table dépend de la définition des objets de la table, mais l'affectation avec une chaîne de caractère "invalid" doit convenir pour une suppression.

Syntaxe de la commande SNMPSET à respecter :
snmpset [arguments communs] objectID type valeur [objectID type valeur]...

1.6.5 SNMP-TRAPv1

Un agent peut observer certains événements, y réaliser des détections et émettre un message "trap" à des stations de gestion pré configurées pour cela. Les messages "traps" sont eux, émis sur le port UDP/162.

- La **première approche** correspond à la notion d'**événement non sollicités**. Lorsqu'un agent détecte un problème urgent, il émet un TRAP vers sa station de gestion sans attendre que celle-ci l'interroge. L'**avantage** de cette approche est de permettre une réception immédiate des situations des agents par la station de gestion. Néanmoins plusieurs **inconvénients** se présentent :
 - Chaque agent doit posséder une application pour gérer ces événements (temps CPU)
 - l'agent considère que la station de gestion est toujours prête.
 - il y a risque de surcharge du trafic sur le réseau. Pour éviter cela, on peut appliquer une politique de seuil qui impose à l'agent de n'envoyer des événements que si l'occurrence de cet événement dépasse une valeur fixée par l'agent.
- La **deuxième approche** consiste à solliciter les événements depuis la station de gestion. Celle-ci **scrute périodiquement** les agents pour obtenir leurs informations d'états. Cette approche présente l'**avantage** de permettre à la station de gestion de contrôler tous les agents.

Néanmoins, le **problème** qui se pose est le choix de la fréquence de scrutation :

 - trop faible, la réaction aux événements risque d'être trop lente
 - trop élevée, le trafic sur le réseau est trop importantCela limite le nombre d'agents qui peuvent être administrés.

Pour bénéficier des **avantages des deux approches**, SNMP propose un compromis qui consiste à imposer à l'agent, d'envoyer un signal vers la station de gestion pour l'avertir d'un problème. La station de gestion a alors la maîtrise pour demander à l'agent de lui envoyer un descriptif complet de sa situation afin de pouvoir résoudre le problème ou avertir l'administrateur.

Cette opération prend comme arguments la valeur de l'objet "sysObjectID" et l'adresse IP de l'agent, le type événement, le numéro d'identification de cet événement spécifique, la valeur de la variable "sysUpTime" de l'agent, la liste des variables contenant des informations sur l'événement ainsi que leurs valeurs.

Syntaxe de la commande SNMPTRAPv1 de NET-SNMP à respecter :
`snmptrap -v 1 [-Ci] [arg commun] oid-enterprise agent generic-trap specific-trap uptime [objectID type valeur]...`