

TABLE DES MATIÈRES

	Page
CHAPITRE 1 INTRODUCTION	1
1.1 Problématique	3
1.2 Objectifs du mémoire	6
1.3 Plan du mémoire	8
CHAPITRE 2 REVUE DE LITTÉRATURES DE LA QUALITÉ DE SERVICE DANS LES RÉSEAUX DE NOUVELLE GÉNÉRATION (NGN)	11
2.1 Introduction	11
2.2 Contexte des réseaux de nouvelle génération	11
2.2.1 Définition et concept	11
2.2.2 Caractéristiques	11
2.3 Qualité de service dans les réseaux IP	12
2.3.1 Définitions de la QoS	12
2.3.2 Type de la qualité de service	13
2.3.3 Les principales métriques de la QoS dans les réseaux IP	14
2.3.4 Exigence de la QoS pour les applications temps réel	14
2.3.5 Exigence de la QoS pour les applications non-temps réel	15
2.3.6 Modèles de gestion de la QoS	16
2.3.6.1 Le modèle Best-effort	16
2.3.6.2 Le modèle InterServ	16
2.3.6.3 Le modèle DiffServ	18
2.4 Qualité de service dans IMS	20
2.4.1 Présentation d'IMS	20
2.4.2 Architecture d'IMS	21
2.4.3 Gestion des identités en IMS	24
2.4.4 Concepts et approches de gestion de la QoS dans IMS	25
2.4.5 Architecture de PCC(Policy and Charging Control)	26
2.4.6 Négociation de la QoS dans IMS	29
2.4.7 Mappage de la QoS dans IMS	31
2.5 Conclusion	32
CHAPITRE 3 ANALYSE DES EXIGENCES D'IMS DANS LE NUAGE	35
3.1 Introduction	35
3.2 Nuage informatique	35
3.2.1 Définition et concept	35
3.2.2 Types de nuages	35
3.2.3 Modèle de service	37
3.2.4 Caractéristique du nuage informatique	38
3.3 Exigences du système IMS déployé dans le nuage	39

3.4	Virtualisation du système IMS	40
3.4.1	Définition	40
3.4.2	Technique de virtualisation	42
3.4.3	Avantage de la virtualisation du système IMS	43
3.4.4	Virtualisation d'IMS	43
3.5	Mise à l'échelle automatique	44
3.5.1	Mise à l'échelle verticale	44
3.5.2	Mise à l'échelle horizontale	45
3.6	Élasticité	46
3.7	Provisionnement des ressource	47
3.7.1	Provisionnement statique	47
3.7.2	Provisionnement dynamique	48
3.7.3	Auto-provisionnement par l'utilisateur	48
3.8	Design et architecture d'IMS dans le nuage	49
3.9	Conclusion	50
CHAPITRE 4 PROVISIONNEMENT DE LA QOS D'IMS DANS LE NUAGE		51
4.1	Introduction	51
4.2	Les systèmes IMS existants	51
4.2.1	Open Source IMS core	52
4.2.2	Clearwater IMS core	54
4.2.3	Infrastructure en tant que service libre : Openstack	57
4.3	Métriques de performance d'IMS	60
4.3.1	Demande d'enregistrement	61
4.3.2	Demande d'établissement d'une session audio	62
4.3.3	Demande d'établissement d'une session donnée	64
4.4	Provisionnement de la QoS d'IMS dans le nuage informatique	65
4.4.1	Provisionnement des ressources dans le nuage informatique	67
4.4.2	Modèle de provisionnement de la QoS proposé	67
4.5	Conclusion	73
CHAPITRE 5 TEST ET VALIDATION		75
5.1	Introduction	75
5.2	Architecture et conception des bancs d'essai	75
5.2.1	Banc d'essai 1 : Cloud open source IMS core	76
5.2.2	Banc d'essai 2 : Cloud Clearwater IMS	78
5.2.3	Banc d'essai 3 : Traditional open source IMS Core	79
5.3	Test de fonctionnement des architectures d'IMS dans le nuage	81
5.3.1	Tableau de bord d'Openstack	81
5.3.2	Demande d'enregistrement	82
5.3.3	Établissement d'une session audio	84
5.3.4	Établissement d'une session donnée	86
5.3.5	Gestion du profil d'utilisateur	86
5.3.6	Test de fonctionnement du service IMS	87

5.4	Protocole expérimental et résultats	88
5.4.1	Scénario d'évaluation des métriques de la QoS	88
5.4.2	Processus de surveillance du trafic dans IMS	91
5.5	Résultats	92
5.5.1	Analyse de délai d'une demande d'enregistrement (RRD)	92
5.5.2	Analyse du délai d'une demande d'établissement d'une session vidéo (SRD)	94
5.5.3	Analyse du délai d'une demande d'établissement d'une session donnée	95
5.5.4	Analyse de flux RTP d'une session audio	96
5.5.5	Analyse de l'utilisation de CPU et mémoire	99
5.5.6	Analyse de l'utilisation de CPU et mémoire virtuelle	101
5.5.7	Comparaison des systèmes IMS déployés dans le nuage informatique	103
5.6	Conclusion	104
CHAPITRE 6	CONCLUSION	105
ANNEXE I	INSTALLATION D'OPEN SOURCE IMS CORE	109
ANNEXE II	INSTALLATION DE CLEARWATER IMS	115
ANNEXE III	CONFIGURATION DE SIPP TRAFFIC GENERATOR	119
BIBLIOGRAPHIE	129

LISTE DES TABLEAUX

		Page
Tableau 2.1	G.1010 – Objectifs de performance pour les applications audio et vidéo	15
Tableau 2.2	G.1010 – Objectifs de performance pour les applications de données	15
Tableau 2.3	Les entités de PCC	28
Tableau 2.4	Identificateur de classe de la QoS dans LTE	29
Tableau 2.5	Mappage de la classe de qualité de service sur le réseaux hétérogène	31
Tableau 5.1	Composants des machines virtuelles OpenIMS.....	76
Tableau 5.2	Caractéristiques des machines OpenIMS	77
Tableau 5.3	Composants des machines virtuelles Clearwater IMS.....	78
Tableau 5.4	Caractéristiques des Machines Clearwater IMS	78
Tableau 5.5	Composants des hosts	79
Tableau 5.6	Caractéristiques des machines	80

LISTE DES FIGURES

	Page
Figure 1.1	Diagramme des différents éléments présentés dans cette mémoire 9
Figure 2.1	Échange des messages RSVP Tirée de rfc2205 (1997) 17
Figure 2.2	Domaine de Diffserv Tirée de Cisco (2013) 19
Figure 2.3	Architecture d'IMS Tirée de El Barachi <i>et al.</i> (2011)..... 22
Figure 2.4	Relation entre l'identité privée et public dans IMS Tirée de MROUEH Lina (2006) 25
Figure 2.5	Architecture de PCC Tirée de 3GPP (2010) 27
Figure 2.6	Signalisation et négociation de la QoS dans IMS (SIP/SDP) Tirée de Skorin-Kapov <i>et al.</i> (2007)..... 30
Figure 2.7	Emplacement des SOMA agent Tirée de Ryu <i>et al.</i> (2006)..... 33
Figure 2.8	Fonctionnement de SOMA Tirée de Ryu <i>et al.</i> (2006)..... 33
Figure 3.1	Modèle de déploiement Tirée de WMWare(2011) 36
Figure 3.2	Architecture de nuage informatique Tirée de Zhang <i>et al.</i> (2010)..... 37
Figure 3.3	Hyperviseur type 1 Tirée de Dominique Revuz (2008) 41
Figure 3.4	Hyperviseur type 2 Tirée de Dominique Revuz (2008) 42
Figure 3.5	Mise à l'échelle et élasticité Tirée de Varia (2012) 45
Figure 3.6	Architecture proposé d'IMS dans le nuage 49
Figure 4.1	Architecture d'Open Source IMS Core Tirée de FOKUS (2006) 53
Figure 4.2	FOKUS Home Subscriber Server Tirée de FOKUS (2009) 54
Figure 4.3	Architecture de Clearwater IMS Tirée de Metaswitch (2013) 55
Figure 4.4	Architecture simplifiée d'OpenStack Tirée de Rackspace (2013) 58
Figure 4.5	Procédure d'enregistrement 61
Figure 4.6	Établissement d'une session audio 62

Figure 4.7	Établissement d'une session donnée	65
Figure 4.8	Composante du modèle proposé	68
Figure 4.9	Modèle de provisionnement de la QoS d'IMS dans le nuage	70
Figure 4.10	Provisionnement des ressources d'IMS	72
Figure 4.11	Provisionnement des ressources du VM	72
Figure 5.1	Cloud Open IMS	77
Figure 5.2	Cloud Clearwater IMS	79
Figure 5.3	Traditional Open IMS architecture.....	80
Figure 5.4	Vue d'ensemble Openstack	81
Figure 5.5	Résumé d'hyperviseur	82
Figure 5.6	Procédure d'enregistrement Cloud OpenIMS	83
Figure 5.7	Procédure d'enregistrement Cloud Clearwater	83
Figure 5.8	Demande d'enregistrement : Méthode "REGISTER"	84
Figure 5.9	Demande d'enregistrement : Méthode "401 Unauthorized"	84
Figure 5.10	Demande d'enregistrement : Code de réponse "200 OK"	85
Figure 5.11	Établissement d'une session audio/vidéo : Méthode "INVITE"	85
Figure 5.12	Établissement d'une session audio : Méthode "INVITE"	86
Figure 5.13	Établissement d'une session donnée : Méthode "MESSAGE"	87
Figure 5.14	Interface de FoHSS	88
Figure 5.15	Interface de Ellis	89
Figure 5.16	Fenêtre du client IMS(Alice/Bob)	90
Figure 5.17	Communication vidéo entre deux clients (Alice/Bob)	91
Figure 5.18	Processus de surveillance du trafic	92
Figure 5.19	Variation du délai d'enregistrement (RRD) sans stress SIP.....	93

Figure 5.20	Variation de délai d'enregistrement (RRD) avec stress SIP	93
Figure 5.21	Variation du délai d'établissement d'une session audio (SRD) sans stress SIP	94
Figure 5.22	Variation de délai d'établissement d'une session audio (SRD) avec stress SIP	95
Figure 5.23	Variation du délai d'établissement d'une session donnée (IMD) sans stress SIP	96
Figure 5.24	Variation du délai d'établissement d'une session donnée (IMD) avec stress SIP	96
Figure 5.25	Analyse de flux RTP par wireshark	97
Figure 5.26	Variation de délai pour le flux audio	98
Figure 5.27	Variation de la Gigue pour le flux audio	98
Figure 5.28	Variation de la bande passante pour le flux audio	99
Figure 5.29	Utilisation de CPU	100
Figure 5.30	Utilisation de la mémoire physique	100
Figure 5.31	Utilisation de VCPU : OpenIMS	102
Figure 5.32	Utilisation de mémoire virtuelle : OpenIMS	102
Figure 5.33	Utilisation du VCPU : Clearwater IMS	103
Figure 5.34	Utilisation de la mémoire virtuelle : Clearwater IMS	103

LISTE DES SYMBOLES ET UNITÉS DE MESURE

AaaS	Application as a Service
AF	Assured forwarding
AS	Application Server
ASM	Application Service Map
BBERF	Bearer Binding and Event Reporting Function
BE	Best Effort
BG	Border Gateway
BGCF	Breakout Gateway Control Function
CL	Controlled-Load
DiffSer	Differentiate Service
DSCP	DiffServ Code Point
EF	Expedited Forwarding
ETSI	European Telecommunications Standards Institute
GBR	Guaranteed Bit Rate
GGSN	Gateway GPRS Support Node
GS	Guaranteed Service
IaaS	Infrastructure as a Service
IBCF	Interconnection Border Control Function
ICT	Information and Communication Technology
IPDV	IP Packet Delay Variation

IPER	IP Packet Error Ratio
IMS	IP Multimedia Subsystem
InterServ	Intergrated Service
IPTD	IP Packet Transfer Delay
IPLR	IP Packet Loss Ratio
ISO	International Organization for Standardization
ITU	International Telecommunication Union
KPI	Key Performance Indicator
LTE	Long Term Evoluation
MOS	Mean Opinion Score
MS	Media Server
NFV	Network Functions Virtualization
NGN	Next generation Network
NIST	National Institute for Standards and Technology
NON-GBR	Non-Guaranteed Bit Rate
OCS	Online Charging System
OFCS	Offline Charging System
OSA	Open architecture Application
PaaS	Platform as a Service
P-CSCF	Proxy Call Session Control Function
PCEF	Policy and Charging Enforcement Function

PCC	Policy and Charging Control
PCRF	Policy and Charging Rules Function
PDF	Policy Decision Function
PEC	Pre-emption capability
PEV	Pre-emption vulnerability
PHB	Per Hop Behaviour
PoC	Push to talk Over Cellular
QOE	Quality of Exeperience
QCI	QoS Class Identifier
QoS	Quality Of Service
RRD	Registration Request Delay
RSVP	Ressource reServation Protocol
RTCP	Real Time Control Protocol
RTSP	Real Time Streaming Protocol
RTP	Real Time Protocol
SaaS	Service Level Agreement
S-CSCF	Serving Call Session Control Function
SDP	Session Description Protocol
SIP	Service Inition Protocol
SLF	Subscription Locator Function
SPR	Subscription Profile Repository

SRD	Session Request Delay
TC	Traffic Class
TOP	Type of service
TrGw	Transition Gateway
UDP	User Datagram Protocol
VM	Virtual Machine
VoIP	Voice over IP
VoLTE	Voice over LTE
EBS	Ericsson Blade Server
BG	Border Gateway

CHAPITRE 1

INTRODUCTION

L'évolution de la technologie de l'information et de la communication vers des architectures basées essentiellement sur le protocole IP et qui sont capables d'unifier les réseaux fixes et mobiles pour donner naissance aux réseaux de nouvelle génération (NGN).

Par ailleurs, l'union internationale des télécommunications (ITU) a prévu que dans certain nombre de pays développés, tous les réseaux de lignes fixes seront des réseaux de nouvelle génération dès 2012 et qu'il en ira de même pour les réseaux mobiles à l'horizon 2020 (ITU, 2009).

Notamment, les opérateurs de télécommunications cherchent à trouver une architecture standard indépendamment de leurs technologies du réseau utilisées, qui optimisent leurs coûts et offrent divers services multimédias sur une même connexion (p.ex téléphonie, accès Internet et divertissement) tout en assurant la convergence fixe/mobile avec la susceptibilité de passage à l'échelle.

Également, cette évolution répond bien aux besoins des opérateurs de télécommunications vu que les réseaux de nouvelle génération offrent une connectivité mondiale transparente indépendamment du type de réseau, du dispositif utilisé, à tout moment et en tout lieu.

De nouveaux acteurs comme les fournisseurs de services Internet à haut débit et de TV, ainsi que des opérateurs virtuels ont commencé à investir dans ce domaine qui a été auparavant monopolisé par les entreprises de télécommunications et des radiodiffuseurs.

Mais encore, le passage d'une architecture traditionnelle basée essentiellement sur la commutation de circuits vers une nouvelle architecture basée sur IP constitue un changement important dans le secteur de technologies d'information et de communication (TIC).

Ainsi, les travaux de normalisation de la troisième génération du projet de partenariat (3GPP) ont mené à l'architecture standardisée IP Multimedia Subsystem (IMS). Ce dernier a été défini

par le 3G.IP, initialement, destiné pour les réseaux mobiles et utilise le General Packet Radio Service (GPRS) comme réseau d'accès. Grâce à l'évolution rapide d'IMS et la migration vers le tout IP avec l'interopérabilité entre les réseaux fixes et mobiles, ce système converge aussi vers les réseaux fixes.

Le but principal du système IMS est d'offrir aux opérateurs une architecture de service multi-média standard. Il est basé sur le protocole IP pour le transport de données et le protocole SIP (Session Initiation Protocol) pour la signalisation et le contrôle de session (Rosenberg *et al.*, 2002).

Aujourd'hui, IMS est utilisé comme infrastructure de services par les opérateurs téléphoniques à travers le monde qui déploient des architectures 3G ou 4G. Une croissance globale dans l'accès aux différents services via les téléphones intelligents (Smartphones) au réseau d'opérateurs est passé de 1.9 milliard en 2013 à 2.3 milliards d'abonnées en 2014. Ce nombre pourrait atteindre 7.2 milliards à la fin de 2019 (Ericsson, 2014).

À l'égard de cette évolution, de nouveaux concepts sont apparus dont le nuage informatique est le plus prometteur. Ce dernier constitue une transformation majeure dans le secteur de technologies de l'information et de la communication (TIC). Les avantages multiples offerts par le nuage informatique mènent les entreprises à embrasser ce dernier.

Les statistiques réalisées par plusieurs firmes de sondage montrent que le pourcentage des utilisateurs, fournisseurs de services ainsi que l'investissement dans le nuage informatique ne cessent de croître au fil du temps.

D'ailleurs, une étude menée par la firme Brocade (Brocade, 2012) montre que 60 % des grandes entreprises envisagent une migration vers le nuage informatique depuis 2012. Ainsi, qu'entre 2009 et 2013 le marché de nuage informatique a connu une croissance continue dans les trois modèles de bases de l'infonuage : Software as a Service (SaaS), Platform as a Service (PaaS) et Infrastructure as a Service (IaaS) qui ont passé de 68 milliards de dollars en 2010 à 150 en 2014 dont les dépenses représente 10 % des investissements informatiques en 2013.

La compagnie Gartner (Gartner, 2013) indique qu'une forte croissance continue dans les services de nuage informatique public avec des dépenses de l'utilisateur final sur ces services de 18 % en 2013 pour un chiffre global de 131 milliards de dollars et prévoit que ce même marché pourrait être plus de 180 milliards de dollars en 2015.

Cette augmentation progressive du nombre d'utilisateurs de nuage informatique provoque une croissance du marché mondial des équipements de nuage pour répondre aux besoins de tous les utilisateurs. Cette croissance atteindra 79,1 milliards de dollars en 2018 tels que mentionnés par la compagnie SiliconANGLE (Siliconangle, 2014) qui a indiqué aussi que plus de 82% des entreprises affirment leurs satisfactions et une économisation de l'argent en se déplaçant vers le nuage.

Étant donné, que la réduction du coût représente la première priorité des entreprises, les fournisseurs de nuage informatique promettent à ces derniers à atteindre cet objectif.

En plus, l'absence de valeur ajoutée dans le secteur de télécommunication mène les opérateurs à chercher d'autres solutions pour augmenter leur revenu. Le nuage informatique représente bien la solution la plus prometteuse aux opérateurs pour apporter de nouveaux services à leurs clients et de les délivrer avec une nouvelle manière à la fois efficace et à faible coût.

1.1 Problématique

Le nuage informatique est en train de changer les façons de fournir et de créer des services à travers le protocole IP. Il représente une transformation majeure dans le domaine de technologies de l'information et de la communication et possède plusieurs avantages qui attirent d'autres acteurs du secteur de télécommunications comme les opérateurs.

Les fournisseurs de nuage offrent aux opérateurs et aux prestataires de services une plus grande flexibilité et facilité dans la livraison des services à un prix compétitif. Le manque de valeurs ajoutées et la diminution du revenu chez les Telcos ont poussé ces derniers à prendre l'initiative d'investir dans le nuage étant donné qu'ils offrent aux utilisateurs l'accès à un large éventail de services avec le moindre coût et dans un temps réduit.

Bien que le choix de l'environnement du nuage semble une bonne opportunité pour les Telcos pour augmenter leurs revenus et pour apporter des valeurs ajoutées à leurs services, la migration de l'infrastructure de service IMS de l'opérateur vers ce dernier est un grand défi, notamment à ce qui a tant aux fait suivants :

Complexité de déploiement des applications et des services : souvent, le déploiement des services et des applications dans le réseau d'opérateurs est un processus complexe et une tâche ardue. En général, l'architecture de l'opérateur est très complexe et l'hétérogénéité des équipements et des périphériques du réseau opérateurs sont parmi ses défis. L'interopérabilité avec d'autres réseaux téléphoniques qui utilisent de diverses technologies complique d'avantage la tâche de déploiement. En effet, l'opérateur doit prendre en compte ces facteurs avant d'intégrer un nouveau service dans son écosystème. Cette situation, suggère de répondre à des problèmes tels que : comment faciliter le déploiement des applications ? Comment réduire la complexité et le temps de déploiement des applications et des services Telco ?

Coûts d'exploitation et de maintenance élevés CAPEX et OPEX : déployer de nouvelles applications et des nouveaux services dans les réseaux d'opérateurs nécessite des dépenses supplémentaires à la fois Operational Expenditure (OPEX) et Capital Expenditure (CAPEX). En effet, l'achat des ressources matérielles ainsi que leurs installations implique des coûts additionnels. Le choix de la stratégie d'installation réseau et la sélection des composants appropriés prennent beaucoup de temps. Des coûts de maintenance additionnelle sont requis dans le cas de la défaillance d'un équipement réseau. Cette situation, suggère à répondre aux problèmes tels que : comment optimiser les coûts d'exploitation ? Comment réduire le temps de déploiement de nouvelles applications et services ? Comment optimiser l'utilisation des matériels et favoriser la composante la plus critique ? Comment améliorer le processus de maintenance ?

Maintenir le niveau de la qualité de service convenu dans le Service-level Agreement (SLA) : le système IMS est le composant le plus important dans l'architecture du NGN, utilisé pour fournir et créer des services. IMS est basé sur IP dont son principal défaut réside dans l'incapacité d'offrir une QoS de bout en bout. Ce système fournit des services multimédias principalement à temps réel dont la gestion de la qualité de service (QoS) est l'élément

fondamental. Des situations imprévisibles comme la surcharge du réseau et le manque de ressources peuvent causer la dégradation de la QoS fournie. Ceci revient à répondre à la question : comment assurer que le système à déployer dans le nuage fournit le service avec une QoS acceptable ?

Agilité des applications : les téléphones mobiles sont devenus de plus en plus un outil incontournable pour accéder aux services. Selon Ericsson une augmentation progressive du nombre total des abonnés téléphoniques devrait croître de 6,8 milliards en 2014 au 9.2 milliards avant la fin de 2019 dont 80% (7.2 milliards) ont un accès haut débit. Cette croissance continue du nombre d'abonnés provoque une augmentation du trafic réseau de l'opérateur et du volume de données moyennes échangées de plus de 10 fois en 2019. Ceci revient à répondre à cette série de questions : comment l'opérateur suit cette croissance et répond à toutes les demandes de services à leurs clients ? Comment assurer la mise à l'échelle du système IMS lors du changement du trafic (diminution/ augmentation) ? Quels sont les paramètres à adopter pour une technique automatique de mise à l'échelle ?

Gestion des ressources : en général, dans le système IMS fixe, les ressources informatiques sont allouées d'une manière statique. L'approvisionnement des ressources est effectué en se basant sur des techniques de prédiction de la charge de trafic selon des intervalles choisis, ainsi les ressources nécessaires sont réservées au préalable. Une mauvaise prédiction des besoins en ressources cause deux graves problèmes : une sur-utilisation des ressources si le trafic réel est inférieur à celle prédite ou une sous-utilisation si la charge prédite est supérieure et dans ce cas il y a une insuffisance des ressources allouées. Dans cette situation, il est judicieusement de répondre aux questions : quelle est la meilleure stratégie d'allocation des ressources pour mieux gérer les ressources informatiques ? Comment résoudre le problème lié à l'approvisionnement statique des ressources ? Comment automatiser la technique de l'approvisionnement des ressources dans le système IMS ?

Disponibilité des services dans IMS : l'augmentation du nombre d'utilisateurs et des terminaux connectés au réseau de l'opérateur augmentent progressivement la charge de trafic. Ceci peut causer des graves problèmes dans le réseau à savoir le non fonctionnement des équipe-

ments, une panne de matériel ou un manque de capacité sur les serveurs physiques. Ce qui mène à poser les questions : comment assurer la disponibilité des services lors de la surcharge des réseaux ? Comment réagir lors d'une panne d'un élément du réseau ? Quelle est la stratégie à adopter pour améliorer la disponibilité de ces ressources ?

1.2 Objectifs du mémoire

L'objectif principal de notre travail de recherche consiste à déployer le système IMS dans l'environnement de nuage informatique. L'architecture déployée dans le nuage doit permettre la mise à l'échelle automatique des ressources informatiques dont une amélioration de la QoS est visée dans le but de satisfaire l'utilisateur final.

Notre tâche consiste à migrer l'ensemble des composants de base de l'architecture de service d'un opérateur téléphonique (IMS) vers le nuage afin de réduire la complexité et le coût de déploiement.

Étant donné, le manque de valeurs ajoutées dans le secteur de la télécommunication et surtout pour les opérateurs téléphoniques. Le déploiement d'IMS dans l'écosystème de nuage permet aux opérateurs de bénéficier des différents avantages de cet environnement.

Les objectifs spécifiques sont alors :

- réduction de la complexité de déploiement des applications et des services Télécom ;
- réduction du CAPEX et OPEX ;
- automatisation de la mise à l'échelle automatique ;
- optimisation de la gestion des ressources informatiques ;
- amélioration de la qualité de service ;
- augmentation de la disponibilité des services IMS.

À travers, le déploiement du système IMS dans le nuage informatique, l'opérateur parvient à réduire le coût de l'infrastructure et de maintenance ainsi de confronter au problème d'augmen-

tation du coût d'investissement et la décroissance de du profit. D'ailleurs, il sera plus facile à intégrer de nouveaux services et les délivrer d'une manière différente et de faire face à la concurrence.

Pour valider la possibilité de déploiement d'IMS dans le nuage, notre recherche a été axée sur l'utilisation des produits IMS libre. Ainsi, nous avons déployé deux architectures IMS libres : Open Source IMS Core (FOKUS, 2006) et Clearwater IMS (Metaswitch, 2013). De même, concernant l'infrastructure nous avons aussi utilisé la solution libre Openstack (Rackspace, 2013) comme notre infrastructure en tant que service.

La mise à l'échelle est l'une des préoccupations majeures des opérateurs téléphoniques pour garantir l'évolutivité du système lors du changement de la charge de trafic sur le réseau. L'automatisation d'une technique pareille va permettre aux opérateurs d'assurer la disponibilité du réseau et de mieux gérer les ressources informatiques.

La bonne gestion des ressources informatiques d'un opérateur va permettre à ce dernier d'optimiser son utilisation et ainsi que de réduire son coût de déploiement et d'exploitation. Étant donnée, que la virtualisation est l'élément de base du nuage, cette technique va faciliter la gestion et le management des ressources virtuelles.

Certes que la réduction de coût est la première priorité des opérateurs téléphoniques et des fournisseurs de services, néanmoins il ne faut pas oublier que le principal service offert par un opérateur téléphonique est la voix. Ce type de service est à temps réel et il nécessite une certaine contrainte de priorité et de délai.

Toutefois, la non-satisfaction d'un utilisateur d'un service donné provoque une perte de client de la part de l'opérateur. De ce fait, maintenir un certain niveau de la qualité de service est l'un des objectifs importants à prendre en considération. Ainsi, l'utilisation des outils d'approvisionnement de la qualité de service va servir à mieux évaluer la qualité fournie par le réseau.

1.3 Plan du mémoire

Notre mémoire s'articule autour de cinq chapitres en plus d'une conclusion comme le montre l'organigramme de la Fig.1.1.

Le Chapitre 1 est une introduction générale qui résume le contexte du travail du mémoire, présenter les problèmes reliés à notre sujet de recherche et les objectifs à atteindre.

Le chapitre 2 est une revue de littérature de la qualité de service dans les réseaux de nouvelle génération. Dans une première partie, nous avons mis en contexte le NGN ainsi que les notions qui sont liées à la QoS dans les réseaux IP, la seconde partie est une présentation globale d'IMS, son architecture et les différentes méthodes de gestion de la QoS dans ce dernier.

Le chapitre 3 constitue une analyse des exigences d'IMS déployées dans le nuage informatique. D'abord, une vue d'ensemble sur le concept du nuage informatique sera développée, ensuite une présentation des exigences de ce dernier pour pouvoir déployer une telle Telco application (IMS) dans le nuage. Une architecture d'IMS sera proposée dans le nuage en se basant sur l'analyse précédente.

Le chapitre 4 est consacré au provisionnement de la QoS d'IMS déployé dans l'environnement nuage, l'étude des différentes techniques de provisionnement dans le nuage ainsi qu'une présentation des systèmes IMS existants. Une description du modèle proposé du provisionnement de la QoS d'IMS dans le nuage fera suite.

Le chapitre 5 est dédié à l'analyse de performance des différents bancs d'essai des architectures d'IMS déployé dans l'infrastructure de nuage d'Ericsson. Après une description détaillée des trois bancs d'essai, de l'environnement de test et de la technologie utilisée, les différents résultats obtenus avec leur interprétation seront exposés et discutés.

Pour conclure, une synthèse du travail effectué est développé avec les différentes perspectives possibles à partir de cette recherche qui feront l'objet des axes de nos travaux futurs.

