L'autonomique dans les réseaux et le besoin en connaissances

Sommaire

1.1 Introduction

Toute architecture doit pouvoir évoluer pour s'accommoder avec les nouvelles tendances, or, Internet n'a que très peu évolué depuis sa création. En effet, le réseau Internet a été conçu pour le transport d'un trafic de type texte très peu volumineux, alors qu'aujourd'hui, il est plutôt utilisé pour le trafic multimédia. Selon Cisco [2013], le trafic vidéo sur Internet représentera 69% de tout le trafic grand public en 2017. En 2012, il n'était que de 57%. Ce pourcentage sera de l'ordre de 80% à 90% du trafic mondial consommé d'ici 2017 en incluant les vidéos échangées via les réseaux pair-à-pair (P2P). Ce

type de trafic impose que l'acheminement des flux soit assuré avec une Qualité de Service (QdS) maîtrisée, chose que l'architecture actuelle des réseaux a du mal à fournir puisqu'elle n'a pas été conçue pour cela.

Partant de ce constat, on assiste, depuis peu, à un engouement de la communauté dans un effort d'améliorer l'architecture d'Internet de manière à l'adapter aux nouveaux types de contenus. Comme le suggèrent Ghodsi et al. [2011], plusieurs études ont été menées sur des réseaux centrés sur le contenu ou les données. Nous trouvons donc dans la littérature des appellations comme *Data-Oriented Network, Concent Centric Network, Content-based Network, Named-Data Network,...*. Malgré les divergences que peuvent avoir ces différentes propositions, elles peuvent être considérées comme faisant partie intégrante des réseaux centrés sur l'information (*Information Centric Networks*, ICN) dans la mesure où elles s'accordent sur deux points clés :

Le premier concerne un mécanisme de gestion du contenu cache. Celui-ci a deux comportements :

i) Si un noeud demande une donnée à un autre noeud qui l'a dans son cache, ce dernier lui renvoie directement la donnée en question,
ii) Si la donnée n'est pas en cache, ce dernier la demande à son tour et la met en cache.

Cette approche a des conséquences sur les mécanismes de sécurité. En effet, là où il suffisait de sécuriser le serveur original des données et le chemin emprunté par les paquets, il faut sécuriser dorénavant le contenu lui même dans une approche à base d'ICN. Cette sécurité est assurée par un mécanisme de certificats : le serveur original signe le contenu, ainsi tous les noeuds du réseau et les usagers peuvent, facilement, en vérifier la validité.

Le second point a trait au mécanisme de diffusion/récupération de l'information (ou requête/réponse). Certes, ce n'est pas le paradigme le plus récent étant donné qu'il est largement étudié depuis une vingtaine d'années, mais il s'agit d'un composant majeur des ICNs. Même si le nom de ce concept change d'une approche à une autre, celles-ci s'accordent toutes sur le fait que ce mécanisme doit être composé de deux primitives :

i) La publication des informations : Elle permet à un noeud de mettre à disposition de l'ensemble du réseau les informations à sa disposition.
ii) La récupération des informations : Elle consiste à permettre à tout noeud de faire des requêtes sur les informations publiées.

Toutefois, et afin de réponde rapidement et de la manière la plus adéquate aux besoins changeants et volatils des utilisateurs, ces nouvelles approches augmentent la complexité du contrôle dans le réseau. Le nouveau paradigme des réseaux logiciels, plus connus sous l'appellation anglo-saxone *Software Defined Networks* (SDN), tente de répondre à cette problématique [Open-Networking-Foundation, 2012]. Il s'agit d'une nouvelle approche dans laquelle le contrôle est découplé du matériel et donné à un logiciel appelé contrôleur. Le but des SDNs est de permettre aux ingénieurs et aux administrateurs de construire des "réseaux à la carte". Les réseaux, ainsi construits, sont pilotés à partir d'un pupitre de commande "centralisé" sans avoir à intervenir sur les commutateurs individuellement. Toutefois, la nature des flux circulant sur le réseau et la constante progression des besoins des utilisateurs font qu'un contrôle centralisé et en partie manuel n'est pas viable à long terme. Il nous semble donc nécessaire d'opérer une évolution technologique du réseau. Cette évolution, qui consiste à intégrer une automatisation de certaines tâches dans la gestion du réseau, est appelée approche autonomique.

1.2 Approche autonomique

Les réseaux autonomiques représentent un concept qui a pour ambition de rendre le réseau indépendant de tout pilotage humain. Une telle approche répond parfaitement à la problématique actuelle qui, compte tenu de l'accroissement des besoins, consiste à continuer à offrir aux utilisateurs un service fiable et sans couture, connu sous l'appellation anglo-saxone "seamless service" (le système doit gérer dynamiquement la session de l'utilisateur et son unicité de bout en bout en temps réel). L'idée repose sur une adaptation du concept du système autonome introduit par IBM en 2001 [Kephart and Chess, 2003 ; Kephart, 2005] à la thématique des réseaux informatiques. Ce type d'approche a pour ambition de répondre à la complexité grandissante des réseaux et de permettre ainsi leur expansion future au-delà de leur taille actuelle.

1.3 Définition des réseaux autonomiques

Les réseaux autonomiques sont des réseaux qui offrent des capacités "d'auto-*" sans aucune intervention externe. D'un point de vue architectural, ils sont basés sur un ensemble d'éléments autonomes appelés *Autonomic Elements* (AEs). Comme décrit dans la figure 1.1, un AE a 4 caractéristiques

principales [Sterritt et al., 2005] :

— se connaître soi-même (*self-awareness*),
— connaître son environnement (*environment-awareness*),
— s'auto-monitorer (*self-monitoring*),
— s'auto-ajuster (*self-adjusting*).

Ces caractéristiques constituent les conditions nécessaires pour qu'un système autonome puisse assurer les 4 objectifs décrits par Sterritt et al. [2005] :

— auto-configuration (*self-configuring*),
— auto-réparation (*self-healing*),
— auto-optimisation (*self-optimising*),
— auto-protection (*self-protecting*),

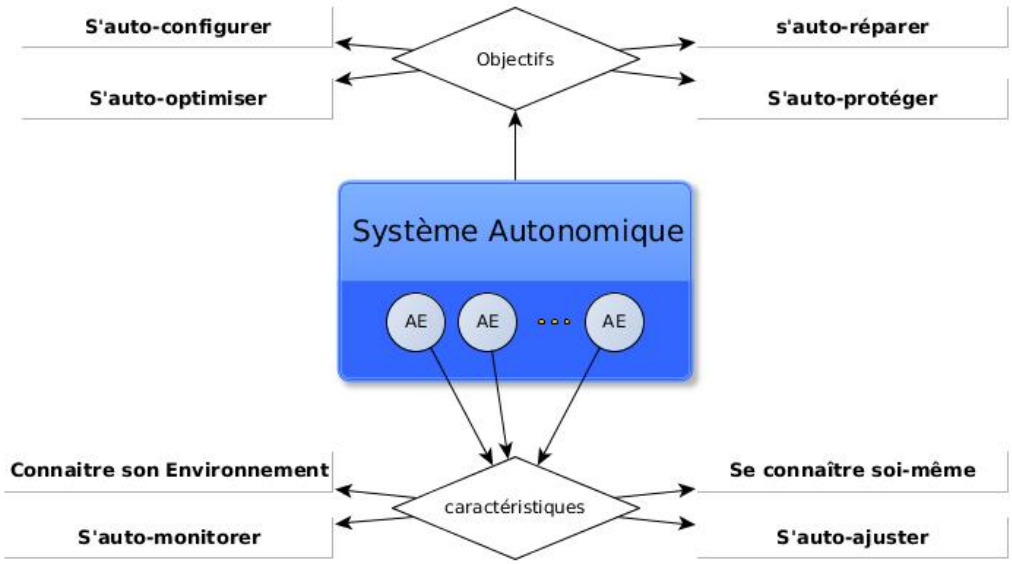

FIGURE 1.1 – Propriétés d'un système autonomique

1.4 Caractéristiques d'un réseau autonomique

Du point de vue architectural, un AE se doit de répondre à quatre caractéristiques : *self-awareness*, *environment-awareness*, *self-monitoring* et *self-adjusting* [Kephart and Chess, 2003 ; Kephart, 2005]. Nous définissons, dans la suite, chacune d'entre elles :

- ***Self-awareness*** est la capacité d'un individu de se connaître soi-même indépendamment de l'environnement et des autres individus.

