Faculté
des Sciences
Département de l'Informatique

UMONS
Université de Mons

Titre

Mémoire réalisé par Prénom NOM
pour l'obtention du diplôme de Master en Sciences Informatiques

Année académique 2009–2010

Directeur : Nom du directeur

Service : Service dans lequel vous avez fait votre mémoire

Faculté des Sciences • Université de Mons • Place du Parc 20 • B-7000 Mons

Table des matières

Introduction générale

Actuellement, plus de $10,300$ [1] sondes Atlas sont déployées dans le monde pour effectuer des mesures réseaux comme le DNS, Ping, Traceroute, etc. Ces sondes sont maintenues par le RIPE NCC (Réseaux IP Européens - Network Coordination Centre). Les données collectées par ces mesures sont stockées et sont disponibles en accès libre [2]. Quotidiennement, plus de 18732 mesures sont planifiées au départ de ces sondes vers de multiples autres destinations. En moyenne, 33 Go de données [3] sont collectées chaque jour pour tous les types de mesures.

Le besoin en stockage des données est en croissance continue avec la quantité de données générées par les transactions des clients, les réseaux sociaux, l'Internet des objets qui collectent constamment les données, etc. Les solutions traditionnelles en terme de stockage, de calcul et de visualisation ne répondent pas aux besoins, surtout des grandes organisations. Ce qui les a encouragé à créer des solutions permettant de répondre aux nouveaux besoins, c'est le Big Data.

L'objectif du présent mémoire est de montrer la capacité des nouvelles technologies du Big Data à fournir des solutions efficaces capables d'assurer le stockage des données massives et d'effectuer des tâches de traitement sur ces quantités de données. Dans notre cas, ce sont des données collectées par les sondes Atlas. Ces données apportant des informations utiles et pertinentes que nous recueillons sur l'état du réseau.

Les articles, les travaux publiés par RIPE Atlas et les présentations durant les rencontres organisées par RIPE NCC permettent d'avoir une idée générale sur les sujets à traiter en vue d'exploiter les données collectées par les sondes Atlas. Plus généralement, les sujets abordés sont : la visualisation de certains indicateurs sur les performances d'un réseau, l'analyse des censures appliquées au niveau de certains pays, l'analyse des détours que subit un trafic local et l'étude des performances d'un réseau, par exemple : le temps de la latence, la perte des paquets,

1. A la date de l'accès à la source `https://atlas.ripe.net/results/maps/network-coverage/`, le $14/08/2018$.

2. Les données des derniers 30 jours, les données des autres périodes sont accessibles avec une API REST.

3. Au format compressé.

l'asymétrie du trafic et la congestion des routeurs.

Les utilitaires ping et traceroute font partie des outils d'analyse de l'état du réseau et de résolution des problèmes dans les réseaux fortement utilisés. En particulier, l'utilitaire traceroute fournit des informations de l'aller et du retour entre une adresse IP source et une adresse IP destination sur un réseau. Il fournit les sauts impliqués tout au long du chemin entre la source et la destination ainsi que le temps requis pour les atteindre. Les détails fournis par traceroute permettent d'avoir des informations sur les réseaux traversés, la latence, etc.

Les traceroutes effectués par toutes les sondes, durant une heure, génèrent des données dont la taille est d'environ 8 Go. La manipulation de cette quantité de données nécessite des ressources de hautes performances. On ne peut pas compter sur les ressources traditionnelles comme les bases de données relationnelles pour le stockage, les processeurs des machines ordinaires [4] pour traiter les données après la récupération de celles-ci.

L'analyse des données collectées par les sondes Atlas a prouvé l'utilité de ces données. Plusieurs cas d'utilisation sont régulièrement publiés. Nous nous intéresserons au sujet du délai d'un lien réseau (lien topologique), car traceroute fournit les sauts impliqués dans un chemin, entre une adresse IP source et une adresse IP destination, avec les informations de la latence. Nous allons étudier la capacité des technologies Big Data à gérer la quantité de données générées par les sondes Atlas. La gestion des données porte sur le stockage, le traitement et la visualisation.

Ce document est structuré comme suit : le premier chapitre reprend une présentation du projet RIPE Atlas où nous allons présenter les sondes Atlas, leur fonctionnement et quelques cas d'utilisation spécifiques. Le deuxième chapitre introduit le terme Big Data, puis, il reprend les disciplines impliquées dans le Big Data, pour ensuite parcourir un ensemble d'outils Big Data. Pour conclure le deuxième chapitre, un choix sera énoncé concernant les outils à utiliser dans l'analyse des traceroutes. En ce qui concerne le troisième chapitre, il reprendra les étapes de l'analyse de données des traceroutes suivant un processus particulier. Depuis la collecte des données Atlas jusqu'à la génération des résultats de l'analyse.

4. Machines avec une mémoire RAM de 4 ou de 8 Go.

Chapitre I

RIPE Atlas

I.1 Introduction

Le présent chapitre commence par une présentation détaillée du projet RIPE Atlas mené par l'organisme RIPE NCC. Ce projet a introduit l'utilisation des sondes pour effectuer des mesures des réseaux dans le monde. Ensuite, ce chapitre reprend une liste non exhaustive de quelques outils similaires aux sondes Atlas en matière d'objectifs. Enfin, expose quelques limites du système RIPE Atlas. La dernière section reprend brièvement quelques travaux basés sur le projet RIPE Atlas.

I.2 A propos RIPE NCC

Le RIPE NCC est un organisme qui alloue les blocs d'adresses IP et des numéros des Systèmes Autonomes dans l'Europe et une partie de l'Asie, notamment au Moyen-Orient.

> Un ***Système Autonome***, appelé AS, est un ensemble de réseaux et de routeurs sous la responsabilité d'une même autorité administrative. Chaque Système Autonome est identifié par un code sur 16 bits uniques. Les protocoles qui tournent au sein d'un Système Autonome peuvent être différents.

RIPE NCC assure différents services relatifs à la gestion des réseaux informatiques. Il maintient multiples projets pour un nombre de protocoles comme DNS (DNSMON), BGP (Routing Information Service ou RIS) et d'autres projets et services. En particulier, nous sommes intéressés par projet RIPE Atlas géré aussi par RIPE NCC. L'objectif du projet RIPE Atlas est de déployer des dispositifs dans le monde, capables de collecter des données réseaux. Nous allons le

détailler dans la section I.3.

I.3 Présentation du projet RIPE Atlas

RIPE NCC a créé le projet RIPE Atlas en 2010. Le nombre de sondes déployées est en augmentation constante, sachant qu'elles sont déployées par des volontaires.

I.3.1 Les mesures actives et passives de l'Internet

Il existe plusieurs approches pour analyser l'état d'un réseau. Les deux approches les plus répandues sont : active et passive. L'approche passive fait référence au processus de mesure d'un réseau, sans créer ou modifier le trafic sur ce réseau. L'approche active repose sur l'injection des paquets sur le réseau et surveiller le flux de ce trafic. Cette injection a pour objectif la collecte des données relatives aux performances du réseau en question. Par exemple, la mesure du temps de réponse, le suivi du chemin des paquets, etc.

Les données collectées permettent de surveiller les réseaux pour ensuite proposer des améliorations de l'Internet. Le projet RIPE Atlas est un des outils s'inscrivant dans l'approche active. Ce sont des dispositifs, appelés sondes, hébergés par des volontaires, ils sont distribués et maintenus par RIPE NCC. Les données collectées par ces dispositifs sont disponibles au public [4].

Actuellement, plus de $10,000$ sondes Atlas sont actives, ces dernières produisent environ 450 millions de mesures par jour, ce qui correspond à $5,000$ résultats par seconde [22].

I.3.2 Généralités sur les sondes Atlas

- Les sondes Atlas mesurent les performances de la couche IP. Une sonde envoie des paquets réels et observe la réponse en temps réel indépendamment des applications en dessus de la couche IP.

- Les sondes Atlas ne sont pas des observatrices des données comme le trafic du routage BGP, ainsi, elles n'observent pas le trafic de leurs hébergeurs.

- Les sondes Atlas se situent dans différents emplacements dans le monde, cette répartition permet de diversifier les mesures (voir les sections des mesures I.3.9 et I.3.11).

- Les sondes Atlas sont déployées volontairement dans une maison, un bureau, un entrepôt de données, etc.

– Les mesures peuvent être lancées à tout moment et pour n'importe quelle période [1].

– La participation au projet RIPE Atlas est ouverte à toute personne qui s'y intéresse, cela inclut les résultats de mesures, les outils d'analyse, l'hébergement des sondes elles-mêmes, les travaux, etc.

– RIPE Atlas simule le comportement de la couche IP. Par exemple, avec RIPE Atlas, il est possible de :

  – Suivre l'accessibilité d'une destination [2] depuis différents emplacements dans le monde et depuis différents réseaux. Car les sondes Atlas sont réparties dans plusieurs pays et déployées dans différents réseaux.

  – Étudier des problèmes du réseau remontés en effectuant des vérifications de connectivité ad-hoc via les mesures effectuées par les sondes Atlas.

  – Tester la connectivité IPv6.

  – Vérifier l'infrastructure DNS.

La section I.5 reprend quelques cas d'utilisation du système RIPE Atlas et les sujets qu'on peut étudier.

I.3.3 Les générations des sondes Atlas

Depuis leur création en 2010, les sondes Atlas ont connu trois générations du matériel. Le tableau I.1 reprend quelques caractérisations de ces trois générations des sondes Atlas et la figure I.4 montre le matériel utilisé dans chaque génération.

1. Si le nombre de crédits (voir la section des crédits I.3.10) disponibles le permet et qu'il n'y a pas de dépassement du nombre de mesures autorisé.
2. Une destination représente une adresse IP dans le présent contexte.

	v1	**v**2	**v**3
Matériel informatique	Lantronix XPort Pro [8]	Lantronix XPort Pro [8]	tp-link tl-mr3020
Début d'utilisation	2010	2011	2013
Mémoire RAM	8 Mo	16 Mo	32 Mo
Mémoire Flash	16 Mo	16 Mo	4 Mo
CPU	32-bit	32-bit	32-bit
Support du Wi-Fi	Non	Non	oui
Support du NAT	oui	oui	oui
Vitesses supportées	10 Mbit/s et 100 Mbit/s	10 Mbit/s et 100 Mbit/s	10 Mbit/s et 100 Mbit/s

TABLE I.1 – Les caractéristiques des trois générations des sondes Atlas

FIGURE I.1 – Génération 1

FIGURE I.2 – Génération 2

FIGURE I.3 – Génération 3

FIGURE I.4 – Les trois générations des sondes Atlas

Source : `https://atlas.ripe.net/docs/`, consultée le 05/08/2018.

Pour précision, les générations 1 et 2 présentent une très faible consommation d'énergie, cependant, elles ont un temps de redémarrage et coûts de production élevés.

En 2015, plusieurs utilisateurs des sondes Atlas ont montré un intérêt aux sondes virtuelles. Ces sondes virtuelles présentent des avantages et aussi des inconvénients. Parmi les avantages, la conception des sondes virtuelles permet d'explorer des emplacements qui sont difficilement accessibles. En effet, cela permet d'étendre le réseau des sondes Atlas. D'autre part, les sondes virtuelles peuvent être installées sans contraintes physiques ou organisationnelles. Parmi les inconvénients, une complexité sera ajoutée au système RIPE Atlas, plus de ressources seront demandées. Ensuite, il y a le problème de la qualité des données ; le manque

de données peut faire référence à une perte de paquets ou bien la machine qui héberge la sonde n'est plus disponible pour continuer les mesures.

I.3.4 La connexion des sondes Atlas à Internet

Les génération 1 et 2 des sondes Atlas ont une interface Ethernet (RJ-45). La génération 3 dispose techniquement des capacités Wi-Fi. Cependant, ces sondes ne sont pas suffisamment prêtes au niveau logiciel pour supporter le Wi-Fi. L'objectif était de garder l'indépendance des sondes Atlas du trafic de celui qui les héberge.

Une fois la sonde se connecte au port d'Ethernet, elle acquiert une adresse IPv4, un résolveur DNS en utilisant DHCP et la configuration IPv6 via *Router Advertisement*. Ensuite, elle essaie de rejoindre l'infrastructure du RIPE Atlas. Pour ce faire, elle utilise le résolveur DNS et se connecte à l'infrastructure à travers SSH sur le port TCP de sortie 443 comme il est illustré dans la figure I.5. L'architecture du système RIPE Atlas est détaillée dans la section I.3.5.

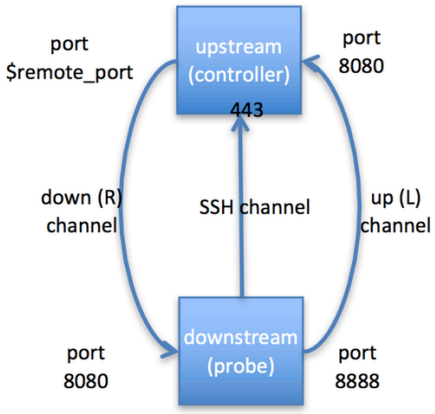


FIGURE I.5 – La connexion des sondes Atlas à l'infrastructure RIPE Atlas [21]

I.3.5 Architecture du système RIPE Atlas

Il existe deux catégories d'outils de surveillance du réseau : des outils matériels et d'autres logiciels. Les sondes Atlas sont parmi les outils matériels. le choix d'utilisation d'un outil matériel au lieu d'un outil logiciel dépend de plusieurs facteurs, par exemple l'indépendance du système d'exploitation, la facilité de déploiement, la disponibilité des sondes tout le temps (au lieu d'être dépendante de la machine qui l'héberge) et d'autres facteurs liés à la sécurité.

Le système RIPE Atlas est conçu pour qu'il soit opérationnel de façon distribuée. La plupart des composantes ont assez de connaissances pour remplir leurs rôles, sans nécessairement avoir besoin de connaître les états des autres composantes du système. Cela assure que le système soit capable d'assurer la plupart des fonctionnalités en cas d'un problème temporaire. Par exemple, si une sonde est déconnectée de l'infrastructure, elle continue les mesures planifiées et les données sont renvoyées dès sa reconnexion au système.

La figure I.6 montre une vue d'ensemble de l'architecture du RIPE Atlas.

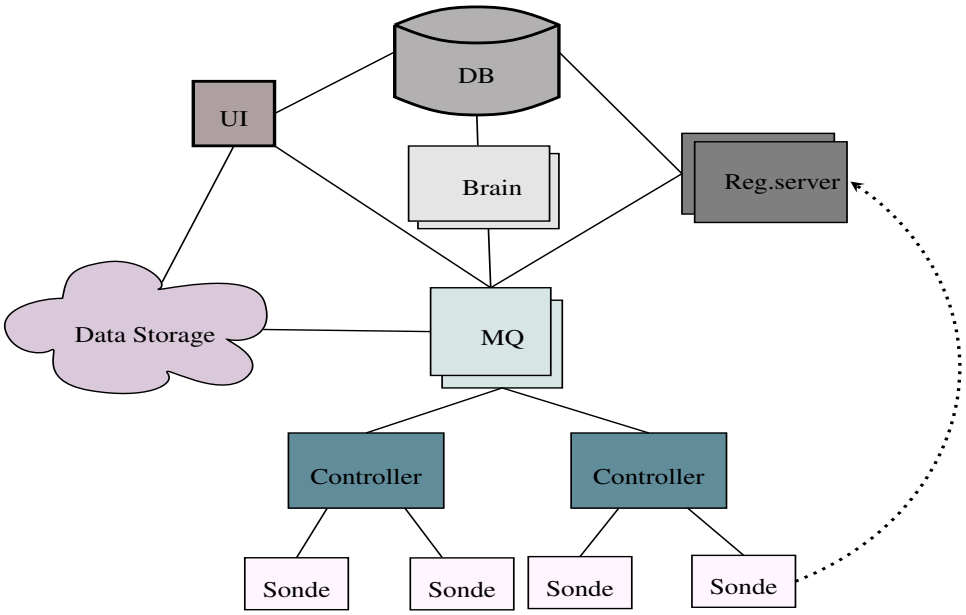


FIGURE I.6 – Architecture du système RIPE Atlas [22]

On distingue les composantes suivantes :

Registration server (Reg.server) : c'est le seul point d'entrée de confiance pour les sondes Atlas. Son rôle est de recevoir toutes les sondes désirant se connecter au système RIPE Atlas. Ensuite, il redirige chaque sonde vers le contrôleur adéquat, celui le plus proche de la sonde et que ce contrôleur soit suffisamment non occupé. Le serveur d'enregistrement a un aperçu de haut niveau du système.