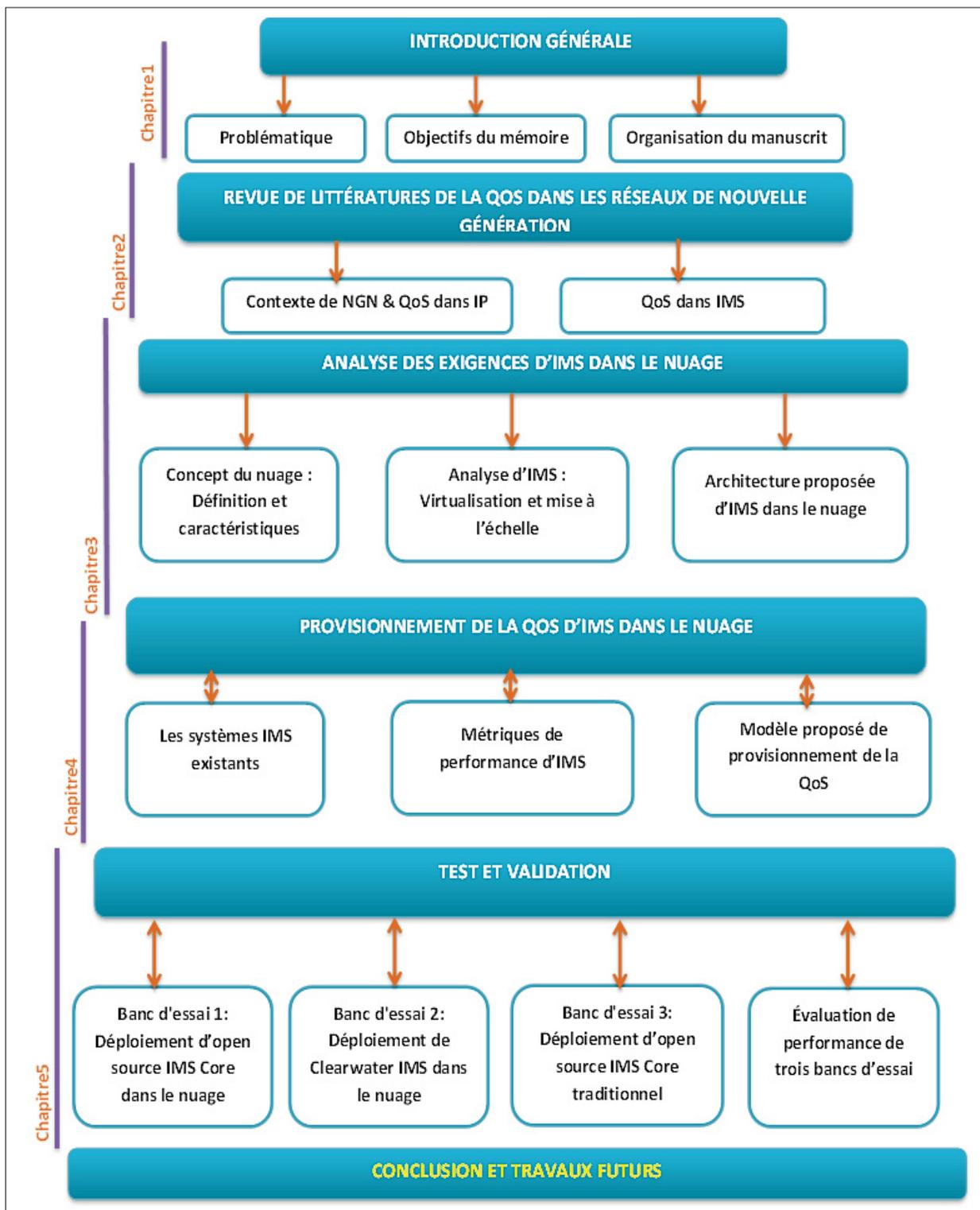


Figure 1.1 Diagramme des différents éléments présentés dans cette mémoire

CHAPITRE 2

REVUE DE LITTÉRATURES DE LA QUALITÉ DE SERVICE DANS LES RÉSEAUX DE NOUVELLE GÉNÉRATION (NGN)

2.1 Introduction

Ce chapitre présente une revue de littérature de la qualité de service dans les réseaux de nouvelle génération. Dans un premier temps, nous abordons les principales caractéristiques de la gestion de QoS dans les réseaux IP, les différentes métriques, les modèles de gestion ainsi que les exigences de la QoS pour les applications temps réel et non-temps réel. Dans un deuxième temps, nous exposons les principales évolutions des architectures de QoS dans le système IMS en se basant sur les architectures des réseaux de nouvelle génération.

2.2 Contexte des réseaux de nouvelle génération

2.2.1 Définition et concept

Les réseaux de nouvelle génération ou en anglais Next Generation Networks (NGN) telle que définit par l'union internationale des télécommunications (UIT) est un réseau capable de fournir les services de télécommunication aux utilisateurs et d'utiliser en même temps multiple large bande toute en se basant la transmission paquet.

Cependant, il doit être en mesure de garantir la QoS dans la couche de transport. Les fonctions liées aux services dans ces réseaux sont indépendantes des technologies liées au transport sous-jacent. Il permet aussi l'accès sans entrave pour les utilisateurs aux réseaux et aux fournisseurs de services concurrents et aux services de leur choix. Il prend en charge la mobilité généralisée qui permet la fourniture cohérente et partout de services aux utilisateurs.

2.2.2 Caractéristiques

Le réseau NGN est caractérisé par les aspects fondamentaux suivants :

- transfert par paquets ;
- séparation des fonctions de contrôle entre capacités de support ;
- découplage de la provision de services de transport et de la provision de l'interface ouvertes ;
- soutien à une vaste gamme de services, d'applications et de mécanismes fondés sur des blocs de construction de services y compris les services en temps réel et non temps réel ;
- capacités à large bande avec QoS et de la transparence de bout en bout ;
- inter fonctionnement avec les réseaux existants via des interfaces ouvertes ;
- mobilité généralisée ;
- libre accès des utilisateurs aux différents fournisseurs de services ;
- variété de schémas d'identification qui est résolu en adresses IP pour des fins de routage dans les réseaux IP ;
- caractéristiques de services unifiés pour le même service tels que perçus par l'utilisateur ;
- convergence des services entre les réseaux fixes et mobiles ;
- indépendance des fonctions liées aux services de technologies de transport sous-jacents ;
- support de multiples technologies ;
- conforme à toutes les exigences réglementaires.

2.3 Qualité de service dans les réseaux IP

Cette section définit la qualité de service selon les organisations de standardisation et présente ces différents types on se basant sur ces définitions. Ainsi, une description des principales métriques et modèles de gestion de la QoS dans les réseaux IP sera présenté.

2.3.1 Définitions de la QoS

Il existe plusieurs définitions de la qualité de service selon les organisations de normalisation : Le standard ISO 8402 (ISO, 2000a) définit la qualité de service comme « l'ensemble des ca-

ractéristiques d'une entité qui porte sur sa capacité à satisfaire des besoins demandés et implicites».

Le standard ISO 9000 (ISO, 2000b) a défini la qualité comme « le degré avec lequel un ensemble de caractéristiques inhérentes répondent à des besoins ».

L'ITU-T (E.800, 1993) et l'ETSI la qualité de service est définie comme « l'effet collectif de la performance d'un service qui détermine le degré de satisfaction d'un utilisateur de ce service demandé»

L'IETF définit la QoS en tant que « la capacité à séparer le trafic ou à différencier différents types de trafic afin de traiter certains flux différemment d'autres flux ».

2.3.2 Type de la qualité de service

Concernant les différents types de la QoS, on peut les classer en trois types selon les définitions des organismes de standardisation : la qualité de service intrinsèque, perçue et évaluée (Hardy, 2001).

La qualité de service intrinsèque

Ce type de QoS est directement fourni par le réseau lui-même. La QoS intrinsèque est décrite par des paramètres objectifs tels que le délai et les pertes des paquets. Et donc l'IETF la considère le point essentiel.

La qualité de service perçue

Appelé aussi qualité d'expérience ou en anglais Quality of Experience (QoE) et qui représente la qualité ressentie par l'utilisateur. En général la QoE est utilisée pour évaluer les applications à temps réels telles que la qualité d'audio et de vidéo. La qualité de service perçue est liée aux performances du réseau, mais elle mesurée par une moyenne attribuée par les utilisateurs selon leurs opinions sur le service offert.

Généralement, la méthode la plus utilisée est le Mean Opinion Score (MOS) qui représente une technique où les utilisateurs accordent une note, valeur entre 1(qualité très mauvaise) et 5(qua-

lité excellente) à une application selon leurs perceptions de la qualité. Ensuite, une moyenne est calculée des notes de ces derniers. Ainsi, la note attribuée par les utilisateurs peut concerner la disponibilité du service, le temps de la réponse de l'application. Par ailleurs, il n'existe pas fortement une correspondance entre la QoS intrinsèque et la qualité perçue vu que cette dernière est très subjective (Hardy, 2001).

La qualité de service évaluée

La qualité de service évaluée est traduite par la volonté d'un utilisateur de garder un service spécifique ou non. Ce type de qualité dépend fortement de la qualité d'expérience et d'autres critères tels que le prix, l'assistance offerte avec le service et le marketing.

2.3.3 Les principales métriques de la QoS dans les réseaux IP

Cette section décrit les principaux paramètres objectifs de la qualité de service intrinsèque (Y.1540, 2011).

Délai de transfert d'un paquet (IPTD) : représente le délai de transfert de paquet, généralement calculé de bout en bout et exprimé en milliseconde.

Variation de délai de transfert d'un paquet (IPDV) : Appelé aussi gigue ou jitter en anglais et qui représente la variation de délai de transfert entre deux paquets consécutifs.

Taux de perte de paquets (IPLR) : désigne est le pourcentage de paquet perdu par rapport à celle émise.

Taux de paquet erroné (IPER) : désigne le taux de paquet erroné à la réception par rapport à celui émis.

Débit : le débit peut ou ne pas être considéré comme une métrique de la QoS, le débit est le nombre de bits par unité de temps qu'une source a le droit d'émettre.

2.3.4 Exigence de la QoS pour les applications temps réel

Parmi les applications multimédias les plus exigeantes en termes de qualité de service sont les applications audio et vidéo. Les contraintes de la QoS de ces applications selon les différentes recommandations de l'ITU-T (Y.1540, 2011) sont illustrées dans le tableau 2.1.

Il faut noter que ces valeurs dépendent aussi d'autres paramètres tels que le type de codec et du débit binaires et supposent qu'il n'existe pas de perte de paquets afin de minimiser ces effets.

Tableau 2.1 G.1010 – Objectifs de performance pour les applications audio et vidéo

Applications	Symétrie	Débit binaire	Délai aller	Gigue	IPLR
Conversation audio	Bilatérale	4-64kbps	Préféréd<150ms Limitéd<400ms	<1ms	<3% IPLR
Messagerie Vocale	Unilatérale	4-32kbps	Reproduction<1s Enregistrement<2s	<1ms	<3% IPLR
Streaming audio	Unilatéral	4-128kbps	<10s	<<1ms	<1% IPLR
Streaming vidéo	Bilatéral	4-384kbps	Reproduction<1ms Enregistrement<2s	<1ms	<1% IPLR
Vidéoconférence	Unilatéral	4-384kbps	Préféréd<150ms Limitéd<400ms	<1ms	<1% IPLR

2.3.5 Exigence de la QoS pour les applications non-temps réel

Les applications non-temps réel sont les moins exigeants en termes de QoS. Les spécifications des exigences de la QoS pour quelques applications non-temps réel selon les différentes recommandations de l'ITU-T (Y.1540, 2011) sont décrite dans le tableau 2.2. Parmi ces applications on peut noter la navigation web, le transfert de données, les jeux interactifs.

Tableau 2.2 G.1010 – Objectifs de performance pour les applications de données

Applications	Symétrie	Quantité de données	Délai aller	Gigue	IPLR	IPER
Navigation web	Unilatéral	10 ko	Préféréd <2 s/page Acceptable<4s/page	NA	0%	0%
Transfert de données en masse	Unilatéral	10 ko-10Mo	Préféréd <15s Acceptable<50s	NA	0%	0%
Transaction (e-commerce)	Bilatéral	10 ko	Préféréd <2s Acceptable<4s	NA	0%	0%
Images fixes	Unilatérales	100ko	Préféréd <1s/page Acceptable<50s/page	NA	0%	0%
Jeux Interactifs	Bilatéraux	1ko	<200Mo	NA	0%	0%
Telnet	Bilatéral	1ko	<200Mo	NA	0%	0%
Courriel	Unilatéral	10 ko	Préféréd <2s Acceptable<4s	NA	0%	0%
Fax	Unilatéral	10 ko	30s/page	NA	0%	0%

2.3.6 Modèles de gestion de la QoS

La plupart des réseaux IP sont basés sur des modèles de gestion de la QoS. Ces modèles se diffèrent par le niveau de garantie qu'il offre ou non la QoS. Les différents modèles proposés pour le management de la QoS dans les réseaux de nouvelles générations sont : Best-effort, InterServ et DiffServ.

2.3.6.1 Le modèle Best-effort

Le modèle Best-effort est le modèle utilisé sur internet, qui ne garantit aucun niveau de qualité ou priorité. Une application peut envoyer des paquets sans limitation et sans demander une autorisation ou informer le réseau avant l'envoi. Le réseau est chargé de délivrer ces paquets en best effort mais ne garantit aucun délai maximal. Le modèle Best-effort est utilisé par défaut dans l'Internet, applicable pour la plupart des applications réseau telles que le transfert de fichier FTP, navigation web (H3C, 2008).

2.3.6.2 Le modèle InterServ

Ce modèle a été défini par l'IETF dans le cadre de proposer une architecture de service intégré. L'objectif principal de ce modèle est d'éviter la congestion en utilisant une technique de réservation de ressources avant de commencer le transfert de données. Ce modèle prend en charge la qualité de service sans changer le protocole IP et son implémentation se fait dans les entités du réseau (routeurs).

Le modèle InterServ utilise le protocole RSVP pour la signalisation de bout en bout. Le principe de fonctionnement de ce protocole est la réservation de ressources de bout en bout avant tout transfert de données. Avant l'émission des données, l'émetteur envoie un message « PATH » vers le récepteur qui contient la spécification du trafic qui sera généré par l'application. Lorsque le message « PATH » arrive à la destination, il répond avec un message « RESV » qui suit le chemin inverse et qui déclenche la réservation des ressources requises.

Une fois la réservation des ressources est effectuée, le flux qui traverse une entité et qui fait l'objet d'une réservation sera identifié en fonction de l'adresse et le port source et destination, ainsi que le protocole de transmission utilisé.

Il faut noter que l'échange de donnée un réseau InteServ doit suivre le même chemin que les deux messages « PATH » et « RESV ». Une fois l'échange de donnée est terminé, l'application doit libérer les ressources déjà réservées à travers les deux messages « PathTear » et « ResvTear ». La fig.2.1 illustre l'échange des messages RSVP dans un réseau qui utilise le modèle InterServ (Zhang *et al.*, 1997).

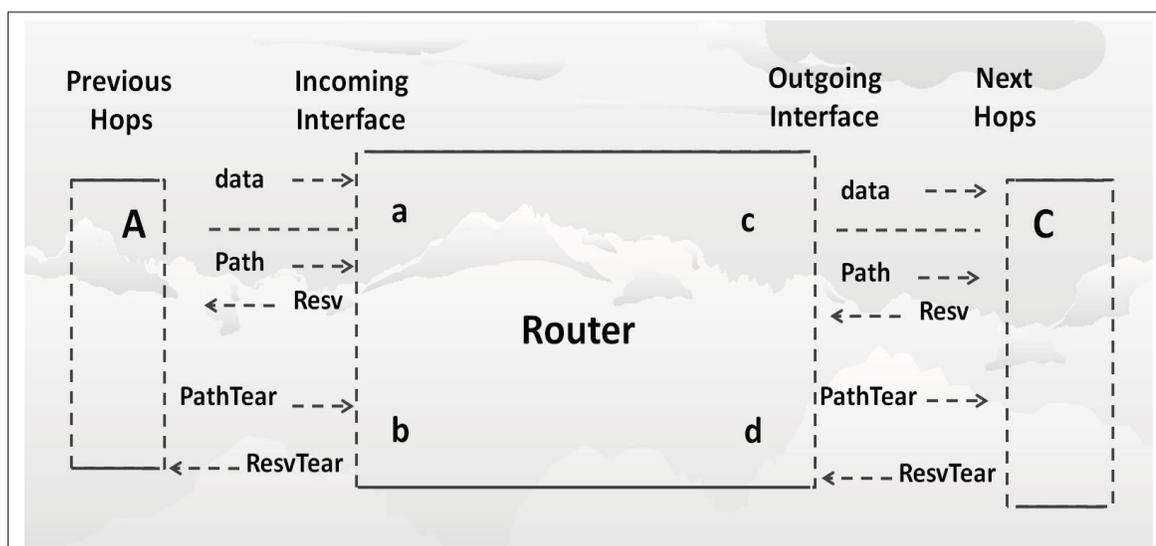


Figure 2.1 Échange des messages RSVP
Tirée de rfc2205 (1997)

Par ailleurs, les routeurs doivent introduire quatre nouvelles fonctionnalités :

- **l'ordonnancement de paquets** : qui détermine l'ordre de service des paquets envoyé ;
- **les classificateurs de paquets** : qui classent les flots des paquets dans les classes spécifiques (BE, CL ou GS) ;
- **le contrôle d'admission** : permet de vérifier l'admissibilité d'un nouveau flot et d'autoriser son arrivée munie d'une certaine QoS sans affecter les flots existants ;

- **le protocole de réservation de ressource RSVP** : ce protocole permet la signalisation hors bande et établir des réservations de bande passante sur chaque routeur traversé dans le réseau.

classe de services InterServ

Deux nouvelles classes ont été définies avec le modèle InterServ en plus de la classe Best-effort :

- **service garanti (GS)** : Cette classe de service permet de garantir les paramètres de performances délai maximal, bande passante. Lorsque, les flux respectent les paramètres déjà réservés, il garantit que les paquets arrivent avec un délai maximal et qu'il n'aura pas de perte. Cette classe de service est généralement applicable pour les applications à temps réelles VoIP, push to talk, etc ;
- **contrôle de charge (CL)** : Cette classe de service est équivalente au service de Best-effort dans un domaine non surchargé.

Avantage et inconvénient du modèle InterServ

Le modèle InterServ offre une garantie de la QoS de bout en bout à travers la réservation préalable des ressources selon les exigences de chaque service. Néanmoins, ce modèle est complexe et ne permet pas le passage à l'échelle vu que chaque nœud du réseau doit garder beaucoup d'informations sur l'état du réseau. La capacité limitée et la complexité de ce modèle rendent ce dernier inadaptable pour les réseaux de grande taille.

2.3.6.3 Le modèle DiffServ

Ce modèle a été proposé pour résoudre le problème de complexité d'InterServ. Contrairement à InterServ qui effectue le traitement de flux de bout en bout, le modèle DiffServ traite de flux au niveau des routeurs situés aux extrémités du réseau (routeurs de frontière) en agrégeant le flux par classe.

Dans ce mode chaque classe de service est identifiée par une valeur dans l'entête IP (TOS ou TC) qui a été redéfini et renommé par DSCP (DiffServ code point). Il s'agit du champ TOS (Type Of Service) pour IPv4 et TC (Traffic Class) pour IPv6. Chaque classe de service possède ces caractéristiques en terme de métrique de la QoS (délai, gigue, perte).

Domaine DiffServ

Le concept de domaine Diffserv consiste au regroupement d'un ou plusieurs réseaux soumis à une seule autorité administrative. Les routeurs de bordure de chaque domaine sont responsables de l'agrégation du trafic en des classes et permettent de se connecter à d'autre domaine DiffServ. La Fig.2.2 illustre un exemple d'utilisation du domaine Diffserv.

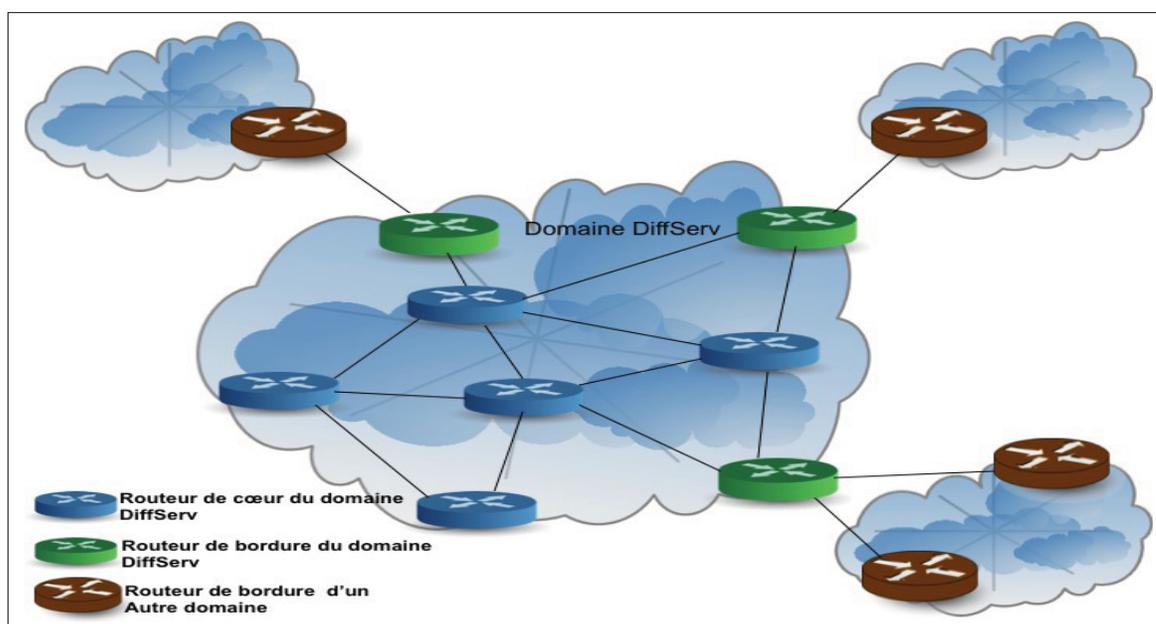


Figure 2.2 Domaine de Diffserv
Tirée de Cisco (2013)

Chaque abonné à un domaine DiffServ possède un contrat appelé SLA (Service Level Agreement) qui fixe les termes et les conditions d'utilisation du service, dont la partie technique (QoS que le fournisseur s'engage d'offrir, sécurité) du service offert aux utilisateurs est spécifiée dans un contrat SLS (Service Level Specification).

Classe de services DiffServ

Dans un réseau DiffServ, chaque entité assure la transmission des paquets par le comportement de saut (PHB : per-hop behavior) en fonction du champ DSCP dans l'entête. Ce pendant trois classes de services ont été définies dans ce modèle chacun d'eux soumis à un traitement spécifique :

- **transfert accéléré(EF)** : Cette classe de service est applicable le trafic qui nécessite un faible retard, faible gigue, et à faibles taux de pertes, généralement, pour le trafic qui exige u débit relativement constant et une transmission rapide ;
- **transfert assuré (AF)** : Cette classe de service assure la transmission quand il ne dépasse pas la largeur de bande maximale autorisée. Si le trafic généré dépasse la bande passante autorisée, ce dernier sera divisé en quatre classes AF. L'IETF recommande d'utiliser quatre files pour chaque classe de service ;
- **best effort (BE)** : Cette classe de service est applicable aux services insensibles à délai, gigue, et perte de paquets.

2.4 Qualité de service dans IMS

2.4.1 Présentation d'IMS

IP multimedia subsystem est une architecture réseau standard qui a été défini dans le contexte des réseaux de prochaines générations pour offrir des services multimédias basés sur IP. Initialement, cette architecture a été conçue le réseau mobile tel que spécifié dans la version 5 du 3GPP dont le but de fournir des services multimédias aux abonnées troisième génération des réseaux cellulaires(UMTS). Le standard IMS développé répond aux besoins des opérateurs offrant de multiples services sur une grande échelle à de nombreux clients avec plus de flexibilité et à faible coût.

Cependant, IMS a rapidement convergé vers le réseau fixe pour assurer l'inter fonctionnement des réseaux fixes et mobiles sur une architecture tout IP. IMS utilise le protocole applicatif

standardisé par l'IETF, SIP(Session Initiation Protocol) pour la signalisation et le contrôle de la session en plus du protocole SDP. Ce protocole ne se charge pas de la transmission des données de la session, et IMS utilise d'autres protocoles tel que RTP/RTCP pour la transmission des services à temps réel comme la voix sur IP est assurée.

2.4.2 Architecture d'IMS

L'analyse de l'architecture de IMS illustré dans la Fig.2.3 et du rôle des différentes entités dans le réseau pour fournir une QoS de bout en bout montre que les composants et les fonctionnalités d'IMS peuvent être classés en six catégories (Poikselkä et Mayer, 2009).

- fonctions de contrôle d'appel session (CSCF) ;
- bases de données (HSS, SLF) ;
- serveurs médias et les serveurs d'application (AS, MRFC) ;
- fonctions d'inter fonctionnement (BGCF, MGCF, IMS-MGW, SGW) ;
- fonctions de support (PCRF, SEG, IBCF, TrGW, LRF) ;
- fonctions de chargement.

Proxy-Call session Control Function (P-CSCF)

Lorsque l'utilisateur est attaché au réseau, le trafic de signalisation SIP de l'utilisateur sera envoyé à la classe P-CSCF qui représente le premier point de contact avec IMS. Le P-CSCF est également responsable de la sécurité pour le trafic SIP et d'interagir avec PCRF (politique et une fonction de recharge rôle) pour échanger des informations de la politique et de facturation à l'utilisateur. En outre, il gère la session d'urgence quand il est détecté (Poikselkä et Mayer, 2009).

Serving-Call session Control Function (S-CSCF)

Le S-CSCF est l'une des entités de base dans le système IMS. En effet, lorsqu'un utilisateur envoie une demande d'enregistrement, il sera acheminé vers le S-CSCF, qui a été choisi par I-CSCF.

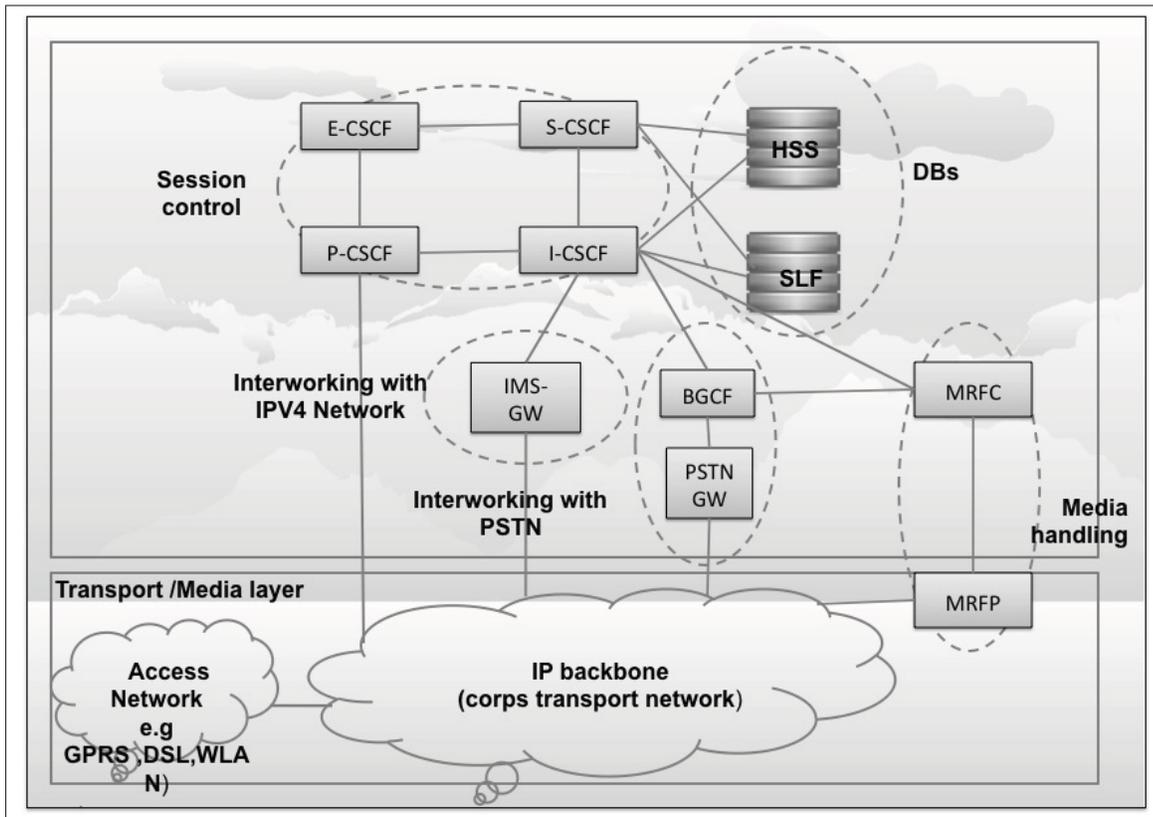


Figure 2.3 Architecture d'IMS
Tirée de El Barachi *et al.* (2011)

La première fonction de la S-CSCF est de vérifier l'identité (authentification processus) de l'utilisateur en fonction du profil stocké dans le HSS. Après identification, le S-CSCF, enregistre l'utilisateur et recontacte le HSS pour télécharger le profil de service afin de décider quel serveur d'application est autorisé (par exemple, audio, présence, push to talk). Le S-CSCF est également responsable de générer les enregistrements de facturation.