  D'un point de vue philosophique, la connaissance de soi même peut être résumée par la citation : « cogito ergo sum » (Je pense, donc je suis).

  Du point de vue purement technique, un AE doit donc avoir connaissance de ses propres ressources, ses composants, ses performances et son état interne.
- ***Environment-awareness*** (connaître son environnement) consiste à connaître l'ensemble des mécanismes avec lesquels le système est en interaction. Cette caractéristique a une incidence sur la faculté du système à s'adapter aux différents contextes.
- ***Self-monitoring*** (auto-monitoring) est la faculté d'un système de suivre par soi-même et régulièrement son comportement et son évolution en vue de pouvoir évoluer et s'améliorer en fonction du contexte.
- ***Self-adjusting*** (auto-ajustement) est la capacité d'un système de s'adapter au changement de son environnement et de rétablir ses fonctionnalités après perturbation, sans aucune intervention externe.

1.5 Objectifs d'un réseau autonomique

Du point de vue fonctionnel, un système autonome se doit d'assurer quatre automatismes à la fois : l'auto-réparation, l'auto-configuration, l'auto-optimisation et l'auto-protection [Kephart and Chess, 2003 ; Kephart, 2005]. Nous définissons, dans la suite, chacune d'entre elles :

- **Auto-réparation :** Les systèmes qui s'auto-réparent peuvent automatiquement identifier, analyser, résoudre et réparer les problèmes. L'objectif étant de maintenir de façon continue la disponibilité des réseaux et des services et de gagner du temps par rapport aux approches actuelles où parfois un certain temps est requis pour le traitement du problème. La plateforme *Open Object Request Broker* (OpenORB) [Blair et al., 2002] décompose le mécanisme d'auto-réparation en deux entités :
  — La première, dite entité de monitoring, observe le comportement des couches fonctionnelles sous-jacentes, collecte les informations liées à la Qualité de Service (QdS) et fait part d'un comportement anormal,
  — La deuxième, dite entité de contrôle, s'occupe de sélectionner une stratégie d'adaptation appropriée basée sur les rapports émis par le mécanisme de monitoring.

- **Auto-configuration :** Elle couvre le besoin de rendre la configuration et la reconfiguration d'un système et de ses entités dynamiques et autonomes, en particulier, dans les phases d'installation et de suppression d'un nouveau composant dans le système. Ce dernier pourra ainsi auto-déterminer la bonne configuration à adopter et éviter les nombreuses erreurs de configuration liées au facteur humain.

  L'auto-configuration doit être réalisée dans le but d'atteindre le comportement attendu. Citons comme exemple la proposition de Molinier et al. [2012] concernant l'auto-configuration des équipements dans un réseau de domicile. Dans cette approche, un agent intelligent est placé sur chaque équipement. Il est censé s'occuper de la configuration des interfaces, de la découverte du voisinage et du choix du chemin et du médium de communication d'une manière tout à fait transparente à l'utilisateur.

- **Auto-optimisation :** L'optimisation d'un système est une tâche difficile à effectuer compte tenu du nombre de paramètres en jeu. Afin d'éviter à l'administrateur de paramétrer lui même les équipements et les réseaux pour fonctionner correctement, les systèmes doivent continuellement chercher à améliorer leurs performances et optimiser leurs opérations. L'auto-optimisation concerne donc l'automatisation de la gestion des performances des systèmes. Celles-ci doivent être optimisées et évaluées de façon continue.

- **Auto-protection :** Elle concerne les capacités d'un système à se protéger et à se sécuriser. L'automatisation de la sécurité a pour but de développer un comportement pouvant détecter les situations où la stabilité des réseaux et des services peut être remise en cause par le biais d'actions volontaires ou non et de se protéger en conséquence pour ne pas perturber les usagers.

  Comme exemple, Luo et al. [2002] proposent un service distribué d'authentification dans le contexte des réseaux ad-hoc. Dans cette solution, de multiples nœuds collaborent afin de traduire le comportement d'un serveur fournissant la certification et l'authentification pour d'autres nœuds du réseau ad-hoc.

- **Autres fonctions** D'autres auto-fonctions ont été définies dans divers travaux de recherche (auto-gestion, auto-localisation, ...) au regard d'un besoin particulier du contexte étudié. Généralement, cette diversification mène à un chevauchement des objectifs. Par exemple, l'auto-gestion couvre à la fois certains attributs de l'auto-configuration et d'autres relatifs à de l'auto-optimisation.

  Après avoir présenté les caractéristiques et les objectifs d'un système autonome, nous décrirons, dans la suite, l'architecture de ce dernier.

1.6 Architecture d'un réseau autonomique

Un système autonome est constitué d'une collection d'éléments autonomes (*Autonomic Elements*, AEs), capables de gérer leur comportement interne ainsi que les relations avec les autres éléments autonomes sans aucune intervention humaine. Pour ce faire, ils s'appuyent sur la connaissance acquise ou apprise par le biais des approches déductives ou inductives déployées dans le réseau. En Intelligence Artificielle (IA), le module qui se charge d'une telle gestion est nommé "agent".

1.6.1 Vision agent du paradigme

Un agent est un système informatique, situé dans un environnement, et qui agit d'une façon autonome pour atteindre les objectifs pour lesquels il a été conçu [Wooldridge and Jennings, 1994].

En distribuant les agents sur chaque équipement (comme le montre la figure 1.2), le système peut alors traiter localement tout problème l'impactant localement et du coup, diminuant de fait sa répercussion sur le système global. Dans la majorité des situations, les problèmes sont plus facilement gérables quand ils sont contenus localement. De plus, un problème local est détecté beaucoup plus rapidement que dans une approche dite "centralisée".

Pour accomplir sa tâche, un agent doit être [Wooldridge and Jennings, 1994] :

- **Situé :** l'agent est capable d'agir sur son environnement à partir des entrées sensorielles qu'il reçoit de ce même environnement,
- **Autonome :** l'agent est capable d'agir sans l'intervention d'un tiers (humain ou agent) et contrôle ses propres actions ainsi que son état interne,
- **Proactif :** l'agent doit exhiber un comportement proactif et opportuniste, tout en étant capable de prendre l'initiative au bon moment,
- **Réactif :** l'agent doit être capable de percevoir son environnement et d'élaborer une réponse dans le temps requis,
- **Social :** l'agent doit être capable d'interagir avec d'autres agents (automatiques ou humains).

Un ensemble d'agents interconnectés entre eux forment un Système Multi-Agents (SMA). Ce système peut doter la machine de l'intelligence requise et des capacités d'apprentissage lui permettant de tirer profit d'une expérience particulière pour faire face à une nouvelle situation.