Controller : les contrôleurs acceptent d'établir une connexion avec une sonde parmi les sondes dont ils ont reçu leurs clés du serveur d'enregistrement (Reg.server). Une fois la connexion est établie entre une sonde et un contrôleur, ce dernier garde cette connexion active pour recevoir les résultats et

prévenir la sonde des mesures à effectuer. Le rôle du contrôleur est de communiquer avec les sondes, associer les mesures aux sondes en se basant sur la disponibilité de la sonde et autres critères, enfin, collecter les résultats intermédiaires des mesures.

Message Queue (MQ) : Tout d'abord définissons MQ :

> « ***Message Queue ou file d'attente de message*** *: est une technique de programmation utilisée pour la communication interprocessus ou la communication de serveur-à-serveur. Les files d'attente de message permettent le fonctionnement des liaisons asynchrones normalisées entre deux serveurs, c'est-à-dire de canaux de communications tels que l'expéditeur et le récepteur du message ne sont pas contraints de s'attendre l'un l'autre, mais poursuivent chacun l'exécution de leurs tâches*[a]. »
>
> a. Source : https://fr.wikipedia.org/wiki/File_d'attente_de_message, consultée le 05/08/2018.

Un cluster de serveurs MQ agit comme un système nerveux central au sein de l'architecture du RIPE Atlas. Il gère la connectivité entre les composantes de l'infrastructure et assure l'échange de messages avec un délai minimal. C'est cette composante qui élimine le besoin que les autres composantes de l'infrastructure soient au courant des états des autres composantes de l'infrastructure. En plus, chaque composante peut être ajoutée ou retirée sans devoir synchroniser cette information à l'infrastructure entière. Si c'est le cas d'une déconnexion d'une composante, les messages seront sauvegardés sur différents niveaux jusqu'au moment de la reconnexion.

UI (User Interface) : elle s'occupe des interactions de l'utilisateur. Elle sert les pages pour l'interface graphique de mesures [3]. Elle traite les appels en provenance de l'API[3] et sert les demandes de téléchargement en provenance de l'API.

Brain : il effectue des tâches de haut niveau dans le système, notamment la planification des mesures. Cette planification est basée sur les demandes reçues via l'interface graphique web de mesures (UI) ou bien via l'API. La planification passe par la présélection des sondes Atlas et la négociation avec les contrôleurs pour voir la disponibilité des sondes Atlas.

3. Source : https://atlas.ripe.net/docs/api/v2/manual/, consultée le 05/08/2018.

DB : c'est une base de données SQL contenant toutes les informations du système RIPE Atlas : les informations sur les sondes et leurs propriétés, les meta-data des mesures, les utilisateurs, les crédits, etc.

Data Storage : c'est un cluster Hadoop/HBase pour le stockage à long terme de tous les résultats . Cette technologie permet aussi d'effectuer des calculs d'agrégation périodiques et d'autres tâches.

> ***Hadoop MapReduce*** est un modèle de programmation qui permet de traiter les données massives suivant une architecture distribuée dans un cluster.
> ***HBase*** est une base de données non relationnelle et distribuée. Elle est adaptée au stockage de données massives.

La figure I.7 présente les étapes d'établissement de la connexion entre une sonde Atlas et l'infrastructure RIPE Atlas.

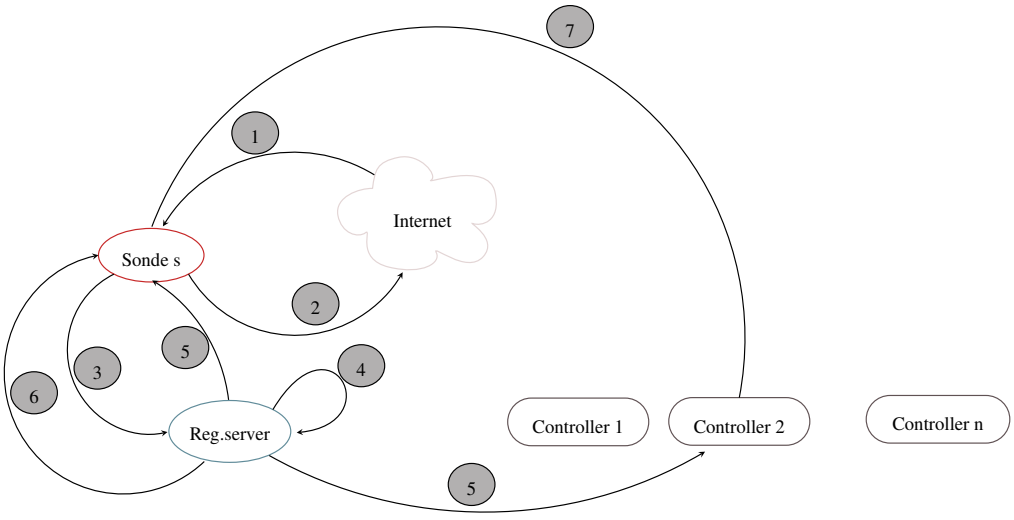


FIGURE I.7 – Les étapes d'établissent d'une connexion entre la sonde Atlas et l'architecture RIPE Atlas

Les étapes suivantes illustrent le déroulement de la connexion d'une sonde Atlas s à l'infrastructure RIPE Atlas.

- La sonde Atlas se connecte à Internet via le câble Ethernet $RJ45$ (1).
- La sonde Atlas acquiert différentes informations : une adresse IPv4, une adresse IPv6 via Router Advertisement et les informations du résolveur DNS via DHCP (2).

- Les informations précédemment acquises permettent à la sonde Atlas de se connecter au serveur d'enregistrement (Reg.server). C'est la première entrée vers l'infrastructure (3).

- En se basant sur la géolocalisation de la sonde Atlas, la charge des différents contrôleurs et d'autres options, le serveur d'enregistrement décide le contrôleur qui va être associé à la sonde Atlas (4).

- Suite à la décision du serveur d'enregistrement, le contrôleur reçoit l'identifiant de la sonde Atlas à gérer et la sonde Atlas reçoit l'identifiant du contrôleur avec à qui elle sera associée (5).

- Une fois l'association entre la sonde Atlas et le contrôleur est faite, la sonde Atlas se déconnecte du serveur d'enregistrement (6).

- La connexion entre la sonde Atlas et le contrôleur est maintenue le plus longtemps possible. Les contrôleurs gardent le contact avec les autres composantes via Message Queue. Dans le cas où une des composantes se déconnecte de l'architecture, les événements sont conservés jusqu'au moment où la connexion est restaurée (7).

La connexion précédemment établie permet aux sondes Atlas d'envoyer leurs rapports de mesures aux serveurs de stockage. C'est la même connexion qui permet de passer les commandes aux sondes pour qu'elles puissent effectuer les mesures et les mises à jour de leur firmware.

I.3.6 Les sondes Atlas et la vie privée

La sonde Atlas n'a pas l'accès au trafic de son hébergeur. Elle maintient sa connexion avec l'infrastructure centrale et elle exécute les mesures planifiées vers les destinations publiques sur Internet.

Les sondes Atlas peuvent révéler l'adresse IP de leur hébergeur. Bien que, les informations personnelles telles que les adresses MAC et les adresses e-mail ne seront jamais affichées. Cependant, l'adresse IPv6 peut exposer l'adresse MAC.

I.3.7 La sécurité dans RIPE Atlas

La connexion entre les composantes de l'infrastructure RIPE Atlas est maintenue le plus longtemps possible comme c'est décrit dans la section I.3.5. De ce fait, la sécurité des différentes connexions est primordiale. Afin de réduire la surface d'attaque contre ces sondes, les précautions suivantes sont prises :

- Les hébergeurs des sondes Atlas ne disposent d'aucun service qui leur permet de se connecter aux sondes (dans le sens de TCP/IP).
- Les sondes Atlas n'échangent aucune clé d'authentification entre elles. En effet, chaque sonde dispose de sa clé qu'elle l'utilise pour se connecter à l'infrastructure.
- Comme les sondes Atlas sont chez les hébergeurs, il est impossible qu'elles soient résilientes au démontage. Cependant, si c'était le cas, cela ne devrait pas affecter les autres sondes Atlas.
- Toutes les communications au sein de l'infrastructure RIPE Atlas se font d'une manière sécurisée. Les connexions entre les composantes sont maintenues grâce aux *secure channels* avec *mutual authentication*.
- Le logiciel qui tourne dans les sondes Atlas peut être facilement mis à niveau ; la sonde Atlas est capable de vérifier l'authenticité d'une nouvelle version du firmware et cela via les signatures cryptographiques.

Le système RIPE Atlas est un système comme les autres, il n'est pas résilient à 100 % aux attaques. Cependant, l'équipe RIPE Atlas propose régulièrement des améliorations et des fixations de bugs surmontées par la communauté RIPE Atlas.

I.3.8 Les ancres VS sondes Atlas

Les ancres Atlas sont des dispositifs agissant comme cibles aux différentes mesures lancées par les sondes Atlas. Il est possible de planifier des mesures entre les ancres RIPE Atlas, ces mesures permettent de vérifier l'état des réseaux qui hébergent ces ancres. Les ancres Atlas peuvent être considérées comme cibles aux mesures suivantes :

- Ping.
- Traceroute.
- DNS : les ancres ont été configurées avec BIND pour qu'elles agissent en tant que serveur DNS faisant autorité.
- HTTP et HTTPS : l'ancre fait tourner un serveur Web, ce dernier utilise un gestionnaire de réponses personnalisé aux requêtes HTTP(S) ayant comme seule option la taille du payload. Cette taille peut prendre une valeur maximale de 4096 et la réponse est fournie sous format JSON. L'exemple d'une requête HTTP avec une taille de 536 depuis une sonde Atlas vers une ancre Atlas est :

http ://nl-ams-as3333.anchors.atlas.ripe.net/536

Les ancres sont configurées avec un certificat SSL auto-signé en utilisant une clé de 2048 bit et un temps d'expiration de 100 ans. Le tableau I.2 reprend une comparaison de certaines caractéristiques communes entre les sondes et les ancres Atlas.

	Sonde Atlas	**Ancre Atlas**
Mesures originaires de	oui	oui
Mesures à destination de	— [4]	ping, traceroute, DNS, HTTP(S).
Nomination	—	structurée [5]
Crédit gagnés	N	$10 * N$
Besoin en bande passante	léger	important
Coût : gratuite	oui	non [6]

TABLE I.2 – Comparaison entre sondes et ancres RIPE Atlas

I.3.9 Les mesures intégrées : Built-in

Une fois une sonde Atlas connectée, elle lance automatiquement un ensemble de mesures prédéfinies, appelées *Built-in Measurements*. Les mesures personnalisées sont détaillées dans la section I.3.11. Le choix du mode IPv4, IPv6 ou les deux, dépend de la capacité du réseau qui héberge la sonde Atlas.

Il existe deux types de mesures : les mesures *One-Off*, ce sont les mesures qui s'exécutent une seule fois. Pour le deuxièmes type, ce sont les mesures qui s'exécutent périodiquement, à chaque intervalle de temps.

De base, les sondes Atlas assurent les mesures intégrées suivantes :

- Les informations sur la configuration du réseau dans lequel la sonde Atlas est déployée.
- L'historique de la disponibilité de la sonde Atlas.
- Les mesures du RTT (Round Trip Time) par traceroute.
- Les mesures ping vers un nombre de destinations prédéfinies.

4. — : Non disponible.
5. Exemple de *de-mai-as2857.anchors.atlas.ripe.net* avec la structure suivante : *pays-ville-ASN.anchors.atlas.ripe.net*.
6. Le matériel est au frais de l'hébergeur.

- Les mesures traceroute vers un nombre de destinations prédéfinies.
- Les requêtes vers les instances des serveurs DNS (Domain Name System) racines.
- Les requêtes SSL/TLS (Secure Socket Layer/Transport Layer Security) vers un nombre de destinations prédéfinies.
- Les requêtes NTP (Network Time Protocol).

Chaque mesure a un identifiant ID unique. Cet identifiant indique le type de la mesure, s'il s'agit du ping, traceroute ou autres. Plus de détails sur la signification des identifiants des mesures sont disponibles dans la section **??** dans l'annexe A.

En plus des mesures intégrées, les sondes Atlas peuvent effectuer des mesures personnalisées. Ces mesures peuvent être lancées via l'interface web [3] ou bien via HTTP REST API. Toutefois, la planification des mesures personnalisées nécessite l'acquisition de ce qu'on appelle les "crédits" au sens RIPE Atlas.

I.3.10 Le système de crédits Atlas

Le système de crédits RIPE Atlas est une sorte de reconnaissance de la contribution des participants à ce projet. Un hébergeur d'une sonde Atlas reçoit un nombre de crédits en contrepartie de la durée pendant laquelle sa sonde reste connectée. D'autre part, il gagne d'autres crédits suivant les résultats de mesures générés par cette sonde. Les crédits gagnés peuvent être utilisés dans la création des mesures personnalisées, appelées *User Defined Measurements* (voir la section I.3.11). Les personnes ayant gagné des crédits peuvent les transférer vers une autre personne ayant besoin de ces crédits. Les crédits peuvent être obtenus via :

- L'hébergement d'une sonde Atlas ; à chaque utilisation d'une sonde, son hébergeur reçoit un nombre de crédits. La connexion d'une sonde Atlas au système durant une minute apporte 15 crédits.
- L'hébergement d'une ancre Atlas [7].
- La recommandation à une personne d'héberger une sonde Atlas.
- En étant un sponsor du RIPE NCC. Le parrainage des sondes Atlas est disponible pour les organisations et les individus. Le sponsor reçoit le même nombre de crédits que les hébergeurs de ces sondes.
- En étant un registre Internet régional (Local Internet Registry).

7. Les ancres Atlas sont décrites dans la section I.3.8.

– La réception des crédits d'une autre personne via un transfert de crédits.

Le lancement des mesures personnalisées exploite les ressources de l'infrastructure RIPE Atlas d'une part, du réseau hébergeur de la sonde d'autre part. Par conséquent, les mesures sont organisées afin d'éviter toute surcharge du système. Le coût d'une mesure dépend du type de la mesure et des options spécifiées. Le système calcule le nombre de crédits nécessaires pour effectuer une mesure donnée. Le nombre de crédits est déduit à chaque résultat reçu. Ci-dessous le coût unitaire des différents types de mesures.

Ping et ping6 :

$$\text{Coût unitaire} = N \times (\lfloor \tfrac{S}{1500} \rfloor + 1)$$

Où N est le nombre de paquets dans le train (par défaut 3) et S est la taille du paquet (par défaut : 48 octets).

DNS et DNS6 :

Coût unitaire pour UDP : 10 crédits/résultat
Coût unitaire pour TCP : 20 crédits/résultat

Traceroute et traceroute6 :

$$\text{Coût unitaire} = 10 \times N \times (\lfloor \tfrac{S}{1500} \rfloor) + 1)$$

Où N est le nombre de paquets dans le train (par défaut 3) et S est la taille du paquet (par défaut : 40 octets).

SSLCert et SSLCert6 :

Coût unitaire = 10 crédits/résultat.

Exemple :
La planification d'une mesure ayant les caractéristiques suivantes nécessite $14,400$ crédits.

La fréquence	: deux fois par heure
La durée	: deux jours (48 heures)
Le nombre de sondes	: 5
Type de mesure	: *traceroute*

Tel que :

$$5 \times 2 \text{ mesures/heure} \times 48 = 480 \text{ ligne résultat}$$
$$30 \text{ credits/result} \times 480 \text{ results} = 14,400 \text{ crédits}$$

I.3.11 Les mesures personnalisées : User Defined mesurement

En plus des mesures intégrées, par défaut, dans une sonde Atlas, il est possible de planifier des mesures personnalisées. Ce sont les mêmes types de mesures : ping, traceroute, HTTP Get, SSLCert , DNS, NTP et TLS. Cette planification coûte des crédits, en effet, il faut avoir assez de crédits pour lancer des mesures. L'interface web dédiée à la création d'une nouvelle mesure offre toute les possibilités comme la précision des éléments suivants :

- Le type de la mesure.
- La sélection des sondes Atlas réalisant la mesure.
- La fréquence de la mesure et sa durée.

