Supervision des réseaux

Rushed Kanawati
Département R&T, IUT de Villetaneuse
http://lipn.fr/∼kanawati
rushed.kanawati@lipn.univ-paris13.fr

Module M3108

November 28, 2014

Plan

Organisation du cours

Module M3 108 — 30h

- 4 séances de cours
- 3 TD
- 4 TP
- **Contrôle continu**

CONTEXTE

Gestion de réseaux

Tâches de gestion — Modèle FCAPS (ISO)

1. Gestion des Fautes
2. Gestion des Configurations
3. Gestion de l'utilisation (Accounting)
4. **Gestion des Performances**
5. Gestion de la Sécurité

Gestion de réseaux

Gestion de Fautes

- Identifier les sources des fautes (ex. inaccessibilité d'un service)
- Résoudre le problème
- Capitalisation de l'expérience de résolution de problèmes.

  Souvent ignoré, au risque de refaire les étapes 1 et 2 encore et encore pour les mêmes problèmes

Gestion de réseaux

Gestion de Configuration

- Déploiement et maintenance de la configuration des ressources du réseau
- Sauvegarde des configurations des ressources dans une base de données, ce qui facilite la mise à jour, la surveillance mais aussi la résolution de problèmes.

Gestion de réseaux

Gestion de l'utilisation

- Garantir que les ressources de calculs et de communication sont utilisées équitablement par les différentes utilisateurs.

Gestion de réseaux

Gestion de la Performance

- Définition de mesures (métriques) de *qualité de service* (QoS).
- Collecte ou estimation continu des valeurs des métriques définies
- Définition de seuils d'alarme que le dépassement peut indiquer un incident qui nécessite intervention.

Gestion de réseaux

Gestion de la sécurité

- Contrôler l'accès aux ressources du réseau (droits d'accès)
- Détection des intrusions
- Outils = Pare-feu, Système de détection/prévention d'intrusion, anti-virus, techniques de chiffrement . . . , etc.
- **Etude en module M4 210**

GÉNÉRALITÉS

Supervision $\subset$ Gestion

Quoi ?

Un dispositif qui permet de surveiller le bon fonctionnement des éléments actifs d'un réseau

Pourquoi ?

- Faciliter la gestion du réseau
- Améliorer la qualité du service (QoS), la fiabilité et la disponibilité du réseau !
- **Garantir le respect d'un *contrat de service* (SLA: Service Level Agreement)**
- Prévenir des tentatives d'intrusions non autorisés

Supervision des réseaux

Supervision $\subset$ Gestion

Comment ?

- Méthodes ad. hoc.

  ping, traceroute, netstat, telnet, whois, . . . , etc.

- Protocole de gestion de réseaux
  - Common Information Management Protocol (CMIP)

    protocole OSI, mais peu utilisé
  - **Simple Network Management Protocol (SNMP)**

    Très simple à mettre en œuvre → large adoption

SLA: Contrat de service

1. Désignation des parties
2. Période
3. Services inclus

   Description technique des services et des **métriques**
4. **SLS (Service Level specification)**

   Seuils de valeurs de métriques
5. Pénalités
6. Reporting et revues
7. Auditabiliié

Qualité de service: définitions

Points de vue

- Pour un utilisateur final : obtenir le meilleur service au meilleur prix
- Pour un administrateur réseaux : obtenir le meilleur service pour un groupe d'utilisateurs
- Pour un ingénieur réseau : conception d'architectures et de mécanismes de mesures de QoS.
- ...

QoS

Mesures

- **Flux** : Une séquence de paquets envoyée d'une source vers une destination.
- Caractérisation de flux:
  1. Fiabilité
  2. Délai (propagation, latence, . . .)
  3. Gigue : variation des temps d'arrivés des paquets
  4. Bande Passante

Exigence de QoS / type d'applications

Application	Fiabilité	Délai	Gigue	BP
e-mail	Haute	Faible	Faible	Faible
FTP	Haute	Faible	Faible	Moyenne
WEB	Haute	Moyenne	Faible	Moyenne
Telnet	Haute	Moyenne	Moyenne	Faible
Audio streaming	Faible	Faible	Haute	Moyenne
Vidéo streaming	Faible	Faible	Haute	Haute
VoIP	Faible	Haute	Haute	Faible
Vidéo conf.	Faible	Haute	Haite	Haute

Quelques techniques de gestion de QoS

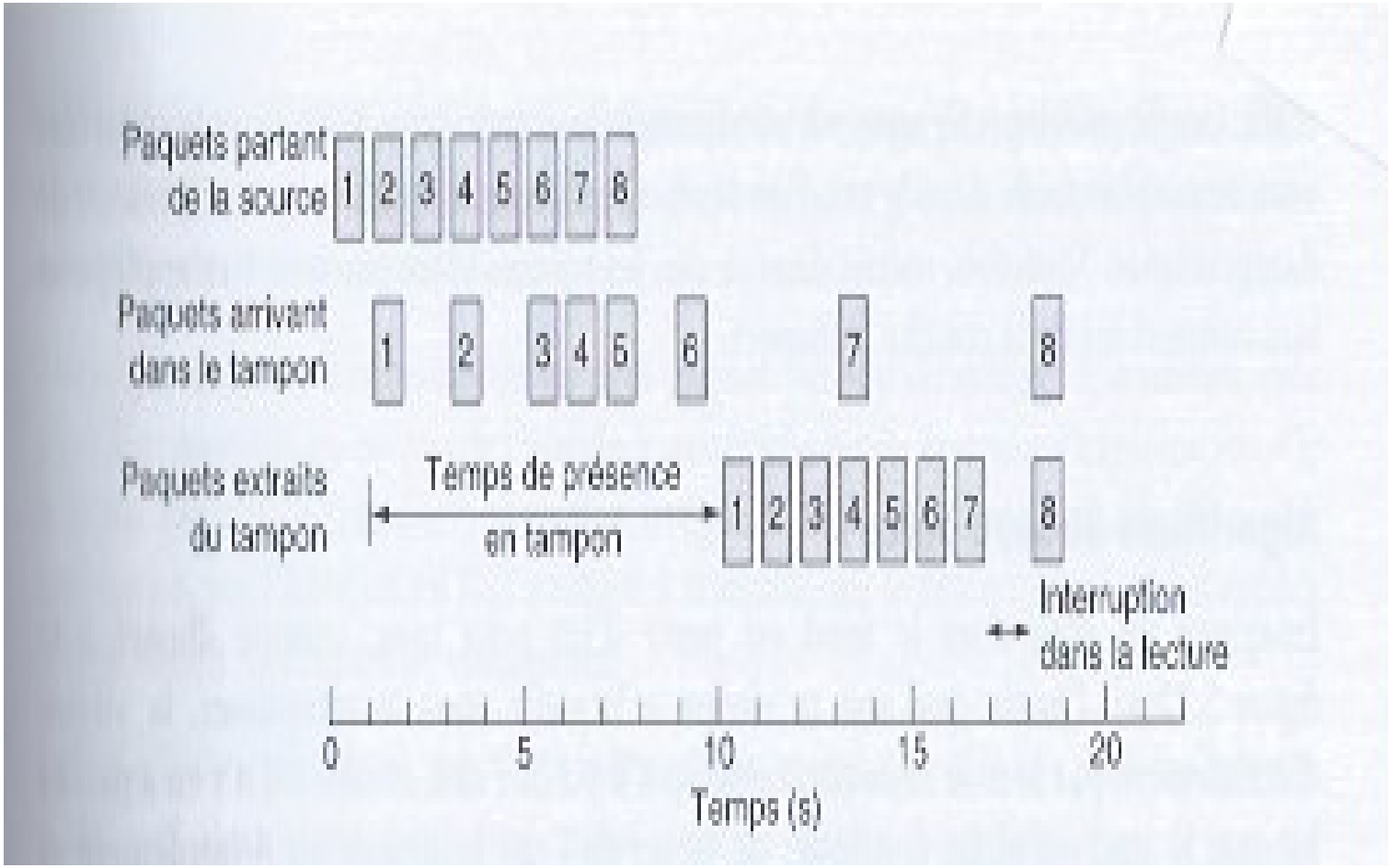
Techniques classiques de gestion de QoS

- Reservation en excès des ressources

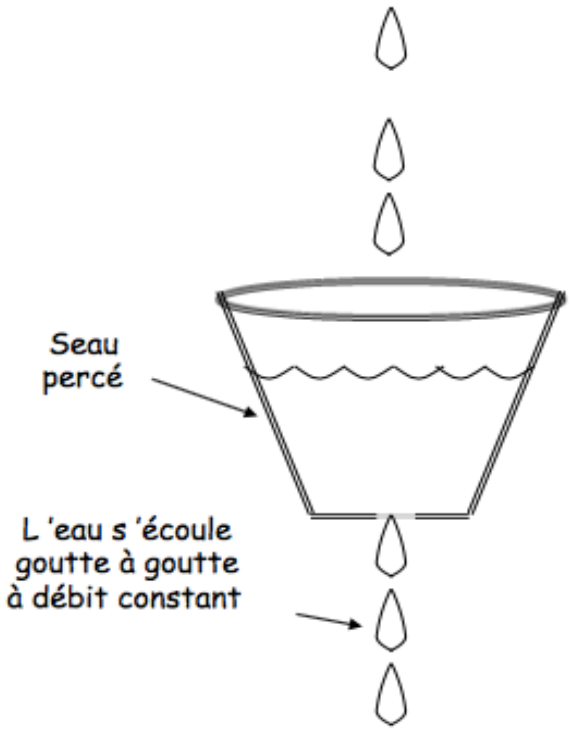
  problème de coût
- Mise en tampon sur le récepteur
- Canalisation du trafic