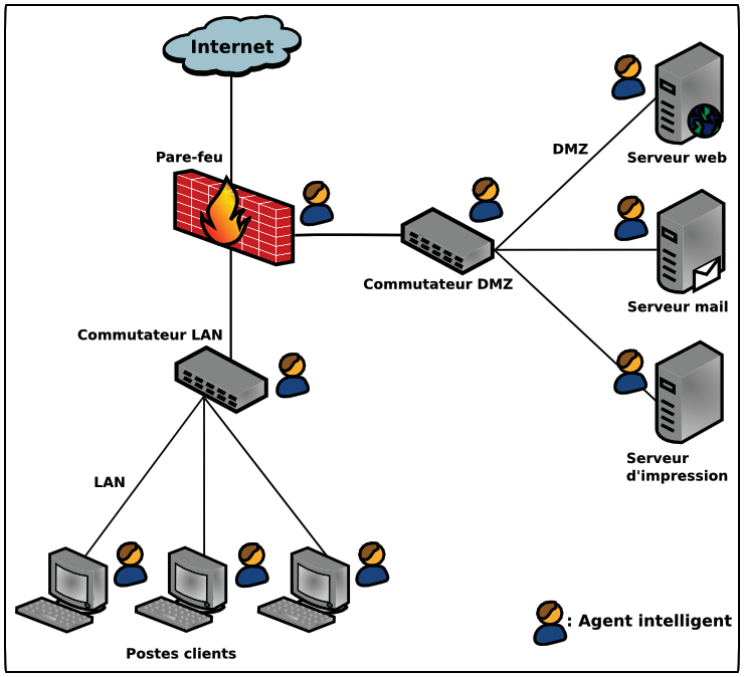

FIGURE 1.2 – Déploiement des agents dans un réseau

Les caractéristiques principales d'un SMA sont :

— Chaque agent a des informations ou des capacités de résolution de problèmes limitées,
— Il n'y a aucun contrôle global du système multi-agents,
— Les données sont décentralisées,
— Le calcul est asynchrone,

Un SMA fournit donc l'intelligence nécessaire pour comprendre le comportement du réseau afin qu'il puisse s'auto-piloter. Il est néanmoins nécessaire de lui fournir un ensemble de connaissances pour offrir des capacités d'auto-gestion élevées et efficaces. Ces connaissances doivent être apprises et acquises à partir de l'ensemble des agents et partagées entre eux de la manière la plus optimisée possible.

1.6.2 Architecture d'un élément autonome

Selon [Kephart and Chess, 2003 ; Kephart, 2005], un élément autonome (*Autonomic Element*, AE) est constitué de deux composants : une ressource gérée et un agent autonome. L'agent autonome est constitué d'une boucle de contrôle (*Control loop*) appelé MAPE (**M**onitor, **A**nalyse, **P**lan and **E**xecute)

capable de gérer le comportement de la ressource. L'interaction entre les deux composants de l'AE se fait à travers une interface constituée de capteurs (Sensors) et d'actionneurs (Effectors). Une interface identique sert à l'interaction de l'AE avec l'environnement extérieur. Les deux interfaces, la MAPE et la ressource sont reliées par un ensemble de connaissances qui décrivent l'état actuel et passé du AE et de son environnement.

La figure 1.3 représente l'architecture d'un AE. Dans ce qui suit, nous définissons chacun de ses composants.

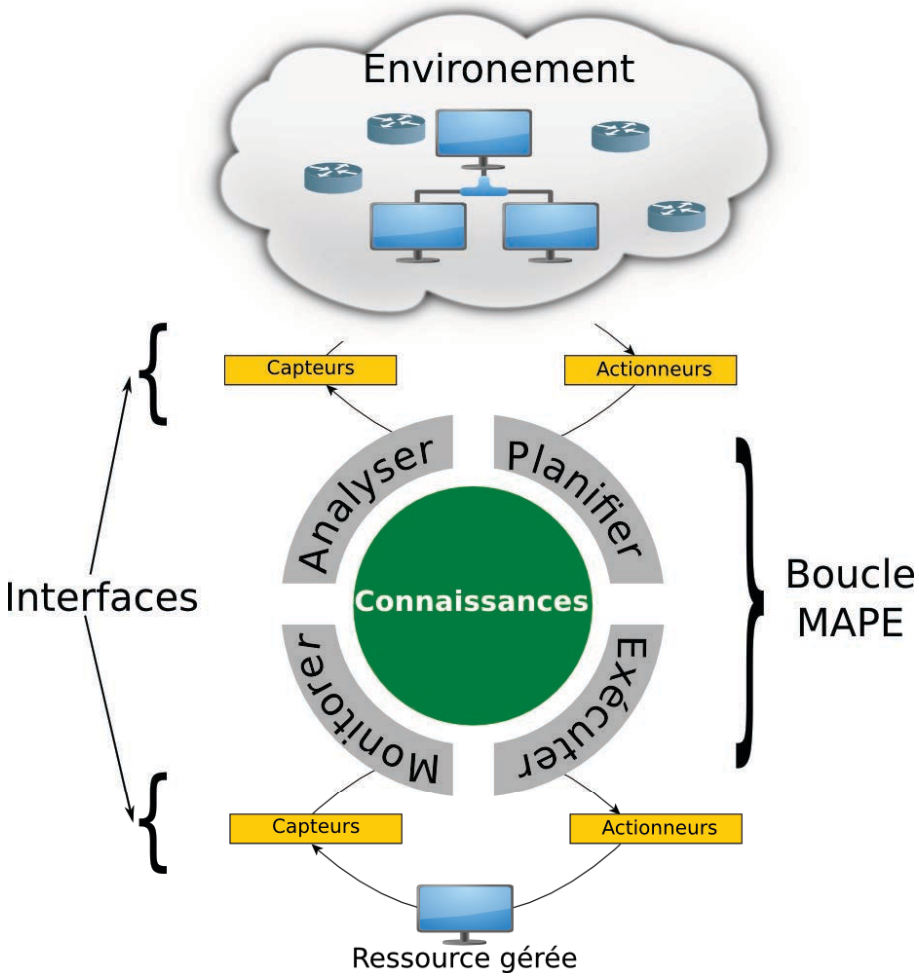

FIGURE 1.3 – Architecture d'un système autonome

1.6.2.1 Ressource

Une ressource gérée est définie comme tout équipement (logiciel ou matériel) pouvant être contrôlé à travers des actionneurs et supervisé à travers des capteurs. Une ressource peut donc être un noeud réseau (routeur, hub, commutateur, ...), un composant (RAM, disque, CPU, ...) ou encore une application (serveur de BD, application de vidéo à la demande, ...).

Toutefois, une ressource doit pouvoir être pilotée au travers d'actionneurs et pouvoir être supervisée au travers de capteurs.

1.6.2.2 L'interface

L'interface est ce qui permet, à la fois, l'interaction entre l'agent et la ressource gérée, et entre l'AE et l'environnement extérieur. Elle est structurée en un ensemble de capteurs et d'actionneurs. L'implémentation de chacun d'entre eux est spécifique à la ressource. L'intérêt de l'interface est donc de standardiser la gestion des ressources et d'améliorer l'interopérabilité.

Capteur : Il s'agit d'un mécanisme pouvant collecter des informations sur un élément géré. Dans les réseaux informatiques, nous pouvons distinguer deux types de capteurs :

- **Les Capteurs actifs** dont le fonctionnement consiste à générer du trafic dans le réseau et à observer l'effet produit sur ses différents composants. Ces capteurs sont intrusifs mais permettent généralement d'obtenir des mesures précises.
- **Les Capteurs passifs** dont le fonctionnement consiste à observer le trafic et les différents composants du réseau afin d'en déduire une certaine compréhension. Ces capteurs sont non-intrusifs, mais les mesures qu'ils fournissent sont, dans la plupart des cas, moins bonnes que celles fournies par leurs homologues actifs.

Actionneur : Il s'agit d'un mécanisme permettant d'agir sur l'élément géré. En d'autres termes, il offre les moyens nécessaires à l'exécution de tâches, éventuellement programmées, d'un système automatisé.

1.6.2.3 La boucle MAPE

La boucle MAPE (**M**onitor, **A**nalyse, **P**lan and **E**xecute) est le composant qui régit le comportement d'un AE. Les parties "Monitorer" et "Analyser" fournissent les caractéristiques liées aux fonctionnalités self-awareness et environnement-awarness. Les deux autres parties "Planifier" et "Executer" décident et réalisent la tâche de self-management (grâce aux actionneurs).