Chaque mesure est suivie via son état. Plusieurs états à distinguer : *specified*, *scheduled*, *ongoing*, *stopped*, *Forced to stop*, *no suitale probes* et enfin *failed*.

I.3.12 La sélection des sondes Atlas

La sélection des sondes Atlas pour effectuer une des mesures repose un des critères suivants :

- Numéro d'AS.
- Zone géographique via l'attitude et la longitude.
- Pays (ou zone géographique comme Europe).
- Préfixe IP.
- Manuellement, avec les identifiants des sondes Atlas.
- Reprendre celles d'une mesure précédente.

Il existe une autre manière de regrouper les sondes avec des étiquettes. Le système d'étiquettes sert comme indicateur des propriétés, des capacités, de la topologie du réseau ou d'autres classifications. On distingue les étiquettes système et utilisateur. Chaque nom d'étiquette est lisible par un humain.

Les étiquettes utilisateurs sont associées à une sonde librement par son hébergeur. Les étiquettes système sont attribuées uniquement par l'équipe RIPE Atlas et sont mises à jour périodiquement, à priori chaque 4 heures. Des exemples d'étiquettes système sont présentés dans la section **??** de l'annexe A.

I.3.13 Les sources de données Atlas

Les sondes Atlas génèrent trois types de données : leurs détails de connexions d'un jour donné, leurs résultats des mesures intégrées et personnalisées et les descriptions des mesures effectuées (meta-data).

Premièrement on trouve les données sur les sondes Atlas par jour. Les détails sur les sondes reprennent les informations des connexions, des réseaux et autres. A priori, les détails des connexions des sondes sont disponibles pour la période du 13 mars 2014 jusqu'à ce jour [8], un fichier JSON par jour (voir un exemple dans la section **??** de l'annexe A). Les données de certains jours sont manquantes. La totalité des archives se trouve dans [5]. La taille d'une seule archive est entre 120 Ko et 921 Ko.

Deuxièmement, les résultats des mesures sont aussi archivés dans un serveur FTP. Seules les données des derniers 30 jours sont conservées en archives [9]. Les fichiers ont été nommés jusqu'au 15 mars 2018 de façon structurée comme suit :

$TYPE-$IPV-$SUBTYPE-$DATE.bz2

- $TYPE peut être traceroute, ping, dns, ntp, http, sslcert.
- $IPV version du protocole IP v4 ou v6.
- $DATE date au format YEAR-MONTH-DAY. (etc. 2017-06-13)
- $SUBTYPE type de mesure builtin ou udm.

En considérant toutes les possibilités des types, la quantité de données générées quotidiennement est environ 25 Go [10] et la taille des archives est entre 281M et 3.2G.

Depuis 15 mars 2018, les résultats des mesures ont été regroupés différemment. 24 archives par jour, une seule archive pour chaque heure et type de mesure.

8. 15/08/2018.
9. Source : https://data-store.ripe.net/datasets/atlas-daily-dumps/, consultée le 05/04/2018.
10. Source : https://ftp.ripe.net/ripe/atlas/data/README, consultée le 26/03/2018.

L'archive ne distingue pas entre mesures IPv4 et IPv6, entre mesures intégrées et personnalisées. Il existe un attribut **"af"** qui distingue entre IPv4 et IPv6 et l'identifiant de la mesure pour distinguer les mesures intégrées et celles personnalisées (identifiant > $1,000,000$).

Streaming API propose un service de récupération des résultats de mesures en temps réel, depuis les sondes publiques. Ainsi, elle fournit continuellement de nouveaux résultats en temps réel, obtenus par les sondes Atlas publiques, via une connexion de type HTTPS web-socket active tout le temps.

Troisièmement, on trouve des archives sauvegardées chaque semaine, elles décrivent les méta-datas des mesures. Une ligne objet JSON pour chaque mesure publique. Au moment de la consultation, la taille de chaque archive était entre 124 Mo et 1.5 Go. L'accès à ce jeu de données se fait de deux façons, via le téléchargement direct depuis un serveur FTP ou bien via streaming API. Les noms des archives sont bien structurés.

I.3.14 Les versions du firmware des sondes Atlas

En principe, toutes les sondes Atlas collectent la même information, indépendamment de leur version du firmware. On trouve les mêmes attributs [11] dans toutes les versions sauf de léger changements : ajout d'un ou de plusieurs attributs, la modification des noms des attributs, etc. Pour la simplification, nous donnons un identifiant entier pour chaque version, entre les parenthèses. Cet identifiant sera utilisé dans la suite de ce document.

Il existe plusieurs versions du firmware :

- La version 1 est identifié par 1 (1).
- La version 4400 est identifiée par une valeur entre 4400 et 4459 (2).
- La version 4460 est identifiée par une valeur entre 4460 et 4539 (3).
- La version 4540 est identifiée par une valeur entre 4540 et 4569 (4).
- La version 4570 est identifiée par une valeur entre 4570 et 4609 (5).
- La dernière version du firmeware [12] est 4610 (6).

I.3.15 Les limitations du RIPE Atlas

De nombreux travaux ayant exploité les données générées par les sondes Atlas. Néanmoins, ce système connaît des bugs et des limitations. Les membres

11. Attribut dans le sens du JSON : chaque résultat de mesure est enregistré comme étant un objet JSON.
12. A la date de consultation $25/01/2018$.

de la communauté RIPE Atlas s'engagent à remonter les bogues liées aux sondes Atlas. Tous les bogues sont répertoriés sous une rubrique dédiée [6].

RIPE Atlas connaît des limitations liées à la visualisation. Actuellement, RIPE Atlas supporte la visualisation des mesures de type ping ayant utilisé au maximum 20 sondes. Cette limitation concerne aussi le type traceroute, en effet, il est possible de visualiser seulement les mesures IPv6 built-in.

Afin d'éviter la surcharge des sondes et de l'infrastructure, RIPE Atlas a limité le nombre de mesures périodiques de 10 à la fois et de 10 mesures de type one-off vers n'importe quelle cible à un moment donné. De plus, il n'est pas possible d'utiliser plus de 500 sondes par mesure.

Pour les mesures one-off (non périodiques), une sonde peut effectuer au plus 10 mesures en parallèle. RIPE Atlas limite aussi la fréquence des mesures personnalisées. Un hébergeur d'une sonde peut effectuer :

- Ping chaque 60 secondes (par défaut 240 secondes).
- Traceroute chaque 60 secondes (par défaut 900 secondes).
- SSL chaque 60 secondes (par défaut 900 secondes).
- DNS chaque 60 secondes (par défaut 240 secondes).

Dans le cas d'une déconnexion, la sonde continue à effectuer les mesures. Pour la version 1 et 2, la sonde est capable de sauvegarder les 6 dernières heures de données. Tandis qu'avec la version 3, une sonde est capable de sauvegarder les résultats de plusieurs mois. Une fois la sonde est connectée, elle envoie les données à l'infrastructure centrale.

Concernant la consommation des crédits, RIPE Atlas limite cette consommation à $1,000,000$ crédits par jour.

I.3.16 Confiance aux données Atlas

De nombreux travaux ont exploité les données Atlas, cependant, peut-on faire confiance à la qualité des données ? les données sont-elles complètes ?

La question de la complétude des données est plus présente pour les mesures périodiques, celles qui se déroulent pendant une durée d et à un intervalle i. W. Shao et al. [27] ont traité les mesures manquantes. L'approche qu'ils ont adopté repose sur la corrélation entre l'absence de certaines mesures et les périodes durant lesquelles les sondes Atlas sont déconnectées. Pour précision, RIPE Atlas maintient les détails des connexions/déconnexions des sondes Atlas. Ils ont étudié les mesures en provenance des sondes v3, effectuées entre le $01/06/2016/$ et le

$01/07/2016/$ (UTC). Ils ont combiné les informations relatives à la connexion/-déconnexion des sondes et leurs mesures planifiées, leur approche se base sur l'attribut $timestamp$ qui est présent dans chaque résultat de mesure et dans les états de connexions.

Nous avons discuté des limitations du RIPE Atlas en terme de mesures autorisées par jour. Cela n'empêche qu'il est possible qu'un nombre important de mesures soit effectué. De plus, plus d'un utilisateur peut s'intéresser à la même sonde Atlas. C'est la question traitée dans le travail de T. Holterbach et al. dans [19], si les mesures lancées par les autres utilisateurs affectent les résultats obtenus par un autre utilisateur, si c'est le cas, comment s'y entreprendre. Les expériences réalisées ont montré la présence de l'interférence entre les mesures à destination des sondes et cela de deux manières. Premièrement, les mesures depuis et à destination des sondes Atlas augmentent le temps reporté par la sonde et ils ont conclu que l'amélioration du CPU a permis de limiter les interférences sur le temps mesuré par les sondes Atlas. Deuxièmement, ils ont conclu que les mesures perdent la synchronisation avec l'infrastructure d'Atlas, pendant plus d'une heure, à cause de la charge concurrentielle que subit le système d'Atlas. Dans ce cas, l'amélioration du matériel ne peut pas résoudre le problème.

I.4 Projets existants de mesures d'Internet

Dans les sections précédentes, on a développé le projet RIPE Atlas comme étant une plateforme pour la collecte des données des réseaux. Toutefois, il existe d'autres projets similaires à RIPE Atlas. Les sections suivantes reprennent une liste non exhaustive des projets similaires à RIPE Atlas.

I.4.1 Test Traffic Measurement Service

Avant l'arrivée du RIPE Atlas, Le RIPE NCC (Réseaux IP Européens Network Cordination Centre) a assuré la mesure de la connectivité entre les réseaux via d'autres plateformes, comme la plateforme Test Traffic Measurement Service (TTM). Il s'agit d'un projet qui permet de mesurer la connectivité entre un nœud source et un nœud destination sur Internet. C'était une des manières pour suivre la connectivité entre le réseau source et le réseau destination.

L'idée était la mise en place d'un dispositif, test-box, qui génère du trafic. Ce dernier n'affecte pas l'infrastructure réseau en matière de bande passante. De plus, il n'a pas l'accès aux données du réseau dans lequel il est mis en place.

Ce service a été assuré et géré, pendant une période de 6 ans, par une équipe au sein du RIPE NCC. Les fonctionnalités assurées par ce service étaient de tester l'accessibilité à une destination via le *ping*, ainsi, les mesures effectuées étaient

indépendantes des applications, elles dépendaient du réseau lui-même. RIPE NCC a arrêté la maintenance du TTM depuis le 1 juillet 2014 [24].

I.4.2 ProbeAPI

ProbeAPI [29] est une plateforme de mesure d'état du réseau, cette plateforme couvre 170 pays et des milliers d'ISPs. *ProbeAPI* est utilisée par les développeurs, les administrateurs des réseaux et les chercheurs, ils peuvent lancer des mesures d'un réseau depuis différents réseaux.

Le logiciel *ProbeAPI* s'exécute dans plusieurs systèmes : dans des ordinateurs (Win32/64), Android via une installation dans les mobiles et les tablettes et dans des routeurs au sein du DD-WRT.

> **DD-WRT** est un micrologiciel libre et gratuit, il est destiné aux routeurs sans fil et aux points d'accès. Il fonctionne avec un système d'exploitation Linux. Le rôle du DD-WRT est de remplacer le micrologiciel intégré aux routeurs par leurs fabricants. Ainsi, il est possible d'étendre des fonctionnalités du routeur en ajoutant d'autres fonctions supplémentaires.

ProbeAPI s'agit d'un logiciel qui tourne dans la machine de l'hébergeur. En conséquence, le suivi des réseaux dépend de la disponibilité de la machine qui le fait tourner. Cette dépendance affecte la disponibilité de la sonde logicielle, sa configuration et aussi les résultats de mesures.

Une étude comparative [30] entre les sondes Atlas et les sondes *ProbeAPI* est résumée dans le tableau I.3. En fin de cette étude, ils concluent qu'en comparant les résultats des mesures ICMP effectuées par les deux plateformes, des contrastes intéressantes ont été constatées. Les sondes Atlas ont montré un comportement stable lors de la réalisation des mesures, les résultats sont peu variables car les sondes sont indépendantes de l'utilisateur. Cependant, il était constaté qu'une forte variabilité au cours du temps pour les sondes logicielles (ProbeAPI), car elles dépendent fortement de l'hébergeur ; sa configuration réseau, sa disponibilité, etc.

Enfin, la force des sondes logicielles comme ProbeAPI réside dans sa capacité à effectuer des mesures depuis la couche application, la plus proche de l'utilisateur. L'exemple de l'évaluation du Time To First Byte et le taux de transfert dans deux pays.

« *Le **Time to First Byte** (TTFB) est le temps de chargement du premier octet, c'est la mesure qui nous permet d'évaluer la vitesse d'accès à un serveur. Plus la mesure est basse et plus le serveur commencera à servir les ressources rapidement.* »[a]

a. Source : `https://www.skyminds.net/calculer-le-time-to-first-byte-ttfb-dun-serveur/`, consultée le 10/08/2018.

RIPE ATLAS	PROBEAPI
Matériel homogène a un comportement prévisible	Matériel hétérogène a un comportement imprévisible
Connexions stables vu l'indépendance du software utilisateur	Connexions instables vu la dépendance du software utilisateur
Indépendance de l'OS et ses limitations ou vulnérabilités	Liaison à l'OS et ses limitations ou vulnérabilités, cependant utile pour les mesures au niveau application
La distribution des sondes est coûteuse, difficile de couvrir certaines régions	Mise en place du logiciel est rapide et moins chère, avec facilité de couvrir plusieurs régions
Les mesures HTTP se limitent aux ancres pour des raisons de sécurité	HttpGet, DNS et page-load sont disponibles via des librairies Mozilla et chromium, et ce pour toutes les destinations

TABLE I.3 – Comparaison entre sondes Atlas et ProbeAPI

Malgré le niveau de couverture assuré par ProbeAPI, cependant ces sondes se connectent et se déconnectent fréquemment, ce qui montre une forte volatilité. Cette volatilité est liée à la dépendance des sondes ProbeAPI de leur hébergeur ; tant qu'il est connecté, la sonde ProbeAPI est prête pour effectuer les mesures. Toutefois, si l'hébergeur est déconnecté, la sonde ProbeAPI ne peut pas effectuer des mesures, d'où le basculement fréquent entre les deux états : connectée et déconnectée.

I.4.3 Archipelago

Archipelago (Ark) [1] est l'infrastructure de mesures actives du CAIDA [2]. Elle est au service des chercheurs en réseau depuis 2007. L'objectif de ce projet est de couvrir un maximum de régions afin de collecter un maximum de mesures.

Ensuite, produire des visualisations qui améliorent la vue globale de l'Internet. Pour précision, c'est un Raspberry Pi 2nd gen.

I.4.4 DIMES

DIMES [28] est un logiciel qui devrait être installé dans une machine. Une fois installé, il fonctionne de sorte que la consommation d'énergie soit minimale et qu'il n'existe aucun impact sur les performances de la machine ou sur la connexion. L'objectif de *DIMES* est de collecter un maximum de données afin d'explorer la topologie d'Internet.

I.4.5 SamKnows

SamKnows [7] est une plateforme globale des performances d'Internet, elle regroupe les ISPs, ingénieurs, universitaires, codeurs et des organismes de régulation. Son objectif est d'évaluer les performances du haut débit des utilisateurs finaux et de trouver les problèmes avant que les clients ne commencent à se plaindre.

I.5 Quelques cas d'utilisation des données collectées par les sondes Atlas

Plusieurs travaux ont exploités les données collectées par les sondes Atlas. Ces travaux peuvent être classés de plusieurs manières, par exemple par thème, par type de mesures utilisé, etc. Nous distinguons les travaux ayant exploité les données collectées par les sondes Atlas à travers les mesures *built-in* ou bien ceux ayant utilisé les données des mesures personnalisées. Pour les premiers, ils permettent d'exploiter au mieux ces données sans surcharger le réseau des sondes Atlas, car ces données sont collectées quotidiennement. Cependant, les autres peuvent introduire une charge sur ces sondes. D'autre part, certains auteurs se sont intéressés aux données traceroute, d'autres aux données ping ou HTTP, etc. Nous allons présenter brièvement quelques travaux par thème.