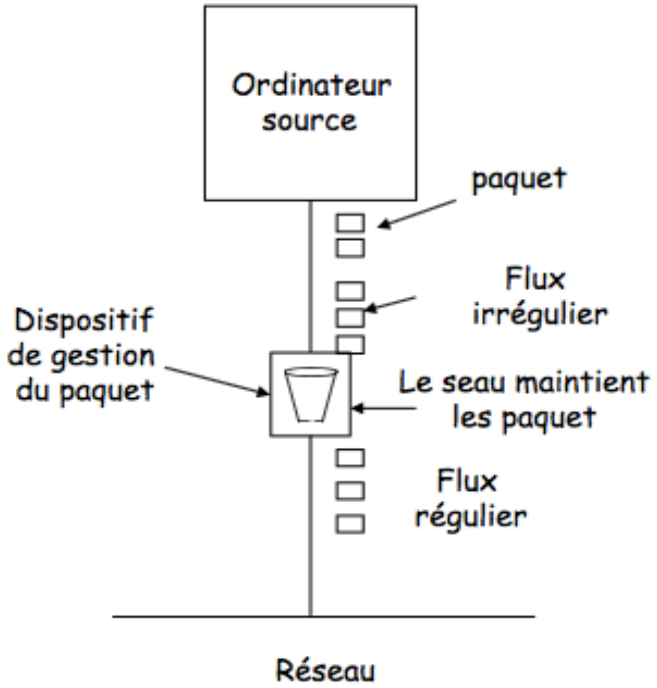
  Algo de seau percé/à jetons
- Réservation de ressources
- ...

Mise en tampon



Canalisation: Algorithme du seau percé

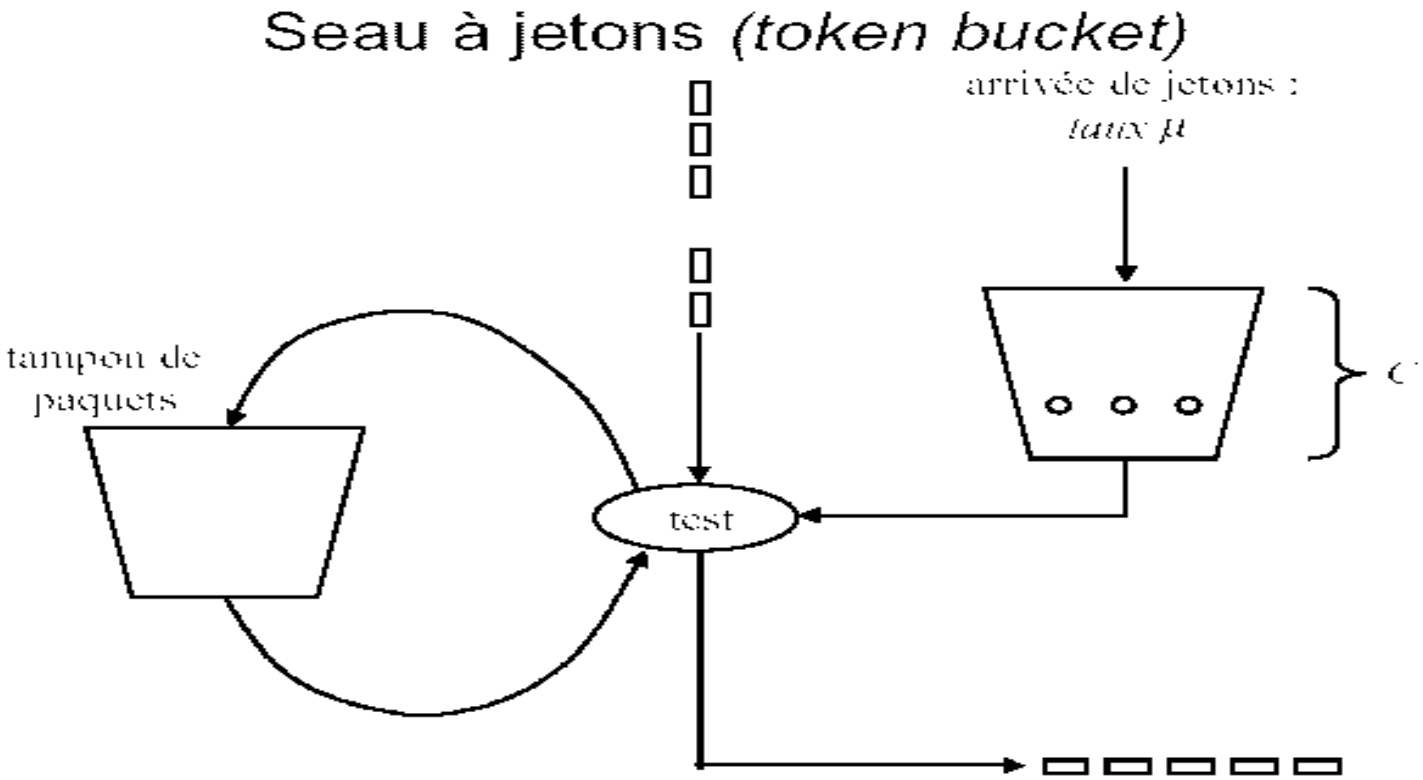




Variations

Le seau peut compter le **nombre de paquets** à passer ou **le volume de données** à passer

CANALISATION: ALGO DU SEAU À JETON



Variations

Un jeton peut correspondre à un **paquet** à passer ou à un **volume de données** à passer

Mesures et métriques

Méthodologie générale

1. Définir l'objectif des mesures
2. Définir de points de mesure et placer des points de contrôle
3. Observer l'exécution du système.
4. Collecter les mesures et **analyser**

MÉTRIQUES : LIMITATIONS

Limitations

- Limitations physiques

  Vitesse de propagation des ondes

- Limitations induites par les équipements

  Files d'attente, charges de liens, routage, . . . , etc.

Techniques de mesure

- Mesures intrusives

  Injection du trafic

- Par estimation

  Capture et analyse de traces

- Simulation

  Modèle mathématique : ex. théorie de files d'attente

Mesures intrusives: +/-

Avantages

+ Evaluation réelle de la QoS
+ Choix du type de trafic à évaluer

Inconvénients

- Nécessite un système opérationnel
- Mesurer modifie ce qui est mesuré !
- Difficile et coûteux

ESTIMATION : +/-

Avantages

+ Pas de modifications du système

Inconvénients

- Difficile de prévoir la QoS
- Faible reproductibilité
- coût de collecte des traces.

SIMULATION : +/-

Avantages

+ Rapidité d'exécution
+ Choix de scénarios d'évaluation

Inconvénients

- Développement d'un modèle (avec approximations forcément)
- Nécessite confirmation par des mesures directes ou estimation !