Le Monitoring : Il consiste en un ensemble de mécanismes pouvant collecter des informations sur un équipement géré. Ces informations peuvent être des paramètres définissant l'état du réseau (trafic, ressources, besoins, ...), les caractéristiques du réseau (topologie, capacité) ou ceux liées à l'utilisateur (périphérique, besoins, ...).

La collecte de ces informations se fait au travers de capteurs. Ces informations peuvent être filtrées, analysées ou bien agrégées avant d'être transmises au module d'analyse.

L'analyse : Elle consiste à observer et à évaluer le fonctionnement d'un système en vue de décider si une action doit être entreprise pour garantir un objectif de haut niveau.

La planification : Cette opération consiste à organiser les actions à entreprendre selon une méthodologie précise dans le but d'atteindre un objectif. Il s'agit de l'étape de formalisation des changements envisagés durant la phase d'analyse.

L'Exécution : Elle consiste à réaliser les actions décidées lors de la phase de planification. Cette réalisation se fait par le biais d'actionneurs. Le résultat obtenu permet d'enrichir les connaissances du système.

1.6.3 Systèmes autonomes existants

Plusieurs travaux de recherche ont été menés dans le but de produire des plateformes autonomes capables de gérer les ressources d'un réseau informatique. Nous en citerons dans la suite quelques exemples.

— **BIONETS :** Le projet BIONETS (**BIO**logically-inspired autonomic **NET**work**S**) [Carreras et al., 2007] s'inspire des mécanismes du monde biologique afin de mettre en oeuvre un système capable de gérer un grand nombre d'éléments hétérogènes sans aucune intervention humaine. Ce système dote les éléments du réseau d'une capacité d'évolution bio-inspirée en leur fournissant la faculté d'auto-adaptation due à l'interaction et à la coopération entre ses différents éléments.

— **GANA :** L'architecture GANA (**G**eneral **A**utonomic **N**etwork **A**rchitecture) a été proposée dans le cadre du projet EFIPSAN [Chaparadza et al., 2009 ; A. Liakopoulos, 2009]. Il s'agit d'une architecture qui implémente des fonctions d'auto-monitoring et d'auto-gestion. GANA est conçue selon une architecture cubique qui comprend quatre plans fonctionnels (plan de données, plan de découverte, plan de décision et plan de dissémination de connaissance). Chaque plan contient quatre niveaux d'abstraction (niveau réseau, niveau noeud, niveau fonction et niveau protocole) (cf. figure 1.4).

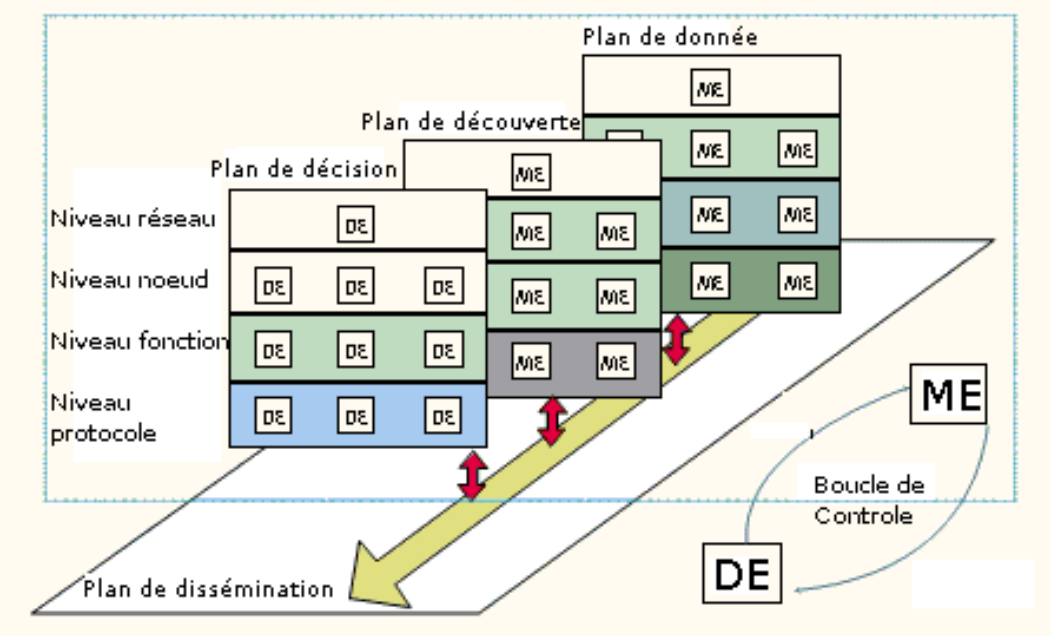

FIGURE 1.4 – Architecture de GANA [A. Liakopoulos, 2009]

GANA utilise les réseaux pair-à-pair dans le but de contrôler les performances et de gérer les ressources d'un système. Ce contrôle est effectué par des agents, appelés *Decision Element* (DEs), qui décident du comportement à adopter. Pour y parvenir, ils utilisent un ensemble d'informations stockées et disséminées à travers une table de hachage

distribuée (DHT) sur l'ensemble des noeuds du réseau.

— **FOCALE :** Strassner et al. [2006] ont proposé l'architecture FOCALE (**F**oundation, **O**bservation, **C**omparison, **A**ction and **L**earning **E**nvironment). Elle est l'une des premières architectures à utiliser les ontologies pour la représentation des connaissances. Comme GANA, FOCALE se base sur une table de hachage distribuée pour gérer ces connaissances.

— **ANA :** Le projet ANA (**A**utonomic **N**etwork **A**rchitecure) [Bouabene et al., 2010] vise à identifier les principes fondamentaux régissant les réseaux autonomes. Dans le but de démontrer la faisabilité d'un réseau autonome, ANA propose une plateforme de démonstration qui implémente ces principes et qui permet un déploiement à large échelle. L'architecture d'ANA repose sur deux composants principaux :
  — Blocs (Bricks) : Ils représentent les composants les plus atomiques d'ANA. Chaque bloc est en charge d'une fonction particulière. Les blocs Ethernet et IP fournissent par exemple le moyen d'interagir avec les réseaux existants.
  — Minimex : Il s'agit de l'élément qui permet aux briques d'interagir. Il contient toutes les unités de gestion permettant à une brique de découvrir les autres blocs locales et d'échanger des messages et des instructions.

  La dissémination des connaissances dans ANA n'est pas clairement définie. Toutefois Schuetz et al. [2007] étudie la gestion décentralisée des réseaux sans fil et propose d'utiliser un modèle de synchronisation dans un voisinage (portée d'une cellule radio).

— **CASCADAS :** Le Projet CASCADAS (**C**omponentware for **A**utonomic **S**ituated-aware **C**ommunication and **D**ynamic **A**daptable **S**ervices) [Marrow and Manzalini, 2006 ; Baresi et al., 2009] propose une plateforme de développement de réseaux autonomes à base d'ACEs (Autonomic Communication Elements) (cf. figure 1.5) capable d'assurer plusieurs auto-fonctionnalités (auto-organisation, auto-configuration, auto-protection et auto-optimisation). Les réseaux ainsi développés devraient être capables d'évoluer dans un environnement dynamique.

