Il est obligatoire que toutes les implémentations de SNMP supportent les cinq PDUs(Protocol Data Units) suivants:

- **GetRequest-PDU,**
- **GetNextRequest-PDU,**
- **GetResponse-PDU,**
- **SetRequest-PDU, et**
- **Trap-PDU.**

Type	***Structure des Trames SNMP V1, V2 et V3***								
	version	community	Somme de PDU SNMP						
get	0->V1	"public"	**0xA0**	ReqId	0	0	Variables liées		
get-next	1->V2		**0xA1**	ReqId	0	0	Variables liées		
set			**0xA3**	ReqId	0	0	Variables liées		
réponse			**0xA2**	ReqId	ES	EI	Variables liées		
Trap			**0xA4**	Entreprise	Addr	Generic-trap	Specific-trap	Timestamp	Variables liées
TrapV2			**0xA7**	ReqId	0	0	Variables liées		
InformV2			**0xA6**	ReqId	0	0	Variables liées		
Get-bulkV2			**0xA5**	ReqId	n	m	Variables liées		
ReportV3			**0xA8**	ReqId	0	0	Variables liées		

Quelques **informations** sur les champs du tableau précédent :

- **Version** : 0 pour indiquer que la trame est de version SNMPv1 ...
- **Community** : SNMPv1 supporte une **authentification triviale** par utilisation en quelque sorte d'un mot de passe appelé "community name". Plus précisément, la communauté doit être utilisée pour accéder à un objet de celle-ci. Un agent définit typiquement 2 communautés, une pour tous les objets pouvant être lus (objet en read-only ou en read-write), et une autre pour les objets modifiables (read-write).
- **Variables liées** : champs qui spécifient le ou les objets devant être collectés ou modifiés.
- **ReqId** : request identifier. Cette valeur doit être la même entre la requête et la réponse.
- **ES** : error status. Indique si un agent a procédé avec succès à la requête.
  - noError(0) : succès
  - tooBig(1) : Plusieurs raisons mais la principale serait que la requête contienne trop d'objets.
  - noSuchName(2) : tentative d'accès à une variable qui n'existe pas ou l'affectation d'une variable accessible en lecture seule.
  - babValue(3) : tentative d'affectation d'un objet avec un mauvais format ou une valeur non valide.
  - readOnly(4) : tentative d'affectation d'un objet non accessible en écriture. En réalité non utilisé car une tentative d'accés à une variable en lecture seule produira l'ES 2.
  - genErr(5) : autres erreurs
- **EI** : error index. Indique, si différent de 0, la position de la première variable dans la requête qui est en erreur.
- **Entreprise** : identifie le composant qui a généré le "trap". Cette valeur vaut "mib-2.system.sysObjectID"
- **Addr** : adresse IP de l'agent qui a généré le "trap".
- **Generic-trap** : numéro indiquant le type d'évènement émis par l'agent.
  - **coldStart**(0) : indique que l'agent a été redémarré
  - **warmStart**(1) : indique que l'agent s'est réinitialisé
  - **linkDown**(2) : indique qu'une interface est passée de l'état "up" à "down". La première variable identifie l'interface.
  - **linkUp**(3) : indique qu'une interface est passée à l'état "up". La première variable identifie l'interface.
  - **authentificationFailure**(4) : indique qu'un message SNMP a été reçu avec un échec d'authentification (mauvais nom de communauté)
  - **egpNeighborLoss**(5) : indique qu'un voisin EGP est passée à l'état "down". La première variable identifie l'adresse IP du voisin EGP.
  - (6) : le trap est spécifique à l'entreprise et son numéro est dans le champs "specific-trap".
- **Specific-trap** : Numéro spécifique de l'évènement. Un problème évident avec ce mode est que différents vendeurs peuvent récupérer les mêmes numéros pour représenter les évènements de leurs matériels. Il faut donc connaître pour chaque concepteur de matériel les numéros d'évènements spécifiques de chaque matériel. SNMPv2 définit un nouveau type de trap permettant de résoudre ce problème.
- **Timestamp** :. Cette valeur vaut "mib-2.system.sysUpTime"