Interrogating- Call session Control Function (I-CSCF)

Cette entité est le point de contact au sein d'un réseau de l'opérateur. Il contacte le HSS et recueille les informations du S-CSCF que l'utilisateur sera affecté pour l'acheminement des requêtes entrantes.

Emergency - Call session Control Function (E-CSCF)

Cette entité est dédiée au traitement des demandes d'urgence dans IMS.

Home subscriber Server(HSS)

Le HSS est la base de données principale pour tous les utilisateurs. Elle stocke les profils des utilisateurs tels que leurs identités, les informations d'enregistrement, les paramètres d'accès, les services autorisés, la localisation de l'utilisateur et le S-CSCF assignés.

Subscription Locator Function (SLF)

Cette base de données est utilisée pour identifier l'adresse appropriée du HSS pour le I-CSCF, le S-CSCF et l'AS lorsque le réseau d'opérateur déploie plusieurs HSS.

Application Server (AS)

Cette entité n'est pas considérée comme une entité pure du réseau cœur IMS parce qu'elle réside dans les réseaux domestiques de l'utilisateur, mais dans (Poikselkä et Mayer, 2009) ils le décrivent comme étant une partie d'IMS, vu qu'elle fournit quelques services multimédias tels que push to talk dans les réseaux cellulaires. En fait, (Khelifi et Gregoire, 2008) définit trois types de serveurs d'applications (AS) dans IMS :

- serveurs d'applications SIP ;
- architecture de serveur d'application libre OSA(Open architecture Application servers) ;
- applications personnalisées pour les réseaux mobiles utilisant méthode de connexion amélioré.

Un serveur d'application (AS) contient un ou plusieurs services et les utilisateurs peuvent associer plus qu'une AS en fonction de leurs profils.

Media Server (MS)

Appelé aussi media ressource function(MRF) et décomposé en deux éléments : MRFC et MRFP qui gèrent les communications SIP depuis et vers la S-CSCF (Khelifi et Gregoire, 2008) et fournit à l'utilisateur les ressources demandées (Poikselkä et Mayer, 2009).

Breakout Gateway Control Function (BGCF)

Cette entité est responsable de sélectionner le MGCF qui va acheminer l'appel vers le PSTN.

Policy and Charging Rules Function (PCRF)

Cette entité est responsable de la gestion et de la décision des politiques et de contrôle de chargement en se basant sur les informations de la session et le média offert à l'utilisateur en fonction des informations collectées de P-CSCF.

Border Gateway (BG)

Cette fonction est décomposée en deux entités (IBCF and TrGW) et assure les interconnexions entre domaines d'opérateurs.

Interconnection Border Control Function (IBCF)

Cette entité a été définie dans la recommandation ETSI et son rôle est de permettre la communication entre IMS IPv6 et IPv4 et le plan contrôle du niveau transport.

Transition Gateway (TrGW)

Il est chargé de fournir la version dans le niveau transport.

2.4.3 Gestion des identités en IMS

Dans le système IMS, un nouveau concept d'identification est adopté. Cependant, l'identification reste compatible avec les anciens réseaux comme le GSM. Les utilisateurs dans IMS sont identifiés à travers un couple d'identifiants : un identifiant public et un identifiant privé.

Identité publique de l'utilisateur

L'Identité publique ou en anglais IP Multimedia public Identity (IMPU) est une identité publique attribuée par l'opérateur à l'utilisateur IMS. Chaque utilisateur peut avoir un ou plusieurs identifiants publics qui est similaire au MSISDN (Mobile Station ISDN Number) dans les réseaux GSM et UMTS et peut être sous deux formats :

Format SIP URI : "sip : Alice @openims.test"

Format TEL URI : "tel : 514 000 0000"

Identité privée de l'utilisateur

L'identité privée ou en anglais IP Multimedia Private Identity (IMPI) est une identité unique pour chaque utilisateur qui est stockée dans la carte à puce. Cette identité permet l'authentification de l'utilisateur lors de la procédure d'enregistrement au sein du réseau IMS. L'identité privée de l'utilisateur IMS est unique et associée aux ensembles des IMPU que l'utilisateur possède telle qu'illustrée par la Fig.2.4.

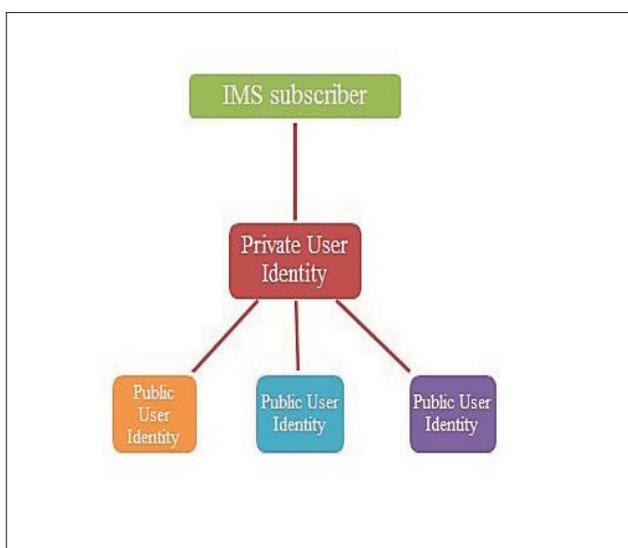


Figure 2.4 Relation entre l'identité privée et public dans IMS
Tirée de MROUEH Lina (2006)

L'ensemble des identités de chaque utilisateur est stocké dans le profil de l'utilisateur au sein de la base de données de HSS.

2.4.4 Concepts et approches de gestion de la QoS dans IMS

En IMS, le problème regardant la QoS au niveau de la couche de transport IP est similaire à celui défini à l'Internet. Plusieurs travaux ont examiné ce problème et les modèles IntServ et DiffServ ont été étudiés dans différentes situations.(Raouyane *et al.*, 2009), (Skorin-Kapov *et al.*, 2007) et(Bertrand, 2007).

Les deux modèles DiffServ et InterServ s'appliquent pour les réseaux IMS, par contre la capacité du DiffServ de maintenir un minimum d'informations concernant l'état du réseau rend ce dernier le plus approprié et le plus évolutif par rapport à IntServ (Bertrand, 2007).

Pour les autres couches, le système IMS gère la QoS selon le service demandé par l'utilisateur tant que chaque service possède ces propres exigences. À cette fin, le 3GPP a introduit une nouvelle architecture dans sa release 7, Policy and Charging Control (PCC)(Skorin-Kapov *et al.*, 2007) pour le management de la QoS dans IMS. Le mécanisme de gestion de la QoS avec le PCC permet de déterminer si la QoS est respectée ou non, ainsi de réserver les besoins requis pour chaque service selon la demande et le profil de l'abonné.

2.4.5 Architecture de PCC(Policy and Charging Control)

Jusqu'à la release 5 du 3GPP l'architecture de PCC spécifiée n'introduit aucun mécanisme de gestion de la QoS. En effet cette architecture a évolué très rapidement, des modifications sont apportées sur l'architecture d'IMS et c'est dans de la release 7 que le 3GPP a spécifié de nouvelles fonctions qui sont associées à l'allocation des ressources et la gestion de la QoS.

Dans la release 9 (3GPP, 2010) l'architecture de PCC comprend quelques nouvelles spécifications et comprend des fonctions de haut niveau, à la fois pour le chargement et la QoS. Fig. 2.5 montre l'architecture du PCC spécifié dans cette release.

La description des entités de l'architecture du PCC est présenté dans le Tableau 2.3.

Quatre fonctions de contrôle de la qualité de service spécifient dans la release 9 dans l'architecture de PCC : allocation de ressource, événement déclencheur, établissement de flux de médias et contrôle de réseau. L'architecture PCC inclut quatre niveaux de service des paramètres de la QoS : Allocation and Retention Priority (ARP), Guaranteed Bit Rate (GBR) /non-Guaranteed Bit Rate (non-GBR), QoS Class Identifier (QCI) et le débit maximum (Maximum Bit Rate : MBR).

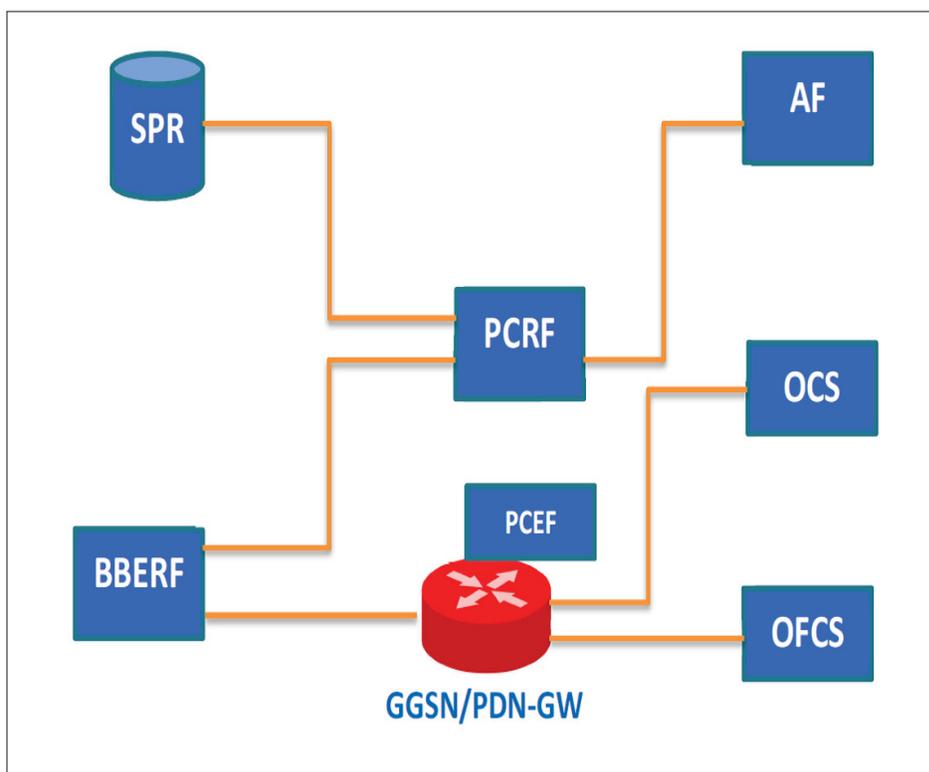


Figure 2.5 Architecture de PCC
Tirée de 3GPP (2010)

Allocation et rétention de priorité (ARP)

Ce paramètre inclut les paramètres de niveau de priorité et de sécurité ; pre-emptions capability (PEC) et pre-emptions vulnerability (PEV) qui sont organisés en valeurs.

- le niveau de priorité a une plage de valeurs de 1 à 15 partagées entre les services de réseau domestique et le réseau visité ;
- le PEC est défini comme étant la capacité d'une session d'obtenir les ressources qui sont déjà assignées à une autre session avec un niveau de priorité inférieur ;
- le PEV est défini comme la vulnérabilité d'une session afin de permettre la perte de ressources qui sont déjà affectées d'une autre session avec un niveau de priorité supérieur.

Tableau 2.3 Les entités de PCC

Composants	Fonctionnalités
AF	Media Format Flow description AF Application Identifier ; AF Application Event Identifier ; AF Record Information ; Flow status (for gating decision) ; Priority indicator, which may be used by the PCRF to guarantee service for an application session of a higher relative priority ; Emergency indicator.
BBERF	Bearer binding. Uplink bearer binding verification. Event reporting to the PCRF. Sending or receiving IP-CAN-specific parameters, to or from the PCRF.
OCS	Online Charging System
OFCS	Offline Charging System
PCRF	Encompasses policy control decision and flow based charging control functionalities. Provides network control regarding the service data flow detection, gating, QoS and flow based charging towards the PCEF. Apply the security procedures before accepting service information from the AF. Decide how a certain service data flow shall be treated in the PCEF Bearer binding.
PCEF	Subscriber Identifier ; IP address of the UE ; IP-CAN bearer attributes ; Request type (initial, modification, etc.) ; Type of IP-CAN (e.g. GPRS, I-WLAN, etc.) ;
SPR	Subscriber's allowed services ; For each allowed service, a pre-emption priority ; Information on subscriber's allowed QoS, including the Subscribed Guaranteed Bandwidth QoS ; Subscriber's charging related information (e.g. location information relevant for charging) ; Subscriber's User CSG Information reporting rules ; Subscriber category ; Subscriber's usage monitoring related information.

Débit binaire garanti(GBR)/Débit binaire non garanti (non-GBR)

Ce paramètre indique si une session a réservé des ressources de débit binaires ou non. Il est associé à la caractéristique de type de ressource de QCI.

Identificateur de classe de la QoS (QCI)

L'QCI est un nombre scalaire associé à un élément de réseau et il est utilisé pour décrire le traitement d'acheminement des paquets en termes de caractéristiques de performance. Cette valeur doit être pré configurée par l'opérateur directement dans le composant du réseau. Puisqu'il existe plusieurs caractéristiques associées aux valeurs QCI, 3GPP normalisés quatre ca-

ractéristiques : type de ressource (GBR, non GBR), la priorité, le budget de délai de paquet, et le taux de perte de paquets. Le tableau 2.4 montre un exemple de QoS Class Identifier dans LTE. réseaux hétérogènes (DiffServ, InterServ et UMTS)(Ryu *et al.*, 2006).

Tableau 2.4 Identificateur de classe de la QoS dans LTE

QCI	Types de Ressources	Priorité	Délai	PELR	Exemples de services
1	GBR	2	100 ms	10^{-2}	conversation audio
2		4	150 ms	10^{-3}	conversation vidéo (live)
3		3	50 ms	10^{-3}	jeux en temps réelle
4		5	300 ms	10^{-6}	non-conversation vidéo
5	Non-GBR	1	100 ms	10^{-6}	IMS signaling
6		6	300 ms	10^{-6}	vidéo streaming email, chat, ftp et partage de fichier
7		7	100 ms	10^{-3}	voix, vidéo (live streaming) et jeux interactive

2.4.6 Négociation de la QoS dans IMS

Dans cette section nous présentons comment les entités du système IMS négocient les paramètres de la QoS. L'objectif de la négociation de QoS vise à déterminer la meilleure configuration de service et d'allocation des ressources réseau qui maximise la qualité du service de l'utilisateur (Skorin-Kapov *et al.*, 2007). Fig. 2.6 illustre la négociation des paramètres de la qualité de service et de réservation de ressources entre UE et le serveur d'application (AS).

Les procédures de négociation des caractéristiques de la session multimédia sont spécifiées par le 3GPP et sont basées sur le mécanisme SIP / SDP et les paramètres qui pourraient être négociés comprennent le type, la qualité, l'encodage des médias, capacités du terminal à utiliser, et QoS désirée (QoS garantie, non garanti QoS, best effort).

La première étape, un UE doit obtenir l'accès à IMS à travers un réseau d'accès (GPRS, UMTS). La seconde étape, consiste à allouer un Proxy-CSCF qui va s'interfacer avec le PDF (Policy Decision Function) pour autoriser l'utilisation du bearer et les ressources de la QoS pour les services IMS dans le réseau d'accès. Après, le P-CSCF sélectionne le S-CSCF qui va interroger le Home Subscriber Server (HSS) pour accéder aux informations de profil utilis-

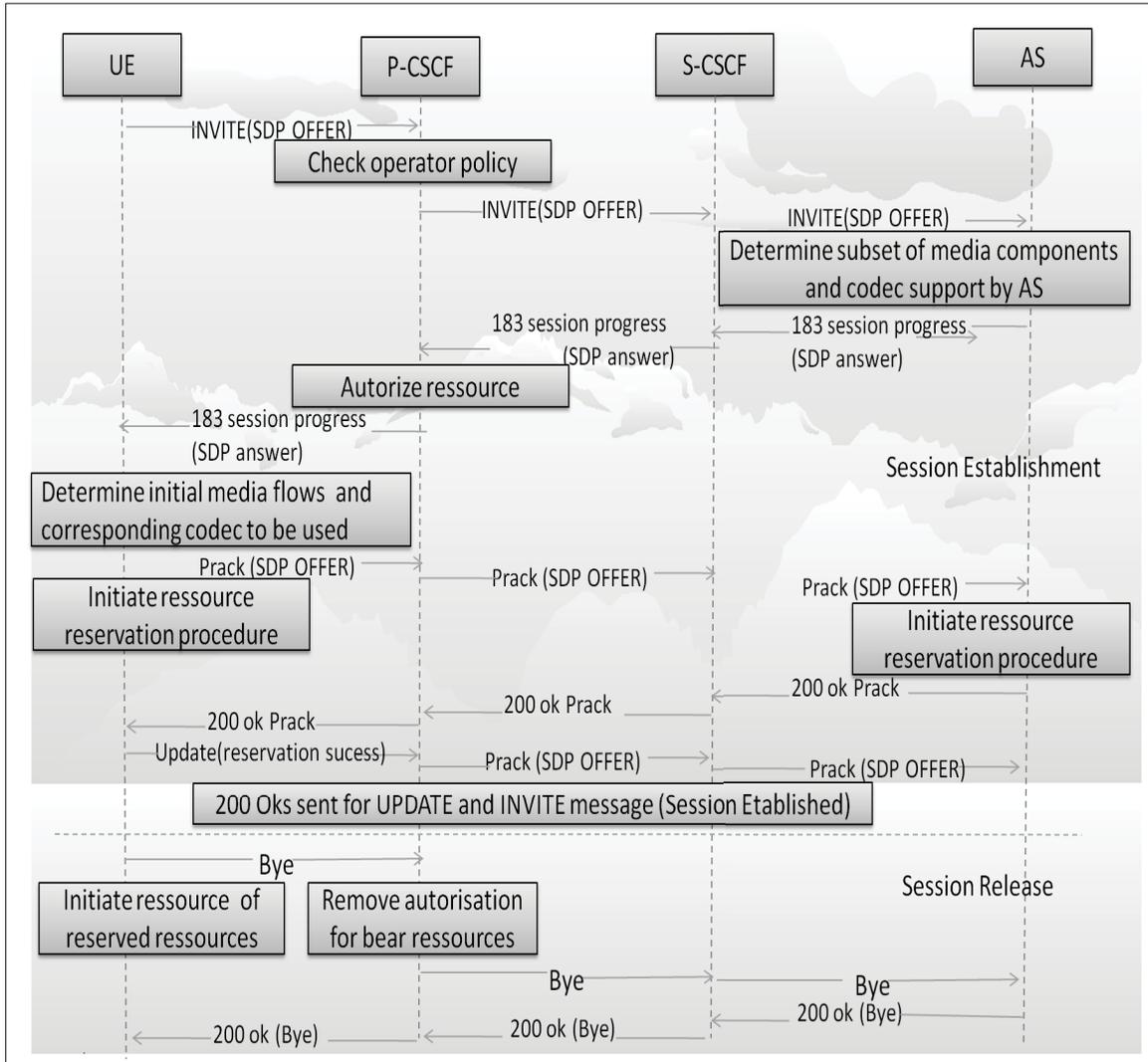


Figure 2.6 Signalisation et négociation de la QoS dans IMS (SIP/SDP)
Tirée de Skorin-Kapov *et al.* (2007)

teur, récupérer les données de l'abonnement, et pour réaliser l'authentification, l'autorisation et de facturation.

La dernière étape, à la fin de la transmission les deux entités communicantes libèrent les ressources déjà réservées.

2.4.7 Mappage de la QoS dans IMS

Lors de l'étude de la QoS dans le contexte des réseaux de prochaines générations il est important de prendre en considération le mappage de la QoS lors de l'inter fonctionnement entre réseaux hétérogènes.

En général, le mappage de la QoS dans ces réseaux catégorisé par le mappage entre les différentes classes de service et paramètres pour les différents réseaux. Cependant, la mise en œuvre de cette technique est difficile et ne garantit pas la granularité de la classe de QoS entre des réseaux différents (Ryu *et al.*, 2006).

De même, chaque opérateur téléphonique utilise sa propre architecture de QoS pour fournir une qualité acceptable a leurs abonnés qui justifie leur abonnement a un service donné. L'aspect hétérogène de ces réseaux pose la question suivante, comment les opérateurs peuvent gérer la QoS lors du mappage entre des réseaux IMS de caractéristiques différentes ?

Différentes approches ont été adressées pour le problème mappage de la QoS dans IMS. Ainsi, le mappage peut être accompli entre deux réseaux IMS utilisant le un même modèle de gestion ou des modèles de gestion différents. Par exemple, le mappage entre deux domaines Diffserv ou le mappage entre IP et UMTS. Le tableau 2.5 montre un exemple de mappage entre les classes de QoS d'IP et les réseaux hétérogènes (DiffServ, InterServ et UMTS)(Ryu *et al.*, 2006).

Tableau 2.5 Mappage de la classe de qualité de service sur le réseaux hétérogène

QoS IP Class	Exemples de services	InterServ	DiffServ	UMTS
0	Real time, jitter sensitive, high interaction (VoIP, video conference)	GS	EF	Conversational
1	Real time, jitter sensitive, interactive (VoIP, video conference)	CS	EF	Streaming
2	Transaction data, highly interactive (Signalling)	CS	AF	Interactive
3	Transaction data, interactive	CS	AF	Interactive
4	Low loss only (data, video streaming)	CS	AF	Interactive
5	Traditional applications of default IP networks	BE	Default	Background

D'autres études spécifient l'utilisation d'une entité à part qui se charge de cette fonction pour assurer plus de granularité lors du mappage.

(Ryu *et al.*, 2006) proposent une méthode mappage de la QoS entre les différentes technologies de transport à l'aide d'une Application Service Map (ASM). Après avoir déterminé l'exigence en paramètre de base (délai, gigue, bande passante, la perte de paquets, etc.) en fonction du service demandé. Le mappage de la QoS en utilisant ASM est basé sur la perception des utilisateurs de bout en bout indépendante de la technologie réseau.

Afin de garder les utilisateurs, les opérateurs développent leur portefeuille avec l'intégration des services avec des différentes exigences de qualité de service pour les applications destinées à plusieurs utilisateurs mobiles tels que l'IPTV. Ce type de services est appelé multi-utilisateur session, car ils peuvent être répartis entre plusieurs utilisateurs en même temps avec des profils de qualité de service différents.

(Cerqueira *et al.*, 2007) présentent une nouvelle approche Session-aware QoS mapping and adaptation (SOMA) qui mappe la session dans le service réseau le plus approprié dans les réseaux hétérogènes (par exemple IntServ, DiffServ et UMTS) et différentes technologies de transport (par exemple IP unicast ou multicast) (Cerqueira *et al.*, 2007). Les SOMA peuvent être implémentés directement dans les entités de frontière ou dans le cœur du réseau. En effet, le schéma proposé SOMA repose sur les exigences de chaque utilisateur. L'agent du réseau avec lequel le récepteur est attaché crée un "Session Object" composé des paramètres de QoS comme le débit binaire, garantie de bande passante, la tolérance à la perte et la gigue et les métriques du trafic (taille des paquets). Fig. 2.7 montre un exemple d'emplacement de SOMA et la Fig. 2.8 montre un exemple de fonctionnement SOMA.

2.5 Conclusion

Les modèles de gestion de la QoS proposés dans le réseau IMS sont les mêmes modèles utilisés dans les réseaux IP InterServ et DiffServ. Ainsi, la négociation des paramètres de la QoS dans ce système est effectuée au préalable avant toute transmission de données en utilisant SIP/SDP.

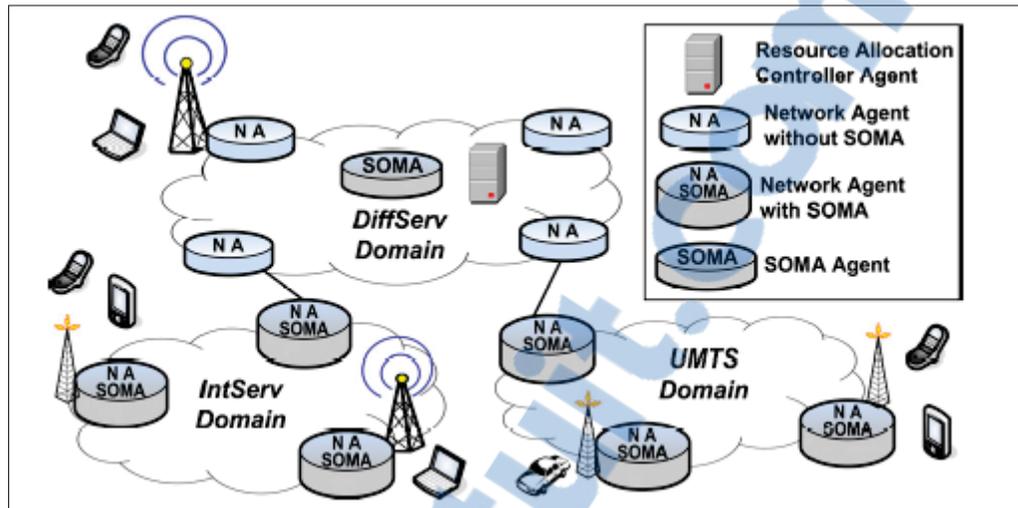


Figure 2.7 Emplacement des SOMA agent
Tirée de Ryu *et al.* (2006)

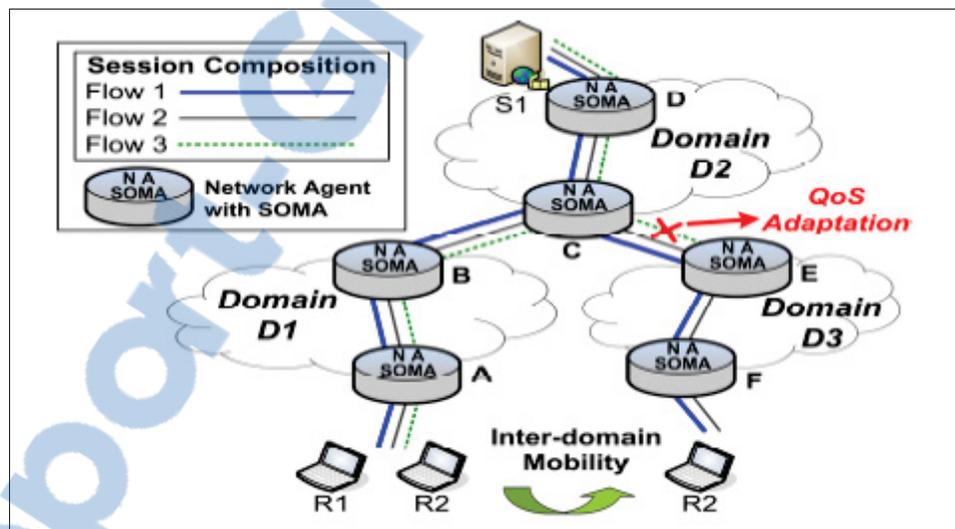


Figure 2.8 Fonctionnement de SOMA
Tirée de Ryu *et al.* (2006)

Les différentes approches de mappage de la QoS étudiées dans ce dernier que ce soit entre les différentes couches ou entre les différents modèles de gestion de QoS (InterServ, DiffServ) est effectué en général de la même manière : mappage entre les classes de services de chaque technologie qui sont opérées par les entités qui se trouvent aux frontières de chaque réseau de différentes technologies. Dans le chapitre suivant, nous présentons le paradigme du nuage informatique et nous analysons les exigences de système IMS dans cet écosystème .

CHAPITRE 3

ANALYSE DES EXIGENCES D'IMS DANS LE NUAGE

3.1 Introduction

Ce chapitre vise à présenter une étude du système IMS déployé dans l'environnement de nuage informatique. Le but ultime de cette analyse est de dégager les exigences de ce système dans le domaine virtuel. Dans une première partie, un tour sur le concept de nuage informatique, ces définitions, ces principales caractéristiques, ainsi que les différents modèles de déploiement seront présentés. Dans une seconde partie, nous exposons les besoins d'IMS pour être déployée dans l'environnement de nuage tout en présentant notre architecture proposée du système IMS déployé dans le nuage.