I.5.1 Détection des coupures d'Internet

Les données collectées par les sondes Atlas ont permis de valider certaines coupures d'Internet, par exemple la coupure concernant le point d'échange AMS-IX (Amsterdam Internet Exchange). En 2015, Robert Kisteleki et al. [20] ont évalué l'état des pings en provenance des sondes Atlas à destination de trois ancres

Atlas qui se trouvent dans AMS-IX. En effet, peu de pings ont réussi d'atteindre leurs destinations, cependant, certains pings n'ont pas réussi à le faire. Ils ont conclu qu'il existe un problème du réseau, et le problème concerne les ancres plutôt que les sondes ayant lancé le ping. De même pour DNS, ils ont constaté l'absence des données DNS sensées être collectées par les ancres Atlas à destination du K-root.

I.5.2 Aide à la prise de décision

L'utilisation des sondes Atlas n'est pas limitée au domaine de recherche seulement, elle a permis aussi d'aider à la prise de décision pour certaines implantations et pour la mise en place des équipements comme les routeurs, les data-centers, les IXPs, etc.

Les ingénieurs de *Wikimedia Foundation* et du RIPE NCC ont collaboré dans un projet [9] pour étudier la latence vers les sites du Wikimedia. L'idée était d'exploiter la distribution des sondes Atlas dans le monde en vue de mesurer la latence vers les sites du Wikimedia. L'étude de la latence va permettre d'améliorer l'expérience des utilisateurs vers ces sites en réduisant la latence. Comme Wikimedia avait l'intention d'étendre son réseau de datacenters, ils ont profité des résultats de cette étude pour choisir les futurs emplacements de leurs data-centers.

Un groupe de chercheurs africains a évalué le routage inter-domaine afin d'étudier les emplacements adéquats pour la mise en place d'un IXP [25]. Après avoir analysé les données des mesures collectées par les sondes Atlas, ils ont constaté que le trafic de et à destination de l'Afrique quitte le continent vers les États-Unis ou bien l'Europe pour revenir en Afrique, d'où l'intérêt d'investir dans la mise en place des IXPs dans ce continent.

I.5.3 Le suivi des censures

En 2014, des chercheurs ont examiné les incidents de type content-blocking en Turquie et en Russie tout en prenant en considération le respect de l'aspect éthique des données. Ils ont aussi élaboré un aperçu comparatif des différents outils permettant de mesurer les réseaux [12]. C. Anderson et al. ont repris deux cas d'études où une censure a été appliquée : la Turquie et la Russie. L'idée de C. Anderson et al. est de créer des méthodes pour analyser ces censures en se basant sur les données collectées par les sondes Atlas.

Il existe plusieurs pratiques pour appliquer la censure. Ces pratiques dépendent des objectifs de cette censure ; bloquer un site web, rediriger le trafic, filtrer l'accès à travers des mots clés, etc.

En mars 2014, des utilisateurs turcs ont été interdits d'accéder au réseau social *Twitter*. Ce filtrage a été fait en utilisant *DNS Tampering* et *IP Blocking*.

Comme ces deux pratiques sont évaluables avec les sondes Atlas, ils ont planifié des mesures vers plusieurs destinations et depuis un nombre de sondes. Ces mesures sont reprises en détail dans le tableau I.4.

Cible	Type	Sondes	Fréquence (s)	Crédits
Twitter	SSL	10	3,600	2,400
YouTube	SSL	10	3,600	2,400
Tor	SSL	10	3,600	2,400
Twitter	DNS (U)	10	3,600	2,400
YouTube	DNS (U)	10	3,600	2,400
Twitter	Tracert	10	3,600	7,200

TABLE I.4 – Les détails des mesures effectuées dans le travail de C. Anderson [12]

L'analyse de données obtenues a permis de détecter six changements concernant les décisions du filtrage. Plus de détails se trouvent dans [12].

Quant à la Russie, les autorités ont décidé de mettre le blog d'*Alexei Navalny* sur *LiveJournal* dans la liste noire. En même temps, certains médias indépendants ont été aussi filtrés, l'exemple du site grani.ru. Pour le site *Grani*, les sondes Atlas ont reçu des réponses DNS aberrantes, d'où l'impossibilité de joindre *grani.ru*. Cependant, le filtrage du site *navalny.livejournal.com* a pris une autre forme, c'était une redirection d'adresse IP. La réponse d'une requête vers ce site donne 208.93.0.190 au lieu de 208.93.0.150. Ces deux adresses sont inclut dans le préfixe 208.93.0.0/22 géré par *LiveJournal Inc*. 208.93.0.190 correspond au contenu non-blacklisted, alors que 208.93.0.150 correspond au contenu correct.

I.5.4 Le suivi des performances d'un réseau

Les ancres Atlas

Les ancres Atlas ont des capacités avancées que les sondes Atlas. Les ancres servent comme cibles aux mesures des sondes. De plus, elles sont capables de fournir des détails sur l'état du réseau dans lequel elles sont déployées. S. Gasmi, un hébergeur d'une ancre Atlas, a développé un outil disponible au public [13]. A partir des données collectées par les ancres Atlas, cet outil permet d'analyser la qualité de la connectivité d'un réseau (ou d'un AS) et permet de suivre les changements relatifs à la topologie des réseaux

Par exemple, il a constaté que la vérification du BGP Prepending et des communautés BGP peut être faite en considérant les éléments suivants : adresse IP

13. Source : `http://ripeanchor.sdv.fr/`, consultée le 08/08/2018.

source, AS source, pays, le RTT du ping, les chemins du traceroute. En particulier, S. Gasmi a évalué deux corrélations. Dans un premier temps, il a visualisé la corrélation entre l'AS path et Round Trip Time (RTT). Il a regroupé des sondes par pays, ensuite, il a calculé, par ce pays, la moyenne du nombre de sauts et la moyenne du RTT des requêtes à destination de l'ancre depuis ces sondes Atlas. La figure I.8 reprend les résultats obtenus. Aucun renseignement sur la période des données. Pour les sondes en provenance de la France, le nombre de sauts et le RTT entre les sondes déployées en France sont faibles car l'ancre (la cible) se trouve aussi en France.

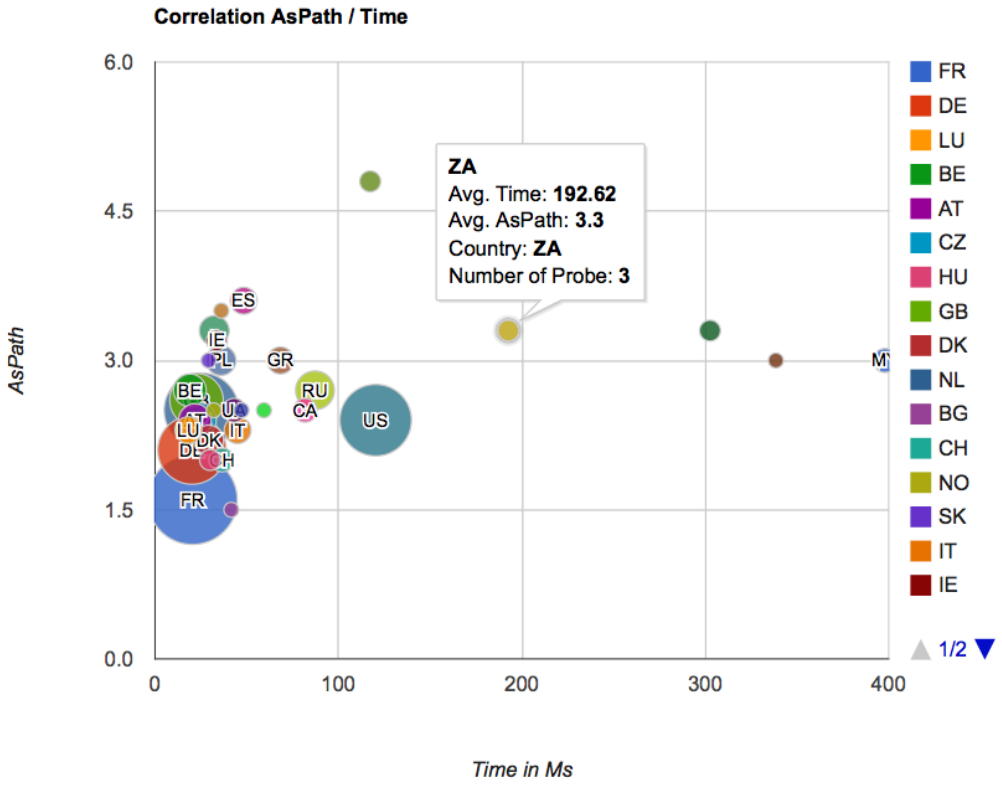


FIGURE I.8 – La corrélation entre la moyenne des AS paths et la moyenne des RTTs [15]

Ensuite, S. Gasmi a mesuré le RTT entre des sondes Atlas dans le monde et son ancre, il a aussi visualisé le nombre de sauts parcourus entre des sondes Atlas à travers le monde et son ancre. Ces deux visualisations permettent d'avoir une idée sur la latence entre certains pays et le pays de l'ancre en question. Plus de détails sur l'approche sont disponibles dans [15].

La vérification de la cohérence du Traceroute

Les chemins parcourus par traceroute pour aller d'une source s vers une destination d changent au cours du temps pour plusieurs raisons. Par exemple, suite à un changement BGP, à une répartition des charges, à des pannes des routeurs, à des pannes des liens physiques, etc.

Traceroute Consistency Check peut reprendre les chemins obtenus via traceroute au cours du temps . L'objectif est de suivre les nœuds apparaissant dans le chemin allant de s à d aux instants t, $t+1$, $t+2$, etc, et cela afin de voir les nœuds traversés plus fréquemment au cours du temps. Le chemin est mis à jour via Atlas streaming API.

L'outil proposé dessine les chemins traceroute comme étant un graphe dirigé, chaque nœud est coloré suivant sa cohérence. Le code source du projet est disponible sur GitHub [13]. La figure I.9 présente un exemple de la visualisation proposée. Ce résultat concerne la mesure 1663314 [14]. Ce sont des traceroutes à destination de l'adresse 213.171.160.1 entre 02/05/2014 13 : 00 et 03/05/2014 15 : 00.

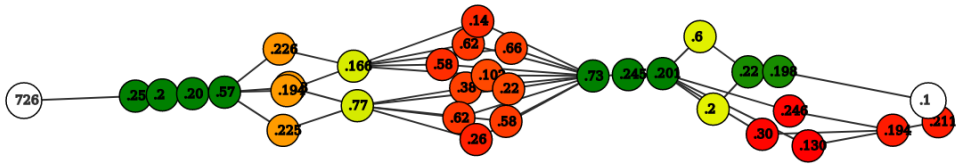


FIGURE I.9 – Visualisation des changements des chemins traceroute [13]

BGP+traceroute

C'est une combinaison des données BGP (RIPE RIS) et traceroute (RIPE Atlas). L'objectif de ce projet était de partir d'un AS path pour enfin géolocaliser les ASs. L'idée est de prendre un AS path des données RIPE RIS, puis, récupérer le préfixe (bloc d'adresses IP) annoncé via cet AS path, ensuite, lancer un traceroute vers une des adresses du bloc. Enfin, géolocaliser les ASs via les données du traceroute. Le code source et la présentation de ce projet sont disponibles GitHub [18, 17].

14. Source : `https://atlas.ripe.net/measurements/1663314/`, consultée le 05/08/2018.

BGP Atlas Monitor "BAM!"

Le projet *BAM* vise la visualisation , en temps réel, des informations utiles pour les opérateurs des réseaux. Par exemple, BAM montre la visibilité des préfixes obtenus par RIPE RIS. De plus, il est possible de voir le délai du *ping* obtenu via les sondes Atlas. Le code source est disponible sur GitHub [16]. En fournissant un ASN (identifiant d'un AS), *BAM* récupère les préfixes IPv4 et IPv6 et leur visibilité et il montre aussi les sondes dans cet AS. L'outil offre les fonctionnalités suivantes :

- Les préfixes annoncés par un ASN.
- La visibilité d'un ASN.
- La visibilité d'un préfixe.
- La liste des sondes par AS.
- Les objets route des préfixes.

Prédiction des routeurs provoquant la perte des paquets

Dans l'étude [14], Romain Fontugne et al. ont modélisé le comportement des routeurs, ils ont développé un modèle qui permet d'estimer l'endroit de la perte des paquets. A partir des traceroutes passant par un routeur r à une destination d, ils ont construit un modèle de forwarding pour ce routeur. Ce modèle reprend les prochains sauts (routeurs) et la fréquence de passage par ces derniers. Si le routeur r change le prochain saut qui a eu "l'habitude" de traverser pour atteindre d, alors, il est possible d'estimer l'origine de la perte de paquets.

I.5.5 Le suivi des détours dans un trafic local

Dans leur travail [11], E. Aben et al. avaient l'objectif de voir comment les mesures du RIPE Atlas peuvent fournir un aperçu sur le chemin du trafic local à un pays. Précisément si ce trafic traverse un autre pays en revenant au pays du départ. Ce qui pourrait aider à améliorer les performances et l'efficacité des IXPs. L'objectif est d'analyser les chemins identifiés dans le trafic d'Internet entre les sondes Atlas dans un pays donné et essayer d'identifier si le trafic traverse les IXPs.

France-IX est un point d'échange Internet (IXP) français créé en juin 2010. Afin d'apprendre la topologie de routage, un RIS route collector (RRC21) a été installé au sein du France-IX. Actuellement, la France compte 755 sondes Atlas et 9 ancres. Une ancre sur les 9 est installée au sein de France-IX.

Une des questions posées c'était si le trafic local de la France reste local, les sondes Atlas ne permettent pas de mesurer le trafic entre deux points, cependant, elles permettent de calculer le chemin entre deux points, adresses IP, ce qui permet d'inférer les sauts par lesquels le trafic passe le trafic. Le travail [10] s'intéresse au trafic depuis et vers une sonde en France en se basant sur l'étude dans [11].

Les résultats obtenus de l'analyse des détours peuvent être intéressants pour les opérateurs des réseaux afin d'améliorer leurs services, ainsi intéressants pour les IXPs tels qu'ils peuvent proposer des services de peering dans les endroits où il le faut.

I.5.6 Visualisation : indicateurs et dashboard

L'objectif de certains travaux était d'exploiter les données collectées par les sondes Atlas pour concevoir des tableaux des indicateurs. Par exemple, à partir des données de connexion/déconnexion des sondes Atlas, visualiser les sondes connectées, déconnectées, abandonnées. Un autre projet avait comme objectif la reconstruction d'un graphe reprenant les routeurs (nœuds) impliqués dans certains traceroutes, ainsi, identifier les nœuds les plus traversés. D'autres travaux ont repris les détails de la latence, essentiellement, sont les valeurs des RTT dans les pings et les traceroutes qui permettent de visualiser ce type d'information.

La liste des travaux basés sur le projet RIPE Atlas est très longue. Nous avons essayé d'énumérer quelques projets, les classer par thèmes, toutefois, ce n'est pas un classement unique, tel qu'on peut retrouver un travail dans plus d'une catégorie, ou bien les classer par un autre classement.

I.6 Conclusion

Dans ce chapitre, nous avons présenté les sondes Atlas et leur fonctionnement, ainsi que quelques travaux qui ont impliqué les données collectées par ces sondes dans plusieurs domaines tels que la prise de décision, le suivi des censures, la conception des tableaux de visualisation, etc. Ces données sont cruciales pour mener à toute analyse. Cependant, ces données sont massives, elles sont dans l'ordre d'une dizaine de Go pour une heure de mesures du type traceroute par exemple, y incluent toutes les destinations, d'où la nécessité d'impliquer des outils du Big Data pour une meilleure extraction d'informations utiles. En effet, le chapitre 2 aborde le sujet du Big Data dans ses différentes dimensions.

Chapitre II

La détection des anomalies dans les délais d'un lien

Dans le présent chapitre, nous allons présenter l'outil de détection des anomalies dans les délais d'un lien conçu dans le cadre du travail de R. Fontugne et al [14]. Nous avons choisi ce travail qui exploite des données massives en vue d'évaluer quelques technologies du Big Data.