— **AUTOI :** Le projet AUTOI (**AUTO**nomic **I**nternet) [Galis et al., 2009 ; Abid et al., 2008] propose une solution autonome basée sur les réseaux virtuels recouvrants (*overlay networks*) peuvant être déployée de manière transparente sur le réseau Internet. Ces réseaux recouvrants permettent non seulement d'interconnecter plusieurs réseaux hétérogènes et mais aussi de supporter la mobilité.
  L'architecture d'AUTOI se compose de 4 plans :

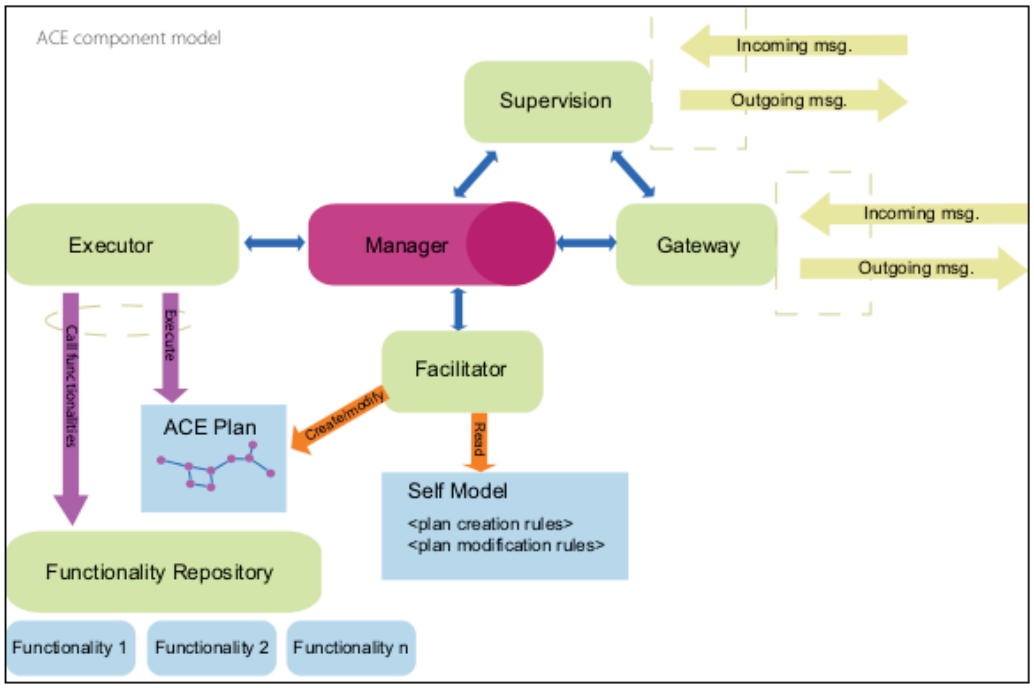

FIGURE 1.5 – Architecture de CASCADAS [Marrow and Manzalini, 2006]

— Plan de gestion : Il s'agit du plan en charge des différentes fonctions autonomes du système.
— Plan de connaissance : Il s'agit d'une base de données distribuée à travers l'ensemble des noeuds du réseau. Les données publiées y sont présentées sous la forme d'une ontologie commune. La distribution de la base de données se fait sur la base d'une approche correspondante à une "vue située" (cf. section 1.7.3.4).
— Plan d'orchestration : Il s'agit d'un plan de gestion capable de gérer plusieurs domaines et de résoudre, ainsi, les différents conflits qui peuvent subsister.
— Plan d'activation de service : Il fournit des fonctions pour le déploiement automatique ou l'activation de nouveaux services, de nouveaux protocoles, ainsi que des nouvelles ressources dans le réseau.

Chaque contribution dans les réseaux autonomes a apporté son lot de nouveautés : Utilisation des ontologies dans FOCALE [Strassner et al., 2006], Virtualisation dans AutoI [Abid et al., 2008] ou bien Conception atomique dans ANA [Bouabene et al., 2010]. Malgré les divergences que peuvent avoir ces différentes propositions, toutes s'accordent sur la nécessité d'une plateforme distribuée de gestion de connaissances. Cette plateforme constitue le

plan de connaissance. Il est donc intéressant de concevoir un nouveau système autonome ayant pour point de départ une plateforme de gestion de connaissances efficace et qui agrège une partie des innovations faites dans les autres propositions.

1.7 Plan de connaissance

Le plan de connaissance a été introduit par Clark et al. [2003] dans l'objectif de remédier aux limites conceptuelles de l'architecture IP classique. Son objectif a été de répondre à la question suivante : "Quelle serait l'architecture du réseau Internet si nous devons la refaire aujourd'hui ?". En fait, les deux objectifs majeurs du réseau Internet et qui ont fait son succès, la simplicité et la transparence, constituent aujourd'hui un frein à son évolution.

En plus clair, l'architecture des réseaux informatiques a été conçue pour répondre à un besoin simple : faire communiquer deux noeuds du réseau indépendamment du type de trafic qui transite et de la nature de ces noeuds. Dans cette architecture, le coeur du réseau se limite à retransmettre des paquets pour assurer leur transport [Saltzer et al., 1991]. Afin de ne pas encombrer les éléments de contrôle du réseau et, du coup, les temps de traitement, il a été décidé que toute la complexité et l'intelligence soient reléguées aux équipements de bordure du réseau. Ce principe répond essentiellement à deux objectifs majeurs :

— La transparence dans le fonctionnement du réseau : les noeuds qui communiquent n'ont pas à se soucier du fonctionnement des couches basses.
— La simplicité du déploiement : tout déploiement d'applications peut se faire sans aucune modification dans les couches sous-jacentes.

Ces attributs, certainement pertinents pour un réseau de taille contenu, comme le réseau Internet à ses débuts, se révèlent aujourd'hui comme des limites à son développement. En effet, la sécurité est remise en cause du fait des virus et autres logiciels malveillants qui usent de la simplicité du réseau pour se propager. De plus, la lenteur et l'inadaptation des prises de décision pour la gestion et l'optimisation du réseau rendent impossible tout type de contrôle réactif d'un réseau ayant subi des événements imprévisibles ayant impacté ses performances. Cela est dû au fait que ces décisions sont prises par un équipement de bord qui n'a aucune information pertinente sur le coeur du réseau et qui ne constate les anomalies que tardivement. Le rajout d'un plan

de connaissance à l'architecture réseau va permettre d'y remédier.

1.7.1 Architecture d'un plan de connaissance

L'architecture des réseaux est représentée, traditionnellement, sous la forme de trois "plans" (cf. figure 1.6) :

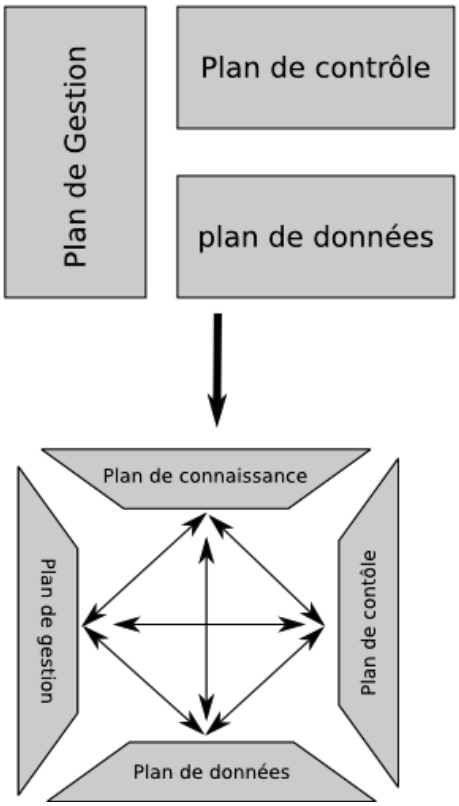

FIGURE 1.6 – Évolution de l'architecture réseau

Le plan de données : Il prend en charge le transit des données échangées entre les utilisateurs (exemples de fonctionnalités : ordonnancement, gestion de files d'attente, contrôle de congestion, façonnage de trafic, marquage de trafic, …).

Le plan de contrôle : Il permet de distribuer les politiques de contrôle et de gérer les données en temps réel (exemples de fonctionnalités : routage, signalisation, contrôle d'admission, réservation de ressources, …).

Un plan de gestion : C'est un plan transversal aux deux autres. Il est en charge de la supervision du plan de données et fournit une vue globale du bon fonctionnement du système (exemples de fonctionnalités : contrôle de supervision de QdS, configuration de QdS, politique de QdS, …).