3.2 Nuage informatique

3.2.1 Définition et concept

Le nuage informatique ou le Cloud Computing en anglais est défini par National Institute of Standards and Technology (NIST) comme l'accès via le réseau, à la demande et en libre-service à des ressources informatiques virtualisées et mutualisées (Mell et Grance, 2011). Ce nouveau concept regroupe l'ensemble des disciplines, technologies et modèles d'entreprises utilisés pour fournir des capacités informatiques (logiciels, plates-formes, matériels) à la manière d'un service à la demande, évolutive et élastique.

3.2.2 Types de nuages

Le nuage informatique peut être déployé selon quatre formes : public, privé, communautaire ou hybride suivant les besoins de l'entreprise. Plusieurs facteurs influent sur le choix de l'entreprise d'un modèle de déploiement. Parmi ces éléments on trouve le coût de déploiement, le type de services offerts, le niveau de sécurité et la qualité de services offerte.

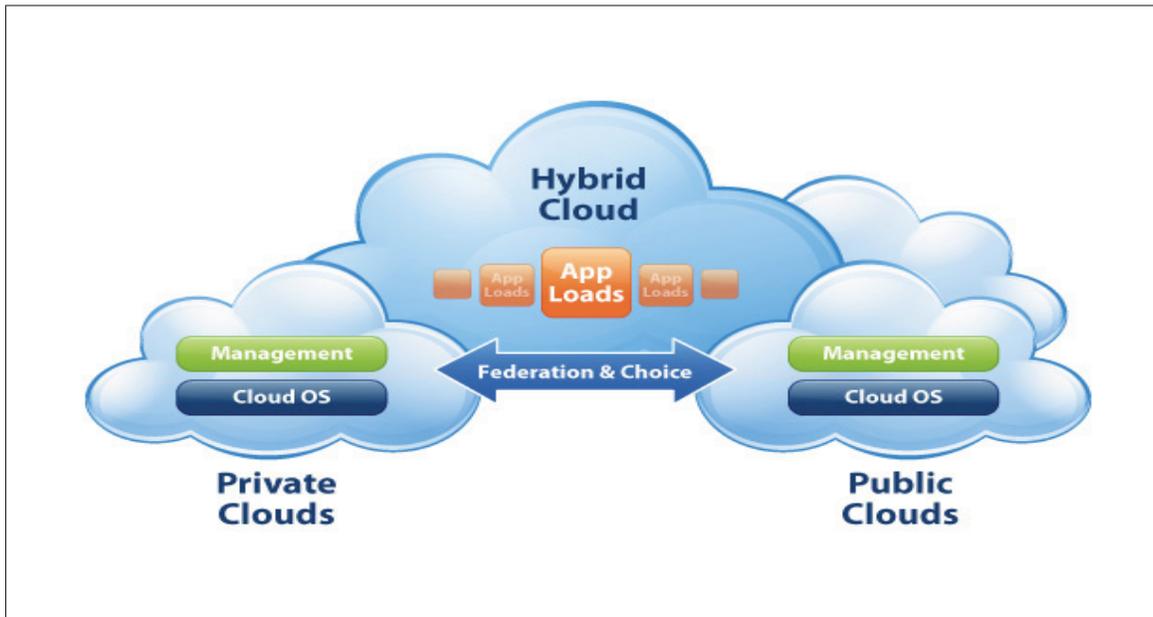


Figure 3.1 Modèle de déploiement
Tirée de WMWare(2011)

- **le nuage public** est hébergé dans les locaux du prestataire de nuage. Il est accessible par Internet et géré par un prestataire externe, les ressources peuvent être partagées entre plusieurs entreprises ;
- **le nuage privé** peut être hébergé, géré et exploité par l'organisation, un tiers, ou une combinaison d'entre eux, et il peut exister sur ou en dehors des locaux. Son infrastructure est provisionnée à l'usage exclusif d'une seule organisation ;
- **le nuage hybride** tel qu'illustré dans la Fig.3.1 dont l'infrastructure de nuage informatique est une composition de deux ou plusieurs infrastructures de nuage distinctes (privé, communautaire ou public) pour une même entreprise, ils sont liés entre eux par des technologies normalisées ou propriétaires ;
- **le nuage communautaire** peut être hébergé, géré et exploité par un ou plusieurs des organismes de la communauté, un tiers, ou une combinaison d'entre eux, et il peut exister sur ou en dehors des locaux. Ce type de nuage est destiné généralement aux professionnels dans le but de travailler de manière collaborative.

3.2.3 Modèle de service

Le nuage informatique comprend trois modèles de service : infrastructure en tant que service, plateforme en tant que service et logiciel en tant que service (Mell et Grance, 2011). Ces trois modèles peuvent être organisés sous forme de quatre couches : matériel, infrastructure, plateforme et logiciel. Une description détaillée de l'architecture du nuage informatique en se basant sur ces modèles est illustré par la Fig.3.2 (Zhang *et al.*, 2010).

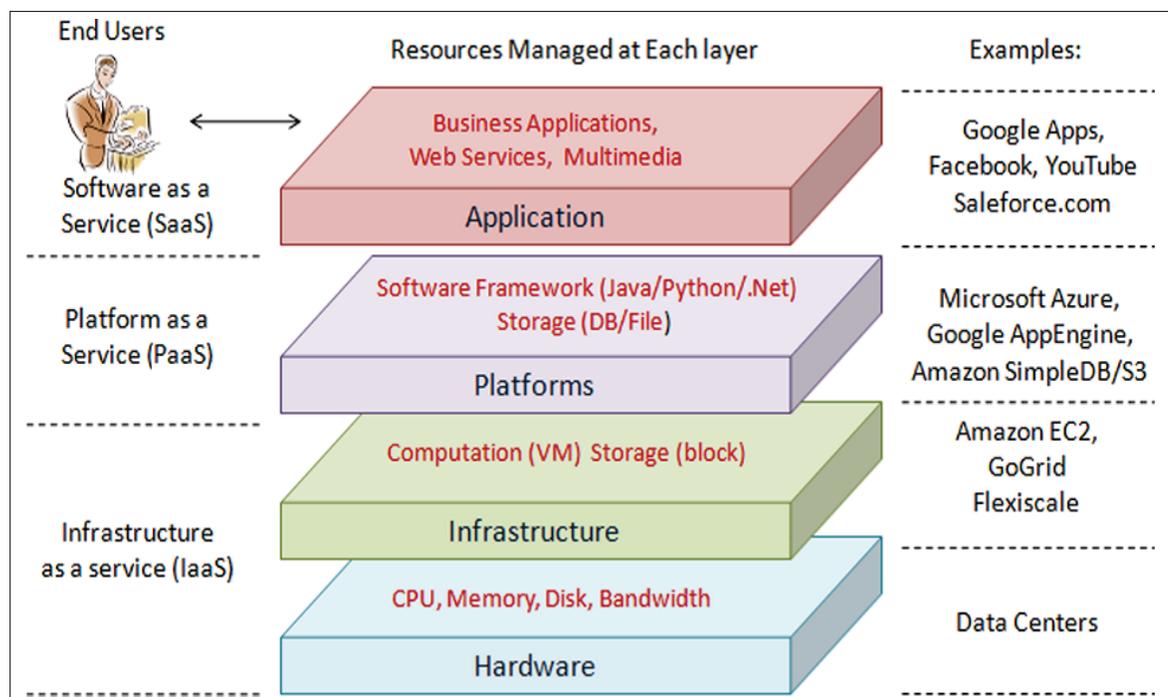


Figure 3.2 Architecture de nuage informatique
Tirée de Zhang *et al.* (2010)

Logiciel en tant que service

Le modèle logiciel en tant que service tel connu par son appellation en anglais software as a service (SaaS) permet à ces clients d'utiliser les applications du fournisseur qui fonctionnent sur une infrastructure de nuage.

Généralement, ces applications sont accessibles à partir de différents périphériques clients à travers une interface client légère, comme un navigateur Web (par exemple, web-based email),

ou une interface de programme. Les consommateurs ne peuvent ni gérer ni contrôler l'infrastructure sous-jacente, y compris les serveurs, systèmes d'exploitation, stockage à l'exception peut-être des limites spécifiques à l'utilisateur et les paramètres de configuration de l'application (Microsoft, 2011). Cependant, l'utilisateur final ne requiert aucune connaissance informatique particulière (Mell et Grance, 2011).

Plateforme en tant que service

Ce modèle est connu en anglais par plateforme as a service (PaaS), permet à ces clients de développer des applications à l'aide d'outils de programmation fournis par le nuage. Google App engine et Windows Azure proposent ce genre de solution. Le client gagne plus de flexibilité, mais dans un environnement technique défini par son fournisseur.

Infrastructure en tant que service

Connu en anglais sous l'appellation infrastructure as a service(IaaS), ce modèle de service met à la disposition des clients les ressources informatiques de base telles que le stockage, les réseaux et éventuellement les systèmes d'exploitation. Amazon propose par exemple ce genre de services.

3.2.4 Caractéristique du nuage informatique

Le nuage informatique possède plusieurs caractéristiques qui sont principalement :

Libre service à la demande (On-demand self-service)

Les clients d'un Cloud peuvent gérer leurs ressources sans interaction avec le fournisseur de service (Mell et Grance, 2011). Ceci permet de mieux gérer les ressources en cas de changement rapide de la charge du travail pour répondre aux besoins.

Accès étendu au réseau (Broad network access)

Les ressources sont disponibles sur le réseau et accessibles via un mécanisme standard qui favorise l'utilisation de plateformes hétérogènes pour les clients lourds ou légers depuis n'importe quel type de périphérique réseau (appareil mobile, station de travail, tablette ou ordinateurs portables). (Microsoft, 2011)

Transparence (Resource pooling)

Les clients n'ont aucun contrôle ni même connaissance de l'emplacement exact, y compris géographiquement, des ressources qui leur rendent le service demandé (Mell et Grance, 2011).

Service mesuré (Measured service)

Le contrôle et l'optimisation des ressources sont gérés automatiquement par le nuage tout en s'appuyant sur une capacité de mesure à un certain niveau d'abstraction approprié pour le type de service (par exemple, le stockage, le traitement, la bande passante et les comptes d'utilisateurs actifs). L'utilisation des ressources peut être surveillée, contrôlée et signalée, à fin d'assurer la transparence à la fois pour le fournisseur et le consommateur du service utilisé (Microsoft, 2011).

Multi locataire (multi tenancy)

Les ressources informatiques du fournisseur sont mises en commun pour desservir multiples consommateurs à l'aide d'un modèle multi locataire, avec différentes ressources physiques et virtuelles assignées dynamiquement et réassignées selon la demande des consommateurs.

Il n'y a un sentiment d'indépendance de l'emplacement que le client n'a généralement aucun contrôle ou la connaissance de l'emplacement exact des ressources fournies, mais peut-être en mesure de spécifier l'emplacement à un plus haut niveau d'abstraction (par exemple pays, État ou centre de données).

Élasticité rapide (Rapid elasticity)

L'élasticité est définie comme la capacité de mettre à l'échelle des ressources à la fois vers le haut et vers le bas si nécessaire. Pour le consommateur, le nuage semble délivrer des ressources de façon infinie, et il peut acheter une puissance de calcul autant qu'il veut sans soucie.

3.3 Exigences du système IMS déployé dans le nuage

L'objectif principal de notre mémoire est de provisionner la qualité de service d'IMS déployé dans l'environnement du nuage informatique. À cette fin, nous avons tout d'abord commencé par étudier les exigences de ce système pour être déployé dans un tel environnement.

Étant donné, les énormes notions qui sont liées au concept de nuage. Dans notre mémoire nous avons focalisé sur deux éléments : la virtualisation et la mise à l'échelle automatique, l'élasticité et le provisionnement. Dans notre travail de recherche, nous avons considéré quelques hypothèses :

- la faisabilité de déploiement du système IMS dans le nuage informatique ;
- les services d'IMS dans le nuage sont fonctionnels.

3.4 Virtualisation du système IMS

3.4.1 Définition

La virtualisation est une technologie apparue, dans les années soixante avec le Control Program/Cambridge Monitor System (CP/CMS) , un système d'exploitation en temps partagé développé par IBM. Cette technique continue à évoluer et converge à d'autres domaines d'utilisation dont le nuage informatique est sur la tête de la liste. En général, la virtualisation est utilisée afin de découpler le système d'exploitation du matériel sous-jacent et pour consolidation des serveurs.

Les objectifs principaux de l'utilisation de cette technique dans les entreprises sont la réduction du coût et temps de la mise en œuvre (Letaifa *et al.*, 2010). Ainsi elle donne la possibilité de faire fonctionner simultanément plusieurs systèmes d'exploitation (Linux, Windows, Mac OS...) sur un même serveur physique à travers l'utilisation de système virtuelle(machine virtuelle) pour chaque système.

Une machine virtuelle dont le terme anglais Virtual machine (VM) peut être vue comme un serveur virtualisé hébergé sur un serveur physique. Pour assurer le fonctionnement de machines virtuelles sur la même machine physique, on fait recours à un hyperviseur, une plateforme de virtualisation qui permet la création et l'exécution de ces VMs. Il existe deux types principaux catégorie d'hyperviseurs :

Hyperviseur type 1

L'hyperviseur type 1 ou encore natif est un logiciel qui s'exécute directement sur une plateforme hardware. ce type d'hyperviseur permet aux systèmes d'exploitation invités de rester relativement près du matériel avec un accès à bas niveau ce qui rend ses performances proches d'un système de manière native.(IBM, 2005). La figure 3.3 montre un système physique avec un hyperviseur de type 1 en cours d'exécution directement sur le matériel du système, et trois systèmes virtuels (VM) en utilisant les ressources virtuelles fournies par l'hyperviseur. Les hyperviseurs de type 1 les plus connu sont VMware ESX/ESXi, XenServer et KVM.

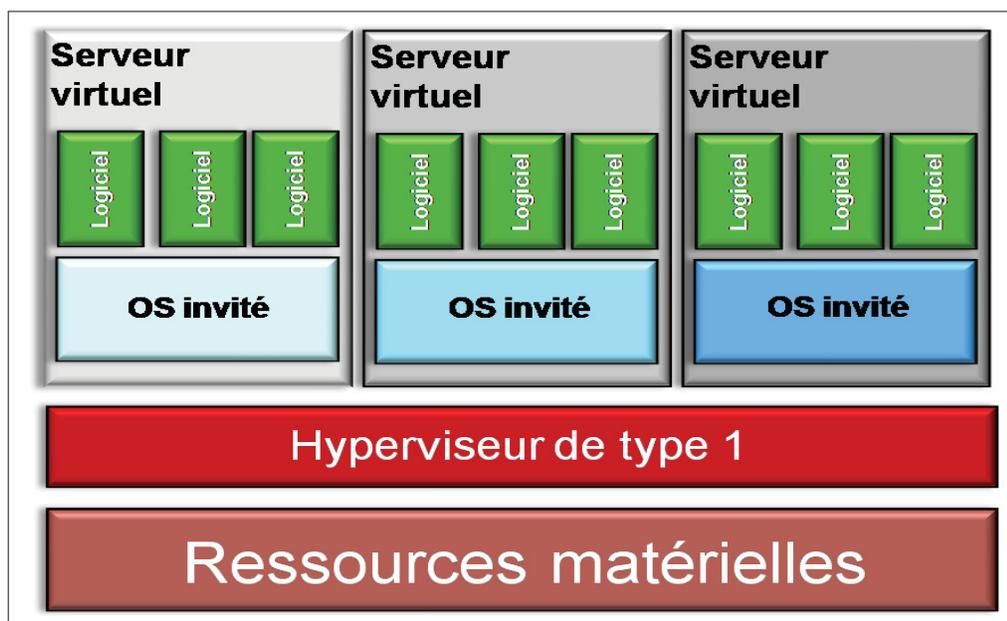


Figure 3.3 Hyperviseur type 1
Tirée de Dominique Revuz (2008)

Hyperviseur type 2

L'hyperviseur type 2 (par exemple VMware GSX Server et VMware Workstation, VirtualBox) s'exécutent sur un système d'exploitation hôte qui fournit des services de virtualisation, tels que le support de périphérique E / S et la gestion de la mémoire. La figure 3.4 montre un système physique avec un hyperviseur de type 2 en cours d'exécution sur un système d'exploitation hôte et trois systèmes virtuels (VM) en utilisant les ressources virtuelles fournies par l'hyperviseur.

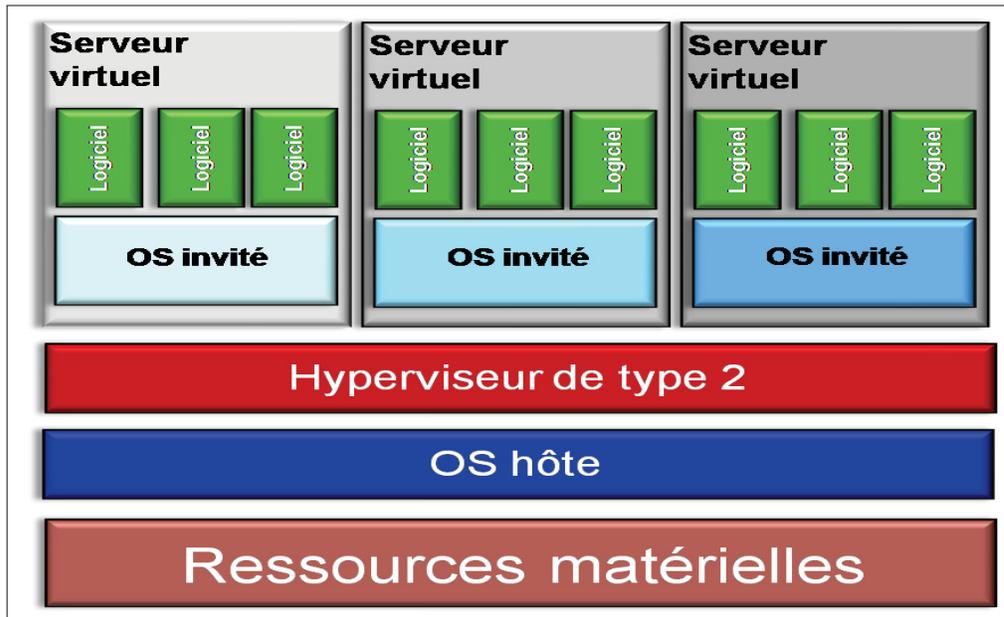


Figure 3.4 Hyperviseur type 2
Tirée de Dominique Revuz (2008)

3.4.2 Technique de virtualisation

Il existe plusieurs techniques de virtualisation dont les plus utilisées sont : l'isolation, la paravirtualisation, la virtualisation complète et le partitionnement matériel :

L'isolation : consiste à mettre en place, sur un même noyau de système d'exploitation, une séparation forte entre différents contextes logiciels. Il s'agit de la technique de virtualisation la plus « légère » qui existe.

La **paravirtualisation** présente aux systèmes d'exploitation une machine générique spéciale, qui requiert donc des interfaces spéciales, intégrées aux systèmes invités sous la forme de drivers ou de modifications du noyau. Il s'agit d'un compromis entre un niveau d'abstraction élevé et un niveau de performance satisfaisant.

La **virtualisation complète** là où l'hyperviseur intercepte de manière transparente tous les appels que le système d'exploitation peut faire aux ressources matérielles, et supporte donc des systèmes invités non-modifiés.

Le **partitionnement matériel** consiste à séparer les ressources matérielles au niveau de la carte mère de la machine. Cette technique est surtout répandue dans les serveurs hauts de gamme, par exemple les domaine logique chez Sun. Elle est assez rare dans le monde x86. Les blades en sont un exemple, mais ils n'offrent pas des fonctionnalités aussi avancées que ce que l'on retrouve sur d'autres architectures matérielles comme SPARC.

3.4.3 Avantage de la virtualisation du système IMS

La virtualisation offre plusieurs avantages :

Consolidation afin de réduire le coût du matériel

La virtualisation permet d'accéder et de gérer efficacement les ressources pour réduire les coûts de gestion des opérations et des systèmes tout en conservant capacités requises. Elle permet aussi de disposer un serveur unique qui fonctionne comme plusieurs serveurs virtuels.

Optimisation de la charge de travail

Virtualisation vous permet de répondre de façon dynamique aux besoins de l'application pour chaque utilisateur. Elle peut accroître l'utilisation des ressources existantes à travers le partage dynamique des pools de ressources.

Flexibilité et réactivité de l'informatique

Virtualisation vous permet d'avoir une vue unique et consolidée avec un accès facile à toutes les ressources disponibles dans le réseau, indépendamment de leur emplacement. Virtualisation vous permet de réduire la gestion de votre environnement en fournissant une émulation pour la compatibilité, améliorée l'interopérabilité, et les fenêtres transparentes de changement.

3.4.4 Virtualisation d'IMS

Pour virtualiser IMS nous avons utilisé la solution de virtualisation complète KVM. C'est solution de virtualisation open source pour Linux sur du matériel x86 et contenant des extensions de virtualisation pour Intel ou AMD. En utilisant KVM, on peut exécuter plusieurs machines

virtuelles fonctionnant avec différents systèmes d'exploitation. Chaque machine virtuelle dispose des ressources virtuelles privées : une carte réseau, disque, carte graphique, etc.

3.5 Mise à l'échelle automatique

La mise à l'échelle ou l'évolutivité est l'une des solutions proposées pour résoudre le problème de manque de ressource lors de la variation de la demande pour les applications multimédias via le mobile. La mise à l'échelle est intégrée dans plusieurs domaines autres que les applications multimédias mobile et fait le sujet de plusieurs travaux de recherche.

Dans notre travail de mémoire nous étudions la mise à l'échelle dans le système IMS déployé dans le nuage informatique. En effet, avec la migration du système IMS vers le nuage informatique la mise à l'échelle dans IMS est l'une des fonctions principales qu'un opérateur veut adopter à son réseau étant donné que l'augmentation de la demande causée par le nombre d'utilisateur continue à augmenter jour après jour.

Le problème concernant l'augmentation de la charge de trafic dans IMS est due au nombre de messages de signalisation circulant dans ce dernier (Bellavista *et al.*, 2009).

Il existe deux approches pour la mise à l'échelle dans IMS :

3.5.1 Mise à l'échelle verticale

La mise à l'échelle verticale ou en anglais *scale-up* est une approche où il y a plus d'investissement en largeur sur des ordinateurs performants pour accommoder la charge demandée. L'entreprise qui déploie ce genre de solution ne prend pas en considération l'architecture des applications. (Varia, 2012). Cette technique de mise à l'échelle est réactive et nécessite un coût d'investissement initial pour sa mise en place. Malgré que cette approche résout le problème de l'insuffisance des ressources en cas d'augmentation de la charge de trafic, elle possède deux inconvénients majeurs :

- il y a un énorme dépense d'investissement (CAPEX) dans l'achat des ordinateurs performants pour répondre aux besoins ;

- perte de client dans le cas où la charge prédit est inférieure à la charge actuelle.

La Fig.3.5 illustre la réaction du système lorsque la charge augmente rapidement avant le déploiement des nouvelles ressources est effectué. (Varia, 2012)

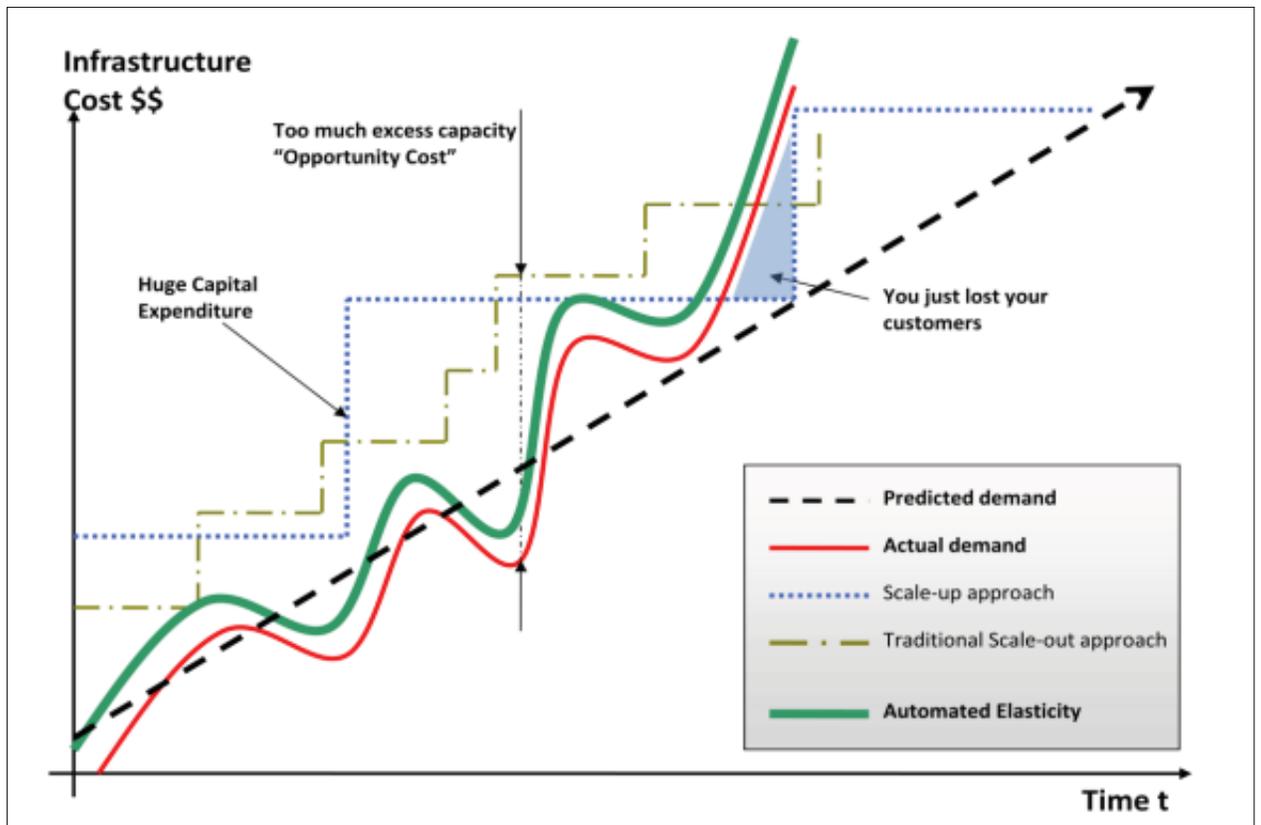


Figure 3.5 Mise à l'échelle et élasticité
Tirée de Varia (2012)

3.5.2 Mise à l'échelle horizontale

La mise à l'échelle horizontale ou en anglais scale-out est l'approche traditionnelle la plus utilisée. Le principe de cette approche est de créer dès le départ une architecture qui peut être horizontalement mise à l'échelle. Identiquement à la mise à l'échelle verticale cette approche est réactive et requiert un coût initial pour la mettre en place. Cependant, contrairement à la mise à l'échelle verticale, l'investissement dans cette technique est effectué sur des petites

morceau, l'entreprise achète plusieurs ordinateurs qui ont des capacités réduites et dont le coût est minimum.

Cette technique de mise à l'échelle est la plus adoptée par les entreprises vu que le coût d'investissement initial pour la mise en place est inférieur à celui de la mise à l'échelle verticale. Le rendement de la mise à l'échelle est pourtant aussi basé sur la prédiction de la charge de trafic qui sera générée en fonction des intervalles de temps. En effet, la prédiction peut ne pas être exacte et cause ainsi des problèmes de perte de clients telle qu'il illustre la Fig.3.5 si la charge de trafic actuelle est supérieure à celle estimée ainsi qu'un excès de capacité.

3.6 Élasticité

L'élasticité est l'une des caractéristiques fondamentales qui a joué le grand rôle dans le succès du nuage informatique. En effet, cette propriété résout les problèmes majeurs qui ne sont pas réglés par la technique de la mise à l'échelle. Son principe de base est d'augmenter et de réduire les ressources informatiques en fonction de la charge actuelle.

Étant donnée la variation de la charge de trafic générée, les ressources requises varient en fonction des intervalles de temps, telle qu'il illustre la Fig.3.5. Cependant, pour bénéficier de cette propriété, l'entreprise doit intégrer ce concept dans son architecture. Pour concevoir une telle architecture, l'entreprise doit prendre en considération plusieurs facteurs qui jouent le rôle dans le succès :

- quel sont les éléments qui seront intégrés dans l'architecture et qui peuvent être élastiques et comment les rendre élastiques ;
- comme notre objectif est d'intégrer l'élasticité dans le système IMS ;
- quel est l'impact de l'élasticité d'un composant sur l'architecture du système global et sur la qualité de service perçue par les utilisateurs.

3.7 Provisionnement des ressource

Le provisionnement dynamique des ressources est l'une des caractéristique principale du nuage informatique. Contrairement au modèle traditionnel qui permet d'ajouter des ressources en cas de pic, dans l'environnement du nuage le provisionnement se fait en fonction de l'état actuel de la demande d'une manière automatique ce qui réduit considérablement la coût d'opération (Zhang *et al.*, 2010).