II.1 Introduction

Le travail de R. Fontugne [14] et al exploite une des mesures effectuées par les sondes Atlas : la requête traceroute. L'idée de ce travail est de collecter les résultats des requêtes traceroutes effectuées par les sondes Atlas, d'en déterminer une valeur de référence sur base de l'historique, et ensuite de comparer la référence avec la valeur courante. La référence pour le délai d'un lien donné est mise à jour au fur et à mesure de l'analyse.

Dans leur travail [14], R. Fontugne et al. ont exploité la distribution répandue des sondes Atlas dans le monde afin d'étudier un des problèmes relatifs aux performances des réseaux informatiques.

En pratique, il est difficile d'avoir une idée globale et exacte sur la topologie de l'Internet. Toutefois, les opérateurs des réseaux informatiques disposent d'un aperçu de l'état des entités qui forment leurs réseaux, les relations entre ces entités ainsi que les éventuels problèmes. Avec la distribution abondante des sondes Atlas dans le monde en terme de type d'adressage : sondes Atlas supportant seulement l'adressage IPv4, d'autres qui supportent en plus l'adressage IPv6, en terme de la diversité géographique, la diversité en terme d'ASs hébergeant les sondes Atlas, etc, il était possible d'aborder les délais dans les réseaux informatiques à travers de nouvelles approches, reposées sur des fondements statistiques. Parmi les points

forts de l'analyse menée par R. Fontugne et al., c'était la possibilité de valider les méthodes proposées avec des événements marquants sur Internet.

Le travail de R. Fontugne et al. reprend trois méthodes basées sur les données collectées par les sondes Atlas, chaque méthode reflète l'approche utilisée pour étudier les performances des réseaux informatiques. Ces méthodes sont les suivantes :

1. la détection des changements des délais que subissent les liens intermédiaires dans les traceroutes ;

2. la conception d'un modèle de forwording pour un routeur donné. Ce modèle prédit l'acheminement du trafic afin d'identifier les routeurs et les liens en panne dans le cas d'un problème de perte de paquets ;

3. la création d'un score par Système Autonome afin d'évaluer l'état de ce dernier.

Dans la suite de ce travail, nous allons reprendre seulement la première méthode. Il s'agit d'étudier le délai d'un lien topologique, c'est le délai entre deux routeurs adjacents sur Internet.

II.2 L'étude des délais des liens

II.2.1 Les données utilisées dans l'analyse des délais

La méthode conçue pour la détection des changements des délais se base sur des fondements statistiques. Ces derniers sont capables de montrer leurs performances si la taille des échantillons [1] considérés est grande. Afin de surveiller un grand nombre de liens sur Internet, il faut avoir un grand nombre de sondes Atlas avec une certaine diversité et qui sont capables de collecter une quantité importante de données relatives aux performances des réseaux informatiques, c'est ce qu'assure le projet RIPE Atlas.

Le travail de référence implique principalement les mesures de traceroutes, ainsi deux catégories de mesures sont utilisées :

– *builtin* : ce sont les traceroutes effectués par toutes les sondes Atlas vers les instances des 13 serveurs DNS racines. Les traceroutes sont effectués chaque 30 minutes. En pratique, certains serveurs racines DNS déploient

1. L'échantillon de la métrique qui caractérise un lien : RTT différentiel.

l'anycast. Au moment de la réalisation du travail de référence, c'étaient des traceroutes vers 500 instances des serveurs DNS racines ;

> **DNS Anycast** est une solution utilisée pour accélérer le fonctionnement des serveurs DNS. Les serveurs DNS adoptant cette approche fournissent des temps de réponse plus courts, et ce partout dans le monde. Les requêtes en provenance de l'utilisateur sont redirigées vers un nœud adéquat suivant un algorithme prédéfini.

– *anchoring* : ce sont les traceroutes effectués par environ 400 sondes Atlas à destination de 189 serveurs [2] et ce chaque 15 minutes.

En ce qui concerne les traceroutes analysés, le tableau II.1 reprend plus de détails.

	Nombre de traceroutes	**Nombre de sondes**
IPv4	2.8 billion	11,538
IPv6	1.2 billion	4,30

TABLE II.1 – Récapitulatif des traceroutes utilisés dans le travail de référence

L'étude des délais ne concerne pas les adresses privées, ainsi, le suivi des délais ne concerne pas les réseaux privés. De plus, ce suivi se base sur les requêtes de type traceroute, et traceroute reprend une partie de la topologie de l'Internet. En effet, les liens considérés sont ceux topologiques et ne sont pas les liens physiques.

II.3 Définition du RTT différentiel d'un lien

II.3.1 RTT différentiel

Avant de définir le RTT différentiel, soit la définition du RTT :

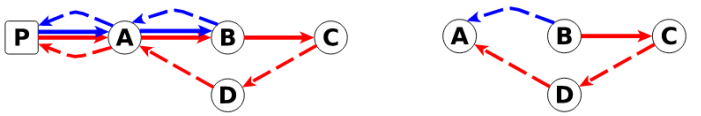
2. Sondes Atlas ayant des fonctionnalités avancées.

ICMP :Internet Control Message Protocol est un protocole utilisé pour véhiculer des messages de contrôle sur Internet.
RTT est obtenu en calculant la différence entre le timestamp associé à l'envoi du paquet sondé et le timestamp associé à la réception de la réponse ICMP. C'est une métrique pour évaluer les performances d'un réseau en matière de temps de réponse. Les mesures du RTT sont fournies par les utilitaires traceroute et ping. En ce qui concerne traceroute, ce dernier fournit les sauts impliqués dans le chemin de forwarding, c'est le chemin parcouru par le trafic entre la source et la destination. RTT inclut le temps de transmission, du quering et du traitement.

La figure II.1 (a) illustre le RTT entre la sonde P et les deux routeurs B et C. Le RTT différentiel entre deux routeurs B et C adjacents, noté Δ_{PBC}, est la différence entre le RTT entre la sonde P et B (bleu) d'une part, et le RTT entre la sonde P et C (rouge) dans la figure II.1 (b).

$$\begin{aligned}\Delta_{PBC} &= RTT_{PC} - RTT_{PB} \\ &= \delta_{BC} + \delta_{CD} + \delta_{DA} - \delta_{BA} \\ &= \delta_{BC} + \varepsilon_{PBC}\end{aligned}$$

où δ_{BC} est le délai du lien BC et ε_{PBC} est la différence entre les deux chemins de retour (B vers P et C vers P).



(a) Round-trip to router B (blue) and C (red).

(b) Difference of the two round-trips (Δ_{PBC}).

FIGURE II.1 – Source : [14]

II.3.2 Le principe de la détection des changements des délais

L'évolution du délai d'un lien est déduit de l'évolution de son RTT différentiel. Reprenons la formule du RTT différentiel du lien BC : δ_{BC} + ε_{PBC}. Supposons qu'on dispose d'un nombre n de sondes Atlas P_i, $i \in [1, n]$, telles que toutes les sondes ont un chemin de retour différent depuis B et depuis C. En effet, les RTTs différentiels pour chacune des sondes Atlas Δ_{P_iBC} partagent la même composante δ_{BC}.

On sait que les n sondes Atlas sont indépendantes ; le chemin de retour de chacune est indépendant, ainsi les valeurs des ε_{P_iBC} sont indépendantes. L'indépendance de ces valeurs implique que la distribution Δ_{P_iBC} est estimé d'être stable au cours du temps si δ_{BC} est constant. Cependant, un changement significatif de la valeur de δ_{BC} influence les valeurs des RTTs différentiels. Dans ce cas, la distribution des RTTs différentiels changes.

La détection des anomalies des délais repose sur un théorème très important en statistiques, c'est le théorème central limite (TCL). Ce théorème annonce que si on a une suite de variables aléatoires X_i indépendantes ayant la même espérance μ et la même variance σ^2, la moyenne de ces variables aléatoires est une variable aléatoire qui suit une loi normale.

II.4 Description des paramètres de l'analyse des délais

La détection des changements des délais nécessite l'ajustement d'un nombre de paramètres. La valeur de chaque paramètre est relative au fondement utilisé théorique ou bien empirique, qui a été justifié par les auteurs du travail de référence. Ci-dessous les paramètres à ajuster avant de lancer une analyse. Chaque paramètre sera défini dans son contexte.

start : c'est la date de début de l'analyse. Ce sont les traceroutes effectués par les sondes Atlas à partir de cette date qui seront analysés.

end : c'est la date marquant la fin de l'analyse. Comme le paramètre *start*, c'est la date des derniers traceroutes capturés par les sondes Atlas à considérer dans la présente analyse.

timeWindow : ce paramètre est exprimé en seconde. La durée de l'analyse, qui est le temps écoulé entre *start* et *end*, est divisée sur des périodes de même taille : *timeWindow*, c'est qui est illustré dans la figure II.2.

minSeen : comme l'analyse est faite sur plusieurs périodes de durée *timeWindow*, le paramètre *minSeen* indique le nombre de fois où le lien doit avoir été identifié. Par exemple, un lien peut être identifié dans 3 d_i, ou bien être identifié une seule fois durant toute la période de l'analyse.

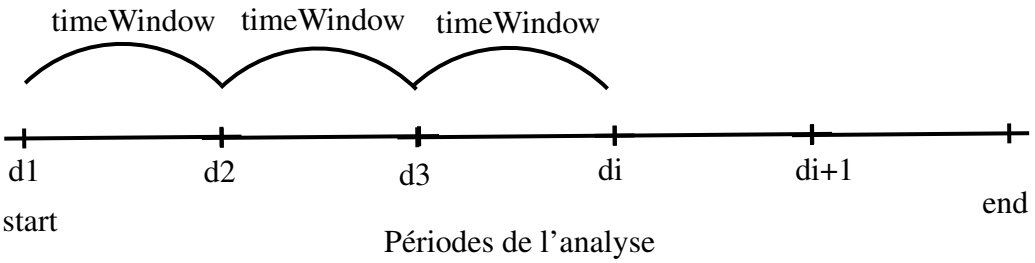


FIGURE II.2

alpha : c'est le paramètre de la moyenne mobile exponentielle calculé.

« *Les méthodes de lissage exponentiel sont un ensemble de techniques empiriques de prévision qui accordent plus ou moins d'importance aux valeurs du passé d'une série temporelle.* [3] »

Pour calculer la prochaine valeur de la médiane des RTTs différentiels de référence $\overline{m}_t$ du timeWindow courant t, soient :

m_t la médiane des RTTs différentiel observée pour un lien durant le timeWindow t.

$\overline{m}_{t-1}$ la médiane des RTTs différentiels de référence durant le timeWindow $t-1$, la prochaine valeur de la médiane de référence $\overline{m}_t$ est obtenue par :

$$\overline{m}_t = \alpha\, m_t + (1- \alpha)\, \overline{m}_{t-1}$$

Pour précision, m_t et $\overline{m}_t$ désignent deux ensembles différents. Le premier est la médiane des RTTs différentiels de chaque période d_i. Toutefois, le deuxième est construit, à partir de *minSeen* , en utilisant la méthode de la moyenne mobile exponentielle où le calcul de la médiane prend en compte les médianes des RTTs différentiels précédents suivant le paramètre α.

Le paramètre $\alpha \in (0, 1)$ est le seul paramètre à définir dans le calcul du $\overline{m}_t$. Ce paramètre contrôle l'importance des mesures précédentes par rapport aux mesures récentes.

«*Plus α est proche de 1 plus les observations récentes influent sur la prévision, à l'inverse un α proche de 0 conduit à une prévision très stable prenant en compte un passé lointain.* [4]». Dans la présente étude, le paramètre α est préféré d'être petit, précisément, il prend par défaut 0.05 comme valeur.

3. Source : https://perso.math.univ-toulouse.fr/lagnoux/files/2013/12/Chap6.pdf, conseultée le 30/09/2018.

4. Source : https://www.math.u-psud.fr/~goude/Materials/time_series/cours3_lissage_expo.pdf, consultée le 30/09/2018.

II.5 L'évolution du RTT différentiel d'un lien et la détection des anomalies

L'entrée de l'algorithme de détection est un ensemble de traceroutes. Un traceroute est un ensemble de sauts auxquels sont jointes l'identifiant de la sonde ayant effectué la requête traceroute et la destination de la requête. Chaque saut est décrit par un ensemble de signaux. Chaque signal décrit le routeur ayant émis une réponse à la sonde parmi les routeurs traversés avant d'atteindre la destination finale. Pour le saut i, on note trois signaux $S_{i,j}; j \in [1,3]$ dont le routeur émettant le signal est $from_{i,j}$ avec un RTT égal à $rtt_{i,j}$, avec $j \in [1,3]$ comme illustré dans la figure II.3.

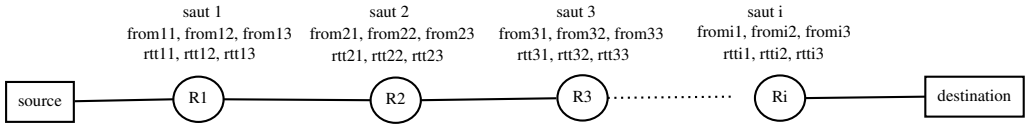


FIGURE II.3

II.5.1 Paramètres de l'algorithme de détection

- Objectif : suivre l'évolution du délais d'un lien au cours du temps en suivant son RTT différentiel par période du temps (*timeWindow*).
- Entrées : date de début de l'analyse *start*, date de la fin de l'analyse *end*, lien à analyser (*link*) et la fenêtre de l'analyse (*timeWindow*).
- Sorties : les dates d_i pendant lesquelles des anomalies ont été détectées.

Soient d_1,d_2, ..., d_N les périodes entre *start* et *end* où

$$d_{i+1} \text{ - } d_i = d_{j+1} \text{ - } d_j = step$$

step est la durée d'une période en secondes, 3600 pour fenêtre d'une heure.

pour tout i et j dans$[1, N]$

II.5.2 Processus de détection des anomalies

Le processus de détection des anomalies passe par plusieurs étapes dont les détails sont donnés à la Figure II.5. D'abord on trie les traceroutes à analyser par période (étape 1). Ensuite, on prépare les traceroute en appliquant sur les traceroutes d'une période un nombre d'opérations (étapes entre 2 et 6). A la fin de la préparation des traceroutes de toutes les périodes, on compare les le délai d'un lien avec les valeurs de référence (étape 7).

1. Trier les traceroutes à analyser par *timeWindow*. En effet, chaque d_i est associé à un ensemble de traceroutes ayant été effectués entre d_i et $d_i + step$.

Les opérations (2 à 6) concernent les traceroutes par tout d_i.

2. Vérification de la validité de chaque traceroute Ces vérifications reprennent les points suivants :

- élimination des traceroutes échoués complètement ;
- élimination des signaux contenant une adresse IP privée ;
- élimination des signaux qui ne contiennent pas un RTT ou qui contiennent un RTT négatif ;
- élimination des signaux échoués.

Il existe deux sortes d'échecs pour un traceroute : échec complet et échec partiel. Dans le premier, la sonde ne réussit pas à atteindre la destination. Par conséquent, la liste des sauts est vide. Toutefois, dans le deuxième cas, l'échec peut concerner un ou plusieurs saut, ou bien il peut concerner un, deux ou trois signaux d'un saut.

3. Calcul de la médiane des RTTs par saut. Pour tout saut d'un traceroute, on calcule la médiane des RTTs par adresse IP. Soit le saut $h = \{s_i\}$ [5] où s est un signal, mediane_rtt (h) = $\{median(\{s_i.rtt_{i,j}\})\}$ [6] pour tout signal s ayant la même adresse IP. Autrement dit, le nouveau saut du traceroute est reconstruit en regroupant les signaux par adresse IP et ensuite en calculant leurs RTTs.

4. Inférence des liens topologiques par traceroute. Un lien topologique est formé par chaque paire de routeurs consécutifs dans un traceroute. De manière générale, la Figure II.4 illustre la constitution des liens possibles dans un traceroute. Soient RAi, avec $i \in 1, 2, ..., N$, l'ensemble des routeurs pour le saut A et RBj, avec $j \in \{1, 2, ..., M\}$, l'ensemble des routeurs pour le saut B, avec N et M deux entiers.

Ainsi, les liens construits sont ceux partant de tout RAi vers tout RBj, où A et B sont deux sauts consécutifs. A l'issue de cette étape, pour tout traceroute, on obtient la liste des liens possibles tout en reprenant des informations générales de la requête traceroute.