Sur la base de cette architecture, chaque algorithme embarqué dans les noeuds du réseau doit donc s'adapter aux politiques demandées en fonction du contexte disponible à son niveau. Partant du constat que la majorité des informations nécessaires aux algorithmes de gestion du réseau est redondante, il s'agit donc de les agréger et de les porter à la connaissance des noeuds qui en font la demande explicite. Notons ici que la périodicité avec laquelle ils nécessitent ces informations n'est pas forcément la même. L'ajout d'un plan de connaissance à l'architecture réseau rend cet aspect possible. De plus, il devient aisé de définir des objectifs globaux qui peuvent s'adapter aux différents contextes locaux. À haut niveau, le plan de connaissance est chargé d'agréger les observations et les contraintes afin d'y appliquer un raisonnement et de générer des réponses pour les diverses situations auxquelles peut être confronté un réseau informatique. Ce nouveau plan requiert la coopération d'un ensemble de noeuds du réseau dans le but de partager les informations nécessaires à leur bon fonctionnement ainsi que les connaissances acquises par chacun d'entre eux. Chaque élément du réseau peut alors choisir la politique à appliquer en fonction de son contexte.

L'architecture d'un plan de connaissance réside dans le concept de séparation des couches [Clark et al., 2003]. En effet, le plan de connaissance n'a pas pour objectif de se substituer aux autres plans, mais confère aux mécanismes de contrôle les moyens de répondre à des objectifs globaux. Il s'agit ici d'un changement total de l'architecture réseau qui passe d'une vue "stratifiée" à une vue "*cross-layer*" comme le montre la figure 1.6. Cette approche permet de faire évoluer la logique de la gestion des réseaux en rendant la complexité locale transparente et en ajoutant un nouveau niveau d'abstraction au-dessus des règles et des politiques de contrôle.

1.7.2 Les concepts clés d'un plan de connaissance

D'après Clark et al. [2003], il faudra reconstruire le réseau autour d'un plan de connaissance afin que celui-ci soit le plus efficace possible. À défaut de pouvoir le faire, nous devons identifier les outils disponibles pour la mise en place de ce plan ainsi que les attributs clés auxquels il doit répondre. Ces attributs sont représentés dans la figure 1.7. Nous définissons, dans la suite,

chacun d'entre eux.

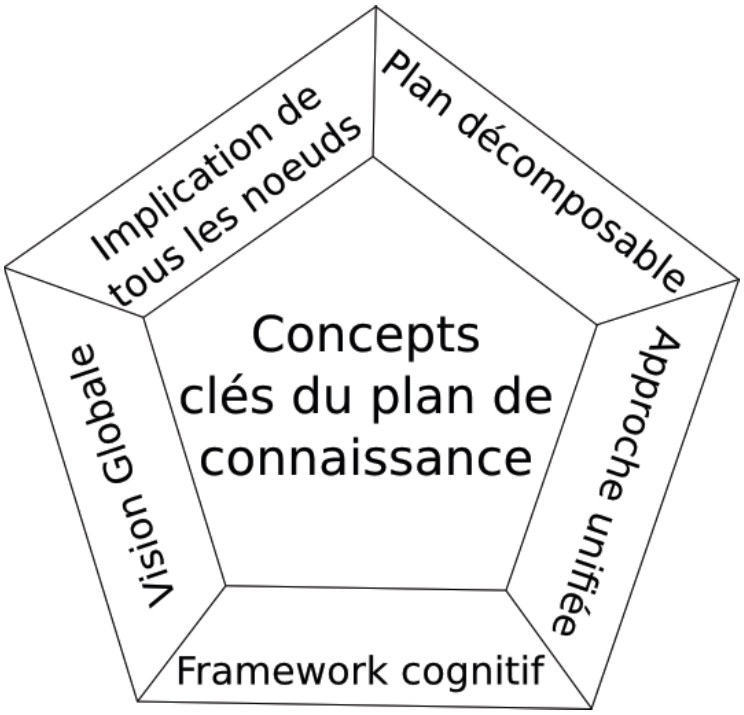

FIGURE 1.7 – Concepts clés d'un plan de connaissance

1.7.2.1 Système cognitif

Les techniques cognitives peuvent servir de fondation à la mise en oeuvre d'un plan de connaissance. En effet, celui-ci a besoin de porter des jugements en présence d'informations partielles ou contradictoires afin de :

- reconnaître et arbitrer les conflits dans les politiques et les objectifs,
- répondre aux dysfonctionnements ou à tout événement impactant le réseau dans des délais très courts,
- effectuer des optimisations d'environnements de grande dimension, dont le traitement, par des administrateurs ou des solutions analytiques classiques, demeurent compliqués,
- automatiser les fonctions qui nécessitent des ressources humaines hautement qualifiées.

Selon Brachman [2002], un système cognitif est un système qui, *"en plus de pouvoir raisonner et apprendre à partir de l'expérience pour améliorer ses performances au cours du temps, peut expliquer et justifier les actions qu'il*

entreprend". Les fonctions d'un système cognitif selon Vernon et al. [2007] peuvent être résumées comme suit : la perception, l'action, l'anticipation, l'adaptation et la motivation.

1.7.2.2 Approche unifiée

Les mécanismes distincts et localisés sont faciles à mettre en oeuvre et la solution à court terme peut être efficace. Toutefois, une approche unifiée est plus efficace à moyen et long termes. En effet, la connaissance du monde réel ne peut pas être strictement découpée en tâches indépendantes et distinctes. Le plan de connaissance doit donc être conçu comme un système unifié.

1.7.2.3 Prise en compte de la connaissance au niveau des extrémités du réseau

Le principe de bout en bout impose que beaucoup d'informations pertinentes sur les performances du réseau ne proviennent pas du coeur du réseau, mais des périphériques et des applications qui l'utilisent. Une partie des connaissances est donc produite, gérée et consommée dans les extrémités du réseau. Elles sont totalement transparentes par rapport aux éléments situés dans le coeur du réseau. Si ce dernier détecte une anomalie, il doit se contenter de raisonner sur une connaissance partielle du problème. L'objectif du plan de connaissance est entre autres celui d'injecter une partie de la connaissance issue des équipements de bordure du réseau dans le contrôle des éléments situés au delà.

1.7.2.4 Vue globale

La plupart des systèmes de gestion sont décentralisés : chaque opérateur gère la partie dont il est propriétaire. Toutefois, l'identification de certains problèmes dépend de la corrélation de différentes observations. Non seulement les données provenant des éléments de bordure doivent être combinées avec les données du coeur du réseau, mais les données de différentes régions (représentées par des différents opérateurs) du réseau peuvent être nécessaires pour reconstruire le puzzle qui constitue la suite d'événements impactant le réseau. Bien qu'utopique, le plan de connaissance doit, dans l'idéal, être en mesure d'étendre sa perspective pour l'ensemble du réseau mondial.

1.7.2.5 Plan décomposable

Un plan de connaissance doit non seulement offrir une vue globale du réseau, mais permettre aussi de structurer cette connaissance en un ensemble de sous plans indépendants. Cette granularité répond à plusieurs objectifs :

La sécurité : Certaines données sensibles ne peuvent pas être divulguées par un opérateur. Il est donc nécessaire de garder la possibilité de gérer une connaissance privée.

Le passage à l'échelle : Certaines connaissances ne sont nécessaires que pour un raisonnement local et il n'est donc pas nécessaire de les partager avec le reste du réseau.

La cohérence : L'incohérence peut résulter de la corrélation de certaines connaissances issues de contextes différents. Pour éviter une possible mauvaise interprétation, certaines connaissances doivent par ailleurs être cloisonnées.

Toutefois, il n'en demeure pas moins que des sous-plans non connectés doivent aussi être capables de fusionner leur point de vue et leurs connaissances si nécessaire. C'est le principe d'interopérabilité.