INTERPRÉTATION DE MESURES

- Une mesure est une suite de données numériques
- Un échantillon de n valeurs est représenté par : $X_1, \ldots, X_n$
- Interprétation : emploi de fonction de statistiques de base : `min, max, moyenne, variance, ..., etc.`

Fonctions statistiques

- **Moyenne**: $\bar{X}_n = \frac{1}{n}\sum_{i=1}^{n} X_i$.
- **Variance**: $V(X) = \frac{1}{n}\sum_{i=1}^{n}(X_i - \bar{X}_n)^2$
- **Ecart type** : $\sigma(X) = \sqrt{V(X)}$
- **Moyenne sur fenêtre** T : $\bar{X}_T = \frac{1}{T}\sum_{i=j}^{j+T-1} X_i$
- **Moyenne mobile exponentielle** : $\bar{X}_{n,\alpha} = \alpha \times X_n + (1-\alpha)\bar{X}_{n-1,\alpha}$

Fonctions statistiques: EDF

- EDF: Empirical distribution function (fréquence empirique de dépassement)
- $F(x) = \frac{1}{n} card\{i | X_i \leq x\}$
- Exemple : Soit $X = -2, 7, 7, 4, 18, -5$
- On a alors :

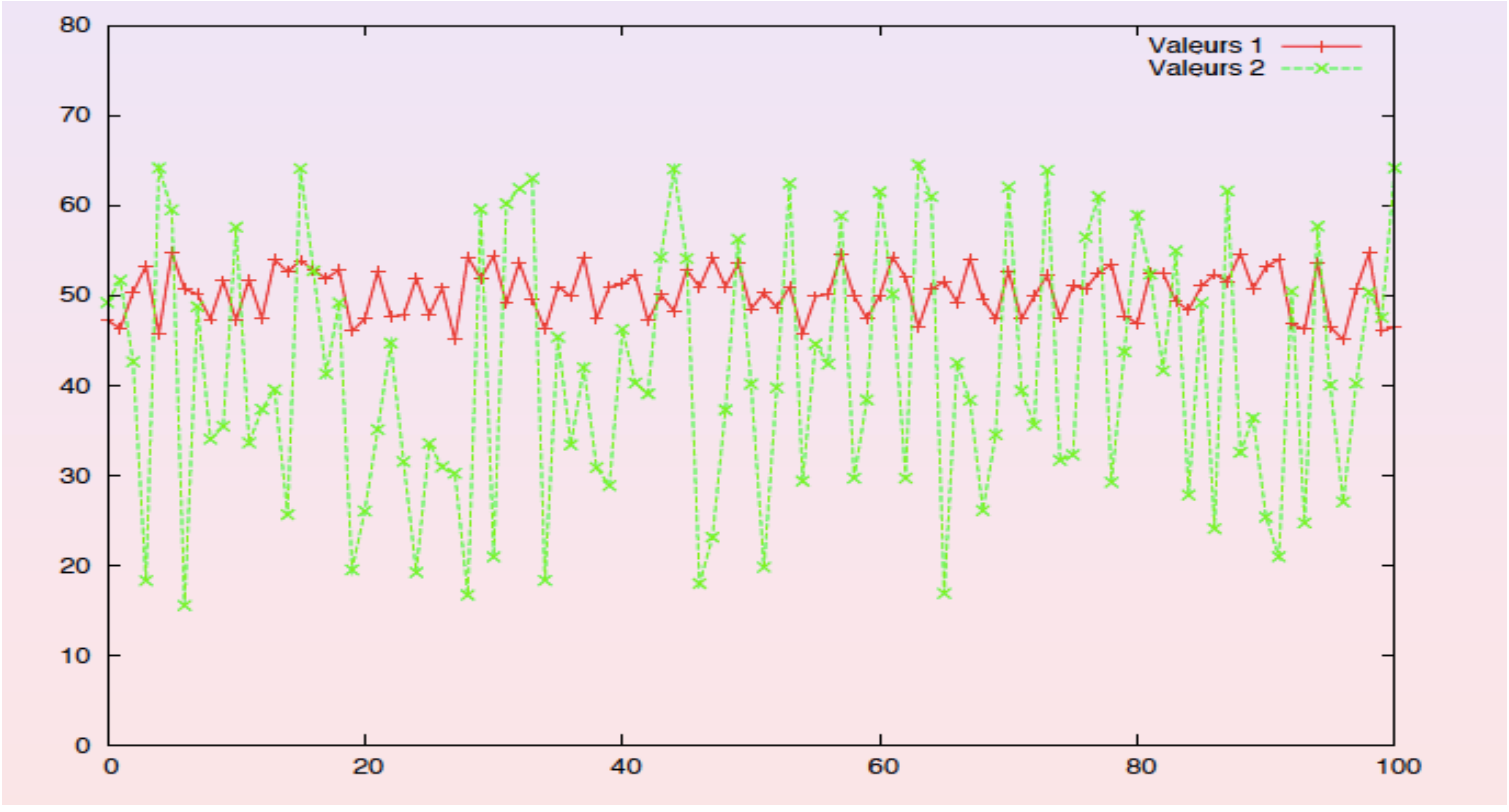
x	-8	-5	-5.0001	-4.999	7	18	239
F(x)	0	$\frac{1}{6}$	0	$\frac{1}{6}$	$\frac{5}{6}$	1	1

Percentile

- Le I^{th} percentile est la plus petite valeur p de l'échantillon telle que $F(p) \geq I\%$
- Exemple : Soit $X = -2, 7, 7, 4, 18, -5$
- on a alors:

  50^{th} percentile = 4 : $F(4) = \frac{3}{6} = 50\%$
  25^{th} percentile = -2 : $F(-2) = \frac{2}{6} \geq 25\%$
  100^{th} percentile =18 (vérifier)

EXEMPLE



Les deux séries sont statistiquement différentes. Mais quelle est la meilleure ?

Exemple: comparaison

Métrique	Série 1	Série 2
MIN	45	15
MAX	55	65
Moyenne	50	40
Ecart Type	2.9	14.4
P (. $>$ 40)	100%	50%
P(. $>$ 50)	50%	30%
P(. $>$ 60)	0%	10%

Métriques réseaux

Definitions

Une métrique est une caractéristique du comportement d'un réseau.
Elle peut servir à effectuer des comparaisons:

- Suite à des changements de configurations/pannes
- entre deux équipements dans un même réseau
- ou entre deux réseaux différents

Quelle métriques réseaux ?

- Les métriques doivent être concrètes et **bien définies**.
- Les métriques doivent être reproductibles

  Une mesure effectuée de multiples fois dans des conditions identiques doit donner des résultats identiques
- Les métriques doivent être utiles, pour les fournisseurs d'accès et les utilisateurs afin qu'ils comprennent les performances qu'ils perçoivent ou fournissent

Quelle métriques réseaux ?

- Les métriques doivent être concrètes et **bien définies**.
- Les métriques doivent être reproductibles

  Une mesure effectuée de multiples fois dans des conditions identiques doit donner des résultats identiques
- Les métriques doivent être utiles, pour les fournisseurs d'accès et les utilisateurs afin qu'ils comprennent les performances qu'ils perçoivent ou fournissent

Deux famille de métriques

1. **IPPM = IP Performance Metrics**

   Groupe de travail de l'IETF :
   `http://datatracker.ietf.org/wg/ippm/`
2. NM-WG: Network Measurements Working Group

   Groupe de travail du OpenGridforum :
   `http://www.ogf.org`

IPPM: Généralités

- Les unités de mesures utilisées sont celles du Système International d'Unités.
- L'unité d'information est le bit. 1 kbits = 1000 bits !
- Le temps utilisé est le temps UTC (Temps universel coordonné)
- Toutes les mesures qui ont étés définies sont des mesures directes, par injection de trafic

MÉTRIQUES IPPM: TERMINOLOGIE

- **Hôte** (Host) : Un équipement utilisant les protocoles de L'Internet (machine, routeur, etc.)
- **Lien** (Link) : Une connexion de niveau 2 entre deux hôtes.
- **Routeur** (Router) : un hôte qui facilite la communication au niveau 3 entre des hôtes.
- **Chemin** (path) : une séquence $h_0, l_1, h_1, \ldots, l_{n-1}, h_n$ où h_i est un hôte et l_j est un lien. **un chemin est unidirectionnel.**.
- **Nuage** (Cloud) : un graphe non orienté dont les sommets sont des routeurs et les arêtes des liens qui connectent des paires de routeurs

MÉTRIQUES IPPM: TERMINOLOGIE

- **sous-chemin** (Subpath) : Un sous-chemin d'un chemin est une sous-séquence d'un chemin qui est elle-même un chemin. (Débute et se termine par un hôte).
- **Point d'échange** (Exchange) : un lien qui interconnecte un hôte à un nuage ou un lien entre deux nuages.