5. La notation $\{a\}$ désigne un ensemble de type a.

6. La notation $a.b$ désigne la valeur de l'attribut b de l'objet a, ainsi $\{a.b\}$ est l'ensemble des b obtenus à partir d'un ensemble de type a .

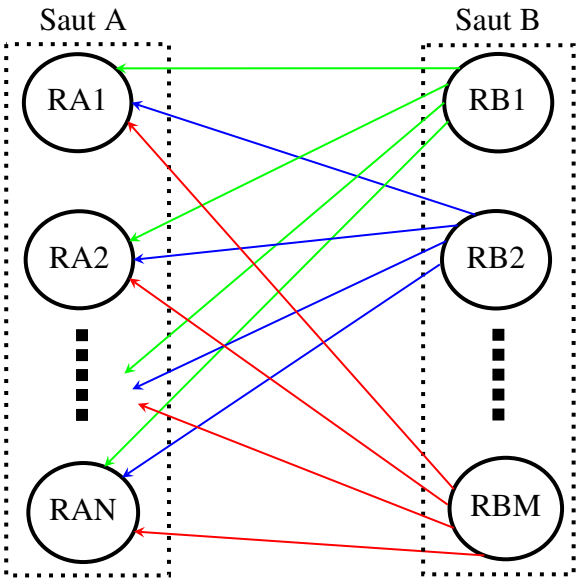


FIGURE II.4 – Inférence des liens possibles entre les routeurs des deux sauts consécutifs RAi et RBj

5. Caractérisation des liens avec leurs RTTs différentiels. A cette étape, on calcule le RTT différentiel d'un lien en calculant la différence entre les RTTs des deux routeurs du lien en question. En plus du RTT différentiel, on note aussi la sonde Atlas ayant effectué la requête traceroute où le lien a été identifié.

6. Fusion des informations d'un lien. Etant donné qu'un lien (IP1, IP2) peut être identifié plusieurs fois pendant une même période d_i d'une part, et le lien (IP2, IP1) est similaire [7] au lien (IP1, IP2) d'autre part, la fusion permet de construire une nouvelle distribution des RTTs différentiels caractérisant le lien (IP1, IP2) qui reprend les RTTs différentiels du (IP1, IP2) et du (IP2, IP1).

A la fin de l'étape 6, tous les traceroutes sont analysés tout en identifiant leurs liens, et ce par d_i. A présent, l'objectif c'est d'identifier les dates pendant lesquelles des anomalies ont été détectées. Pour ce faire, l'idée du travail de référence c'est de conserver, pour un lien donné, une référence du RTT différentiel médian qui sera d'abord comparée avec la médiane courante du RTT différentiel et ensuite mettre à jour cette référence tout au long de la période d'analyse.

7. Calcul de la médiane des RTTs différentiels et l'intervalle de confiance courant du lien analysé. Pour un lien donné, on calcule la médiane des RTTs dif-

7. La similarité est mesurée par le RTT différentiel.

férentiels d'une d_i, ensuite on calcule les deux bornes de l'intervalle de confiance pour d_i.

8. Mise à jour de la médiane et de l'intervalle de référence du lien analysé. La médiane des RTTs différentiels de référence sont d'abords comparés avec ceux de la période d_i courante. Ensuite, ces références sont mises à jour pour prendre en compte ces nouvelles valeurs. A l'issue de cette comparaison, la liste des dates des anomalies est mise à jour.

II.5.3 Vue globale des étapes de la détection des anomalies à travers l'évolution des RTTs différentiels

La figure II.5 présente la succession des étapes de la détection des anomalies dans les délais d'un lien donné.

II.6 La caractérisation des anomalies dans les délais d'un lien

La section II.5 décrit la détection des anomalies dans les liens à travers l'évolutions des RTTs différentiels d'un lien tout au long d'une période donnée. Toutefois, il existe une autre exploitation des traceroutes pour la détection des anomalies similaire à celle présentée précédemment, la différence se voit au niveau l'importance d'un lien. Ainsi, on analyse pas tout liens, plutôt les liens ayant certaine diversité en matière d'ASs ayant identifié ce lien, le nombre de sondes ayant identifié ce lien, etc.

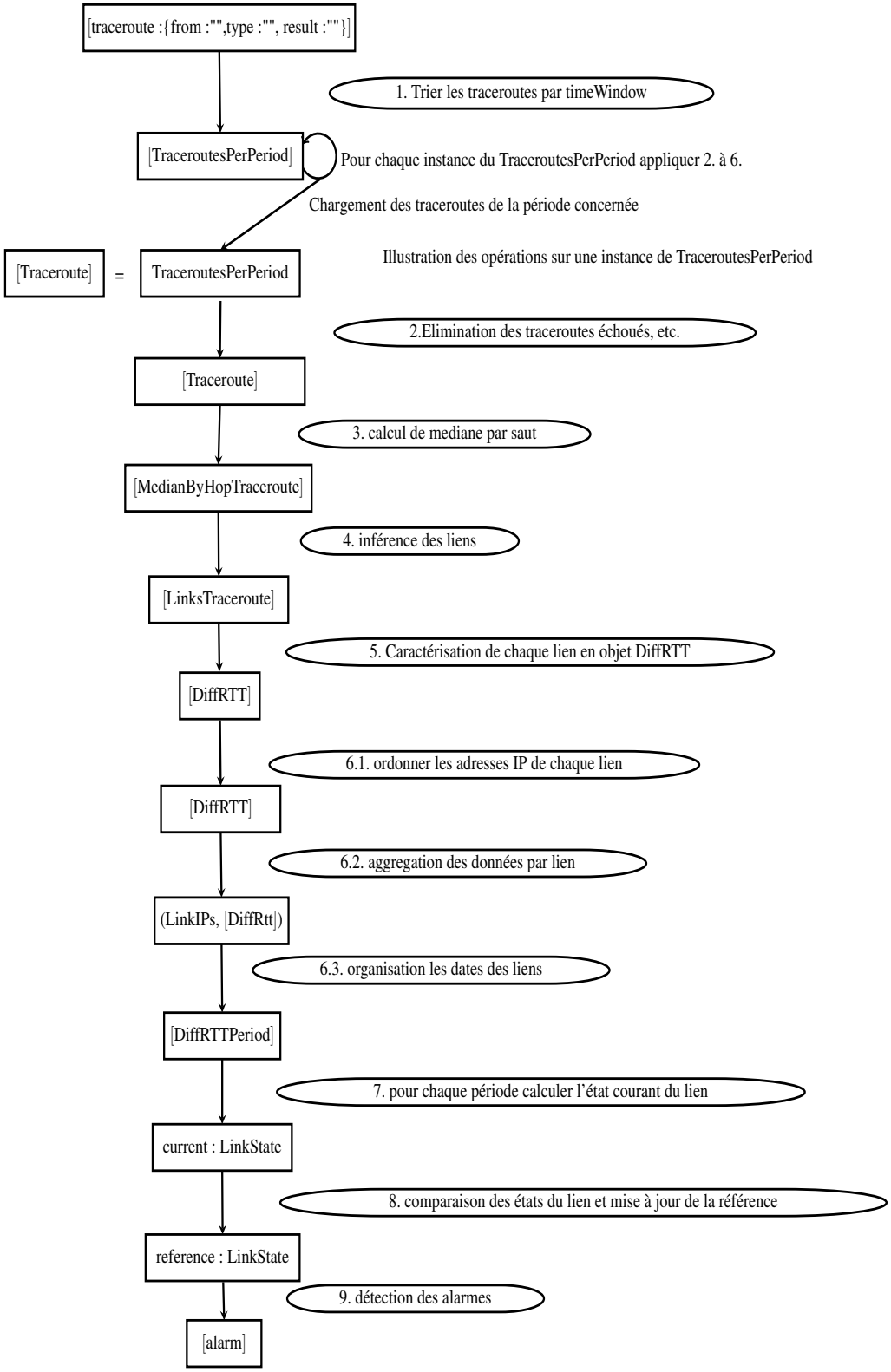


FIGURE II.5 – Le processus de la détection des anomalies dans les délais des liens

Chapitre III

Introduction au Big Data

III.1 Introduction

Big Data est un terme associé aux données massives, rapidement générées, ayant une grande diversité où les outils traditionnels sont incapables de gérer ces données. La complexité du Big Data vient du fait que tout type de données peut être utilisé, en vue de livrer la bonne information à la bonne personne et au bon moment afin d'aider à prendre les bonnes décisions.

III.2 Quelques concepts associés au Big Data

III.2.1 Définition du Big Data : Volume, Vélocité, Variété et Véracité

IBM définit Big Data suivant les quatre dimensions suivants : volume, variété, vélocité et véracité.

Volume de données La quantité de données manipulées par les outils traditionnels de la gestion des données est de l'ordre de Gigaoctets (GB) et de Téraoctets (TB). Toutefois, le Big Data est mesuré en Pétaoctets (PB) et Exaoctets (EB). Une des premières applications du Big Data est la recherche dans Word-Wide Web (WWW). Selon une étude ([26], 2013) de l'International Data Corporation (IDC), le volume de données va atteindre 40 Zettaoctets [1] par entreprise en 2020 .

Vélocité de données le Big Data est généré à travers des milliards d'appareils, ces données générées sont communiquées avec la vitesse de la lumière à travers

1. $1ZB = 1000000000000GB$

l'Internet. L'augmentation de la vitesse de l'Internet est une des raisons ayant favorisé l'augmentation de la vitesse de la génération des données. Par exemple, Wallmart (international discount retail chain) génère environ 2.5 Pétaoctets de données chaque heure via les transactions de ses consommateurs [2].

Variété de données Le Big Data inclut toutes les formes des données, des fonctions diversifiés des données et des sources multiples des données.

Le premier aspect de la variété des données massives est la **forme** de celles-ci, les données manipulées incluent du texte, des graphes, des cartes, des vidéos, des photos, etc.

Le deuxième aspect de la variété des données massives concerne les **fonctions** assurées par ces données. Des données sont issues des conversations humaines, d'autres des transactions des consommateurs, ou bien des données archivées, etc.

La source du Big Data est le troisième aspect de la variété. Des données sont en provenance des téléphones mobiles, des tablettes ou des ordinateurs portables, des fichiers journaux, du réseau de capteurs, etc. Les sources du Big Data peuvent être classées en trois grandes catégories : communications ***human to human*** comme les conversations échangées dans les réseaux sociaux, communications ***human to machine*** comme l'accès des utilisateurs aux données dans le web et enfin communications ***machine to machine*** comme les données issues de la communication entre les capteurs dans un réseau de capteurs.

Véracité de données La véracité concerne la crédibilité et la qualité des données. La mauvaise qualité de données est due à de nombreuses raisons telles que les pannes techniques comme le dysfonctionnement des appareils comme les capteurs, les erreurs humaines, etc. De plus, les données peuvent être intentionnellement erronées pour des raisons de concurrence ou des raisons stratégiques.

III.2.2 Une architecture du Big Data

L'architecture présentée dans ce qui suit est celle proposée dans [23], elle est illustrée dans la figure III.1.

Il existe plusieurs sources de données. Ces données passent ensuite par *ingest system*. Ensuite, les données peuvent passer par deux systèmes : un *stream processing* et un *batch processing*. Les résultats de ce traitement peuvent être envoyées vers les bases de données NoSQL pour une utilisation ultérieure, ou bien utiliser ces résultats comme entrées pour d'autres applications. Ainsi, une solution Big

2. Source : `https://www.bernardmarr.com/default.asp?contentID=690`, consultée le 30/06/2018.

Data comprend typiquement ces couches logiques. Chacune des couches peut être représentée par une ou plusieurs technologies disponibles. Reprenant chacune des composantes logiques :

Big Data sources layer Le choix des sources de donnés pour une application donnée dépend des objectifs qui dirigent l'analyse en question. Les sources avec leurs différents aspects sont détaillées dans la section III.2.1.

Data Ingest layer Cette couche permet de récupérer les données depuis les différentes sources de données. Ainsi, les données sont accueillies à travers des points d'entrées multiples. Ces points sont capables d'acquérir ces données ayant une vélocité variable ainsi qu'une quantité aussi variable. Après avoir traversé *Data Ingest layer*, les données sont envoyées au *batch processing system*, au *real-time processing system*, ou bien à un système de stockage particulier.

Batch processing layer Les données reçues sont celles en provenance du *Data Ingest* ou bien d'une des bases de données NoSQL. Ces données sont ensuite traitées en utilisant les techniques de la programmation parallèle, en vue de fournir les résultats souhaités. La présente couche doit avoir connaissance des sources de données, les types de données, les algorithmes qui vont travailler sur ces données et enfin les résultats souhaités. Les résultats des traitements peuvent être utilisées par une des applications ou bien sauvegarder ces données dans une des bases de données adaptées.

Stream Processing layer Cette couche approvisionne les données directement d'une des entrées du *Data Ingest layer*. Pareillement à *Batch processing layer* en matière des techniques de la programmation parallèle utilisées ainsi que la nécessité d'avoir les détails sur les sources des données, les types de données et les résultats souhaités.

Data organizing layer Le rôle de cette couche est d'organiser les données afin de faciliter l'accès à ces dernières. Ce sont les données obtenues de la part de la couche *Stream Processing* ainsi que la couche *Batch processing*. Cette couche est représentée par les bases de données NoSQL. Il existe plusieurs catégories de bases de données NoSQL.

Infrastructure layer Cette composante est responsable de la gestion des ressources de stockage, les ressources du calcul et la gestion de la communication. Les fonctionnalités de cette couche sont fournies à travers le cloud computing.

Distributed File System Layer Cette couche assure le stockage d'une grande quantité de données, de sorte que ces données soient rapidement et facilement accessibles à toute les couches qui forment un système du Big Data. C'est ce que assure Hadoop Distributed File System (HDFS).

Data consumption Cette dernière couche utilise les sorties produits par les couches de l'analyse. Les résultats fournis peuvent être sous format de rapport, des dashboards, des visualisations, un moteur de recommandation ou tout autre format.

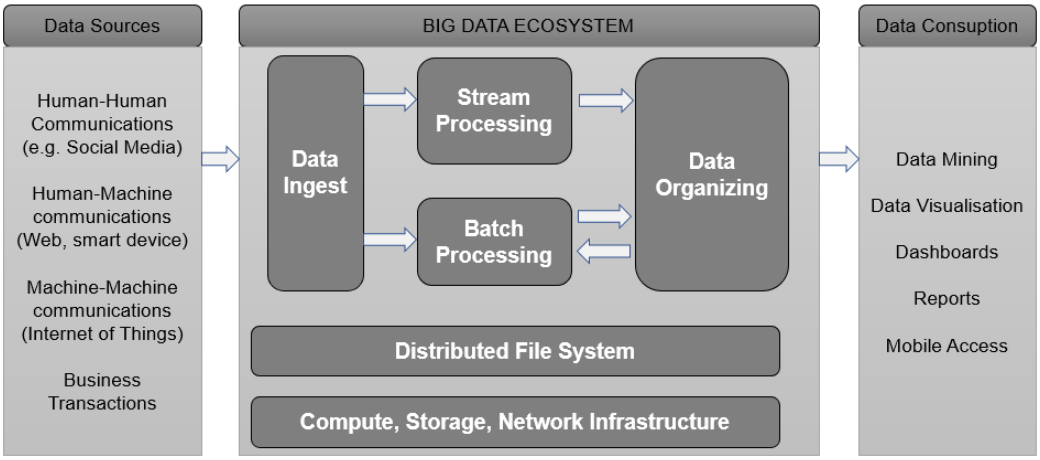


FIGURE III.1 – Architecture générique du Big Data [23]

III.2.3 Les bases de données NoSQL (Not Only SQL)

Introduction

Au cours de ces dernières années, on constate une révolution dans le stockage de données non structurées ayant une taille importante. Car les objets à sauvegarder sont complexes ; ils sont issus de sources hétérogènes. Cette complexité a mis en question les performances des bases de données relationnelles.

Le terme NoSQL est apparu pour la première fois en 1998. Carlo Strozzi a parlé des bases de données relationnelles qui n'utilisent pas le SQL comme langage d'interrogation des tables. Des années plus tard, des solutions open source basées sur ce concept ont vu le jour.