Pour adopter une technique de provisionnement des ressources certaine paramètres doivent être pris en considération à fin de déterminer la quantité requise pour chaque application. Parmi ces paramètres on trouve : temps de réponse, coûts, augmentation de revenus, tolérance de panne, paramètre de la QoS, réduction de la consommation des ressources et d'énergie. Les techniques de provisionnement des ressources dans le nuage informatique existante peuvent être classées en trois catégories : statique, dynamique et auto-provisionnement.

3.7.1 Provisionnement statique

En général, pour les applications, dont la charge de travail invariable, il est conseillé d'utiliser la technique de provisionnement statique. Avec cette technique, les ressources appropriées sont préparées à l'avance par le fournisseur de services et allouées lors du démarrage de service. Le client est facturé avec un forfait ou sur une base mensuelle.

L'enquête menée par (Nagesh *et al.*, 2014) présente quelques exemples de provisionnement statique, dont "Aneka deadline Driven" une technique de provisionnement utilisée pour les applications scientifiques qui nécessitent une grande puissance de calcul. C'est une plateforme d'applications de nuage qui est capable de faire le provisionnement des ressources qui sont obtenues à partir de diverses sources telles que les nuages publics et privés.

Cette technique permet d'allouer efficacement les ressources, ainsi réduisant le temps d'exécution des applications. Inter Cloud Resource Provisioning (ICRP) (Nelson et Uma, 2012) la où les ressources et les tâches sont décrites sémantiquement et stockés en utilisant l'ontologie des ressources et à l'aide d'un ordonnanceur sémantique et un ensemble de règles d'inférence

des ressources sont affectées. Toutefois, la fonctionnalité et la complexité croissantes dans le nuage informatique ainsi que l'insuffisance des ressources ne peuvent pas être évitée.

3.7.2 Provisionnement dynamique

Dans les cas où la demande d'applications peut changer ou varier, les techniques de provisionnement utilisées sont dynamiques. Avec cette technique, le fournisseur alloue plus de ressources s'ils sont nécessaires et les retire quand elles ne sont pas. Le client est facturé sur une base du paiement à l'usage. (Iqbal *et al.*, 2011) ont proposé un algorithme de provisionnement des ressources qui est adapté pour les applications Web où le temps de réponse est l'un des facteurs importants.

Pour les applications Web, garantissant un temps de réponse moyen est une tâche difficile parce que les modèles de trafic sont très dynamiques et difficiles à prévoir avec précision et aussi en raison de la nature complexe des applications Web multi-tier, il est difficile d'identifier les goulots d'étranglement et de les résoudre automatiquement.

(Iqbal *et al.*, 2011) propose un prototype de travail de détection automatique et de résolution des goulots d'étranglement pour les applications multi-tier hébergés dans le nuage. Ce qui améliore le temps de réponse et aussi identifie les ressources provisionnées. Étant donné que la gestion des ressources des VMs est une tâche difficile, une approche appelée Elastic Application Container (EAC) (He *et al.*, 2012) est utilisé pour le provisionnement des ressources.

3.7.3 Auto-provisionnement par l'utilisateur

Une troisième catégorie est l'auto-provisionnement ou également connu sous le libre-service de nuage, là où le client achète des ressources du fournisseur de nuage à travers un interface web. Les ressources du fournisseur sont disponibles à l'utilisation par le client en quelques heures, si ce n'est pas quelques minutes. Plusieurs fournisseurs de service de nuage offrent ce type de provisionnement (e.g Amazon).

3.8 Design et architecture d'IMS dans le nuage

Après avoir dégagé les exigences du système IMS pour être déployé dans le nuage informatique. Dans cette section nous présentons le design et l'architecture de notre système IMS dans l'écosystème de nuage informatique. Bien que, l'architecture complète du réseau d'un opéra-

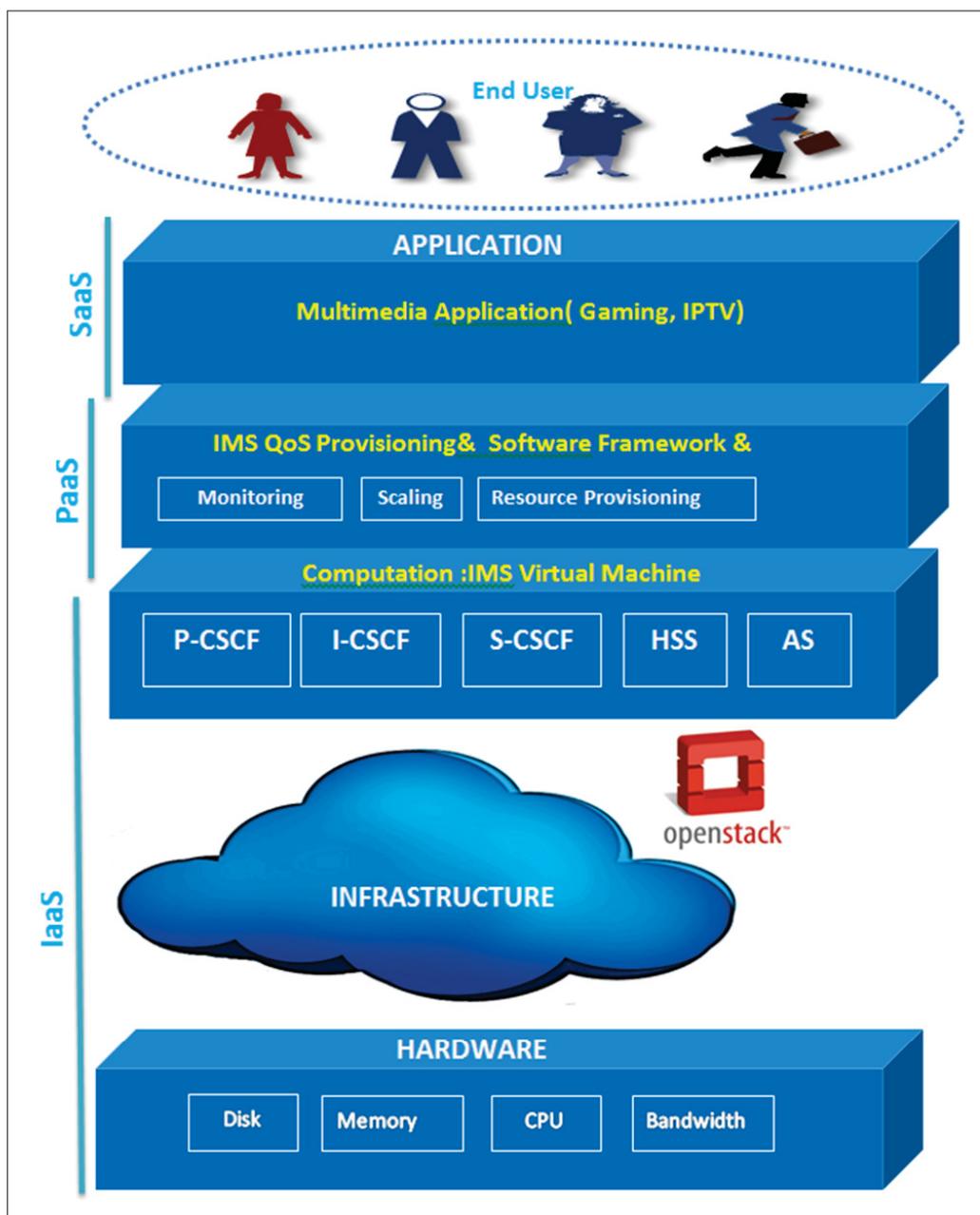


Figure 3.6 Architecture proposé d'IMS dans le nuage

teur téléphonique est constituée de trois parties : la partie réseau d'accès, la partie réseau cœur composé de la couche de transport et la couche IMS et la partie service et application. Dans ce mémoire nous focalisons sur les composantes d'IMS de la partie cœur d'un réseau d'opérateur. Dans l'environnement du nuage informatique l'architecture d'IMS peut être vue comme une décomposition de trois couches. Le design et l'architecture d'IMS dans le nuage est illustré par la Fig.3.6.

3.9 Conclusion

Dans ce chapitre nous avons analysé les exigences du système IMS pour être déployé dans un environnement de nuage informatique. Étant donnée, les concepts liés au nuage informatique sont nombreux, dans ce mémoire nous avons mis l'accent sur les notions de la virtualisation, l'évolutivité et le provisionnement. Dans le prochain chapitre nous focaliserons sur le principe de provisionnement de la Qos d'IMS dans le nuage.

CHAPITRE 4

PROVISIONNEMENT DE LA QOS D'IMS DANS LE NUAGE

4.1 Introduction

Étant donné, que notre travail de recherche consiste principalement au déploiement et au provisionnement de la qualité de service d'IMS dans le nuage informatique. Dans ce chapitre, nous présentons en premier le choix des solutions technologiques. Dans la section 4.2, nous présentons les différentes solutions libres d'IMS existantes, dans la section 4.2.3 nous décrivons les différents composants de l'infrastructure « OpenStack » qu'on a utilisé pour déployer ces différentes architectures système. En second lieu, nous donnons une description des métriques de performance analysé dans cet mémoire et nous présentons les différentes méthodes utilisées qui visent à provisionner les ressources dans le système IMS. Nous commençons en premier lieu par décrire les méthodologies de provisionnement des ressources dans les systèmes traditionnels, ensuite nous étudierons les différentes techniques utilisées dans l'environnement de nuage informatique. Une comparaison entre les différentes méthodes existantes est effectuée. À la lumière de l'étude des différentes solutions de provisionnement existantes, nous proposons la méthodologie qu'on a adopté pour provisionner les ressources de nos systèmes déployés dans le nuage.

4.2 Les systèmes IMS existants

Depuis sa standardisation en 2005, plusieurs systèmes ont été commercialisés par les fournisseurs de télécommunications comme celui d'Ericsson, Motorola, Alcatel-Lucent. Quant aux solutions libres :il existe deux solutions majeures qui sont Open Source IMS Core (FOKUS, 2006) et Clearwater IMS (Metaswitch, 2013). Les deux systèmes ont été développés pour être utilisés comme tests. Le système IMS commercial le plus vendu est celui de la compagnie Ericsson qui a débuté en 2006 avec l'opérateur Vodafone dans plusieurs pays comme le Japon, la République tchèque et Allemagne. Aujourd'hui, l'architecture IMS d'Ericsson est la plus

déployée dans le monde, dans plus de 180 pays comme Network in Taïwan, TELE Greenland, Télécom Italia Mobile, China Beijing Netcom, etc.

Au Canada, plusieurs opérateurs téléphoniques ont mis en place l'architecture IMS d'Ericsson dont le plus récent est le contrat qui a été signé en 2013 avec TELUS mobile. Le déploiement de ce système multimédia permettra à TELUS d'offrir les services convergents avancés dans la voix, les données et des appels vidéo sur les deux réseaux filaires et sans fil aux clients partout au Canada. Dans les sections 4.2.2 et 4.2.1 nous présentons les deux solutions IMS libres existantes qui sont déployées dans le cadre de notre mémoire.

4.2.1 Open Source IMS core

Le projet d'open source IMS core a été lancé en 2006, développé par l'université FOKUS (Fraunhofer Institute for Open Communication System), basé à Berlin-Allemagne là où plus de 80 organisations ont été impliquées dans ce projet dont 57 sont des universités. Les objectifs fondamentaux du projet sont les tests d'interopérabilité, l'analyse comparative et le prototypage d'extension de technologie des applications multimédia innovantes pour le réseau de télécommunications émergentes.

Cette solution a été adoptée par plusieurs opérateurs et fournisseurs de télécommunications dans le monde comme une banque d'essais pour tester les fonctionnalités de système IMS avec l'intégration des nouveaux services sur IP comme la télévision sur IP (IPTV). Open source IMS Core est formé par l'ensemble des éléments de base d'une architecture IMS définie dans les réseaux de nouvelle génération et telle qu'indiquée dans 3GPP, 3GPP2, ETSI TISPAN et l'initiative PacketCable. Les composants de base de l'architecture d'open source IMS Core sont illustrés par la Fig.4.1 (FOKUS, 2006).

Toutes les entités de ce framework sont basées sur des logiciels libres. Ainsi, open source IMS CSCFs est composé de trois éléments : le Proxy (P-CSCF), Interrogating (I-CSCF) et Serving (S-CSCF) Call Session Control Functions. Ces trois éléments sont des extensions de SIP Ex-

press Router (SER) qui ont été testés avec des produits commerciaux pour l'interopérabilité.

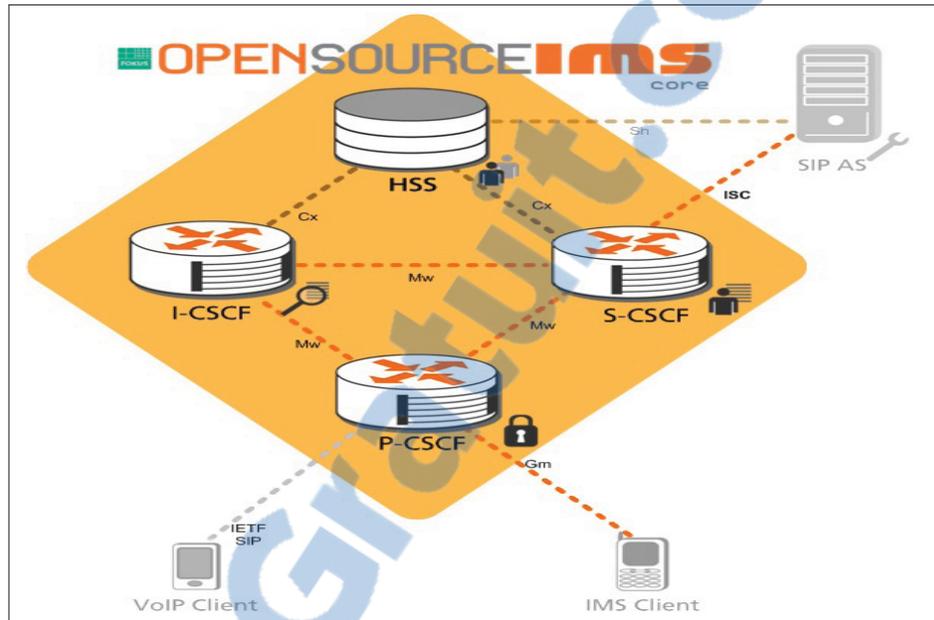


Figure 4.1 Architecture d'Open Source IMS Core
Tirée de FOKUS (2006)

La base de données d'open source IMS Core, FHoSS (FOKUS Home Subscriber Server) est basé sur MySQL la Fig4.2 montre les composantes de cette base de données. La logique du FHoSS d'application est entièrement écrite avec Java en utilisant le conteneur de servlet du logiciel libre Tomcat. La composante principale de cette base de données est l'utilisateur maître (Master) à base de MySQL, supportant des entités de réseaux qui gèrent les communications sur IMS. Plus précisément, le FohSS offre les fonctions suivantes :

- stocke et gère les informations relatives au profil d'abonnés ;
- génère des données pour l'authentification et l'autorisation des utilisateurs ;
- maintenir les tables de routage sous forme de répertoires de localisation de l'abonné ;
- fournir des informations sur les points de service de l'abonné ;
- gestion de déclencheur de service et de l'information de base de l'orchestration ;

- enregistre les informations de facturation.

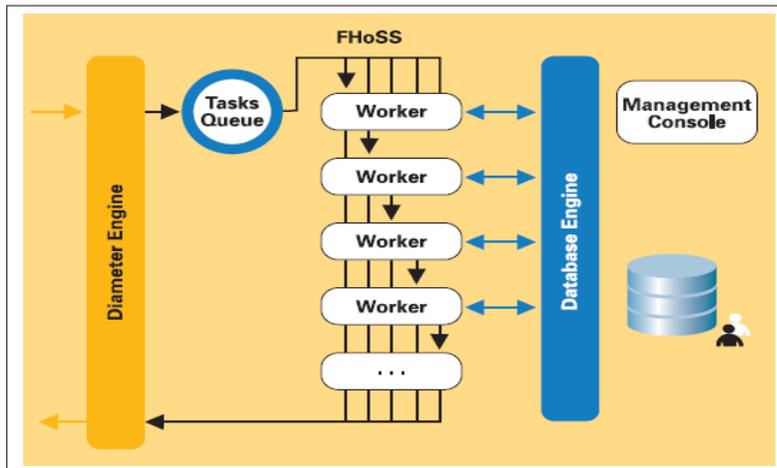


Figure 4.2 FOKUS Home Subscriber Server
Tirée de FOKUS (2009)

FhoSS est configuré pour supporter plusieurs millions d’abonnés et peut servir de multiples centaines d’utilisateurs enregistrés par seconde.

4.2.2 Clearwater IMS core

Le projet de Clearwater IMS est sponsorisé par la compagnie Metaswitch Networks (Networks, 2014). C’est une solution libre publiée en mai 2013. Clearwater IMS suit les principes d’architecture IMS et supporte toutes les interfaces clés standardisées attendues d’un cœur de réseau IMS. À la différence des implémentations traditionnelles de IMS, Clearwater IMS a été conçu dès le départ pour être optimisé pour le déploiement dans les environnements virtualisés et dans le nuage informatique d’où la particularité de cette solution par rapport à open source IMS Core.

Le framework de Clearwater IMS s’appuie sur des patrons de conception et de composants logiciels libres ce qui donne la possibilité à l’évolutivité et la rentabilité de cette solution. Tous les composants de Clearwater permettent le passage à l’échelle horizontale en utilisant un équilibrage de charge simple. L’état à long terme ne sont pas stockées sur les nœuds du cluster,

en évitant la nécessité des systèmes de réplication de données complexes. Au lieu de cela, l'état à long terme est stocké dans des nœuds de service back-end en utilisant les technologies de stockage nuage optimisé tel que Cassandra.

L'interface entre les composants SIP frontaux et les services back-end utilisent «RESTFULL» API de services Web. Ainsi, que l'interface entre les différents composants utilise le regroupement de connexion avec recyclage statistique de connexions pour assurer que la charge est répartie uniformément comme des nœuds sont ajoutés et supprimés dans chaque couche.

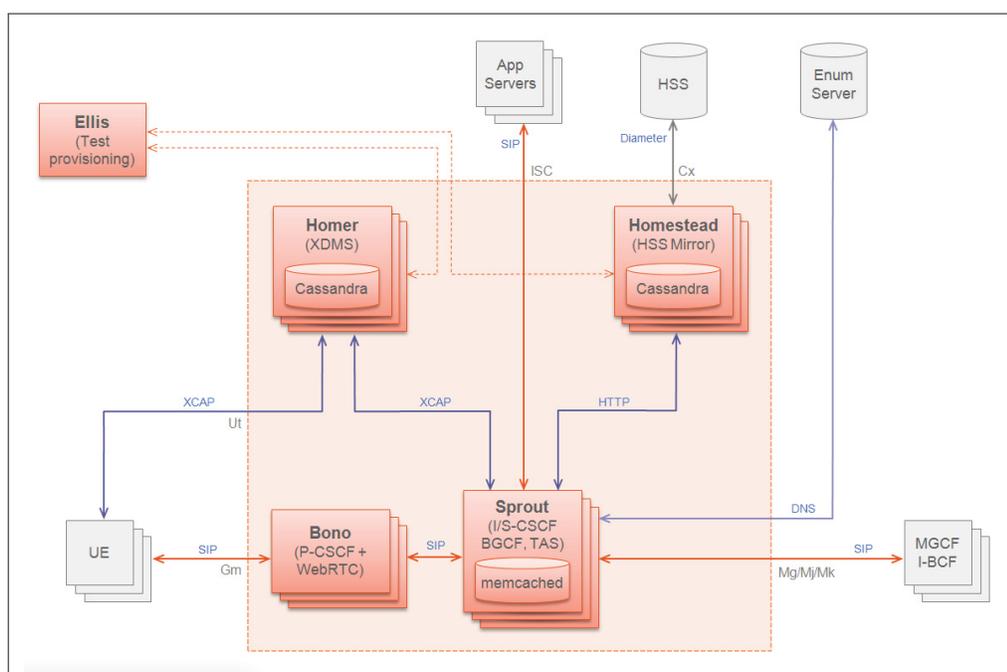


Figure 4.3 Architecture de Clearwater IMS
Tirée de Metaswitch (2013)

Bono (Edge Proxy)

Le nœud Bono forme un proxy SIP horizontalement évolutive offrant à la fois une interface compatible SIP IMS Gm (P-CSCF) et une interface de WebRTC aux clients. Les requêtes clientes sont partagées sur les nœuds on utilisant un équilibrage de charge. Ce nœud constitue le premier point de contact du client avec le système Clearwater , y compris le soutien pour les divers mécanismes de traduction d'adresse réseau NAT (Network Address Translation) pour

les différents nœuds traversée . Un client est donc ancré à un nœud Bono particulier pour la durée de son enregistrement , mais peut se déplacer vers un autre nœud Bono si la connexion ou le client échoue. Les clients peuvent se connecter à Bono en utilisant le protocole SIP/UDP ou SIP/TCP. Bono prend en charge tout client WebRTC qui effectue la signalisation d'établissement d'appel utilisant le protocole SIP sur WebSocket. Les nœuds Bono ne sont pas nécessaires si Clearwater est déployé avec un tiers P-CSCF ou SBC (Session Border Controller) mise en œuvre de P-CSCF.

Sprout (routeur SIP)

Les nœuds de Sprout agissent comme registraire SIP horizontalement évolutive qui gère l'authentification des clients et l'interface ISC pour les serveurs d'applications ainsi que l'interfaçage avec le reste des serveurs tel que Homestead et Homer. Les nœuds de Sprout contiennent également une option pour construire un serveur d'application MMTel. Le cluster Sprout comprend un cluster redondant Memcached pour stocker les données d'enregistrement des clients avec l'état à long terme. Les Transactions SIP sont équilibrées sur le cluster Sprout et il n'y a pas d'association à long terme entre un client et un nœud de Sprout particulier . Sprout utilise des interfaces de services Web fournis par Homestead et Homer pour récupérer les informations de configuration du HSS telles que les données d'authentification et du profil utilisateur ainsi que les paramètres de service MMTel.

Homestead (HSS Mirror)

Homestead fournit une interface de services Web à Sprout pour récupérer des informations relatives au profil d'utilisateur comme les informations d'authentification. Cette composante gère directement les données à travers l'interface web ou les extraire à partir d'un HSS à travers l'interface Cx conforme à IMS. Les nœuds de Homestead fonctionnent comme un cluster à l'aide de Cassandra pour stocker les données. Dans l'architecture d'IMS , la fonction de miroir HSS est considérée comme partie des composants I-CSCF et S-CSCF , cependant dans Clearwater IMS les fonctions des deux composants I-CSCF et S-CSCF sont implémentée avec une combinaison de cluser de Sprout et Homestead.

Homer (XDMS)

Homer est un serveur de management des documents XML standard, en anglais XML Document Management Serve(XDMS) permet de stocker les paramètres du service MMTel dans des documents XML pour chaque utilisateur du système. Les documents sont créés , lus , mis à jour et supprimés à l'aide d'une interface XCAP (XML Configuration Access Protocol) standard. Comme Homestead , les nœuds Homer sont implémentés comme un cluster à l'aide Cassandra.

Ellis

Ellis est un portail simple d'approvisionnement qui fournit une auto-inscription des utilisateurs, gestion de mot de passe et le contrôle des paramètres de service MMTel. Ellis n'est pas destiné pour être une partie de la production des déploiements Clearwater .Et la mise à l'échelle horizontalement n'est pas facile pour cette entité en raison de l'utilisation de MySQL.

4.2.3 Infrastructure en tant que service libre : Openstack

OpenStack est une collaboration mondiale de développeurs et des technologues de nuage informatique produisant une plateforme de nuage informatique open source pour les nuages publics et privés. Le projet de Openstack est initié par Rackspace et la NASA. Aujourd'hui, il est devenu une grande communauté mondiale de développeurs de logiciels qui collaborent sur un système standard qui permet aux entreprises de créer et d'offrir des services de nuage informatique fonctionnant sur du matériel standard.

L'objectif de ce projet est de fournir des solutions pour tous les types de nuages informatiques en étant simple à mettre en œuvre, hautement évolutive et riche en fonctionnalités. La technologie consiste en une série de projets inter-liés qui fournissent différents composants pour une solution d'infrastructure de nuage.

Comme précédemment mentionné OpenStack est un logiciel libre qui permet la construction de Cloud privé et public et joue le rôle d'une infrastructure de service hautement évolutif. Il est composé d'une série de logiciels et un projet libre qui est maintenu par la fondation de Openstack. L'architecture simplifiée d'openstack illustré par la figure 4.4 montre les principales

composantes de ce IaaS. Ces entités peuvent être classées en trois catégories : compute, réseaux et stockage.

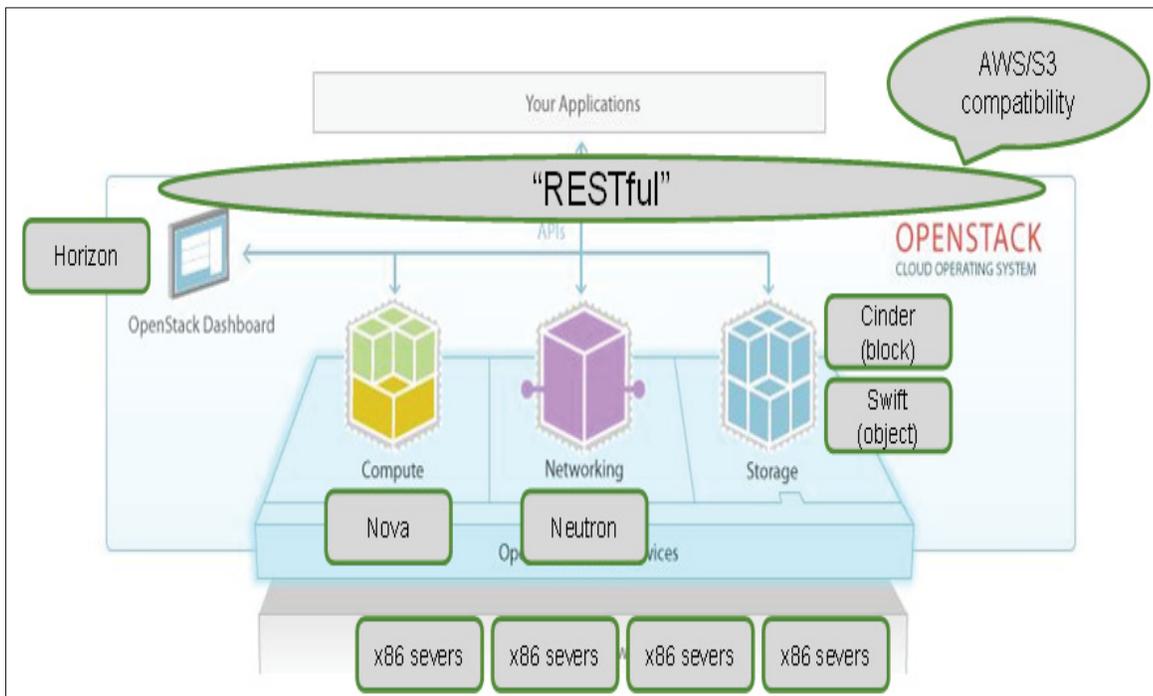


Figure 4.4 Architecture simplifiée d'OpenStack
Tirée de Rackspace (2013)

Openstack Compute

Openstack compute est composé de OpenStack Compute (Nova) et OpenStack Image service (Glance) qui permettent de fournir et de gérer des grands réseaux de machines virtuelles.(Fig.4.4) L'architecture de Compute est conçue pour évoluer horizontalement sur du matériel standard, permettant aux entreprises une économie calculée. Souvent OpenStack Compute est déployé avec l'un de multiples hyperviseurs pris en charge dans un environnement virtualisé. KVM et XenServer sont les choix appréciés par les administrateurs pour la technologie d'hyperviseur et recommandés pour la plupart des cas d'utilisation.

Réseau d'OpenStack (Networking)

Le réseau d'openstack est un système évolutif axé sur l'API pour la gestion des réseaux et

des adresses IP qui peut être utilisé par des administrateurs et des utilisateurs, dont le but d'augmenter le nombre actif des centres de données informatiques . Le réseau d'openStack garantit que le réseau ne serait pas une cause d'étranglement ainsi qu'un facteur qui limite le nombre de déploiements en nuage et offre à l' utilisateur un libre service. Il fournit des modèles de réseaux flexibles pour s'adapter aux besoins des différentes applications ou groupes d'utilisateurs. Les modèles standards notamment les réseaux VLAN pour la séparation des serveurs et du trafic réseau.