Mesures du temps

- **Clock Offset** : décalage entre une horloge et le temps UTC
- **Accuracy** : plus l'offset est proche de zéro, plus l'horloge est précise
- **Résolution** : la valeur minimale de temps qu'une horloge sait mesurer
- **Skew** : différence de fréquence entre l'horloge et le temps UTC
- **Drift** : variation du skew au cours du temps

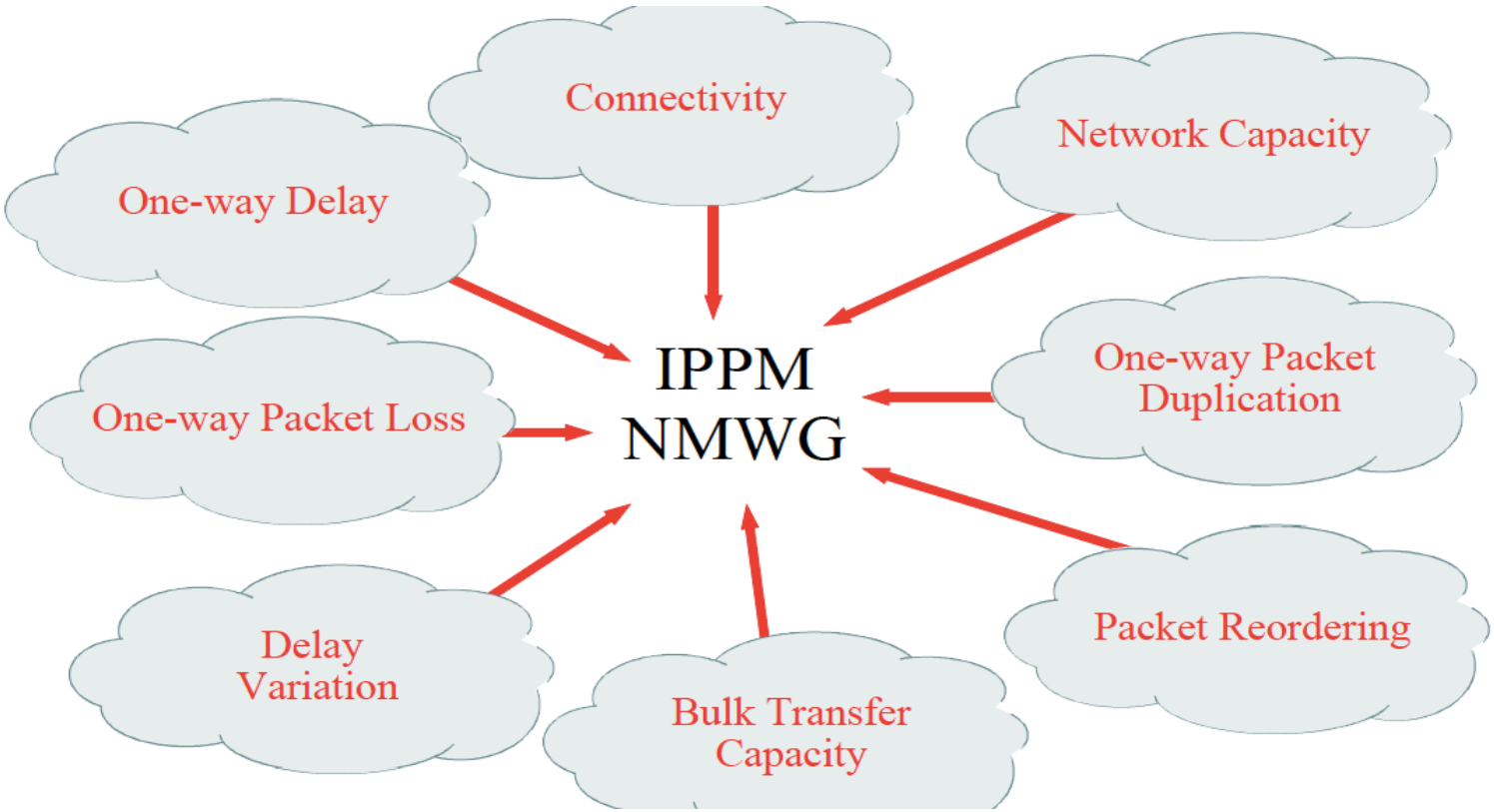
Mesures du temps

- Les mesures de temps ne doivent pas être faites au niveau du système ou au niveau applicatif
  - Les interruptions matérielles introduisent des délais inacceptables
  - La gestion du temps partagé entre les applications par les OS introduit aussi des délais trop importants
- **Temps filaire** : La mesure du temps de réception ou d'émission d'un paquet doit être réalisée le plus près possible du fil

Temps filaire

- Soient *P* un paquet, *H* un hôte et *L* l'endroit où *H* observe un lien Internet
- **Le temps d'arrivée filaire** de P sur H au niveau de L est le premier instant T où un bit quelconque de P est apparu en L
- **Le temps de départ filaire** de P de H est le premier instant T où tous les bits de P sont apparus en L

Les métriques: IPPM



MÉTRIQUE: TYPE-P

- La plupart les métriques d'Internet sont dépendantes du type de paquet utilisé (protocole, taille, ports src/dst)
- Quand une métrique dépend du type de paquet utilisé, son nom doit contenir l'expression Type-P pour la rendre générique
- Pour faire une mesure, il faut d'abord spécifier plus précisément le type de paquet qui sera utilisé
  `type-P-metric` → `TCP-src1550-dst80-size800-metric`

IPPM: Les principaux RFC

- RFC 2330 : Framework for IP Performance Metrics
- RFC 2678 : IPPM Metrics for Measuring Connectivity
- RFC 2679 : A One-way Delay Metric for IPPM
- RFC 2680 : A One-way Packet Loss Metric
- RFC 3357 : One-way Loss Pattern Sample Metrics
- RFC 2681 : A Round-trip Delay Metric for IPPM
- RFC 3393 : IP Packet Delay Variation Metric
- RFC 4737 : Packet Reordering Metric

Mesure de connectivité

Type-P-Instantaneous-Unidirectional-Connectivity(Src, Dst, T) : Booléen

- Src et Dst ont une connectivité à sens unique instantanée de type-P à l'instant T si un paquet de type-P transmis de Src à l'instant T arrive à Dst

Type-P-Instantaneous-Bidirectional-Connectivity(A1, A2, T): Booléen

- A1 et A2 ont une connectivité bi-directionnelle instantanée de type-P à l'instant T si un paquet de type-P transmis de A1 à l'instant T arrive à A2 et si un paquet de type-P transmis de A2 à T+dT arrive à A1

Mesure de connectivité

Type-P-Interval-Unidirectional-Connectivity(Src, Dst, T, dT) :Booléen

Type-P-Interval-Bidirectional-Connectivity(A1, A2, T, dT) :Booléen

- il suffit que la métrique instantanée correspondante soit vraie pour un seul instant dans l'intervalle [T, T+dT] pour que la métrique soit vraie

Mesure de connectivité

Type-P1-P2-Interval-Temporal-Connectivity(Src, Dst, T, dT) :Booléen

- S'il existe $t_i \in [T, T + \Delta T]$ tel que :

  Src a une connectivité unidirectionnelle instantanée de type-P1 avec Dst à l'instant t_i
  Dst a une connectivité unidirectionnelle instantanée de type-P2 avec Src à l'instant $t_i + \delta_t$ où δ_t est le temps de transit entre Src & Dst pour un paquet de type P1

  Alors il y a une connectivité bi-directionnelle de type-P1-P2 entre Src et Dst dans l'intervalle [T, T+dT]

Exemple

TCP-port-N1-port-N2-Interval-Temporal-Connectivity(A1, A2, T, dT)

- Envoi par A1 d'un paquet SYN vers A2
- Vrai si Réception par A1 d'un paquet SYN-ACK en provenance de A2
- Vrai si Réception par A1 d'un paquet RST en provenance de A2
- Vrai si Réception par A1 d'un paquet ICMP Port unreachable en provenance de A2 Faux si Réception par A1 d'un paquet ICMP host unreachable ou network unreachable en retour du paquet aller

OWD

- La borne inférieure du One-Way Delay est la somme des temps de propagation et de sérialisation le long du chemin
- Les valeurs de cette métrique au-dessus de la borne inférieure donnent une indication de la charge du chemin
- Le OWD donne une indication plus fine que le Round Trip Delay
  - Routage asymétrique
  - Il peut y avoir du queuing asymétrique
  - La performance d'une application peut être impactée par le temps de transit dans un sens, pas par celui dans l?autre (Transfert sur TCP)
  - Dans un réseau avec de la qualité de service, les marquages ne sont pas forcément symétriques

OWD

Type-P-One-way-Delay(Src, Dst, T) = dT

- T = wire time émission
- T+dT = wire time réception
- Si le paquet est dupliqué sur le chemin, dT est calculé au wire time de l'arrivée du premier paquet
- Si le paquet est fragmenté et non réassemblé, il est estimé perdu

OWD

Type-P-One-way-Delay-Percentile(Stream, X) = dT

- Valeur de dT qui sépare les échantillons en 2 parties (X% et (100-X)%)
- Stream0 = $(< T0, 100ms >, < T1, 110ms >, < T2, indéfini >, < T3, 90ms >, < T4, 500ms > \rightarrow TPOWDP(Stream0, 50\%) = 110ms$

OWD

Type-P-One-way-Delay-Median(Stream) = dT

- La valeur médiane de l'échantillon. Diffère du 50ème percentile seulement sur un flux avec un nombre pair d'échantillons (moyenne des deux échantillons du milieu)
- $Stream_0 = (< T0, 100ms >, < T1, 110ms >, < T2, indéfini >, < T3, 90ms >, < T4, 500ms$
  $\rightarrow TPOWDM(Stream0) = 105ms((100 + 110)/2)$

Type-P-One-way-Delay-Inverse-Percentile(Stream, treshdold) = X%

- Le pourcentage des échantillons dont la valeur dT est ¡= threshold
- $Stream0 = (< T0, 100ms >, < T1, 110ms >, < T2, indéfini >, < T3, 90ms >, < T4, 500ms > \rightarrow TPOWDIP(Stream0, 100) = 2/5 = 40\%$

Round-trip Delay

Type-P-Round-trip-Delay(Src, Dst, T) = dT

T = temps filaire d'émission du paquet au départ de Src

T+dT =temps filaire de réception du paquet retour par Src

- Type-P-Round-trip-Delay-Percentile(Stream, X%) = dT
- Type-P-Round-trip-Delay-Median(Stream) = dT
- Type-P-Round-trip-Delay-Minimum(Stream) = min(dT)
- Type-P-Round-trip-Delay-Inverse-Percentile(Stream, dT) = X%

One-way Packet Loss

- La métrique One-way Packet Loss est plus précise que le Round Trip Packet Loss, le chemin aller n'est pas forcément le même que le chemin retour
- Certaines applications sont plus sensibles aux pertes de paquets dans un sens que dans l'autre, les transferts de fichiers, par exemple :

  Acquittements dans l'autre sens pour quelques paquets seulement
- Mise en file d'attente asymétrique
- ...