Les bases de données relationnelles sont utilisées par la majorité des entreprises pour plusieurs raisons : la facilité d'utilisation, la disponibilité de plusieurs produits et développeurs, etc. Ces dernières années, avec l'augmentation exponentielle de la quantité de données générées par certaines entreprises, ces dernières

ont constaté l'insuffisance des Systèmes de Gestion de Bases de Données Relationnelles (SGBDR) pour répondre à leurs besoins.

Les bases de données NoSQL sont conçues pour gérer des volumes de données importants. L'idée de base de ces bases, c'est d'abord assurer la capacité de stocker des données de grande échelle dont leur quantité évolue rapidement, voire exponentiellement. En deuxième lieu, les données stockées doivent être interrogées avec efficacité. Les données stockées dans les bases de données NoSQL n'obéissent pas à un modèle prédéfini comme le cas des bases de données relationnelles. Cette flexibilité est une des caractéristiques des bases de données NoSQL.

Types de base de données NoSQL

Il existe quatre catégories distinctes de bases de données NoSQL. Chaque catégorie répond à des besoins particuliers. Ainsi, on distingue les bases de données clé-valeur, document, graphe et colonne.

Clé-valeur Une base de données de type clé-valeur repose sur le paradigme clé-valeur ; chaque donnée, que ce soit un nombre, du texte ou tout autre élément est associé à une clé unique. Cette clé est le seul moyen d'accéder aux données stockées. Dans les bases de données NoSQL de type clé-valeur, les enregistrements n'adhèrent pas à une structure prédéfinie. Par exemple, on peut avoir le premier enregistrement de type entier et le deuxième enregistrement de type texte. Cela assure une forte évolutivité grâce à l'absence d'une structure ou de typage. La figure III.2 reprend un exemple d'une base de données NoSQL de type clé-valeur.

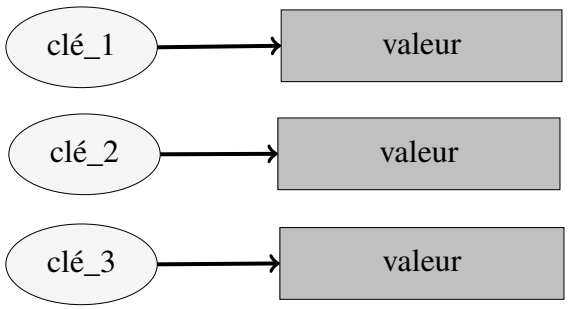


FIGURE III.2 – Illustration d'une base de données NoSQL de type clé-valeur

Document Une base de données NoSQL de type document permet de stocker les données en reposant sur le paradigme clé-valeur. Toutefois, les valeur stockées sont complexes, il s'agit de documents de type JSON, XML, etc. L'accès aux données d'un enregistrement peut se faire de manière hiérarchique. La possibilité de stocker des objets complexes et hétérogènes est un des points forts des

bases de données NoSQL de type document. Un exemple est fourni dans la figure III.3.

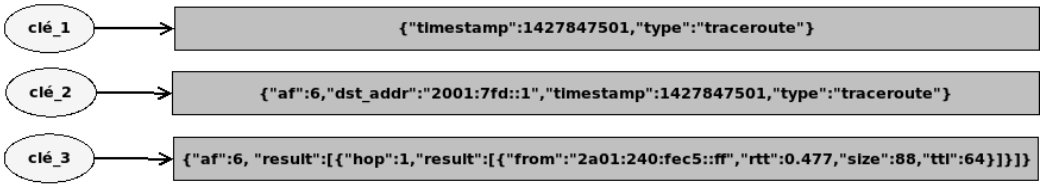


FIGURE III.3 – Illustration d'une base de données NoSQL de type document

Colonnes Dans les bases de données traditionnelles, les données sont stockées sur des lignes. Dans le cas d'une base NoSQL orientée colonne, les données sont stockées par colonne. L'interrogation de ce type de bases travaille sur une colonne particulière sans devoir passer par les autres colonnes comme dans les bases de données relationnelles classiques. Une base de données de type colonne, illustrée dans la figure III.4, est adaptée pour les requêtes analytiques comme les requêtes d'agrégation (moyennes, maximum, etc).

clé	prob_id
1	233
4	10

clé	af
1	4
2	6
3	4
4	4

clé	msm_id
1	5001
2	5006
3	16345
4	6026

FIGURE III.4 – Illustration d'une base de données NoSQL de type colonne

Graphe Dans une base de données de type graphe, les données stockées sont les nœuds, les liens et les propriétés sur les nœuds et sur les liens. Un exemple concret des bases de données NoSQL de type graphe est le réseau social ; chaque entité représente une personne et les relations entre ces personnes peuvent prendre plusieurs formes. Comme il est illustré dans la figure III.5.

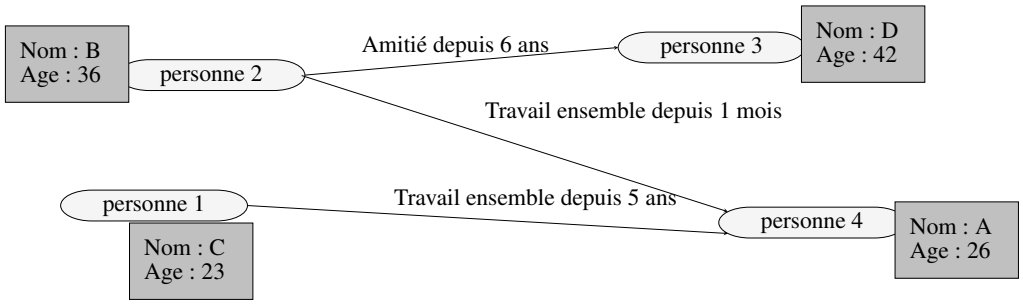


FIGURE III.5 – Illustration d'une base de données NoSQL de type graphe

Il existe plusieurs implémentations des quatre types de bases de données. Chaque implémentation favorise un ou plus des éléments suivants : la disponibilité des données, la cohérence des données et la tolérance au partitionnement. C'est ce qu'explique le théorème CAP.

Big Data et le théorème CAP :

Le théorème CAP annonce que dans le cadre d'un système distribué où les données sont réparties sur plusieurs machines (ou nœuds), une base de données ne peut pas garantir les trois attributs suivants : *Consistency*, *Availability*, et *Partition Tolerence* en même temps.

Consistency (ou intégrité) Chaque donnée a un seul état visible depuis l'extérieur. Par exemple, les différents serveurs hébergeant la base de données voient tous les mêmes données. Ainsi, une lecture faite après une écriture doit renvoyer la donnée précédemment écrite ;

Availability (ou disponibilité) Une base de données doit toujours fournir une réponse à une requête d'un client ;

Partition tolerance (ou la tolérance au partitionnement) Une coupure du réseau entre deux nœuds ou l'indisponibilité d'un de ces nœuds ne devrait pas affecter le bon fonctionnement du système. Tout de même, ce dernier doit répondre à la demande d'un client.

Les trois attributs du théorème CAP s'opposent entre eux. On distingue les trois scénarios possibles :

– Le couple **CA** : les SGBDR adoptent les deux attributs C et A, qui sont une forte cohérence et disponibilité. Cependant, l'attribut partitionnement réseau n'est pas toujours pris en compte.

– Le couple **CP** : les implémentations du C et du P assurent la tolérance aux pannes en distribuant les données sur plusieurs serveurs. Malgré cette réplication, ces implémentations assurent la cohérence des données même en présence de mises à jour concurrentielles.

– Le couple **AP** : les implémentations du A et du P assurent un temps de réponse rapide et une réplication des données. Cependant, les mises à jour étant asynchrones, la garantie que la version d'une donnée soit bonne, ne peut pas être assurée.

La figure III.6 présente des implémentations des différents types de bases de données NoSQL pour chaque couple CA, CP et AP.

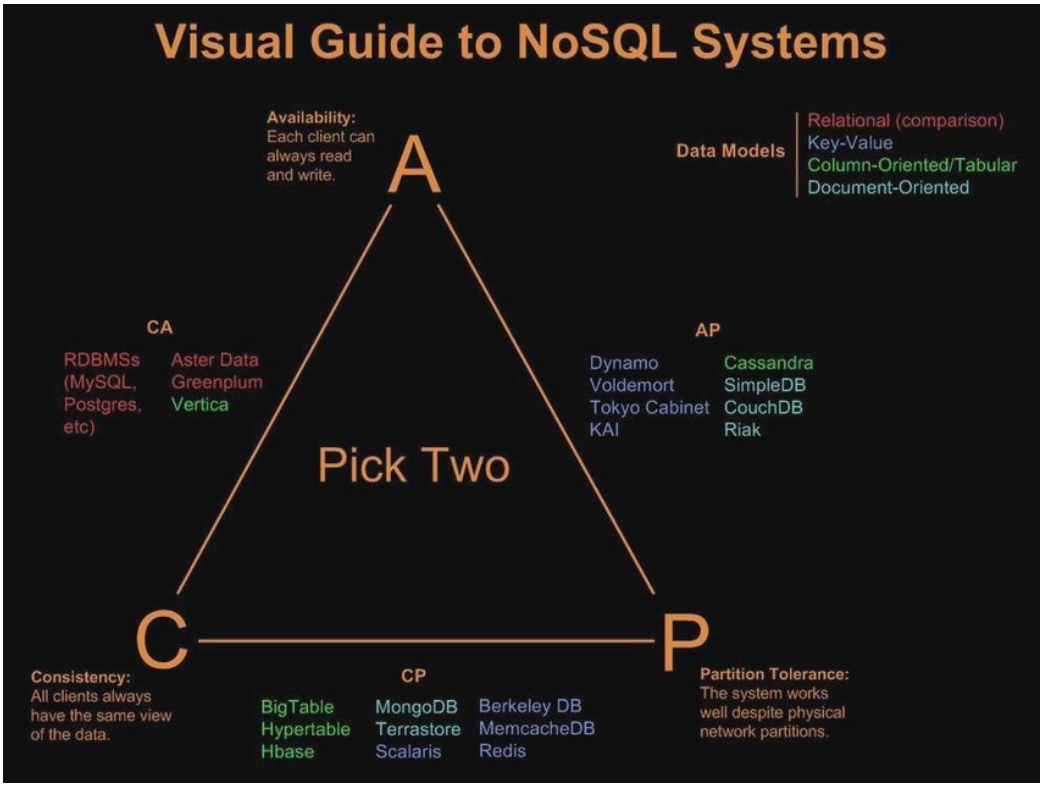


FIGURE III.6 – Bases de données NoSQL suivant le théorème de CAP

Source : `https://payberah.github.io/files/download/p2p/nosql.pdf`, consultée le 05/08/2018.

Le choix d'une base de données relationnelle ou NoSQL dépend des besoins des entreprises. En terme de tendances, la figure III.7 reprend un classement des

SGBDs au 1 août 2018. La suite de la liste ainsi que la méthode qui dirige ce classement sont disponibles sur le site web *DB-Engines Ranking* [3]. Parmi les critères de classement, on trouve le nombre de références du SGBD sur les sites Internet.

343 systems in ranking, August 2018

Rank Aug 2018	Rank Jul 2018	Rank Aug 2017	DBMS	Database Model	Score Aug 2018	Score Jul 2018	Score Aug 2017
1.	1.	1.	Oracle	Relational DBMS	1312.02	+34.24	-55.85
2.	2.	2.	MySQL	Relational DBMS	1206.81	+10.74	-133.49
3.	3.	3.	Microsoft SQL Server	Relational DBMS	1072.65	+19.24	-152.82
4.	4.	4.	PostgreSQL	Relational DBMS	417.50	+11.69	+47.74
5.	5.	5.	MongoDB	Document store	350.98	+0.65	+20.48
6.	6.	6.	DB2	Relational DBMS	181.84	-4.36	-15.62
7.	7.	↑ 9.	Redis	Key-value store	138.58	-1.34	+16.68
8.	8.	↑ 10.	Elasticsearch	Search engine	138.12	+1.90	+20.47
9.	9.	↓ 7.	Microsoft Access	Relational DBMS	129.10	-3.48	+2.07
10.	10.	↓ 8.	Cassandra	Wide column store	119.58	-1.48	-7.14
11.	11.	11.	SQLite	Relational DBMS	113.73	-1.55	+2.88
12.	12.	12.	Teradata	Relational DBMS	77.41	-0.82	-1.83
13.	13.	↑ 16.	Splunk	Search engine	70.49	+1.26	+9.03
14.	14.	↑ 18.	MariaDB	Relational DBMS	68.29	+0.78	+13.60
15.	↑ 16.	↓ 13.	Solr	Search engine	61.90	+0.38	-5.06
16.	↓ 15.	↓ 14.	SAP Adaptive Server	Relational DBMS	60.44	-1.68	-6.48
17.	17.	↓ 15.	HBase	Wide column store	58.80	-1.97	-4.72
18.	18.	↑ 20.	Hive	Relational DBMS	57.94	+0.32	+10.64
19.	19.	↓ 17.	FileMaker	Relational DBMS	56.05	-0.33	-3.60
20.	20.	↓ 19.	SAP HANA	Relational DBMS	51.93	+0.33	+3.96

FIGURE III.7 – Un classement des SGBDs sur *DB-Engines Ranking* du 1 août 2018

Source : `https://db-engines.com/en/ranking`, consultée le 01/08/2018.

III.2.4 Schema on Write VS Schema on Read

Lors du chargement des données depuis leurs sources de stockage pour tout type de manipulation, on distingue deux approches : *Schema on Write* et *Schema on Read*.

Dans la première, il faut définir les colonnes, le format de données, les types, etc. La lecture des données est rapide et moins coûteuse étant donné l'effort entreprit pour définir la structure. C'est le cas des bases de données relationnelles.

Dans la deuxième, les données sont chargées telles qu'elles sont, sans transformations ou changements. L'interprétation de ces données se fait lors de la lec-

3. Source : `https://db-engines.com/`, consultée le 01/08/2018.

ture, et cela dépend des besoins pour lesquels les données sont analysées. Ainsi, les mêmes données peuvent être lues de différentes manières. Par exemple, l'action de lire les données d'une colonne, qu'elles soient de type entier ou bien chaîne de caractère d'un fichier CSV est la même, mais le type de la donnée qui diffère.

Les figures III.8 et III.9 illustrent la différence entre ces deux approches.

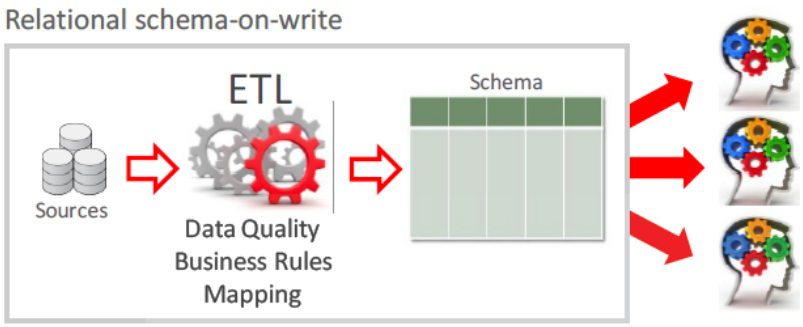


FIGURE III.8 – Schema on Write (Traditionnelle)

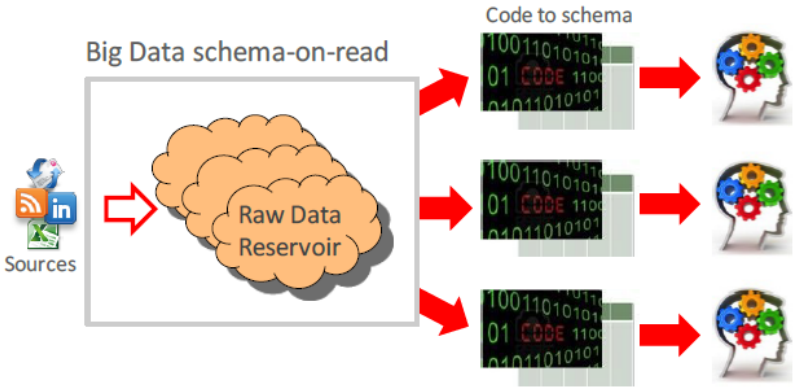


FIGURE III.9 – Schema on Read (Big Data)

Source : https://blogs.oracle.com/datawarehousing/big-data-sql-quick-start-schema-on-read-and-schema-on-write-part11, consultée le 05/08/2018.