Ce réseau gère les adresses IP, ainsi il permet l'allocation des adresses IP statiques ou à travers le DHCP. Ainsi la notion des adresses IP flottantes (floating IP address) autorise le trafic réseau d'être redirigé dynamiquement à n'importe quelle ressource de calcul pendant la maintenance ou en cas de panne. Les utilisateurs peuvent créer leurs propres réseaux, contrôler le trafic et connecter des serveurs et des périphériques à un ou plusieurs réseaux. Alors que les administrateurs peuvent profiter de la technologie (SDN) qui intègre OpenFlow pour autoriser des niveaux élevés de multi-locataires (multi-tenant) d'où une grande évolutivité. Il est doté d'une structure extensible permettant l'ajout d'autres services des réseaux supplémentaires, tels que les systèmes de détection d'intrusion (IDS), l'équilibrage de charge(load balancing),le pare-feu et les réseaux privés virtuels (VPN) à être déployés et gérés.

Stockage d'Opensatck (Openstack Storage)

Le Stockage est constitué de OpenStack Object Storage (Swift) et OpenStack Block Storage (Cinder) (Fig.4.4).En plus de la technologie de stockage de la classe d'entreprise traditionnelle, de nombreuses organisations aujourd'hui ont une variété de besoins de stockage avec des performances variables et des exigences de prix. OpenStack supporte deux types de stockage :le stockage de type objets et le stockage de type bloc, avec de nombreuses options de déploiement pour chaque cas d'utilisation.

Le stockage de type objet est caractérisé par un prix réduit et une évolutivité d'espace de stockage. Il fournit une plate-forme de stockage entièrement distribuée qui offre Plusieurs points d'accès et qui peuvent d'être intégré directement dans les applications ou utilisé pour la sauvegarde, l'archivage et la conservation des données à long terme. Le stockage de type bloc

permet aux périphériques d'un bloc à être exposés et connectés aux instances de calcul, dont le but d'avoir un stockage étendu, une meilleure performance et une intégration avec les plateformes de stockage d'entreprise, tel que NetApp, Nexenta et SolidFire.

Keystone

Le composant Keystone est chargé de la gestion des utilisateurs et des services.

Tableau de bord (Horizon)

Le tableau de bord d'OpenStack permet à l'utilisateur et aux administrateurs l'accès et la gestion des ressources d'informatiques par une interface Web. La conception extensible facilite l'exposition et le branchement des produits et des services de tiers, tels que la facturation, le suivi et les outils de gestion supplémentaires.

Le tableau de bord attire également les fournisseurs de services et d'autres vendeurs commerciaux qui veulent s'en servir. Le tableau de bord est juste un moyen d'interagir avec les ressources de OpenStack. Les développeurs peuvent automatiser l'accès ou construire des outils pour gérer leurs ressources en utilisant l'API natif OpenStack ou l'API de compatibilité EC2. En tant qu'administrateur du nuage, le tableau de bord fournit une vue globale de l'espace et de l'état des ressources de nuage informatique, il est également possible de créer des utilisateurs et des projets, ainsi d'affecter des utilisateurs à des projets et définir une limite pour les ressources attribuées à chaque projet.

4.3 Métriques de performance d'IMS

Il existe plusieurs procédures dans le standard IMS, dans notre analyse nous avons mis l'accent sur trois principales procédures qui sont l'enregistrement et l'établissement d'une session audio et donnée. Les demandes d'enregistrement et d'établissement d'une session sont similaires dans les deux solutions open source IMS. Nous avons analysé aussi le média audio, RTP

4.3.1 Demande d'enregistrement

Pour vous connecter à un système IMS, l'utilisateur final doit d'abord s'attacher au réseau : ce processus est appelé «procédure d'enregistrement» qui est principalement basé sur l'échange de messages de signalisation SIP. La procédure d'enregistrement dans IMS est illustrée par Fig.4.5, l'UE (User Equipment) envoie une première demande d'enregistrement au proxy P-CSCF à l'aide d'un message «REGISTER». Le P-CSCF doit d'abord vérifier l'identité de

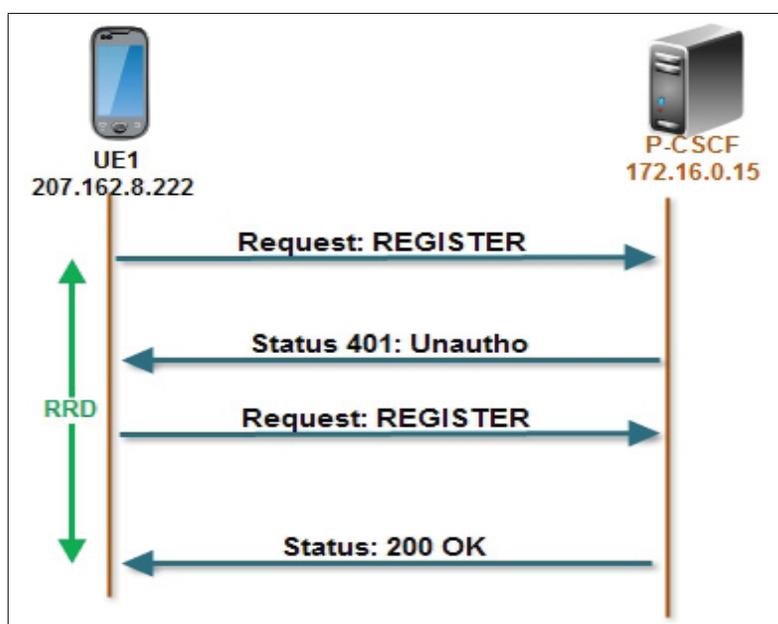


Figure 4.5 Procédure d'enregistrement

l'utilisateur final en fonction de son profil stocké dans la base de données à l'aide 3GPP-Alias et en communication avec les autres entités S-SCSF et I-CSCF. Après cette vérification, l'UE envoie un second message d'enregistrement «REGISTER» qui sera acquitté par «200 OK» qui précise que la demande d'enregistrement est réussie et que l'utilisateur peut établir une session voix, vidéo, ou données.

Délai d'enregistrement (RRD)

Le RRD (Registration Request Delay) dans une tentative d'enregistrement réussie, sujet de 4.1 est définie comme l'intervalle de temps entre le moment où le premier bit du «REGISTER»

initial jusqu'à ce que le dernier bit du message réponse «200 OK» reçue indiquant la tentative d'enregistrement a réussi.

$$RRD = \text{Time of Final Response} - \text{Time of REGISTER Request} \quad (4.1)$$

4.3.2 Demande d'établissement d'une session audio

Une fois l'utilisateur est enregistré au sein du réseau, il peut avoir un access à n'importe quel services inclus dans son abonnement avec l'opérateur. La voix est le service principale offerte par l'opérateur téléphonique. La procédure d'établissement d'une session audio dans IMS est illustrée par Fig.4.6. D'abord, une demande «INVITE» est envoyée par l'UE1 pour lancer un appel téléphonique avec UE2. Ensuite, les deux participants (Alice et Bob) négocient les

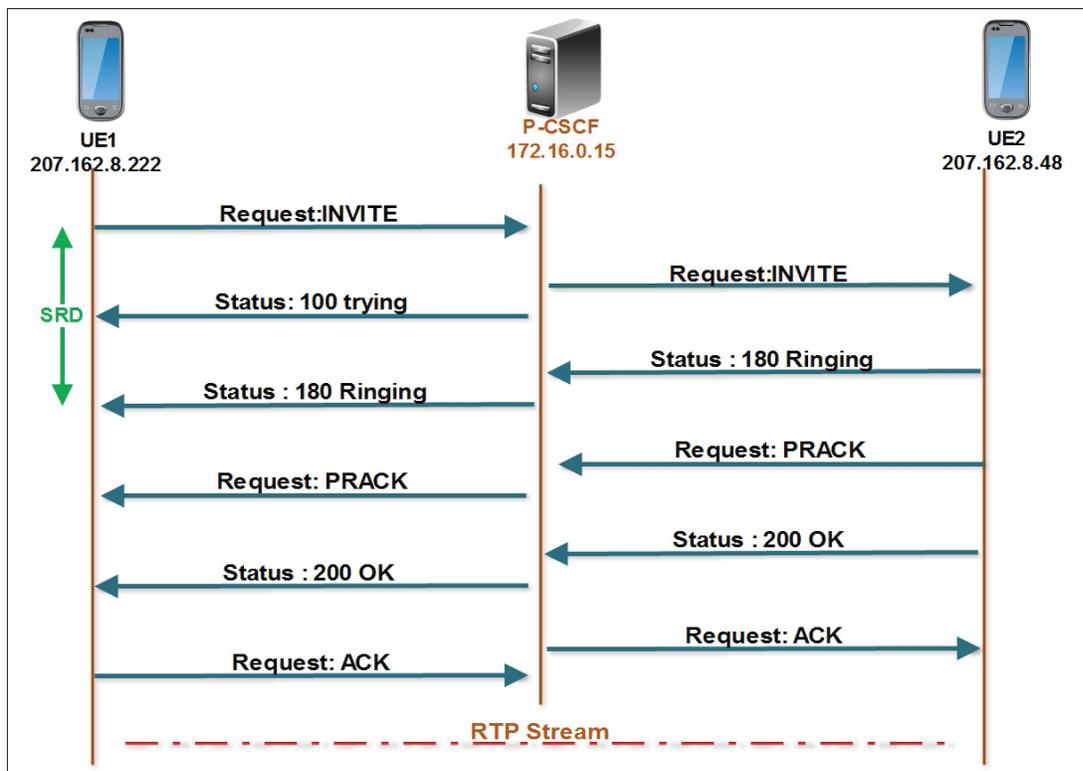


Figure 4.6 Établissement d'une session audio

paramètres de session (de qualité, d'encodage et de médias, les capacités du terminal à être utilisées), ainsi que la réservation de ressources par les messages «SIP/SDP».

Lorsque tous les paramètres de la session sont négociés et mis en place, les deux UE échangent un message «200 OK», puis ils échangent un message «ACK» qui indique que la création de la session vocale a réussi. À cet instant la session est déjà établie entre les deux clients, le trafic audio est acheminé entre eux en utilisant RTP comme protocole de transport.

Remarque : cette procédure est la même lorsque l'utilisateur veut établir un appel vidéo, la seule différence sera au niveau du message SIP/SDP la ou les deux participants négocient en plus les paramètres de l'audio et celle de la vidéo. Au niveau RTP le flux audio et vidéo sont transmis sur deux connexions différentes.

Délai d'établissement d'une session audio(SRD)

Le SRD (Session Request Delay) d'une session de réussite est défini par l'UIT-T est l'intervalle du temps entre le moment où l'UE envoie un message «INVITE» et le moment où il reçoit une réponse «Ringing» illustrée dans la Fig.4.6 et sujet de 4.2.

$$SRD = \text{Time of Status Indicative Response} - \text{Time of INVITE} \quad (4.2)$$

Délai du flux RTP

Le délai $D(i,j)$ entre deux paquets i et j pour le flux RTP peut être exprimé comme suit :

$$D(i, j) = (R_j - R_i) - (S_j - S_i) = (R_j - S_j) - (R_i - S_i) \quad (4.3)$$

Là où S_i est le Timestamp du paquet i et R_i est l'heure d'arrivée du paquet i .

Gigue du flux RTP (Inter arrival jitter)

La gigue est la variance statistique du délai de transmission. Pour les applications téléphoniques la gigue est la variation de délai entre l'émission et l'écoute de la voix. La présence d'une gigue élevée cause une dégradation de la QoS. Ceci est due aux deux facteurs qui sont principalement

les paquets perdus et le délai pour chaque paquet. RTP est utilisé comme protocole de transport pour la téléphonie en IMS donc il faut bien surveiller cette métrique. Le calcul de la gigue est effectué selon RFC3550 et exprimé comme suit.

$$J(i) = J(i - 1) + (|D(i - 1, i)| - J(i - 1))/16 \quad (4.4)$$

Bande passante du flux RTP(BW)

La bande passante au niveau IP pour le flux RTP est égale à la somme de tous les octets, y compris les têtes IP (20 octets) et UDP (8 octets) de tous les paquets du flux RTP au cours de la dernière seconde.

4.3.3 Demande d'établissement d'une session donnée

La messagerie instantanée (IM) se réfère à la transmission des messages entre les utilisateurs en temps quasi réel. Ces messages sont souvent utilisés dans un mode conversationnel dont le transfert est assez rapide pour les participants de maintenir une conversation interactive.

Pour établir une session de données entre les deux abonnées l'UE1 envoie un message de demande «MESSAGE» contenant le message instantané à envoyer. Le message est reçu par UE2, affiché une réponse «200 OK» est généré et envoyé au UE1. Cette procédure est illustrée dans la Fig.4.7.

Délai d'établissement d'une session donnée :

Le délai d'une session données est défini par l'intervalle du temps entre le moment où l'UE envoie la requête «MESSAGE» et le moment de la réception du dernier bit du message réponse «200 OK» reçue indiquant que la session a été établi. L'établissement d'une session donnée est illustrée dans la Fig.4.7. Nous avons définie IMD (Instant Messaging Delay), sujet de 4.5 comme le delai d'établissement d'une session donné.

$$IMD = TimeofStatusIndicativeResponse - TimeofMESSAGE \quad (4.5)$$

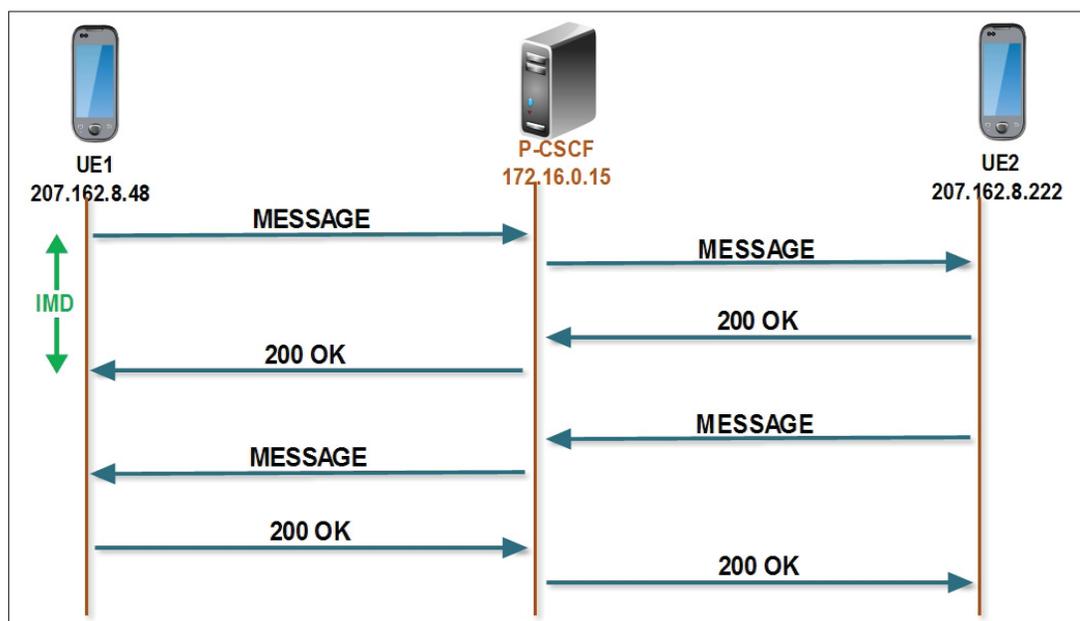


Figure 4.7 Établissement d'une session donnée

4.4 Provisionnement de la QoS d'IMS dans le nuage informatique

Lors d'un abonnement téléphonique, un accord de service est établi entre l'opérateur et l'utilisateur. Le niveau de la QoS offert à l'abonné est défini dans un contrat SLA dont les paramètres sont la disponibilité, le débit, le temps de réponse, la sécurité, la fiabilité, le délai et le taux de perte, etc. Ainsi, pour ne pas violer ce contrat, les paramètres convenus entre les deux parties doivent être respectés par l'opérateur.

Cependant, le manque des ressources informatiques suffisantes peut affecter le fonctionnement du système qui génère de rejets des requêtes clientes. Ceci implique une violation du contrat SLA et la non-satisfaction du client du service fourni. Effectivement, l'opérateur doit s'assurer que les ressources informatiques disponibles sont suffisantes pour le bon fonctionnement du système. L'une des méthodes utilisées pour remédier à ce problème est le provisionnement des ressources qui consiste principalement à la sélection, le déploiement des ressources (par exemple, CPU, stockage et mémoire) pour garantir une meilleure performance pour les applications.

Le processus de provisionnement dans les environnements distribués est une tâche complexe, il est considéré comme un problème NP-difficile. En effet, il est important de trouver la meilleure configuration logicielle et matérielle pour assurer le bon fonctionnement du système et atteindre les objectifs de la qualité de service demandés sans violer le contrat au SLA convenu avec les utilisateurs finaux.

Toutefois, une mauvaise estimation des besoins en ressources peut engendrer une sous-utilisation ou sur-utilisation des ressources. Il est encore plus compliqué si un comportement inattendu est apparu lors de l'exécution des différentes tâches va empêcher le bon fonctionnement du système et la fourniture de services avec la qualité demandée. En plus, lorsque la charge de travail du système est très dynamique c'est à dire la demande de l'application est très variable en terme des ressources et par conséquent les ressources informatiques deviennent insatisfaisantes. Cette transition rapide des besoins complique aussi le processus de provisionnement

En général, dans les réseaux traditionnels comme le réseau IMS les méthodes de provisionnement de la QoS utilisées sont des techniques de provisionnement statiques basées sur des prédictions des besoins en ressources. Également, l'opérateur estime les ressources nécessaires en se basant sur la charge de trafic réseau selon des intervalles horaires bien spécifiques. Ensuite, la réservation des ressources requises est effectuée au préalable de manière statique. Une telle technique peut engendrer deux problèmes :

- une sous-utilisation des ressources si la charge réelle de trafic est inférieure à celle prévu ;
- une sur-utilisation des ressources si la charge réelle de trafic est supérieure à celle prévu.

Dans le cas la ou il y a une sous-utilisation, les ressources allouées dépassent les besoins de l'application ce qui engendre une perte de coût de la part de l'opérateur. Tandis que dans le cas de sur-utilisation les ressources réservées sont insuffisantes pour répondre à toutes les requêtes clientes. Par conséquent, des rejets des nouveaux appels entrants, dégradation de qualité d'où le non-respect de la QoS définit dans les SLAs pour les appels en cours.

Un tel résultat provoque insatisfaction des clients de la qualité offerte et peut causer perte des abonnés de la part de l'opérateur et par conséquence une réduction des revenus.

4.4.1 Provisionnement des ressources dans le nuage informatique

Le nuage informatique est considéré l'environnement le plus populaire qui nécessite un minimum d'intervention manuelle pour la gestion et le provisionnement de la QoS et la gestion des ressources informatiques. Un aspect important du nuage informatique qui le différencie des services traditionnels c'est la quantité infinie des ressources (e.g CPU, mémoire, stockage, réseau) offertes aux grands publics à un prix compétitif.

Selon (Calheiros *et al.*, 2011) le processus de déploiement et la gestion des applications sur les infrastructures de nuage est composé de trois étapes :

Provisionnement de la machine virtuelle : qui consiste à l'instanciation d'une ou plusieurs machines virtuelles (VM) qui correspondent à des caractéristiques spécifiques comme la capacité de stockage, nombre de processeurs , nombre d'interfaces réseaux et la taille mémoire.

Provisionnement des ressources : qui implique l'orchestration et le mappage du VMs sur les serveurs physiques. Cette tâche est généralement effectuée par le fournisseur de nuage là où le prestataire n'a aucun accès.

Provisionnement de l'application : qui consiste au déploiement des applications spécialisées dans des machines virtuelles et le mappage des requêtes de l'utilisateur final aux instances d'application. Dans cette thèse nous avons concentré sur le provisionnement de la machine virtuelle et le provisionnement de l'application, étant donné que ces deux types sont contrôlés par le fournisseur de services.

4.4.2 Modèle de provisionnement de la QoS proposé

Les objectifs de qualité de service dans le système IMS sont liés aux types de services demandés par l'utilisateur final et qui comprennent le temps de réponse, la gigue et le taux de rejet des appels, la qualité de l'audio et de la vidéo, etc. Les sections 2.3.4 et 2.3.5 présentent les exigences de chaque service dans les réseaux IP.

Dans IMS les paramètres de la QoS sont établis et la réservation des besoins est effectuée selon les services demandés par l'utilisateur. Ces dernières sont importantes, car ils ont un effet direct sur l'expérience de l'utilisateur sur le service demandé. Par exemple, si le délai pour établir un appel téléphonique est trop élevé ou si la demande est rejetée, certains utilisateurs pourraient abandonner le service de façon permanente ce qui va contribuer à la réduction des revenus de l'opérateur. Par conséquent, répondre aux objectifs de la QoS pour les services à temps réels offerts par IMS est nécessaire. Pour se faire, nous avons proposé un modèle de provisionnement de la QoS d'IMS dans le nuage. La Fig.4.8 illustre une vue logique du processus et les différentes composantes de modèle de provisionnement de la QoS proposé.

Le processus de provisionnement de la QoS du modèle proposé comprend trois phases : **Monitoring** : consiste à surveiller le trafic SIP et collecter les données nécessaires suivantes lesquelles nous allons subir une l'action.

Scaling : définition les règles de monitoring et de la mise à l'échelle automatique du système IMS et de ses services.

Resource Provisioning : allocation des ressources informatiques nécessaires pour le bon fonctionnement du système IMS dans le nuage.

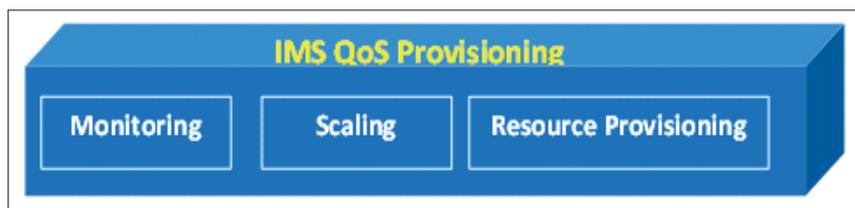


Figure 4.8 Composante du modèle proposé

En général, notre modèle de provisionnement doit respecter les objectifs de la qualité de service selon le service demandé par l'utilisateur en allouant efficacement les ressources nécessaires pour son traitement. Il doit permettre l'adaptation des ressources nécessaires pour le bon fonctionnement du système tout en évitant la sur-utilisation des ressources informatiques. La technique de provisionnement doit optimiser l'utilisation des ressources informatiques virtualisées,

d'où la réduction de la consommation des ressources et de l'énergie physique. Le modèle doit s'assurer de la mise à l'échelle automatique du système lors de l'augmentation de la demande. Plus spécifiquement, les objectifs principaux du modèle proposé sont les suivantes :

- une technique de provisionnement basée sur l'analyse des ressources virtuelles et de la charge de travail dynamique ;
- le provisionnement doit être réalisé d'une façon dynamique sans intervention manuelle du fournisseur de services ;
- mise à l'échelle automatique du système IMS ;
- amélioration de la performance globale du système et garantir le niveau de la qualité de service convenu ;
- optimisation de l'utilisation des ressources virtuelles et physiques ;
- réduction de l'énergie ;
- réduction du coût et augmentation de revenus ;
- satisfaction de l'utilisateur final.

Le système IMS dans le nuage est déployé dans des machines virtuelles. Ces VMs sont hébergées sur des serveurs physiques qui sont caractérisés par leur capacité de mémoire, l'espace de stockage et le nombre de CPUs fixes. Tandis que chaque VM a des ressources virtuelles (VCPU, mémoire virtuelle, espace de stockage, nombre d'interfaces réseau) allouées lors de sa création.

Les paramètres de provisionnement du modèle proposé sont les ressources virtuelles (CPU et mémoire) du VM. Les paramètres de la QoS évalués sont le délai d'une demande d'enregistrement (RRD) définie dans la section 4.3.1 et le délai d'établissement d'un appel téléphonique (SRD) défini dans la section 4.3.2. Nous avons modélisé le comportement des ressources et des performances du système pour les différents types de service et de ressources informatiques avec plusieurs niveaux de charge de travail dont la variation est très dynamique. Le modèle global de provisionnement de la QoS proposé est illustré par la Fig.4.9.

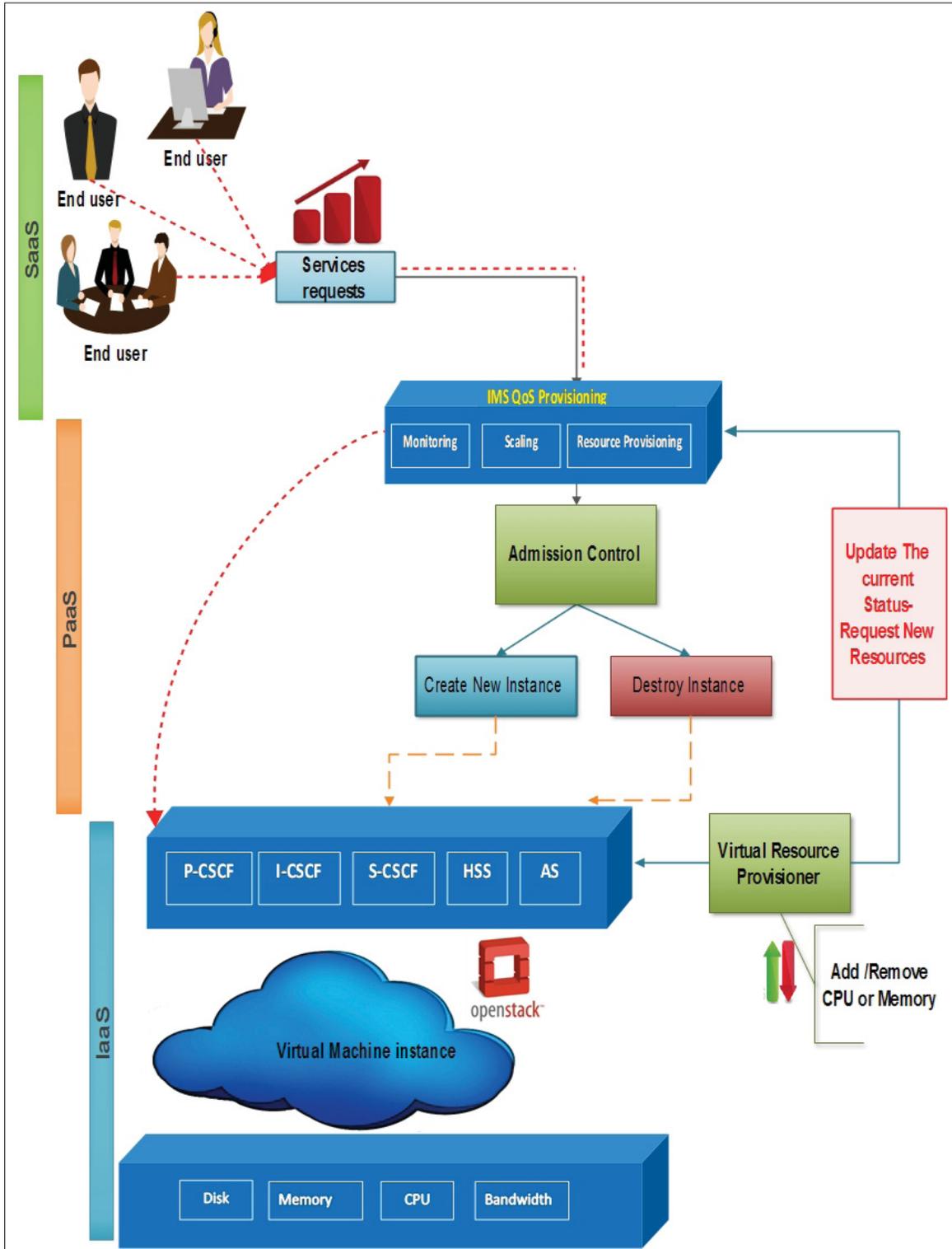


Figure 4.9 Modèle de provisionnement de la QoS d'IMS dans le nuage

L'«**IMS QoS provisioning**» se positionne entre l'application et l'environnement de nuage jouant le rôle d'un gestionnaire des ressources nécessaires pour le fonctionnement du système avec une demande variable. Il est constitué de trois modules principaux : surveillance des ressources et de la QoS, mise à l'échelle et provisionnement des ressources. Il surveille régulièrement les ressources utilisées par chaque instance d'IMS dans le nuage. Cette vérification est effectuée toutes les 30 secondes. Le contrôleur d'admission permet de vérifier d'abord si les ressources disponibles dans chaque instance d'IMS sont suffisantes pour répondre aux besoins de tous les appels systèmes.

Dans le cas où les ressources satisfaisantes aucune action à faire selon l'«IMS QoS provisioning» continue à surveiller le système. Le cas échéant où les ressources sont insuffisantes pour le traitement actuels la «Ressource provisioning» déploie des nouvelles ressources en ajoutant une nouvelle instance de l'application et redirige les nouveaux appels vers cette dernière. Toutefois le choix de l'ajout ou de la réduction des ressources est basé sur les règles définies dans le module de mise à l'échelle «scaling».