1.7.3 Verrous technologiques du plan de connaissance

Afin de développer un plan de connaissance qui permet de répondre aux concepts clés décrits par Clark et al. [2003], nous devons d'abord relever certains défis et verrous technologiques que nous détaillons ci-dessous :

1.7.3.1 Définition de la connaissance

Au regard de l'usage de certains termes de plus en plus répandus dans les publications scientifiques, il n'est pas trivial de différencier les données, les informations et les connaissances. Afin de lever cette confusion, il est important de préciser que le cycle de vie d'une mesure ou d'une observation passe par 3 étapes comme le montre la figure 1.8.

Chaque étape fait appel à un concept précis décrit ci-dessous :

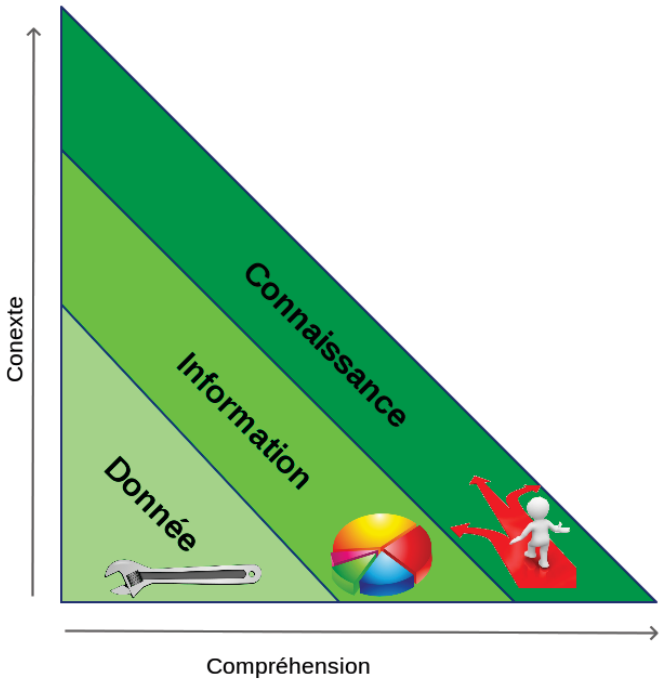

FIGURE 1.8 – Cycle de vie d'une donnée

1.7.3.1.1 Donnée Selon la définition du Larrousse [Jeuge-Maynart, 2010] : « Une donnée est le résultat d'observations ou d'expériences faites délibérément ou à l'occasion d'autres tâches et soumis aux méthodes statistiques. »

Dans un contexte de réseau informatique, elle peut être, par exemple, mesurée à travers des outils de supervision ou fournie par une personne utilisant le système. Citons les exemples suivants pour en comprendre le sens :

— Un opérateur de tripleplay a enregistré 100 tickets d'utilisateurs se plaignant d'une mauvaise qualité de la réception télévisuelle.
— Un serveur de diffusion de vidéo s'est éteint à 21h15 à la date du 10 octobre 2013 pendant 32 minutes.

1.7.3.1.2 Information Selon la définition du Larrousse [Jeuge-Maynart, 2010] : « L'information est tout événement, tout fait, tout jugement porté à la connaissance d'un public plus ou moins large, sous forme d'images, de textes, de discours, de sons. »

D'une manière plus générale, il s'agit d'une donnée à laquelle un contexte et

une interprétation sont fournis afin qu'un administrateur réseau puisse prendre la décision appropriée. Reprenons les exemples cités précédemment, les informations qui en découlent sont comme suit :

— La perception des usagers du service TV mesurée en MOS [1], est passée sous la barre des 3/5 cette semaine.
— La fiabilité du serveur est de l'ordre de 70% pour la semaine en cours.

Ces deux informations permettent à l'administrateur système de décider s'il est nécessaire d'apporter une action corrective si, bien évidement les perturbations ne sont pas le fruit d'événements extérieurs et non contrôlés comme une coupure électrique. À ce stade, il ne s'agit que d'une réaction ponctuelle, sans aucune considération à long terme.

1.7.3.1.3 Connaissance Selon la définition du Larrousse [Jeuge-Maynart, 2010] : « La connaissance est l'opération par laquelle l'esprit humain procède à l'analyse d'un objet, d'une réalité. » Un administrateur réseau réfléchirait à une situation au regard des informations à sa disposition ainsi qu'en fonction d'une expérience acquise avec le temps.

Typiquement, dans les exemples cités plus haut, l'opérateur aurait pu avoir une analyse plus fine : *L'insatisfaction des usagers est probablement dûe au manque de fiabilité du serveur de diffusion vidéo.* La corrélation de ces deux événements est essentielle pour apporter une solution rapide et adéquate, car, par expérience, l'administrateur sait qu'une insatisfaction prolongée conduirait, inéluctablement, à une augmentation du taux attrition [2] (Churn rate) et par conséquent, à une perte financière.

1.7.3.2 Construction des connaissances

Le processus de construction des connaissances commence par la collecte des données et des informations. En ce sens, Madhyastha et al. [2006] ont proposé une plateforme nommée iPlane (**i**nformation **Plane**). Elle consiste en un système distribué à travers le réseau mondial PlanetLab [Culler et al., 2002] qui effectue des mesures continues des métriques de qualité de service réseau (latence, délai, bande passante, taux de perte) dans le but de générer une carte annotée du réseau Internet (pour plus de détail, voir la figure 1.9

1. **M**ean **O**pinion **S**core est un score allant de 1 à 5 décrivant la perception de l'usager d'un service donné [Rec, 2006].
2. L'attrition désigne la perte de la clientèle.

tirée de [Madhyastha et al., 2006]).

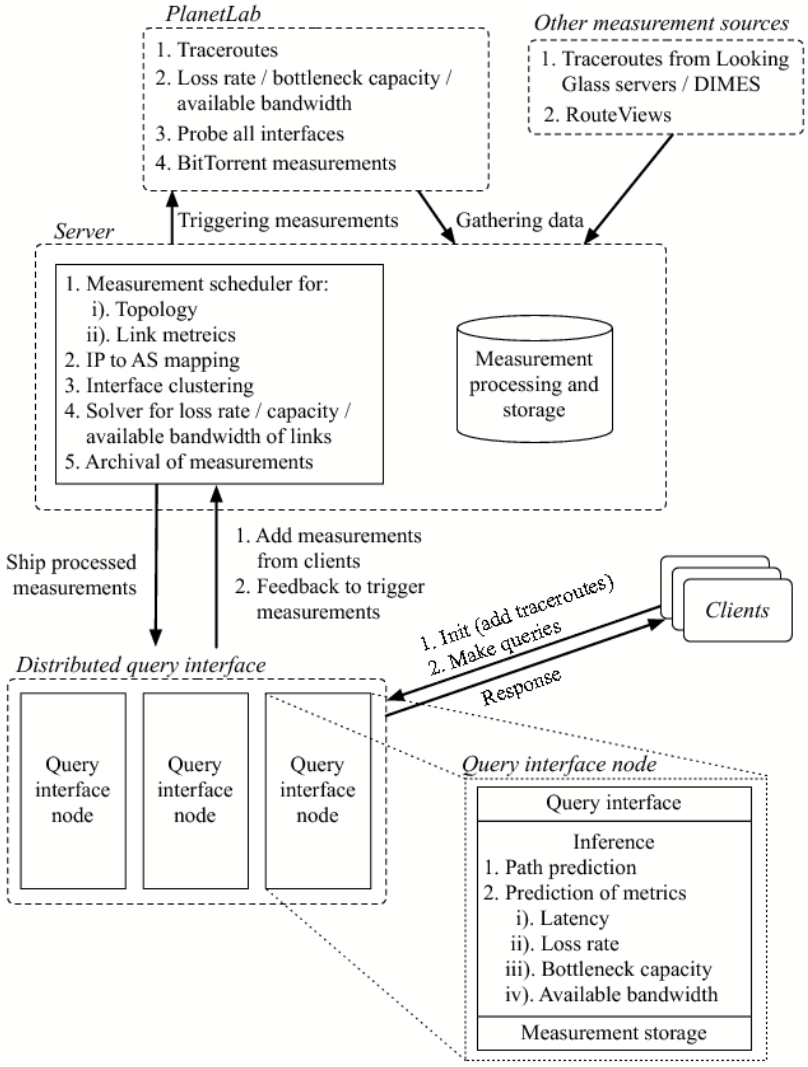

FIGURE 1.9 – Architecture de iPlane

Malgré le fait que iPlane utilise le terme information et non le terme de connaissance, il n'en demeure pas moins que certaines connaissances ont été déduites par corrélation entre les observations des différentes parties du réseau. L'ensemble de ses informations est stocké au niveau d'une base de données centralisée de type SQL-like.