One-way Packet Loss

Type-P-One-way-Packet-Loss(Src, Dst, T): boolean
Type-P-One-way-Packet-Loss-Average = Avg(L0...Lf)

- $Stream = (< T0, 0 >, < T1, 1 >, < T2, 0 >, < T3, 1 >, < T4, 0 >, < T5, 0 >, < T6, 0 >) = 0,286 = 28\%$ de pertes

VARIATION DE DÉLAI (GIGUE)

$$Type - P - One - way - ipdv(Stream(Src, Dst, I1, I2, \lambda), L, F) = ddT, T_1, T_2$$

- F : fonction de sélection de deux paquets émis à t_1 et t_2
- L : longueur des paquets émis
- ddT = dt_2 - dt_1 où dt_1 est le OWD(Src, Dst, t_1) et dt_2 = OWD(Src, Dst, t_2)

RÉORDONNANCEMRNT DE PAQUETS

- Les paquets peuvent arriver dans un ordre différent de l'ordre d'émission
- Exemples:

  Quand les paquets d'un même flux empruntent des chemins avec des durées différentes d'acheminement des paquets (load balancing, par ex)
  Équipements réseaux avec plusieurs processeurs par ports
  Retransmission par un protocole de niveau 2
  Algorithme de gestion de files d'attente traversés par les paquets
  Assignation de paquets successifs dans des buffers différents avec des taux d'occupation différents

RÉORDONNANCEMRNT DE PAQUETS

Type-P-Reordered(Src, Dst, SrcTime, s, NextExp): Booléen

- SrcTime : Temps d'émission du paquet
- s : numéro de séquence
- NextExp : Le numéro de séquence attendu

```
if s >= NextExp, then /* s est dans l'ordre */
        if s > NextExp then
                SequenceDiscontinuity = True;
                SeqDiscontinuitySize = s - NextExp;
        else
                SequenceDiscontinuity = False;
        NextExp = s + 1;
        Type-P-Reordered = False;
else /* quand s < NextExp */
        Type-P-Reordered = True;
        SequenceDiscontinuity = False;
```

RÉORDONNANCEMRNT DE PAQUETS

Type-P-Reordered-Ratio-Stream(Src, Dst, T0, Tf, K, L) = R

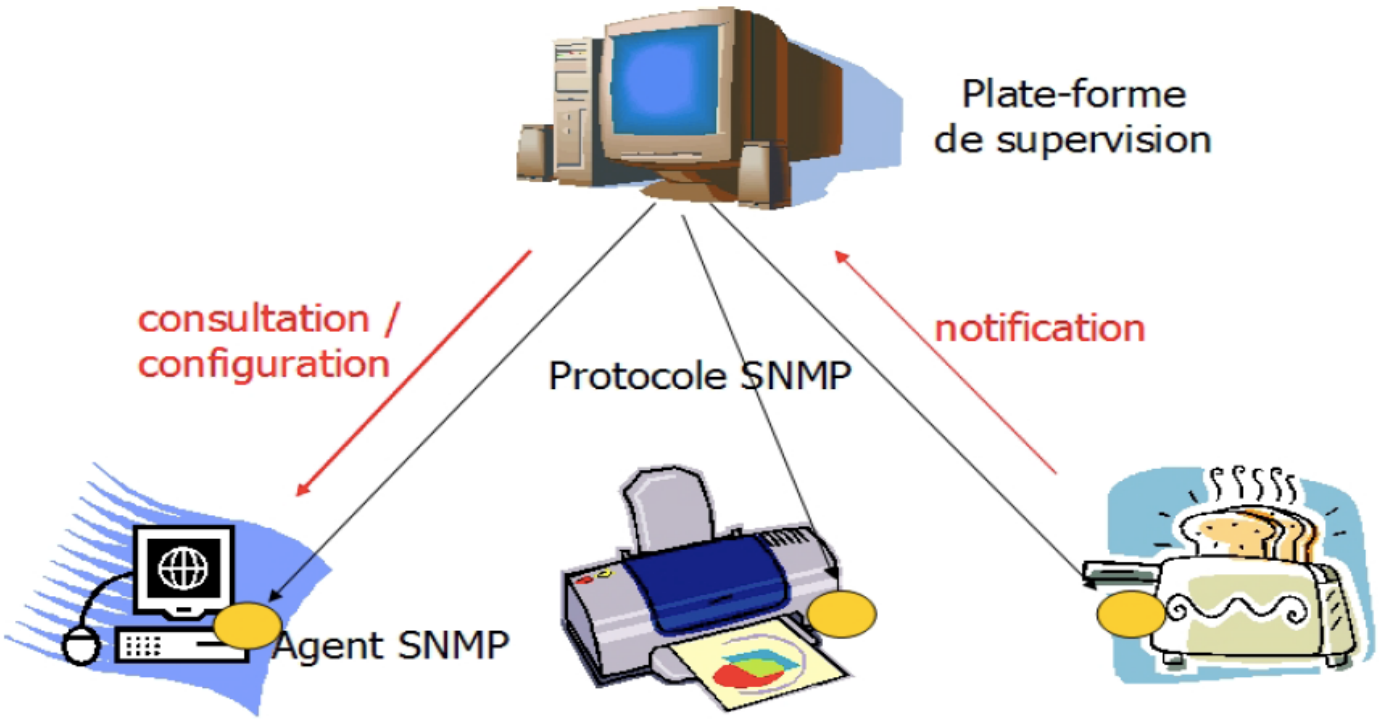
- T0, Tf temps de début et de fin
- dT = temps de transit max (au-delà de dT, le paquet est déclaré perdu)
- K le nombre de paquets envoyés de Src
- L, le nombre de paquets reçus parmi K $L <= K$
- R = le ratio du nombre de paquets réordonnée par rapport au nombre de paquets reçus :

$$R = \frac{Count(Type - P - Reordered = True)}{L}$$

Le protocole SNMP

- SNMP: Simple network Management Protocol
- Trois versions :
  - `SNMPv1`: RFC 1157 (1989 draft), est toujours en activité !
  - `SNMPV2c`: RFC 3417, 3417, 3418. (1993), la version la plus utilisée
  - `SNMPv3`: RFC 3410-18n RFC2576. Apport renforcer la sécurité de l'échange.

SNMP: Principe



- Plate-forme de supervision: NMS (Network Management Station)

SNMP: TERMINOLOGIE

- Chaque agent SNMP maintiens un ensemble d'objets d'observation qui permettent au NMS de qualifier les performances de l'équipement supervisé par l'agent.
- Un groupe d'objets d'observation forme un **MIB**: *Management Information Base*
- Les objets d'un MIB sont définis d'une manière formelle en utilisant une langage: **SMI** (*Structure of Management Information*)
- Un agent peut implementer différentes MIB
  ATM (RFC 2515), MAIL (RFC 2619), DNS (RFC 1611)
- Chaque agent implémente le **MIB-II** (RFC 1213): TCP/IP management information

SNMP: Généralités

- SNMP s'appuie sur UDP.
- **Port 161** : pour envoi de requêtes et réception de réponses
- **Port 162** : pour réception de notifications
- Mécanismes de Time-out pour détection de perte de datagramm UDP.
- Les agents SNMP fonctionnent en mode : *sans état*.