Le principal avantage de ce mécanisme est que la nouvelle machine virtuelle n'est pas nécessaire d'être créée dès le départ. La création de VM est effectuée que lorsque le système a besoin de plus de ressources dans le temps critique où il y a une charge de travail élevé. Une fois la charge de trafic diminue et que le système n'a plus besoin à cette instance, la machine virtuelle sera supprimée. Cette action optimise l'utilisation des ressources. Par conséquent, les paramètres de la QoS seront maintenus tels que définis dans le SLA. Le diagramme donne une vue d'ensemble sur le mode de fonctionnement dans le système. Le diagramme de provisionnement des ressources est illustré par la Fig.4.10.

Le «**Virtual resource Provisioning**» est responsable du provisionnement des ressources virtuelles à l'intérieur de VM plus précisément le CPU virtuel et la mémoire virtuelle. Le principe de ce mécanisme est d'activer et de désactiver les processeurs alloués au VM lors de sa création en fonction de la charge de travail. Le principal avantage de ce mécanisme est que les processeurs ne sont pas activés que seulement s'il est nécessaire. Une fois la charge de trafic diminue et que le système n'a plus besoin de ce processeur et sera désactivé (Turn off).

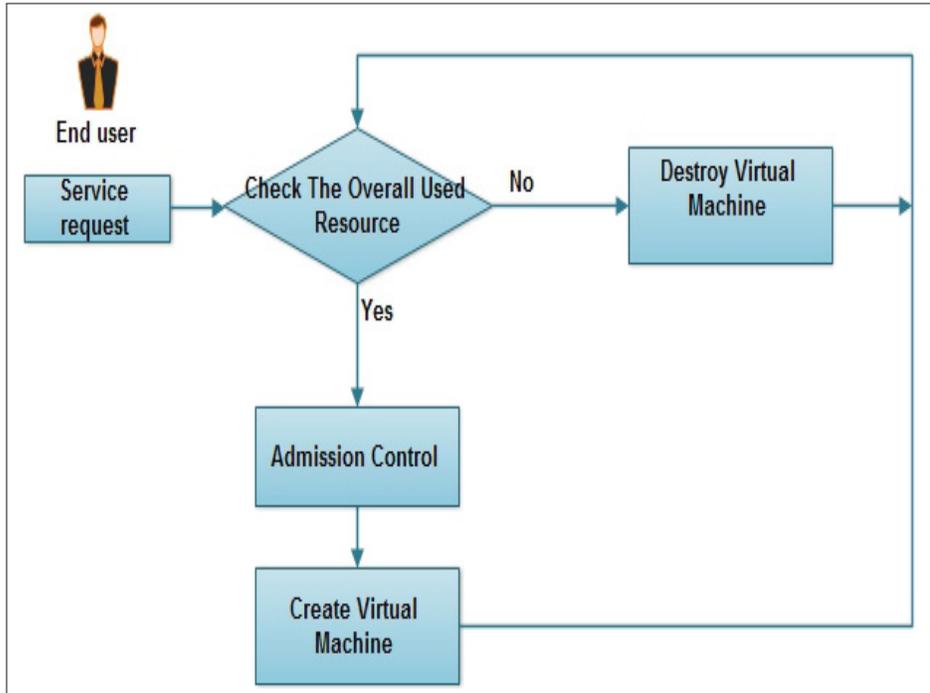


Figure 4.10 Provisionnement des ressources d'IMS

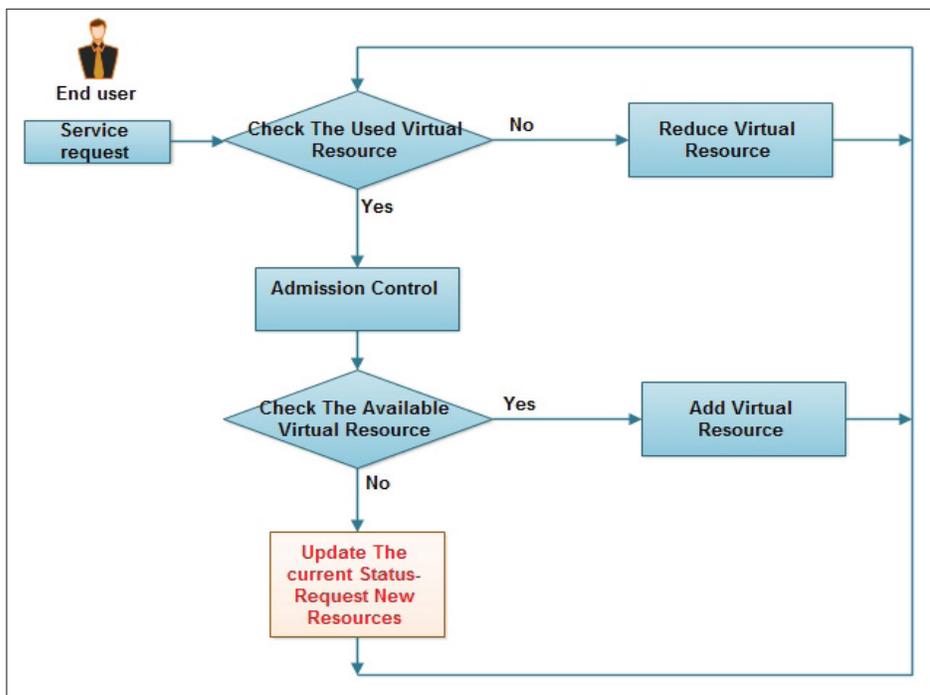


Figure 4.11 Provisionnement des ressources du VM

Le «Virtual ressource Provisioning» effectue la mise à jour régulièrement du statut courant des ressources utilisées, ses informations seront utiles pour l'IMS QoS Provisioning» pour prendre décision. Le diagramme de provisionnement des ressources virtuelles est illustré par la Fig.4.11.

4.5 Conclusion

Dans ce chapitre nous avons présenté les solutions d'IMS existantes opens sources qui sont Clearwater IMS et open source core IMS. Étant donné que les services offerts par le système IMS sont des services en temps réel principalement la voix nous avons proposé un modèle de provisionnement dynamique des ressources des machines virtuelles. Les ressources surveillées sont les ressources virtuelles (CPU et mémoire). Nous avons aussi décrit les métriques de la qualité de services relatif à une session audio qui sont surveillés durant le processus de provisionnement de la QoS.

CHAPITRE 5

TEST ET VALIDATION

5.1 Introduction

Ce chapitre présente une analyse de performance des différentes architectures d'IMS déployées dans le nuage informatique du laboratoire Synchromedia au sein de l'environnement de production de la compagnie Ericsson. La première partie de ce chapitre décrit et détaille l'environnement de test et les trois implémentations réalisées qui seront citées dans la section 5.2. La seconde partie est une étude comparative des paramètres de performance des systèmes déployés dans le nuage avec celui implémenté dans un environnement physique. Une analyse des ressources informatiques physiques et virtuelles utilisées est effectuée en se basant sur les différents niveaux de trafic réseau.

5.2 Architecture et conception des bancs d'essai

Pour tester la fonctionnalité de l'architecture d'IMS dans une plateforme de nuage informatique, nous avons déployé trois bancs d'essai au sein de l'environnement de production de la compagnie Ericsson (2013).

Les deux premiers cas d'études sont basés sur l'intergiciel libre OpenStack (section 4.2.3) en tant qu'infrastructure IaaS pour les deux solutions libres d'IMS (Metaswitch, 2013),(FOKUS, 2006). Les figures Fig.5.1 et Fig.5.2 illustrent respectivement les deux architectures "*Cloud OpenIMS*" et "*Cloud Clearwater*" déployées dans l'infrastructure du nuage privé de la compagnie Ericsson(2013).

Le troisième banc d'essai est une implémentation de la solution libre «open source IMS core» (FOKUS, 2006) dans un environnement physique qui représente une étude de cas de l'architecture d'IMS traditionnel déployée dans des serveurs physiques. La Fig.5.3 illustre l'architecture de banc d'essai. Le but de cette troisième architecture est de créer un cas d'utilisation réelle du

système traditionnel déployé sur des serveurs physiques qui sera un banc d'essai de comparaison avec les deux systèmes déployés au sein de l'environnement de nuage informatique pour évaluer les performances des architectures déployées dans ce dernier.

5.2.1 Banc d'essai 1 : Cloud open source IMS core

Tel que mentionné précédemment nous avons utilisé l'infrastructure libre OpenStack comme notre IaaS pour déployer le framework d'open source IMS core (FOKUS, 2006) dans le nuage privé. Open source IMS core est composé de quatre éléments de base : le proxy P-CSCF, le serving S-CSCF, l'interrogating I-CSCF et la base de données FoHSS telle que décrite dans la section 4.2.1.

Pour déployer ces quatre composants, nous avons choisi d'utiliser trois machines virtuelles. Le module de P-CSCF est installé dans une machine virtuelle, les deux composants I-CSCF et S-CSCF ensemble dans une deuxième machine virtuelle et la base de données FoHSS en troisième machine virtuelle. La Fig.5.1 illustre l'architecture de ce banc d'essai selon cette décomposition. Les trois machines virtuelles sont hébergées sur le même serveur physique et leurs images résident dans une zone partagée (storage) tous situées dans le centre de données de l'ÉTS sur la plateforme d'Ericsson EBS. Le tableau 5.1 spécifie les composants de chaque machine virtuelle. Afin de tester le service de téléphonie dans ce banc de test nous avons

Tableau 5.1 Composants des machines virtuelles OpenIMS

Virtual machine	Composant
VM1	P-CSCF et Serveur DNS (BIND)
VM2	S-CSCF et I-CSCF
VM3	FoHSS

utilisé deux hôtes physiques contenant de client IMS gratuit. Les caractéristiques systèmes des différentes machines, ainsi que leurs ressources virtuelles et physiques allouées en terme de mémoire et CPU sont présentées dans le tableau 5.2.

Pour faciliter la communication entre les différentes entités IMS, le framework d'open source IMS Core fait recours a un serveur DNS. Pour ce faite nous avons choisi d'utiliser un serveur DNS libre *BIND* qui est le serveur le plus utilisé sur la plateforme Linux. L'installation de BIND est réalisée sur la même machine virtuelle qui contient le module de P-CSCF. Nous avons gardé le même nom du domaine "*openims.test*" spécifié par (FOKUS, 2013).

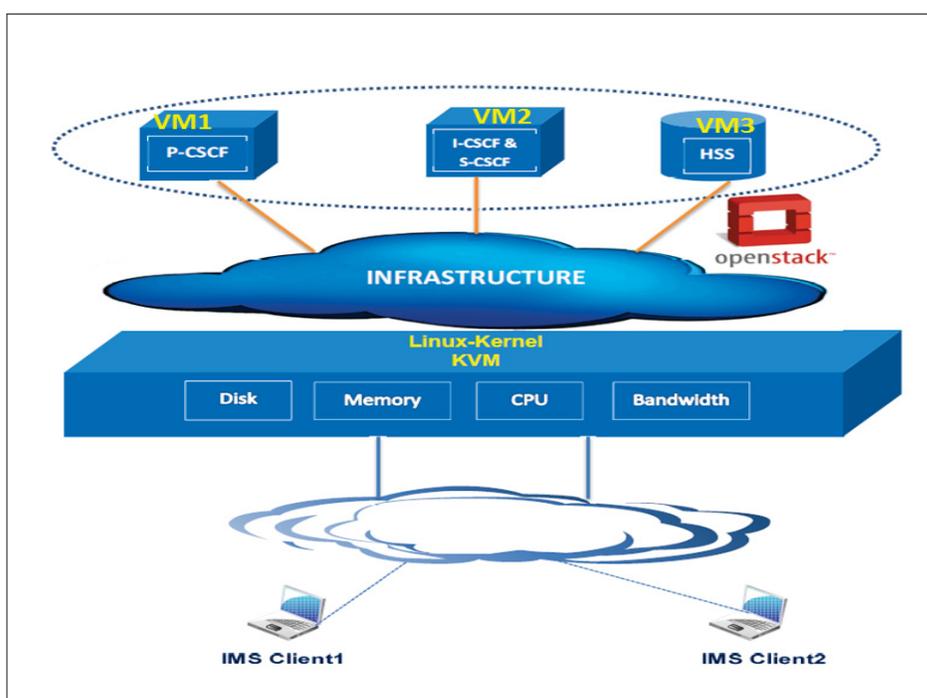


Figure 5.1 Cloud Open IMS

Tableau 5.2 Caractéristiques des machines OpenIMS

	Operating system	CPU/VCPU	Memory/Vmemory
Physical server	Ubuntu 12.04	12CPU	24G
Virtual machine	Ubuntu 12.04	4VCPU	8G
IMS client Machine	Windows7	4VCPU	4G

5.2.2 Banc d'essai 2 : Cloud Clearwater IMS

De même dans ce banc d'essai nous avons aussi utilisé OpenStack comme infrastructure en tant que service. Clearwater IMS contient les mêmes entités standard qu'open source IMS Core(CSCF et HSS) en plus de deux autres composants qui sont **Ellis** et **XDMS**. Nous avons besoin de deux machines virtuelles supplémentaires pour déployer ces deux derniers. Ainsi, Clearwater IMS est déployé dans cinq machines virtuelles comme le montre la Fig.5.2 qui illustre l'architecture de cette mise en place.

L'entité **P-CSCF** et le serveur **DNS** sont installés dans une machine virtuelle, les deux entités **I-CSCF** et **S-CSCF** ensemble dans une seconde machine virtuelle, la base de données **HSS** dans une troisième machine virtuelle, le serveur **XDMS** dans une quatrième machine virtuelle et la cinquième machine virtuelle contient l'entité de provisionnement **Ellis** comme le spécifie le tableau 5.3. Les cinq machines virtuelles sont hébergées sur un serveur physique sur la plateforme d'Ericsson EBS situé dans le centre de données de l'ÉTS.

Tableau 5.3 Composants des machines virtuelles Clearwater IMS

Virtual machine	Composant
VM1	Bono(P-CSCF) et Serveur DNS(BIND)
VM2	Sprout(S-CSCF et I-CSCF)
VM3	Homer(HSS)
VM4	Homestead(XDMS)
VM5	Ellis(Provisionnement)

De même, pour tester le service de la voix dans ce banc de test nous avons utilisé deux hôtes physiques qui contiennent un client IMS gratuit. Les caractéristiques systèmes des différentes machines, ainsi que leurs ressources virtuelles et physiques allouées en terme de mémoire et CPU sont décrites dans le tableau 5.4.

Tableau 5.4 Caractéristiques des Machines Clearwater IMS

	Operating system	CPU/VCPU	Memory/Vmemory
Physical server	Ubuntu 12.04	12CPU	24G
Virtual machine	Ubuntu 12.04	4VCPU	8G
IMS client Machine	Windows7	4VCPU	4G

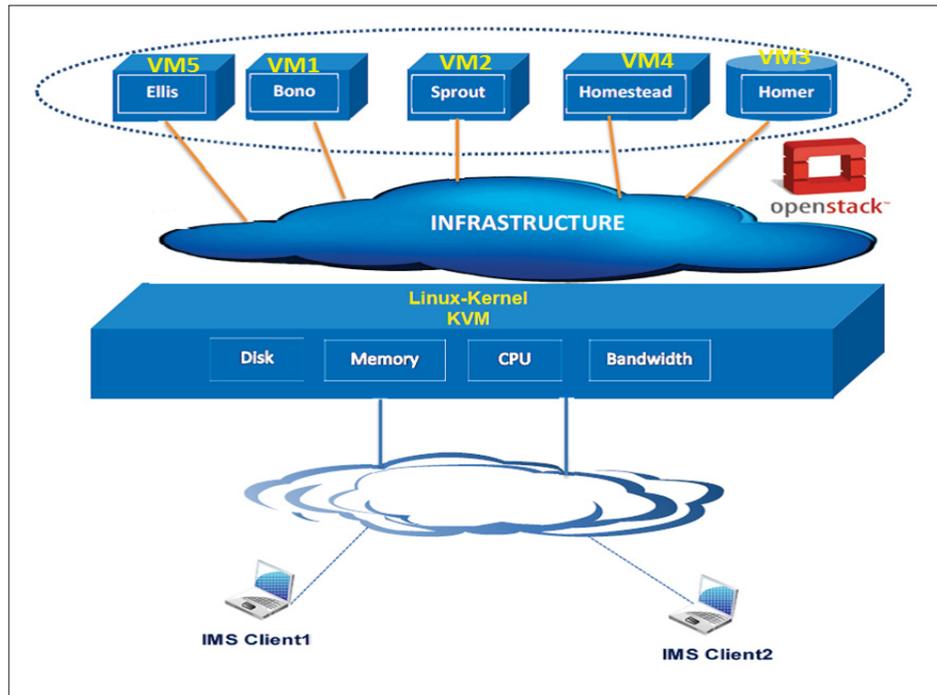


Figure 5.2 Cloud Clearwater IMS

5.2.3 Banc d'essai 3 : Traditional open source IMS Core

Contrairement aux bancs d'essai 1 et 2 qui ont été déployés dans un environnement virtuel, ces deux bancs d'essai sont déployés sur des serveurs physiques. L'architecture de ce banc d'essai est illustrée dans la Fig.5.3. L'objectif principal de cette implémentation est de construire un banc d'essai d'IMS traditionnel. Nous avons utilisé la même solution open source développée par FOKUS (FOKUS, 2013). Ainsi le P-CSCF et le serveur DNS sont déployés dans un serveur physique, le I-CSCF et S-CSCF ensemble dans un second serveur physique, et finalement la base de données FoHSS dans un troisième serveur comme le décrit le tableau 5.5.

Tableau 5.5 Composants des hosts

Physical machine	Composant
Host1	P-CSCF et Serveur DNS(BIND)
Host2	S-CSCF et I-CSCF
Host3	FoHSS

Les trois serveurs physiques sont l'environnement de production réel d'Ericsson EBS . De même, pour tester le service de téléphonie dans ce banc de test nous avons utilisé deux hôtes physiques qui contiennent un client IMS gratuit (Softphone). Les caractéristiques des différentes machines physiques en terme de ressources informatiques allouées (mémoire et CPU) sont spécifiées dans le tableau 5.6.

Tableau 5.6 Caractéristiques des machines

	Operating system	CPU	Memory
Physical server	Ubuntu 12.04	12CPU	24G
IMS client Machine	Windows7	4CPU	4G

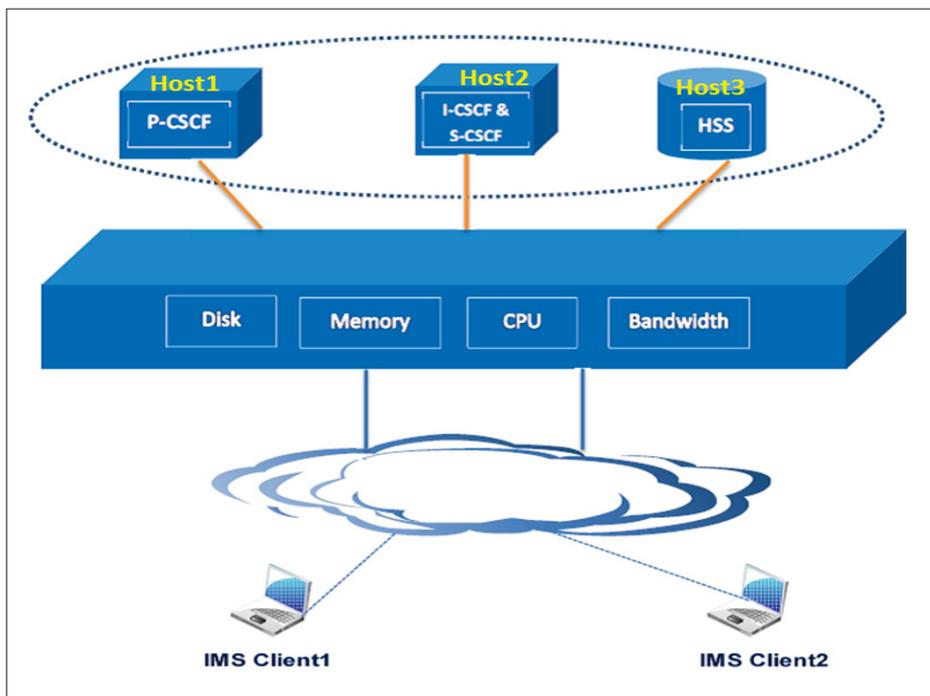


Figure 5.3 Traditional Open IMS architecture

5.3 Test de fonctionnement des architectures d'IMS dans le nuage

5.3.1 Tableau de bord d'Openstack

L'interface web du tableau de bord *horizon* de l'infrastructure de OpenStack déployé est illustré par la Fig.5.4. Cette interface offre aux utilisateurs et aux administrateurs l'accès aux différentes architectures d'IMS déployées et à la gestion des ressources informatiques. Un

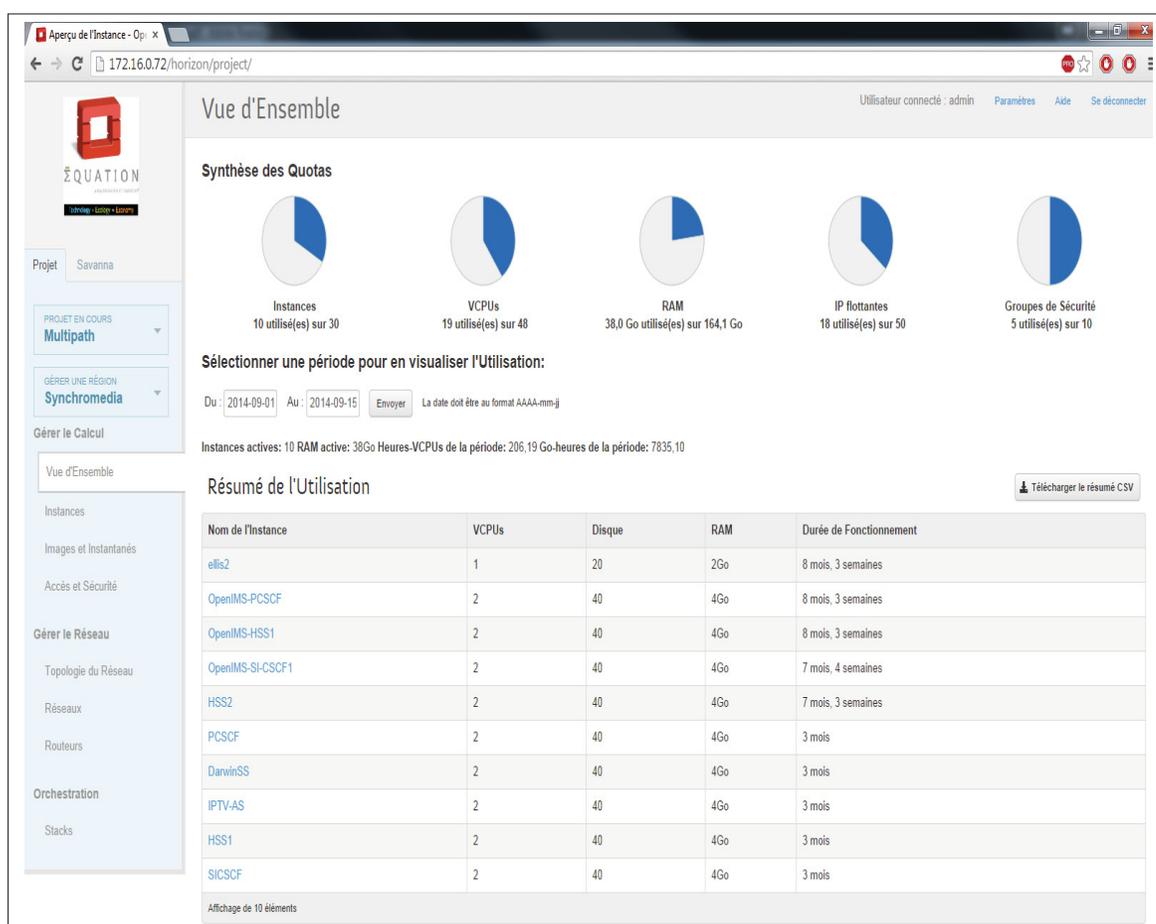


Figure 5.4 Vue d'ensemble Openstack

aperçu sur le quota des ressources utilisées et restantes telles que le VCPU, la RAM, le nombre d'instances et le stockage peut être visualisé dans le résumé d'hyperviseur comme l'illustre la Fig.5.5. Comme le framework de Openstack est déployé dans six serveurs physiques (Fig.5.5),

un résumé global sur les ressources physiques et virtuelles disponible sur chaque serveur est affiché.

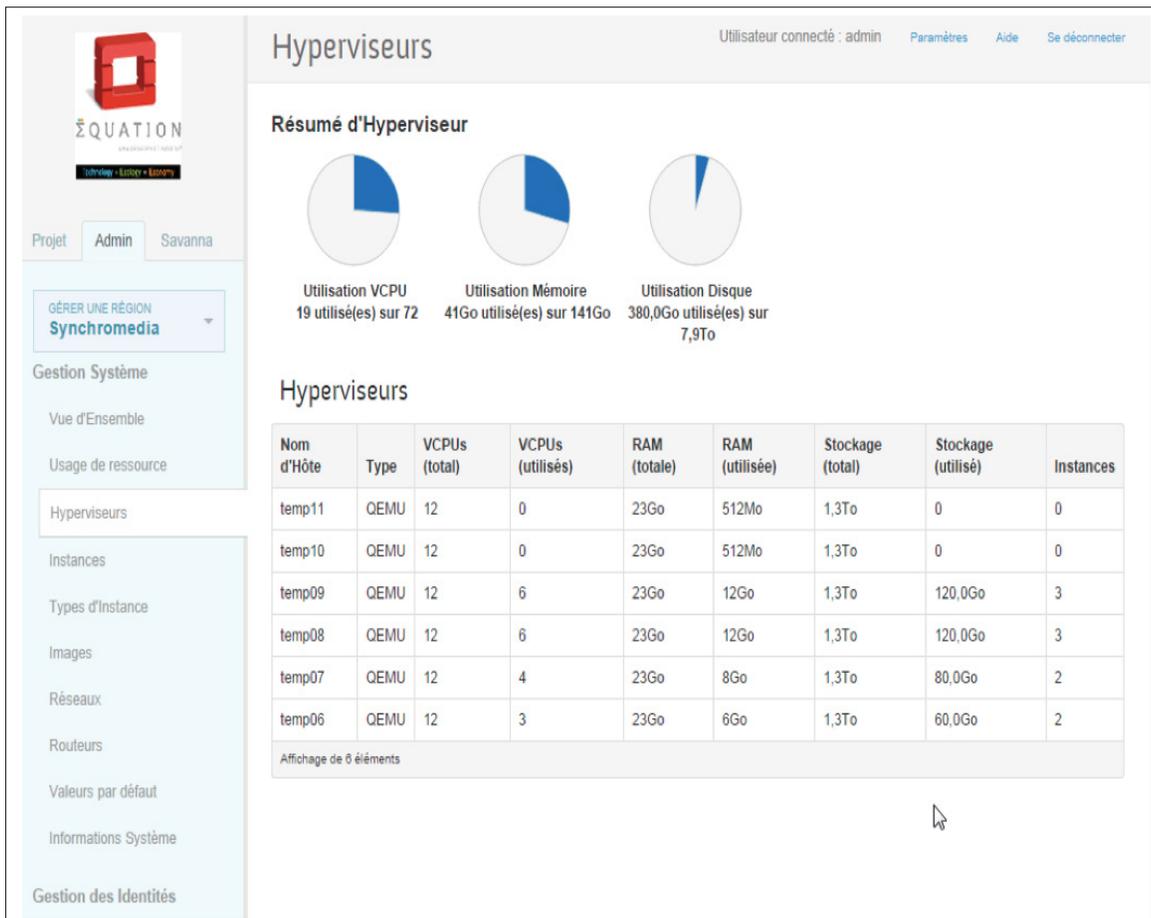


Figure 5.5 Résumé d'hyperviseur

5.3.2 Demande d'enregistrement

Le processus d'enregistrement d'un utilisateur avec les deux systèmes IMS déployés est effectué de la même façon et il est décrit dans la section 4.3. Les messages SIP échangés entre le client et le nuage IMS lors d'une demande d'enregistrement sont capturés du côté client par l'analyseur de paquet Wireshark (Wireshark, 2004) et illustré par les deux Fig.5.6 et Fig.5.7 qui représentent respectivement la procédure d'enregistrement avec le Cloud OpenIMS et la procédure d'enregistrement Cloud Clearwater. La première étape d'un processus d'enregis-

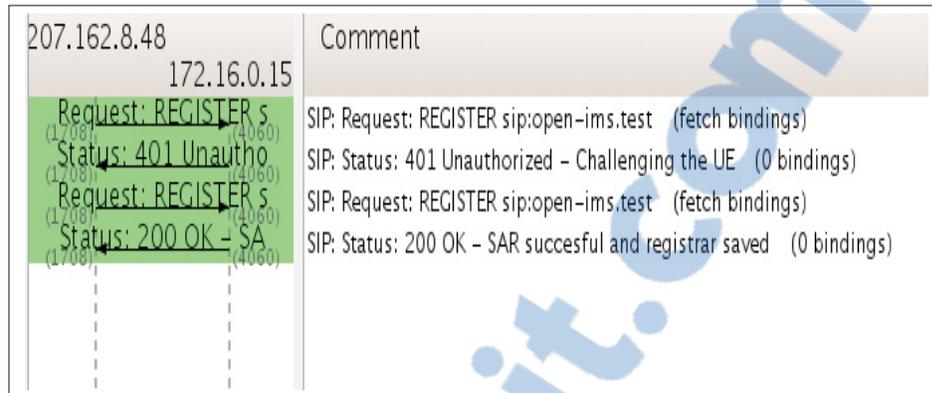


Figure 5.6 Procédure d'enregistrement Cloud OpenIMS



Figure 5.7 Procédure d'enregistrement Cloud Clearwater

trement est l'envoi d'une requête SIP "REGISTER" vers le P-CSCF. Un exemple de demande "REGISTER" est illustré par la Fig.5.8. Après la vérification de l'identité du client en utilisant son profil stocké dans la base de données HSS. L'utilisateur reçoit un message réponse "401 Unauthorized" qui contient les nouveaux paramètres d'authentification. Le message réponse "401 Unauthorized" est illustré par la Fig.5.9.