Chen et al. [2007] proposent une autre approche d'évaluation des performances. Elle consiste à sélectionner un sous-ensemble de liens recouvrants et à évaluer leur taux de perte. Cette opération est répétée un nombre suffisant

de fois afin de pouvoir en déduire les taux de perte sur l'ensemble du réseau.

1.7.3.3 Réprésentation de la connaissance

Fournir une représentation de la connaissance dans le domaine des réseaux informatiques consiste à fournir une vue unifiée et évolutive (à défaut d'être exhaustive) de tous les concepts qui régissent ce domaine. Des organismes de standardisation ont proposé des ontologies et des modèles pour répondre à cette problématique :

MIB (Management Information Base) a été défini en 1991 par IETF [3] [McCloghrie and Rose, 1991]. Il consiste en une base d'informations sur les ressources du réseau géré.

Le Modèle CIM (Common Information Model) est un modèle conceptuel proposé par le DMTF [4] [DMTF, 1999]. Il permet de décrire les informations de gestion des équipements informatiques et des réseaux.

INDL Ontology (Infrastructure & Network Description Language) est une ontologie proposée afin de réponde à la nécessité d'un vocabulaire commun et partagé permettant de décrire les réseaux informatiques [Ham, 2010]. Cette ontologie est décrite dans le format *Resource Description Framework* (RDF) [Lassila et al., 1998].

Les nombreuses propositions faites dans ce cadre reflètent la difficulté de la tâche. En effet, vu le nombre de composants en vigueur, il n'est pas trivial d'identifier et de standardiser l'ensemble des connaissances pertinentes dans le réseau.

1.7.3.4 Dissémination de la connaissance

Le passage à l'échelle et la robustesse représentent des caractéristiques importantes qui doivent être prises en compte dans la construction d'un plan de connaissance. En effet, la distribution des connaissances au sein d'un réseau intégrant un nombre élevé d'équipements diversifiés ne peut pas se faire par des approches de type diffusion, ceci impliqueraient une surcharge importante du réseau. C'est pour cette raison que plusieurs approches de dissémination ont été proposées. Elles sont décrites ci-dessous :

3. Institute of Electrical and Electronics Engineers
4. Distributed Management Task Force

1.7.3.4.1 Diffusion locale La première solution consiste à propager les connaissances uniquement dans un voisinage immédiat. Les partisans de cette approche suggèrent que pour certaines architectures, il n'est pas nécessaire d'avoir une connaissance globale du système, mais uniquement une vue locale. Par exemple, dans [Tang et al., 2007], les auteurs proposent de diviser le réseau en zones géographiques ; les connaissances sont ainsi diffusées à l'intérieur de chaque zone. Dans [Schuetz et al., 2007], les auteurs proposent que chaque station de base synchronise ses connaissances avec les noeuds qui sont dans sa cellule de diffusion. De tels systèmes offrent une vue réduite des connaissances du réseau. Leurs performances dépendent donc du contexte d'utilisation.

1.7.3.4.2 Vue située La diffusion en vue située est décrite comme un nouveau concept de dissémination des connaissances [Nguengang et al., 2008 ; Bullot et al., 2008 ; Marrow and Manzalini, 2006]. Le terme "vue située" (Situatedness) est importé de la thématique multi-agents. Dans celle-ci, la vue située est une propriété des agents autonomes qui consiste à se situer dans son environnement, le percevoir et interagir avec lui [Florian, 2003].

Dans le cas du plan de connaissance, la vue située représente la limite que l'agent s'impose pour aller chercher ou diffuser une connaissance. Du fait que cette approche ne diffuse les connaissances que dans cette zone, elle permet un déploiement à large échelle.

L'inconvénient ici est celui de déterminer de manière distribuée la portée idéale de la vue située qui permet d'avoir le maximum de connaissances pertinentes sans surcharger le réseau. En effet, plus cette zone est importante, plus les connaissances sont globales et pertinentes, mais cela a comme inconvénient celui de générer un plus grand nombre de requêtes augmentant ainsi les temps de réponse. Toutefois, une zone trop petite conduira à une prise de décision inadaptée qui serait la conséquence de connaissances incomplètes.

1.7.3.4.3 Table de hashage distribuée Une table de hachage distribuée (ou DHT) [Stoica et al., 2001], est une technique qui permet de gérer une très grande base d'informations. Elle consiste à répartir la table de hachage [5] sur tous les éléments du réseau, qui en possèdent chacun une partie.

Par exemple, le noeud A va être responsable de toutes les connaissances qui commencent par A, de même que les noeuds B et C …. Lorsqu'un noeud souhaite récupérer une connaissance, il commence par rechercher le noeud qui

5. Une table de hashage est une structure de données de type clé-élément.

en est responsable avant de la demander.

De par sa simplicité de mise en oeuvre, cette approche est utilisée dans plusieurs travaux [Strassner et al., 2006 ; Li et al., 2008]. Toutefois, la nécessité de rechercher le détenteur de la connaissance et ensuite la récupérer pour son traitement fait que cette méthode n'est pas forcément adaptée à tous les cas d'usage, particulièrement dans le cas où le système nécessite un temps de réponse très court.

1.7.3.4.4 Cloud La proposition de stocker et de gérer le plan de connaissance dans le Cloud est assez nouvelle. Yu et al. [2012] proposent d'utiliser une "bigtable" [Chang et al., 2008] pour gérer les connaissances produites par un très grand ensemble de capteurs.

Une bigtable est un système de gestion de base de données distribuée, compressée et à haute performance, de type *Not only SQL* (NoSQL) [6], proposé par Google [Chang et al., 2008]. Les données sont modélisées sous la forme d'un couple clé/valeur. Elles sont stockées dans un système de fichiers (Google FS) dans un format propriétaire (SSTable : Sorted String Table).

Toutefois, le fait que les connaissances soient stockées dans un emplacement distant (datacenter), ajoute une certaine latence lors de leur récupération, ce qui peut ne pas convenir à certaines applications. De plus, le réseau n'est pas à l'abri d'une panne du lien le reliant avec la bigtable. Enfin, le stockage des connaissances chez un hébergeur tiers peut poser certains problèmes de confidentialité des données.

1.8 Conclusion

Les concepts liés à l'autonomique permet aux réseaux de s'auto-organiser à la manière du monde du vivant. Ce concept est avant tout un besoin des opérateurs pour gérer au moindre coût des réseaux qui deviennent de plus en plus complexes. Dans un premier temps, nous avons présenté ce concept ainsi qu'une étude de l'éxistant. De plus, nous avons mis en évidence la nécéssité d'un nouveau plan nommé "plan de connaissance" qui vient se rajouter aux trois plans traditionnels existants. L'émergence de ce plan a impliqué la définition de plusieurs aspects liés à la gestion de cette connaissance : sa dé-

6. NoSQL désigne une catégorie de systèmes de gestion de base de données qui ne sont pas fondés sur l'architecture des bases relationnelles.

finition, sa construction, sa représentation ou encore sa dissémination. Dans une seconde partie de ce chapitre, nous avons défini chacun de ses verrous et nous nous sommes attardés sur la problématique liée à la dissemination des connaissances sur les éléments du réseau.

Dans le chapitre suivant, nous présentons nos contributions dans le cadre de cette dissémination.