SNMP Les communautés

- SNMPv1 et SNMPv2 s'appuient sur le concept de communauté pour authentifier l'échanger entre NMS et un agent
- Trois communautés sont définies:

  `read-only` (par défaut public)
  `read-write` (par défaut private)
  `trap`
- Communauté $\sim$ mot de passe
- L'échange se fait en claire !
- A changer régulièrement

ce n'est plus le cas dans la version 3

SMI: Structure of management Information

- Langage de définition de contenu des MIB
- `SMIv1` (RFC 1155)
- `SMIv2` (RFC 2578) : enrichissement de SMIv1 par de nouveaux types
- Un objet est défini par:

  `Identifiant (OID)`:un nom qui permet d'identifier d'une manière unique un objet
  `Type` et `Syntaxe` : type de données
  `Codage (encoding)` : Définition de schéma de codage pour la transmission des objets

Gestion de noms (OID)

- Les objets à gérer sont organisés dans une hiérarchie (arbre) standardisé et extensibles
- Chaque nœud dans l'arbre porte un nom symbolique et un numéro
- le nom d'un objet est donné par la suite des noms symboliques (ou numéro), séparés par "." des nœdus qu'il faut traverser pur localiser l'objet depuis la racine.

Arbre OID


Root-Node
ccitt(0)
iso(1)
joint(2)
org(3)
dod(6)
internet(1)
directory(1)
mgmt(2)
experimental(3)
private(4)
mib-2(1)
system(1)
interfaces(2)
at(3)
ip(4)
icmp(5)
tcp(6)
udp(7)
egp(8)
transmission(10)
snmp(11)

Définition de OID: exemples

```
internet OBJECT IDENTIFIER ::= { iso org(3) dod(6) 1
}
directory OBJECT IDENTIFIER ::= { internet 1 }
mgmt OBJECT IDENTIFIER ::= { internet 2 }
experimental OBJECT IDENTIFIER ::= { internet 3 }
private OBJECT IDENTIFIER ::= { internet 4 }
```

OID internet: 1.3.6.1

OID mgmt: 1.3.6.1.2

...

OID: Extension

- Le nœud `private/entreprises` peut être étendu pour representer des OID fournies par des entreprises.
- Les numéros des entreprises sont gérés par l'IANA
- Exemple: CISCO a le numéro 9
- Le prefix des objets CISCO est alors: `1.3.6.1.4.1.9`

MIB II

- `system` : information générale sur le système

  `OBJECT IDENTIFIER ::= { mib-2 1 }`
- `interfaces` : informations sur chacune des interfaces reseau

  `OBJECT IDENTIFIER ::= { mib-2 2 }`
- `at` : table de correspondance IP - Physique

  `OBJECT IDENTIFIER ::= { mib-2 3 }`
- `ip` informations sur le fonctionnement d'IP

  `OBJECT IDENTIFIER ::= { mib-2 4 }`
- `icmp` : informations sur le fonctionnement d'ICMP

  `OBJECT IDENTIFIER ::= { mib-2 5 }`

MIB II

`http://rc.net/rfc.1213.html`

- `tcp`

  `OBJECT IDENTIFIER ::= { mib-2 6 }`

- `udp`

  `OBJECT IDENTIFIER ::= { mib-2 7 }`

- `egp`

  `OBJECT IDENTIFIER ::= { mib-2 8 }`

- `transmission` informations sur les modes de transmission et les protocoles d'accès à chaque interface

  `OBJECT IDENTIFIER ::= { mib-2 10 }`

- `snmp`

  `OBJECT IDENTIFIER ::= { mib-2 11 }`

Le groupe System: Exemples

- `sysDescr (1.3.6.1.2.1.1.1)`

  Description textuelle du système (hardware et SE)

- `sysObjectID (1.3.6.1.2.1.1.2)`

  Type d'équipements

- `sysUpTime (1.3.6.1.2.1.1.3)`

  temps depuis le dernier démarrage du système

- `sysContact (1.3.6.1.2.1.1.4)`

  coordonnées de l'administrateur

- `sysName (1.3.6.1.2.1.1.5)`

  Nom symbolique de l'équipement FQDN

- `sysLocation (1.3.6.1.2.1.1.6)`

  emplacement de l'équipement

- `sysServices (1.3.6.1.2.1.1.7)` niveaux des services rendus par l'équipement :

  $\sum_{i=1}^{7} a_i \times 2^{i-1}$ $a_i = 1$ si le système rends un service de niveau i, 0 sinon

SMIv1: DÉFINITION D'OBJET

$< name >$ **OBJECT-NAME**
SYNTAX $< data - type >$
ACCESS $< read - only | read - write | write - only | not - accessible >$
STATUS $< mandatory | optional | obsolete >$
DESCRIPTION `description de l'objet`
$::=$ { $< OID - unique >$ }

DÉFINITION D'OBJET: EXEMPLE

sysLocation **OBJECT-NAME**
SYNTAX `DISPLAYString (SIZE (0..255))`
ACCESS `read-only`
STATUS `mandatory`
DESCRIPTION *`description de l'objet`*
`::=` { < *system* 6 > }

SMIv1: Types de données simples

- INTEGER
- OCTET STRING
- Counter : un compteur codé sur 32 bits qu'on peut incrémenter
- OBJECT IDENTIFIER
- IpAddress
- NetworkAddress
- Gauge un compteur qu'on peut incrémenter et décrémenter
- TimeTicks
- Opaque image binaire d'un type de données

SMIv1: Types de données composés

- SEQUENCE

  Un ensemble d'éléments de types variés

- SEQUENCE OF

  Une séquence d'éléments de même type

TYPES COMPOSÉS : EXEMPLES

MIB II : 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 }
```

TYPES COMPOSÉS : EXEMPLES

MIB II : `IfEntry`

```
IfEntry ::= SEQUENCE {
ifIndex INTEGER,
ifDescr DisplayString,
ifType INTEGER,
ifMtu INTEGER,
ifSpeed Gauge,
... }
```

Désignation d'instances d'objets

- Variable scalaire : `OID.0`.

```
snmpget -v 1 -c public 127.0.0.1
1.3.6.1.2.1.1.4.0
```

- Tableaux : `OID.C.I`

C est le numéro de colonne, I indice dans la colonne C

SMIv2: DÉFINITION D'OBJET

$< name >$ **OBJECT-NAME**
SYNTAX $< data - type >$
UnitParts $< unitédemesures >$
MAX-ACCESS $< read - only | read - write | \mathbf{read - create} | not - accessible | \mathbf{accessible - for - notify} >$
STATUS $< current | deprecated | obsolete >$
DESCRIPTION `description de l'objet`
AUGMENTS `nom de table`
$::= \{ < OID - unique > \}$

Les notifications (trap)

- **trap** (une notification) : message envoyé par un agent à un NMS à l'initiative de l'agent.
- `traptype` **NOTIFICATION-TYPE**
  OBJECTS $\{oid_1, \ldots, oid_n\}$
  STATUS $< current|deprecated|obsolete >$
  DESCRIPTION `description de la notification`
  `::=` { $< OID - unique >$ }

TRAP: UN EXEMPLE