La meilleure approche dépend des besoins de l'analyse. La première approche est meilleure en performances, cependant, la deuxième est tolérante aux erreurs humaines.

III.2.5 L'informatique distribuée et l'analyse de données massives

Il existe deux stratégies pour appliquer des traitements sur un grand ensemble de données :

- Par distribution des traitements (*scaling* des traitements) : les traitements sont distribués sur un nombre de nœuds important. De ce fait, les données sont amenées jusqu'à ces nœuds ;
- Par distribution des données (*scaling* des données) : les données sont distribuées sur un nombre important de nœuds. De plus, cela permet de stocker un maximum de données. Il s'agit d'amener les traitements aux machines sur lesquelles les données sont stockées. Du fait que le stockage de données est réparti sur plusieurs machines, il est possible de traiter des données très volumineuses. La première mise en œuvre de cette approche est le schéma Map-Reduce.

III.3 Parcours de quelques technologies du Big Data

Chapitre IV

Implementation

IV.1 MongoDB

IV.2 DynamoDB

IV.3 AWS

IV.4 Spark Apache avec Scala

IV.4.1 Complément d'information du processus de la détection avec le langage Scala

Comme complément aux étapes décrites dans II.5.2, on présente les différentes classes permettant de modéliser les données tout au long du processus de l'analyse. La définition de ces classes est liée au langage *Scala*.

Soient les classes suivantes utilisées :

La classe Signal modélise un signal[1]. Ainsi, *from* est l'adresse IP du routeur émettant ce signal, *rtt* est le Round Trip Time entre la sonde Atlas et ce routeur et enfin *x* est un indicateur de l'échec du signal.

```
case class Signal(
rtt :  Option[Double],
x :    Option[String],
from : Option[String])
```

1. Un signal dans le contexte d'un traceroute.

La classe Hop modélise un saut dans un traceroute. On caractérise un saut par son identifiant noté *hop*. Celui-ci prend comme valeur un entier commençant à 1 et la liste des signaux relatifs à ce saut notée par *result*. Généralement un saut est représenté par 3 signaux.

```
case class Hop(
var result: Seq[Signal],
hop:        Int)
```

La classe Traceroutes modélise le résultat d'une requête traceroute effectuée par une sonde Atlas. Cette modélisation se limite aux données qui nous intéressent dans la présente analyse.

dst_name représente l'adresse IP de la destination de la requête traceroute, *from* est l'adresse IP de la sonde, *prb_id* est l'identifiant de la sonde, *msm_id* est l'identifiant de mesure, *timestamp* est le temps auquel la requête traceroute a été effectuée et enfin on trouve la liste des sauts qui représentent les routeurs traversés par le trafic entre la source et la destination.

```
case class Traceroute(
dst_name:  String,
from:      String,
prb_id:    BigInt,
msm_id:    BigInt,
timestamp: BigInt,
result:    Seq[Hop])
```

La classe TraceroutesPerPeriod permet de présenter les traceroutes après les avoir trié suivant la période pendant laquelle ils ont été effectués. *timeWindow* est le temps unix marquant le début de la période [2] et *traceroutes* est la liste des traceroutes effectués pendant cette période.

A l'étape 2, l'objectif était d'agréger les signaux par routeur source et ensuite calculer la médiane des RTTs par ce routeur. Par conséquent, un traceroute est présenté différemment, ce qui est illustré par la classe *MedianByHopTraceroute*.

La classe PreparedSignal est une agrégation de tous les signaux, d'un saut donné, par le routeur *from*, la médiane des RTTs calculée est présentée par *medianRtt*.

2. Pour précision, la fin de la période peut être inférée en prenant deux débuts de deux périodes car la durée d'une période est fixe tout au long de l'analyse.

```
case class PreparedSignal(
medianRtt: Double,
from:      String)
```

La classe PreparedHop modélise un saut après avoir agrégé ses signaux.

```
case class PreparedHop(
var result: Seq[PreparedSignal],
hop:        Int)
```

La classe MedianByHopTraceroute modélise un traceroute après avoir agrégé ses sauts. Par rapport au traceroute d'avant l'agrégation, seule la liste des sauts a subi un changement.

```
case class MedianByHopTraceroute(
dst_name:  String,
from:      String,
prb_id:    BigInt,
msm_id:    BigInt,
timestamp: BigInt,
result:    Seq[PreparedHop])
```

La classe Link modélise un lien topologique. Ce dernier est défini par deux adresses IP *ip1* et *ip2* et par son RTT différentiel calculé *rttDiff*.

```
case class Link(
ip1:     String,
ip2:     String,
rttDiff: Double)
```

La classe LinksTraceroute permet de modéliser un traceroute après avoir inféré tous les liens de ce dernier. Ainsi, la liste des sauts est remplacée par la liste des liens (*links*).

```
case class LinksTraceroute(
dst_name:  String,
from:      String,
prb_id:    BigInt,
msm_id:    BigInt,
timestamp: BigInt,
links:     Seq[Link])
```

A l'étape 5, l'objectif était de passer d'un traceroute à une liste de liens caractérisés par les informations générales sur la sonde Atlas, la mesure Atlas, etc. Chaque élément de cette liste est représenté par la classe *DiffRtt*, où *LinkIPs* représente les deux adresses IP d'un lien donné.

La classe LinkIPs permet représenter un lien par seulement ses deux adresses IP *ip1* et *ip2*.

```
case class LinkIPs(
ip1: String,
ip2: String)
```

La classe DiffRtt est une représentation plus détaillée d'un lien, en plus de son RTT différentiel, on ajoute d'autres informations. Les adresses IP d'un lien sont modélisées par la classe *LinkIPs*.

```
case class DiffRtt(
rtt:       Double,
var link: LinkIPs,
probe:     BigInt)
```

A l'étape 6.3, on souhaite normaliser les dates de chaque lien ; peu importe le moment pendant lequel le traceroute a été effectué durant une période d_i, on note seulement le début de cette période. Ainsi, la classe *DiffRTTPeriod* reprend un *lien* donné, les différentes sondes Atlas ayant identifié ce lien (*probes*), les RTTs différentiels de ce lien tout au long de la période et enfin les dates associées à chaque RTT différentiel.

La classe DiffRTTPeriod

```
case class DiffRTTPeriod(
link:       LinkIPs,
probes:     Seq[BigInt],
rtts:       Seq[Double],
var dates: Seq[Int])
```

A la fin des opérations de l'étape 6, on reprend pour chaque période, pour un lien donné, les RTTs différentiels ainsi que leurs dates. Ensuite, on construit les bornes de l'intervalle de confiance courants pour ce lien et les bornes de l'intervalle de confiance de référence, et ce afin de comparer ces deux intervalles en vue d'inférer les anomalies possibles du délais de ce lien.

La classe LinkState permet de modéliser les intervalles de confiance d'un lien pendant une période d_i donnée. *valueLow* est la borne inférieur de l'intervalle de confiance, *valueHi* est la borne supérieure de l'intervalle de confiance, *valueMedian* est la médiane des RTTs différentiels et enfin *valueMean* est la moyenne des RTTs différentiels. Pour précision, les données concernant l'état d'un lien sont sous forme d'une liste. L'idée est de garder l'historique de ces valeurs durant toute la période de l'analyse. Cette historique est exploitée pour tracer l'évolution du RTT différentiel du lien. Cependant, la comparaison utilise les valeurs du dernier état du lien.

```
case class LinkState(
var valueMedian: Seq[Double],
var valueHi:     Seq[Double],
var valueLow:    Seq[Double],
var valueMean:   Seq[Double])
```

La comparaison des intervalles de confiance La comparaison de l'état courant du lien avec celui de référence est effectuée en analysant le chevauchement d'intervalles de confiance courant et de référence. Le délai d'un lien est jugé anormal si son intervalle de confiance courant est inclus dans l'intervalle de confiance de référence. C'est le cas 1 dans la Figure IV.1, *referenceLow* et *referenceHight* sont les bornes de l'intervalle de confiance de référence. *currentLow* et *currentHight* sont les bornes de l'intervalle de confiance courant.

D'après le travail de référence, on distingue quatre cas possibles illustrés dans la Figure IV.1 :

Cas 1 : le délai du lien est normal.

Cas 2 : le délai du lien est anormal.

Cas 3 : le délai du lien est anormal.

Cas 4 : le délai du lien est anormal.

Dans le cas où le délai est jugé anormal, on introduit ce qu'on appelle la *déviation*. Cette métrique caractérise l'anomalie détectée. Elle est calculée différemment dans le cas où le délai est anormal.

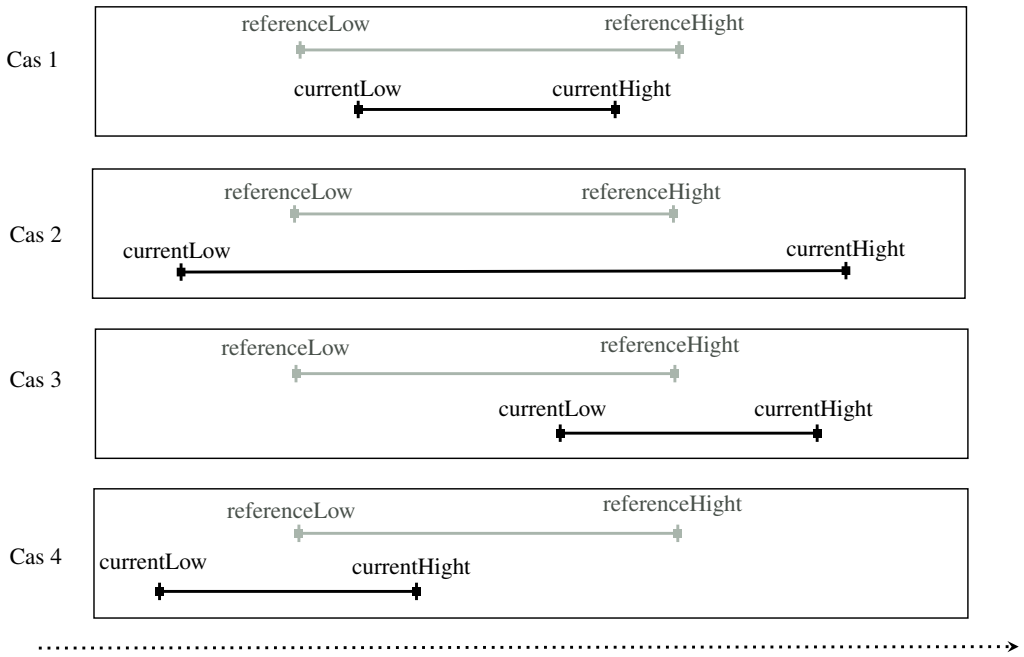


FIGURE IV.1 – La comparaison des deux intervalles de confiance : courant et référence

Bibliographie

[1] Archipelago (Ark) Measurement Infrastructure. URL : http://www.caida.org/projects/ark/. (consulté le 19/01/2018).

[2] Center for Applied Internet Data Analysis (CAIDA). URL : http://www.caida.org/. (consulté le 06/04/2018).

[3] Create a New Measurement - RIPE Atlas. URL : https://atlas.ripe.net/measurements/form/. consulté le 05/08/2018.

[4] Le dépôt des données RIPE Atlas. URL : https://data-store.ripe.net/datasets/atlas-daily-dumps/. (consulté le 26/07/2018).

[5] Les archives des détails des sondes atlas. URL : https://ftp.ripe.net/ripe/atlas/probes/archive/. (consulté le 28/01/2018).

[6] RIPE Atlas - Known Bugs and Limitations. URL : https://atlas.ripe.net/docs/bugs/. (consulté le 05/04/2018).

[7] Samknows. URL : https://www.samknows.com/global-platform. (consulté le 23/01/2018).

[8] Xport pro. URL : https://www.lantronix.com/products/xport-pro/. (consulté le 08/08/2018/).

[9] ABEN, E. How RIPE Atlas Helped Wikipedia Users. URL : https://labs.ripe.net/Members/emileaben/how-ripe-atlas-helped-wikipedia-users, 2014. (consulté le 18/08/2018).

[10] ABEN, E. Looking at France-IX with RIPE Atlas and RIS. URL : https://labs.ripe.net/Members/emileaben/looking-at-france-ix-with-ripe-atlas-and-ris, 2015. (consulté le 08/08/2018).

[11] ABEN, E. Measuring Countries and IXPs with RIPE Atlas. URL : https://labs.ripe.net/Members/emileaben/measuring-ixps-with-ripe-atlas, 2015. (consulté le 08/08/2018).

[12] ANDERSON, C., WINTER, P., AND ROYA. Global network interference detection over the RIPE atlas network. In *4th USENIX Workshop on Free and Open Communications on the Internet (FOCI 14)* (San Diego, CA, 2014), USENIX Association.

[13] DONATO, V. D. Traceroute Consistency Check. URL : https://github.com/vdidonato/Traceroute-consistency-check, 2015. (Consulté le 08/08/2018).

[14] FONTUGNE, R., ABEN, E., PELSSER, C., AND BUSH, R. Pinpointing delay and forwarding anomalies using large-scale traceroute measurements. *CoRR abs/1605.04784* (2016).

[15] GASMI, S. Visualising RIPE Atlas Anchor Measurements. URL : https://labs.ripe.net/Members/salim_gasmi/visualising-ripe-atlas-anchor-measurements, 2015. (consulté le 08/08/2018).

[16] GUILLAUME VALADON, FRANCOIS CONTAT, M. H., AND HOLTERBACH, T. BGP Atlas Monito (BAM). URL : https://github.com/guedou/bam, 2015. (consulté le 08/08/2018).

[17] HEROLD, J. Bgp + traceroute presentation. URL : https://labs.ripe.net/Members/becha/ripe-atlas-hackathon-presentations/bgp-traceroute, 2015. (consulté le 08/08/2018).

[18] HEROLD, J. BGP + Traceroute using RIPE NCC Atlas. URL : https://github.com/wires/bgp-traceroutes, 2015. consulté le 08/08/2018.

[19] HOLTERBACH, T., PELSSER, C., BUSH, R., AND VANBEVER, L. Quantifying interference between measurements on the ripe atlas platform. In *Proceedings of the 2015 Internet Measurement Conference* (New York, NY, USA, 2015), IMC '15, ACM, pp. 437–443.

[20] KISTELEKI, R. The AMS-IX Outage as Seen with RIPE Atlas. URL : https://labs.ripe.net/Members/kistel/the-ams-ix-outage-as-seen-with-ripe-atlas, 2015. (consulté le 23/01/2018).

[21] KISTELEKI, R. RIPE Atlas Architecture - how we manage our probes. URL : https://labs.ripe.net/Members/kistel/ripe-atlas-architecture-how-we-manage-our-probes, 2017. consulté le (08/08/2018).

[22] KISTELEKI, R. RIPE Atlas probes as IoT devices. URL : https://labs.ripe.net/Members/kistel/ripe-atlas-probes-as-iot-devices, 2017. (consulté le 21/12/2017).

[23] MAHESHWARI, A. *Big Data*.

[24] RIPE NCC. Test Traffic Measurement Service (TTM). URL : https://www.ripe.net/analyse/archived-projects/ttm. (consulté le 14/01/2018).

[25] RODERICK, F. On the Diversity of Interdomain Routing in Africa. URL : https://labs.ripe.net/Members/fanou_roderick/on-the-diversity-of-interdomain-routing-in-africa, 2015. (consulté le 11/01/2018).

[26] SAGIROGLU, S., AND SINANC, D. Big data : A review. In *2013 International Conference on Collaboration Technologies and Systems (CTS)* (May 2013), pp. 42–47.

[27] SHAO, W., ROUGIER, J., DEVIENNE, F., AND VISTE, M. Missing measurements on RIPE atlas. *CoRR abs/1701.00938* (2017).

[28] SHAVITT, Y., AND SHIR, E. Dimes : Let the internet measure itself. *SIGCOMM Comput. Commun. Rev. 35*, 5 (Oct. 2005), 71–74.

[29] SPEED CHECKER. Global Internet testing - ProbeAPI. URL : http://probeapi.speedchecker.xyz/. (consulté le 06/08/2018).

[30] VARAS, C. A Practical Comparison Between RIPE Atlas and ProbeAPI. URL : https://labs.ripe.net/Members/cristian_varas/a-practical-comparison-between-ripe-atlas-and-probeapi, 2016. (consulté le 19/01/2018).