```
linkup NOTIFICATION-TYPE
OBJECTS {ifIndex}
STATUS current
DESCRIPTION "A linkup trap signifies that the
entity has detected the the iffierStatus object has
changed to up"
::= { snmptraps 3 }
```

Traps génériques

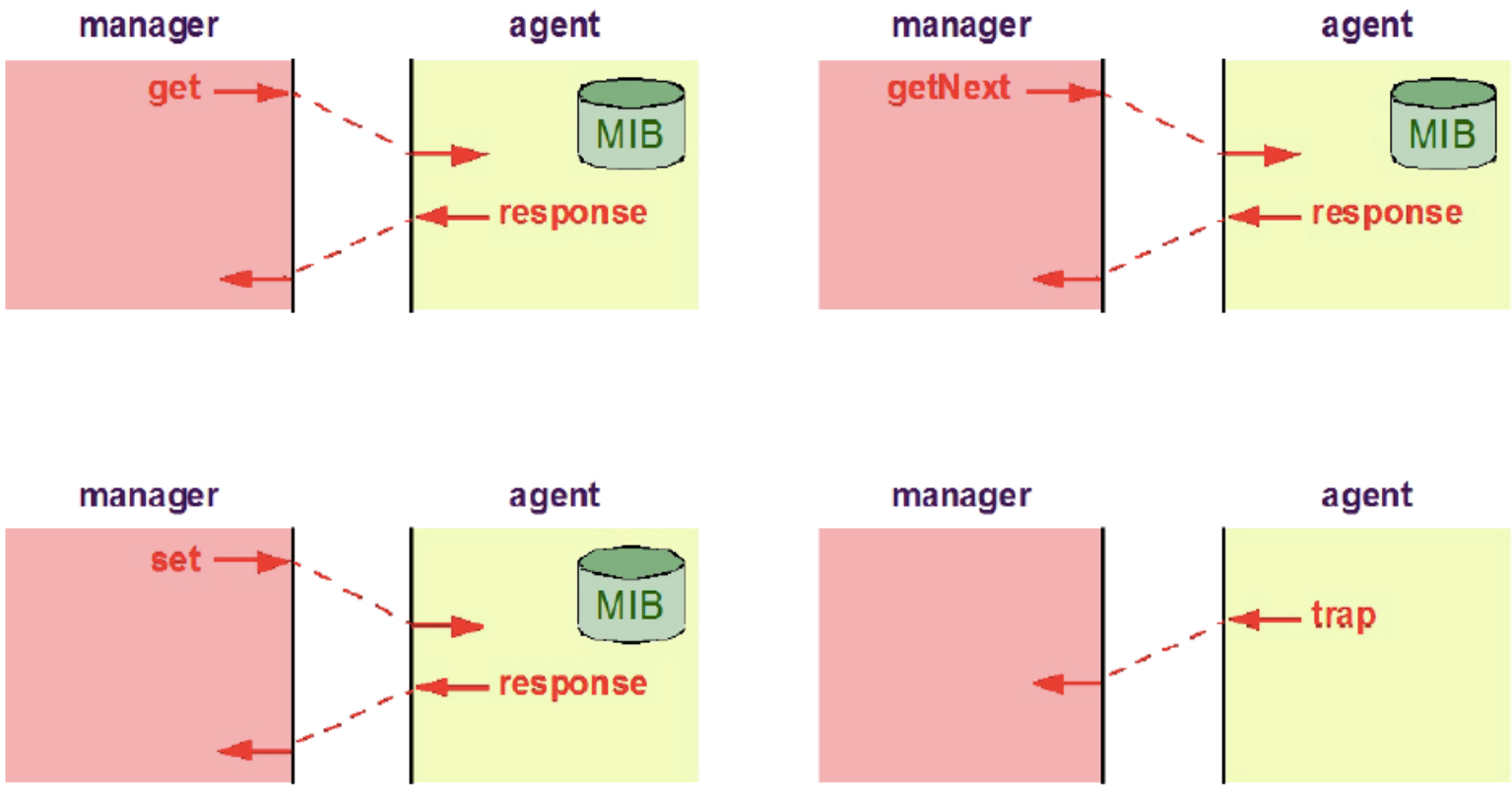
- `coldstart (0)` : démarrage de l'agent (ou de l'équipement)
- `warmstart (1)` : reinitialisation de l'agent.
- `linkDown (2).`
- `linkUp (3).`
- `authentificationFailure (4) .`
- `...`

Définition de groupes

- Un groupe d'objets = ensemble d'objets qui doivent être toujours ensemble.
- Un objet peut appartenir à différents groupes
- Exemple :

  `Groupname` **OBJECT-GROUP**
  OBJECTS $\{oid_1, \ldots, oid_n\}$
  STATUS $< current|deprecated|obsolete >$
  DESCRIPTION `description de la notification`
  $::= \{ < OID - unique > \}$

SNMPv1 : Opérations



OPÉRATIONS `Get`

- **GetRequest** : permet d'interroger un agent sur une ou plusieurs variables
  - **Opération atomique**: Toutes les variables indiquées dans la requête doivent exister, sinon retour de code d'erreur.
  - Erreurs : `noSuchName`, `tooBig`, `...`, etc.
- **GetResponse** : Message envoyé par un agent en réponse à une `GetRequest`.

Opération `GetNext`

- Permet d'interroger un agent sur la variable suivante de la MIB.
- Le parcours de l'arbre se fait en ordre lexicographique.
- Opération utile pour découvrir la structure d'une MIB, de récupérer les lignes d'un tableau
- Attention: Cette commande peut générer un trafic important
- Erreurs : `noSuchName` (fin de la MIB), `tooBig`, `...`, etc.

Opération `setRequest`

- Permet d'assigner une valeur à une ou plusieurs variables (scalaires, création d'une ligne dans un tableau)
- **Opération atomique**
- Erreurs : `noSuchName, badValue, tooBig,...`, etc.

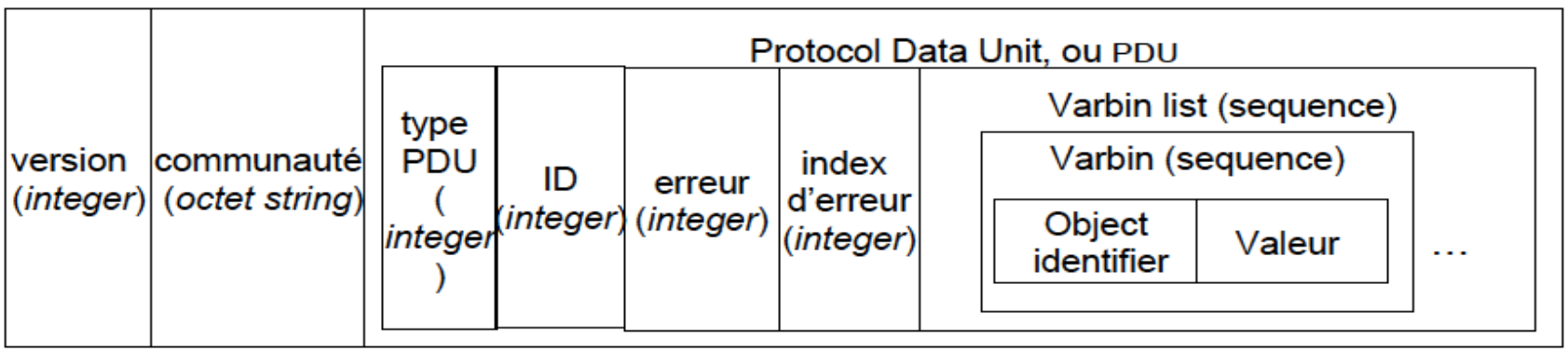
Opération `trap`

- Permet à un agent d'énvoyer une notification à un NMS
- Pas d'acquittement !
- L'agent envoie une notification avec un `TrapID` et un `OID`.

SNMPv1: Format de message

- Un message SNMP est formé de 3 champs :
- **version** : Numéro de version du protocole snmp : **0 pour SNMPv1**
- **communauté** : mot de passe pour un groupe de manager. Un agent ne connaissant pas le nom de la communauté est exclu
- **PDU** : (Protocol Data Unit) dont la composition varie en fonction du typ d'opération : requête, réponse ou notification.

SNMPv1: PDU requête/réponse



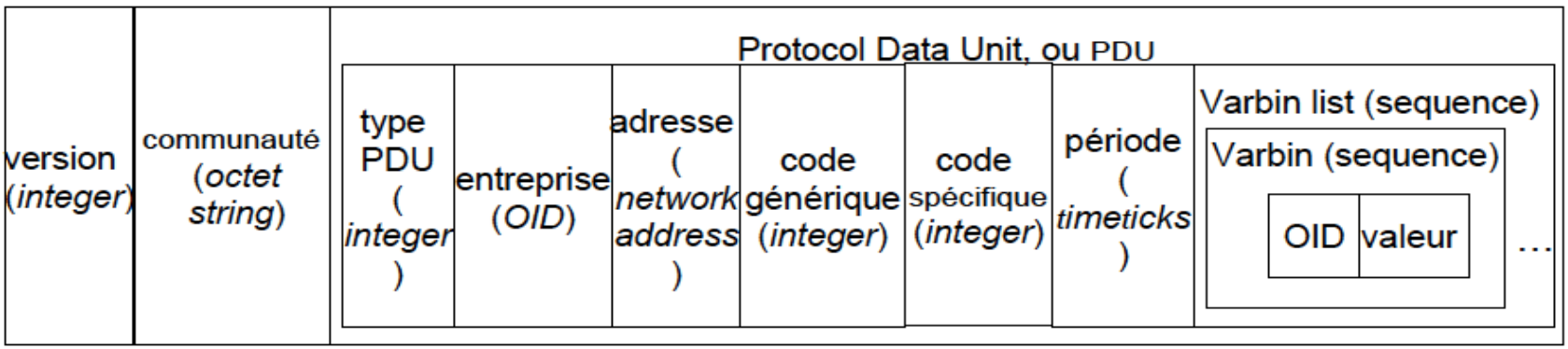
SNMPv1: PDU requête/réponse

- **ID** : Identifiant de la requête et la réponse associé.
- **Erreur** : 0 dans un message envoyé par un MNS.
- **Index** : pointeur vers le premier objet ayant causé l'erreur. 0 si pas d'erreur.
- **Varbin** : séquence de couples $< OID, Valeur >$.

SNMPv1: codes d'erreur

- `0x00` : pas d'erreur
- `0x01` : toobig ! réponse trop longue pour être transportée
- `0x02` : le nom de l?objet demandé n'a pas été trouvé
- `0x03` : le type de donnée de la requête ne corresponds pas à celui se trouvant dans l'agent SNMP
- `0x04` : tentative de modification d'une variable accessible en read-only
- `0x05` : Erreur générale
- `0x06` : Accès non permis
- ...

SNMPv1: PDU TRAP



SNMPv1: PDU TRAP

- **Entreprise** : identification de l'objet administré générant la notification.
- **Adresse** : adresse de l'agent ayant généré la notification.
- **code générique**: code standard de trap.
- **code spécifique** : code non standard (défini par l'entreprise)
- **Période** : temps écoulé depuis la dernière trap.

SNMPv2

- Nouvelles opérations : `GetBulk, notification, inform, report`
- Opération `GetRequest` et `setRequest` ne sont plus atomiques.
- Enreichissement des codes d'erreurs et de codes de trap
- Possibilité de communication entre NMS et pas seulement entre NMS et agents.
- Simplification du PDU trap.
- Indépendance de UDP : SNMPv2 opère aussi sur DDP (Appletalk), IPX, CLNS.

SNMPv2: OPÉRATION `GetBulk`

- Permet à NMS de récupérer d'une manière plus efficace qu'avec `getNext` de grande quantité de variables de la MIB.
- éviter la génération d'un code `toobig` en informant l'agent de la possibilité de diviser la réponse en *bulks*.
- Deux paramètres :

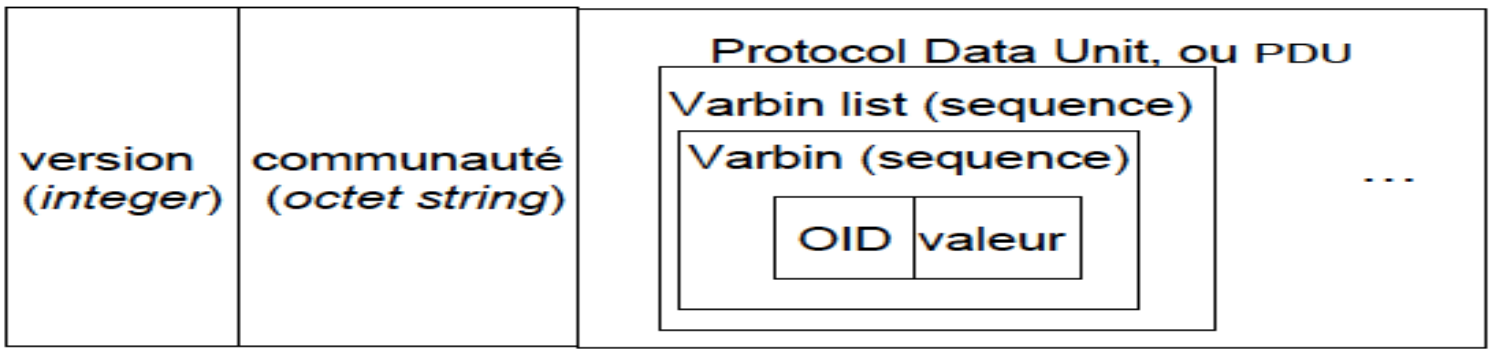
  `nonrepeaters n`: indique que le *n* premiers variables sont des scalaires

  `max-repetitions m`: indique qu'il faut utiliser au max. M `getNext` opérations pour avoir les autres variables.

SNMPv2: OPÉRATIONS `inform, report`

- `inform` : Une notification (trap) mais avec acquittement
- Les deux opérations peuvent être échangées entre deux NMS.
- `report` : Pas de sémantique prédéfinie
- permettent de construire une **architecture hiérarchique** de supervision.

SNMPv2: PDU TRAP

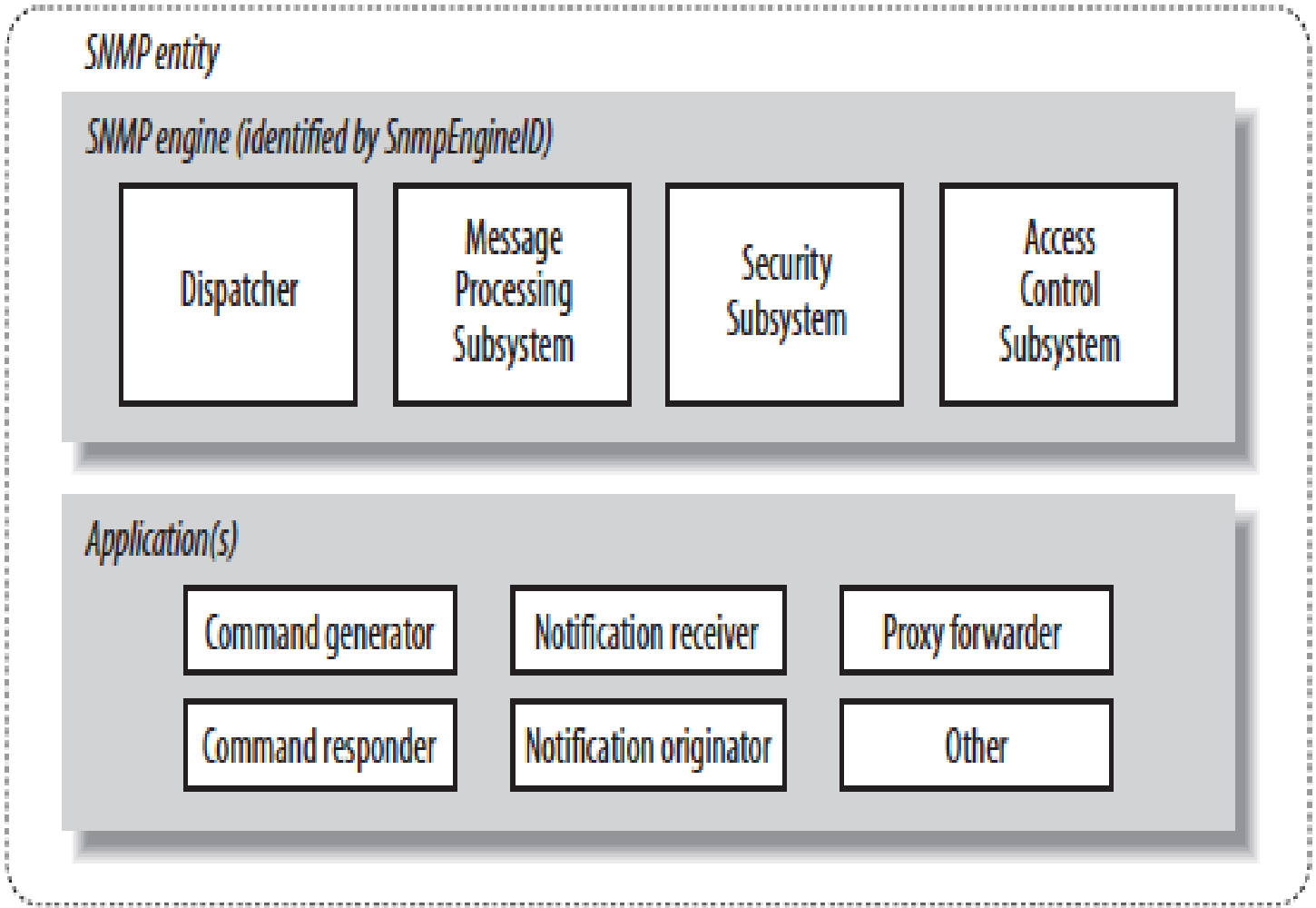


même format que les requêtes/réponses

SNMPv3

- Vers une architecture `P2P` (peer-to-peer)
- Que des `entités` snmp et plus de NMS et agents.
- Une entité est composée d'un engin de base et des applications
- Renforcer les aspects liés à la sécurité : cryptage des échanges, droits d'accès, certificat d'authentification d'entités, synchronisation d'horloges.

SNMPv3: Architecture d'une entité



NET-SNMP

- Une boîte-à-outils linux pour l'utilisation de `snmp`
- Exemples de commandes :
  - `snmptranslate, snmpbulkwalk` : explorer l'arbre d'une MIB
  - `snmpget, snmpgetnext, snmpbulkget` : pour envoyer des requêtes.
  - `snmptrap` : envoyer et traiter des notifications.
  - `snmptable` : afficher le contenu d'une table
  - `snmpset` : effectuer de requêtes de type set.
  - ...

NET-SNMP : EXEMPLES

- **snmpget -v1 -c public localhost .1.3.6.1.2.1.1.6.0**
  `SNMPv2-MIB::sysLocation.0 = STRING: Villetaneuse`
- **snmpset -c private 127.0.0.1 .1.3.6.1.2.1.1.4.0 octetstring "R. Kanawati rushed.kanawati@lipn.fr"**
  `system.sysContact.0 : DISPLAY STRING- (ascii): R. Kanawati rushed.kanawati@lipn.fr`
- d'autres exemples aux prochains TP !