Ensuite, un second message "REGISTER" est envoyé avec les nouveaux paramètres qui sera acquitté par une réponse "200 OK" dont le corps de ce message est illustré par la Fig.5.10 , celui-ci indique que l'enregistrement est réussi et que l'utilisateur peut établir une session voix, vidéo, ou données.

```

▽ Session Initiation Protocol (REGISTER)
  ↳ Request-Line: REGISTER sip:clearwater.test SIP/2.0
  ▽ Message Header
    ↳ Via: SIP/2.0/UDP 207.162.8.222:58168;branch=z9hG4bK193618169;rport
    ↳ From: <sip:6505550533@clearwater.test>;tag=193609339
    ↳ To: <sip:6505550533@clearwater.test>
    ↳ Contact: <sip:6505550533@207.162.8.222:58168;transport=udp>;expires=600000;+g.oma.sip-im;language="en,fr";+g.3gpp.smsip;+g.oma.sip-im
      Call-ID: 8cb88ebc-bbcc-4744-4414-174534166622
    ↳ CSeq: 3845 REGISTER
      Content-Length: 0
      Max-Forwards: 70
    ↳ Authorization: Digest username="6505550533@clearwater.test",realm="clearwater.test",nonce="",uri="sip:clearwater.test",response=""
      Allow: INVITE, ACK, CANCEL, BYE, MESSAGE, OPTIONS, NOTIFY, PRACK, UPDATE, REFER
      Privacy: none
      P-Access-Network-Info: ADSL;utran-cell-id-3gpp=00000000
      User-Agent: IM-client/OMA1.0 Boghe/v2.0.132.808
    ↳ P-Preferred-Identity: <sip:6505550533@clearwater.test>
      Supported: path

```

Figure 5.8 Demande d'enregistrement : Méthode "REGISTER"

```

▽ Session Initiation Protocol (401)
  ↳ Status-Line: SIP/2.0 401 Unauthorized
  ▽ Message Header
    ↳ Via: SIP/2.0/UDP 207.162.8.222:58168;rport=58168;received=207.162.8.222;branch=z9hG4bK193618169
      Call-ID: 8cb88ebc-bbcc-4744-4414-174534166622
    ↳ From: <sip:6505550533@clearwater.test>;tag=193609339
    ↳ To: <sip:6505550533@clearwater.test>;tag=z9hG4bKjpwDTm0oLlLM-1FLfoN0uivyOXok-pfgl
    ↳ CSeq: 3845 REGISTER
    ↳ WWW-Authenticate: Digest realm="clearwater.test",nonce="5e2541911a1904fe",opaque="3b8b2155103a2948",algorithm=MD5,qop="auth"
      Content-Length: 0

```

Figure 5.9 Demande d'enregistrement : Méthode "401 Unauthorized"

5.3.3 Établissement d'une session audio

Maintenant que l'utilisateur a été enregistré dans le réseau d'IMS déployé dans le nuage, le client pourrait effectuer une session voix, vidéo ou donnée. Enfin, le client doit initier une session en envoyant la requête "INVITE" qui permet à un client de demander une nouvelle session.

```

▽ Session Initiation Protocol (200)
  ▸ Status-Line: SIP/2.0 200 OK
  ▾ Message Header
    Service-Route: <sip:sprout.clearwater.test:5054;transport=TCP;lr;orig>
    ▸ Via: SIP/2.0/UDP 207.162.8.222:58168;rport=58168;received=207.162.8.222;branch=z9hG4bK193594929
      Call-ID: 8cb88ebc-bbcc-4744-4414-174534166622
    ▸ From: <sip:6505550533@clearwater.test>;tag=193609339
    ▸ To: <sip:6505550533@clearwater.test>;tag=z9hG4bKpJSZW2Xw2UbDZBZelqpZpEy52rFgBpMRv0
    ▸ CSeq: 3846 REGISTER
      Supported: outbound
    ▸ Contact: <sip:6505550533@207.162.8.222:58168>;expires=300;transport=udp;+g.oma.sip-im;language="en,fr";+g.3gpp.smsip;+g.oma.sip
      Require: outbound
    Path: sip:ZW0dEPpI70@172.16.0.202:5058;transport=TCP;lr;ob
    P-Associated-URI: <sip:6505550533@clearwater.test>
    Content-Length: 0
  
```

Figure 5.10 Demande d'enregistrement : Code de réponse "200 OK"

```

▽ Session Initiation Protocol (INVITE)
  ▸ Request-Line: INVITE sip:bob@open-ims.test SIP/2.0
  ▾ Message Header
    ▸ Via: SIP/2.0/UDP 207.162.8.222:4241;branch=z9hG4bK260409924;rport
    ▸ From: <sip:alice@open-ims.test>;tag=260429457
    ▸ To: <sip:bob@open-ims.test>
    ▸ Contact: <sip:alice@207.162.8.222:4241;transport=udp>;+g.oma.sip-im;language="en,fr";+g.3gpp.icsi-ref
      Call-ID: cfdb0325-5173-4345-2936-edcb11272524
    ▸ CSeq: 22671 INVITE
      Content-Type: application/sdp
      Content-Length: 506
      Max-Forwards: 70
    ▸ Route: <sip:orig@scscf.open-ims.test:6060;lr>
      Accept-Contact: *;+g.3gpp.icsi-ref="urn:urn-7:3gpp-service.ims.icsi.mmtel"
      P-Preferred-Service: urn:urn-7:3gpp-service.ims.icsi.mmtel
      Allow: INVITE, ACK, CANCEL, BYE, MESSAGE, OPTIONS, NOTIFY, PRACK, UPDATE, REFER
      Privacy: none
      P-Access-Network-Info: ADSL;utran-cell-id-3gpp=00000000
      User-Agent: IM-client/OMA1.0 Boghe/v2.0.132.808
    ▸ P-Preferred-Identity: <sip:alice@open-ims.test>
      Supported: 100rel
  ▾ Message Body
    ▾ Session Description Protocol
  
```

Figure 5.11 Établissement d'une session audio/vidéo : Méthode "INVITE"

Cette méthode SIP sert à la fois pour initier une session voix ou une session vidéo pour un client déjà enregistré. Un exemple d'une demande "INVITE" envoyé par l'UE (bob) pour initier un appel téléphonique avec UE (Alice) illustré par la Fig.5.11 . Les paramètres de la session sont négociés en utilisant le protocole SDP dont le message sera encapsulé dans la requête SIP "INVITE".

```

. . .
▼ Message Body
  ▼ Session Description Protocol
    Session Description Protocol Version (v): 0
    ▶ Owner/Creator, Session Id (o): doubango 1983 678901 IN IP4 207.162.8.222
    Session Name (s): -
    ▶ Connection Information (c): IN IP4 207.162.8.222
    ▶ Time Description, active time (t): 0 0
    ▶ Session Attribute (a): tcap:1 RTP/AVPF
    ▶ Media Description, name and address (m): audio 23336 RTP/AVP 0 3 8 9 101
    ▶ Media Attribute (a):ptime:20
    ▶ Media Attribute (a): silenceSupp:off - - - -
    ▶ Media Attribute (a): rtpmap:0 PCMU/8000/1
    ▶ Media Attribute (a): rtpmap:3 GSM/8000/1
    ▶ Media Attribute (a): rtpmap:8 PCMA/8000/1
    ▶ Media Attribute (a): rtpmap:9 G722/8000/1
    ▶ Media Attribute (a): rtpmap:101 telephone-event/8000/1
    ▶ Media Attribute (a): fmp:101 0-16
    ▶ Media Attribute (a): pcf:1 t=1
    Media Attribute (a): sendrecv
    Media Attribute (a): rtcp-mux
    ▶ Media Attribute (a): ssrc:2819502208 cname:ldjWoB60jbyQ1R6e
    ▶ Media Attribute (a): ssrc:2819502208 mslabel:6994f7d1-6ce9-4fbd-acfd-84e5131ca2e2
    ▶ Media Attribute (a): ssrc:2819502208 label:Doubango

```

Figure 5.12 Établissement d'une session audio : Méthode "INVITE"

La session de la audio est un exemple d'établissement d'une session. Les deux participants (Alice et Bob) négocient les paramètres de la session (de qualité, l'encodage de médias et des capacités du terminal à être utilisées), ainsi que la réservation de ressources par les messages "SIP/SDP". Exemple de message SDP est montré par Fig.5.12. Lorsque tous les paramètres de la session sont négociés et mis en place, les deux UE échangent un message "200 OK", puis ils échangent un message "ACK" qui indique que la création de la session vocale a réussi.

5.3.4 Établissement d'une session donnée

Un autre exemple d'établissement de session dans IMS est la session donnée utilisée pour bénéficier du service de messagerie instantanée. Le processus d'établissement de cette session est décrit dans la Fig.4.7. L'exemple de la méthode "MESSAGE" est montré par la Fig.5.13.

5.3.5 Gestion du profil d'utilisateur

Les deux frameworks d'IMS déployés dans le nuage offrent des interfaces web pour la gestion des utilisateurs et des services fournis par le réseau IMS. Le portail web de l'architecture de Cloud OpenIMS est installé en parallèle avec la base de données FoHSS. L'accès à cette interface web de gestion est limité aux administrateurs une authentification préalable est requise

```

Session Initiation Protocol (MESSAGE)
  Request-Line: MESSAGE sip:bob@open-ims.test SIP/2.0
  Method: MESSAGE
  Request-URI: sip:bob@open-ims.test
  [Resent Packet: False]
  Message Header
    Via: SIP/2.0/UDP 207.162.8.222:57286;branch=z9hG4bK190841037;rport
    From: <sip:alice@open-ims.test>;tag=190851251
    To: <sip:bob@open-ims.test>
    Call-ID: 3e8805a0-c9b2-bfac-57ff-20320ddc9c9f
    CSeq: 25539 MESSAGE
    Content-Length: 278
    Max-Forwards: 70
    Accept-Contact: *;+g.oma.sip-im
    Accept-Contact: *;language="en,fr"
    Route: <sip:orig@scscf.open-ims.test:6060;lr>
    Content-Type: application/im-iscomposing+xml
    Allow: INVITE, ACK, CANCEL, BYE, MESSAGE, OPTIONS, NOTIFY, PRACK, UPDATE, REFER
    Privacy: none
    P-Access-Network-Info: ADSL;utran-cell-id-3gpp=00000000
    User-Agent: IM-client/OMA1.0 Boghe/v2.0.132.808
  Message Body
    <?xml version="1.0" encoding="utf-8"?>\r\n
    <isComposing xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="urn
    :ietf:params:xml:ns:im-composing" state="idle"/>\r\n
    <contenttype>text/plain</contenttype>\r\n
    </isComposing>
  
```

Figure 5.13 Établissement d'une session donnée : Méthode "MESSAGE"

Fig.5.14. L'administrateur gère à la fois l'identité des abonnés et le profil de service. Pour Clearwater, l'entité "Ellis" représente le portail web configuré comme un composant séparé qui fait partie de l'architecture de Clearwater IMS. L'entité "Ellis" est installée dans une machine unique connectée à toutes les autres instances et qui représente l'entité de provisionnement. De même l'accès à cette interface est limité, un nom d'utilisateur et un mot de passe sont requis. L'administrateur peut ajouter d'autres comptes et gère l'ensemble d'abonnés.

5.3.6 Test de fonctionnement du service IMS

La Fig.5.16 illustre l'interface de IMS client (Bob-Clearwater) enregistré sur Cloud Clearwater. Fig.5.17 illustre un exemple de communication vidéo entre deux clients IMS (Alice-Clearwater et Bob-Clearwater).

The screenshot shows the FHoSS web console interface. The browser address bar displays `172.16.0.210:8080/hss.web.console/`. The page title is "FHoSS - The FOKUS Home Subscriber Server (Rel. 7)". The navigation menu includes: HOME, USER IDENTITIES, SERVICES, NETWORK CONFIGURATION, STATISTICS, and help.

The main content area is titled "Application Server -AS-". On the left, there is a "Services" sidebar with the following options:

- Service Profiles: Search, Create
- Application Servers: Search, Create
- Trigger Points: Search, Create
- Initial Filter Criteria: Search, Create
- Shared IFC Sets: Search, Create
- DSAI: Search, Create

The main configuration form contains the following fields:

ID	2
Name*	iptv_as
Server Name*	sip:172.16.0.209:8010
Diameter FQDN*	iptv.open-ims.test
Default Handling*	Session - Continued
Service Info	
Rep-Data Limit	1024

On the right, there is a "Sh Interface - Permissions" table:

Permission for	UDR	PUR	SNR
Allowed Request	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Repository-Data	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
IMPU	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>
IMS User State	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>
S-CSCF Name	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>
iFC	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>
Location	<input checked="" type="checkbox"/>		
User-State	<input checked="" type="checkbox"/>		
Charging-Info	<input checked="" type="checkbox"/>		
MS-ISDN	<input checked="" type="checkbox"/>		
PSI Activation	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
DSAI	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Aliases Rep Data	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

At the bottom, a note states: "Mandatory fields were marked with '*'"

Figure 5.14 Interface de FoHSS

5.4 Protocole expérimental et résultats

5.4.1 Scénario d'évaluation des métriques de la QoS

Nous considérons également que la consommation de ressources de l'hôte et des machines virtuelles sont essentiellement l'utilisation du CPU et la mémoire. L'évaluation est faite en deux étapes :

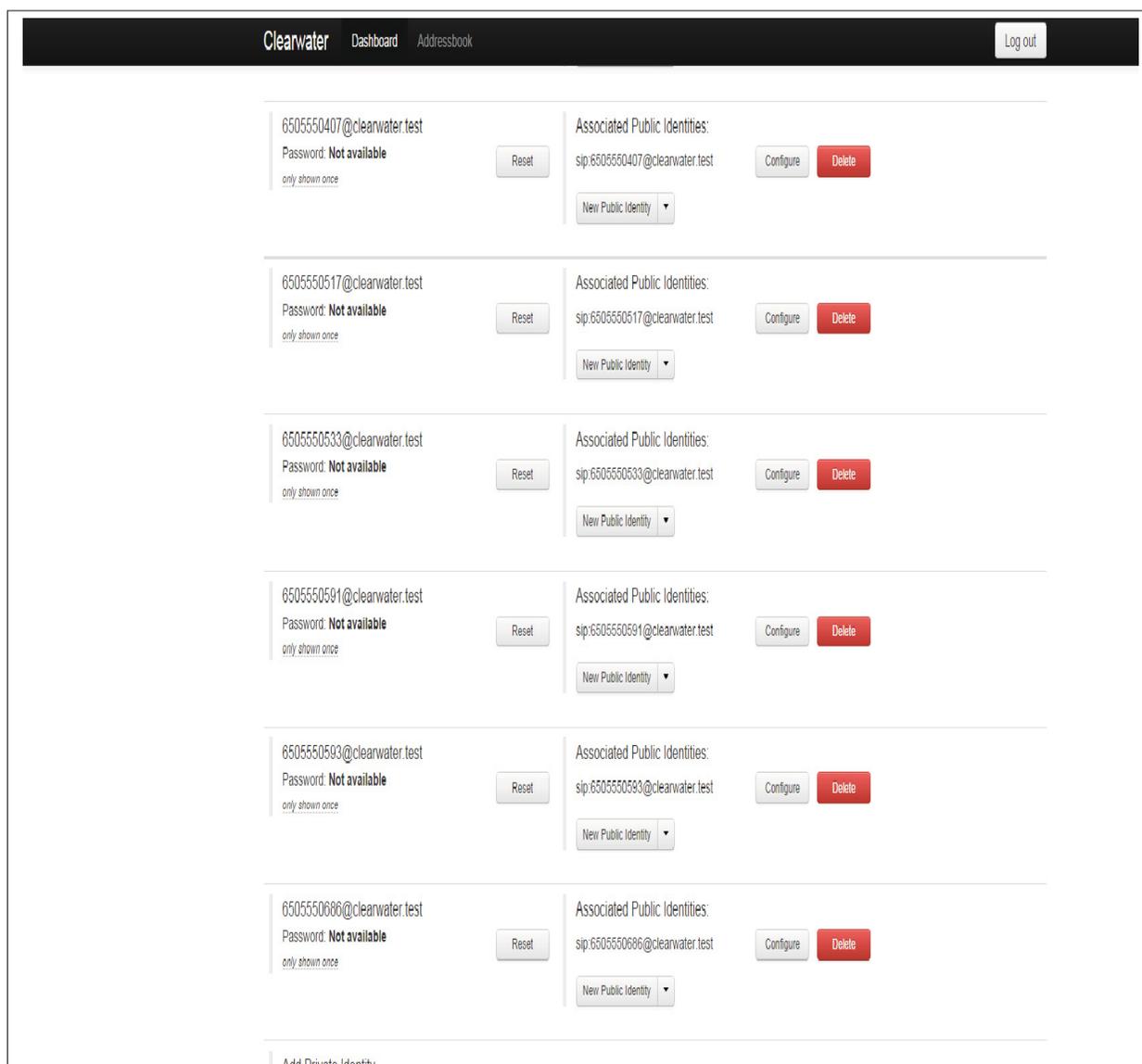


Figure 5.15 Interface de Ellis

Dans l'étape 1, nous avons analysé le délai d'une demande d'enregistrement (RRD), le délai d'établissement de la session audio (SRD) et celui d'établissement de la donnée (IMD). Nous avons également analysé les métriques relatives au flux RTP pour l'appel téléphonique (délai, gigue et bande passante). À cette fin, d'abord nous avons effectué dix expériences pour la demande d'enregistrement. Après nous avons performé dix appels audio entre deux abonnés utilisant leurs profils stockés dans les bases de données HSS. Ensuite nous avons analysé la consommation des ressources physiques, CPU et mémoire physique pour les deux architectures

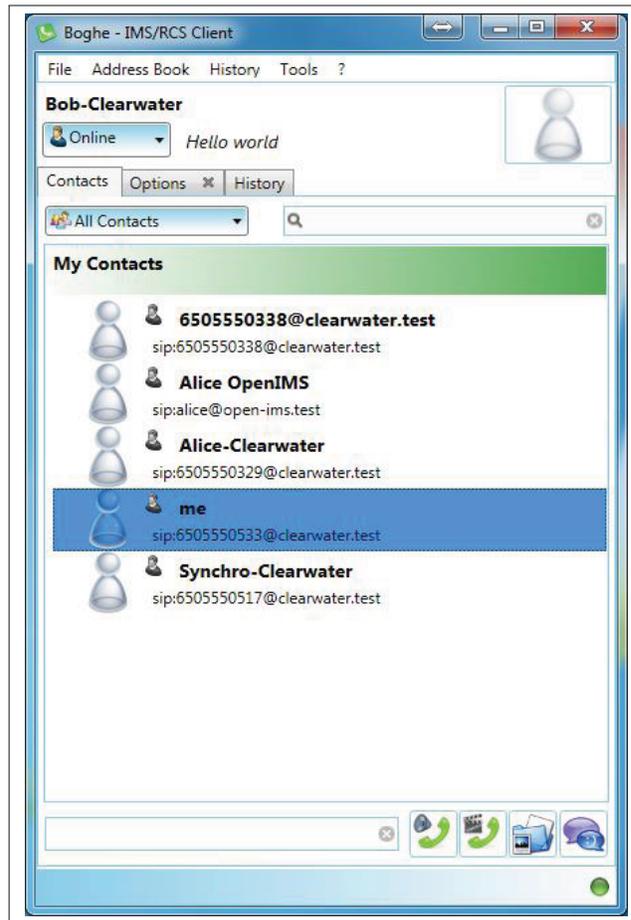


Figure 5.16 Fenêtre du client IMS(Alice/Bob)

déployées dans le nuage. Enfin nous avons analysé l'utilisation des ressources virtuelles pour chaque entité d'IMS.

L'étape 2 consiste à générer plusieurs appels téléphoniques simultanément à l'aide d'un générateur de trafic SIP pour créer une charge de trafic dynamique sur les architectures IMS dans le nuage, le but de ce scénario est d'évaluer le comportement global du système. Le nombre d'appels générés commence à partir de 10 appels par seconde jusqu'à 1200 appels par seconde. D'abord nous avons répéter les mêmes expérimentations de la première étape et collecter les mêmes données relatives aux métriques de la QoS des fins de comparaison. Finalement, nous avons effectué une étude comparative des résultats obtenue dans l'étape 1 et 2.

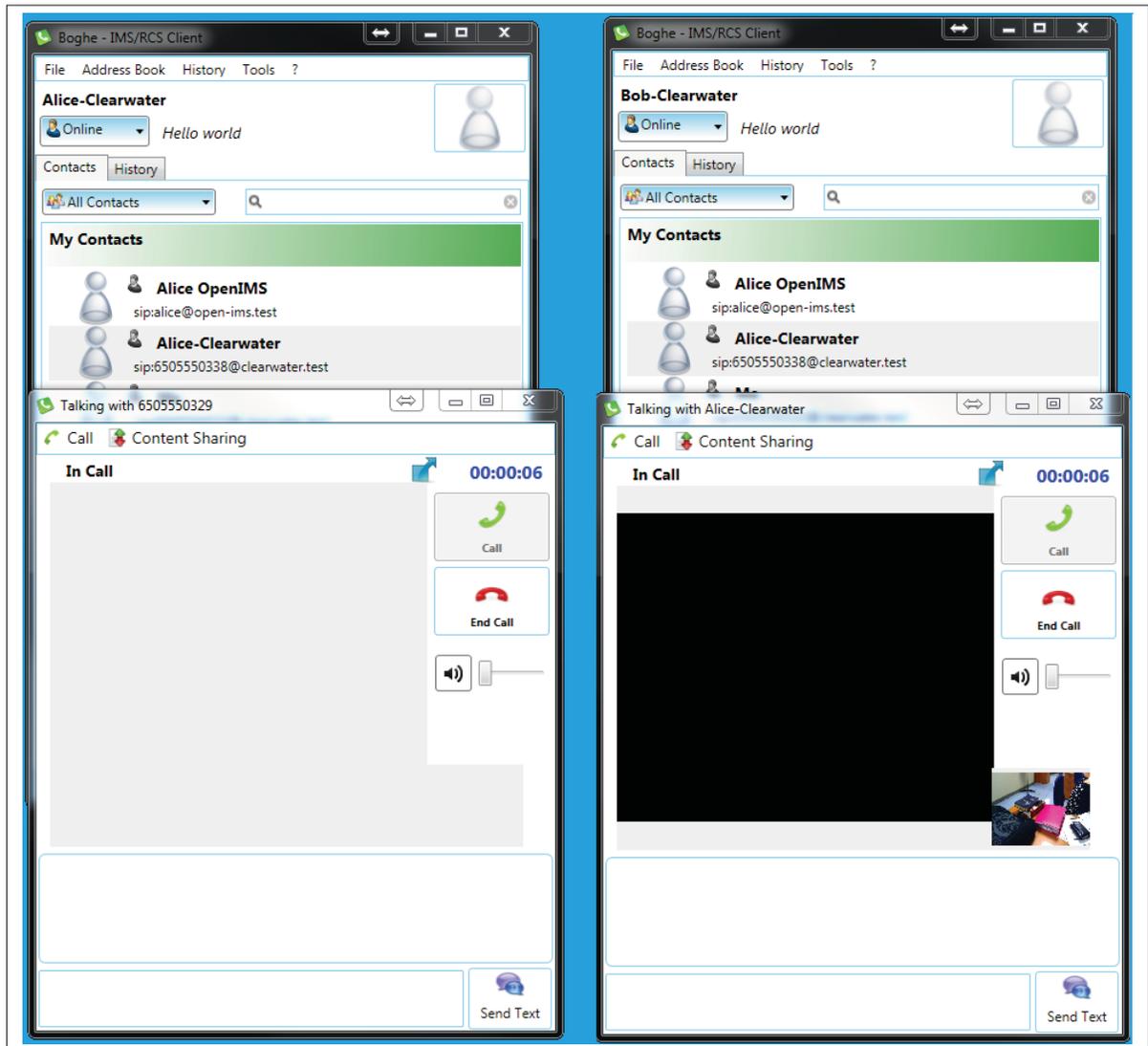


Figure 5.17 Communication vidéo entre deux clients (Alice/Bob)

5.4.2 Processus de surveillance du trafic dans IMS

Pour évaluer les paramètres du qualité de service dans les architectures d'IMS dans le nuage, nous avons surveillé le trafic réseau dans les trois bancs d'essai. De ce fait nous avons utilisé le module "**Monitoring**" du modèle de provisionnement de la QoS proposé, et nous avons développé un outil pour collecter et analyser le trafic de signalisation et de contrôle SIP et le trafic de données RTP. Les étapes de surveillance du trafic réalisé par cet outil sont décrites par la Fig.5.18. Une fois, la collection des trafics réseaux est effectuée, un filtrage de paquet de signalisation et de contrôle SIP est performé à fin d'extraire les paquets dont on a besoin.

Après le filtrage de l'information utile requise, une analyse des données est faite pour calculer les différentes métriques de la QoS (RRD, SRD et IMD). Enfin, les différentes métriques seront affichées à travers l'interface web que nous avons développé.

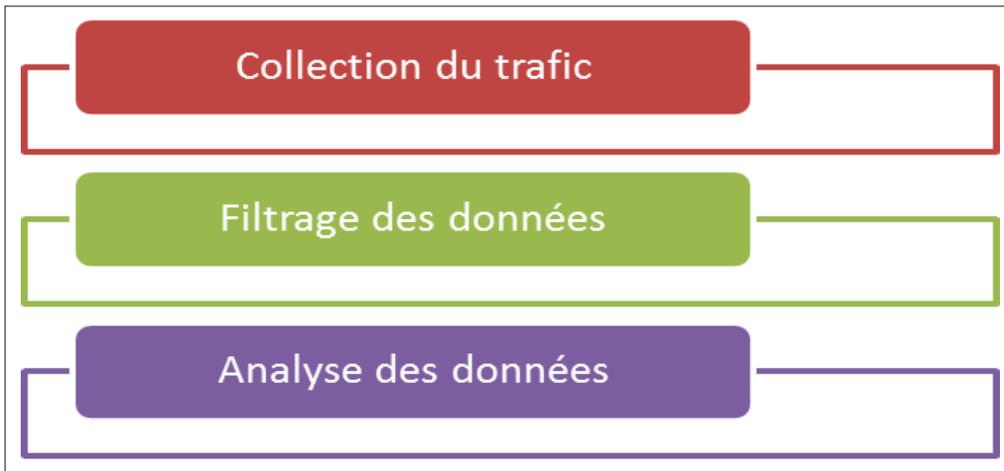


Figure 5.18 Processus de surveillance du trafic

5.5 Résultats

5.5.1 Analyse de délai d'une demande d'enregistrement (RRD)

Le délai d'une demande d'enregistrement décrite dans la section 4.3.1 est mesuré pour les dix expérimentations et pour les deux architectures d'IMS déployés dans le nuage informatique. La variation de RRD est illustré par la Fig.5.19. La variation de RRD dans les architectures d'IMS dans le nuage est comparé avec celui de l'architecture d'IMS fixe (banc d'essai trois) déployée dans le serveur physique. Le résultat global montre que la variation de RRD dans "Cloud OpenIMS" et "Cloud Clearwater IMS" donne des performances meilleures ou similaires que celui de "Traditionnel OpenIMS". Le délai d'enregistrement maximum est de 100 ms mesuré avec traditionnel Open source IMS pour le sixième appel "Call6". Cependant, au sein de Cloud Clearwater IMS l'ensemble des résultats montrent que les délais d'enregistrement pour cette architecture est toujours moins de 40 ms et il est meilleur que les deux autres architectures. Lorsqu'on a envoyé plusieurs requêtes de demande d'enregistrement vers le P-CSCF, qui varie

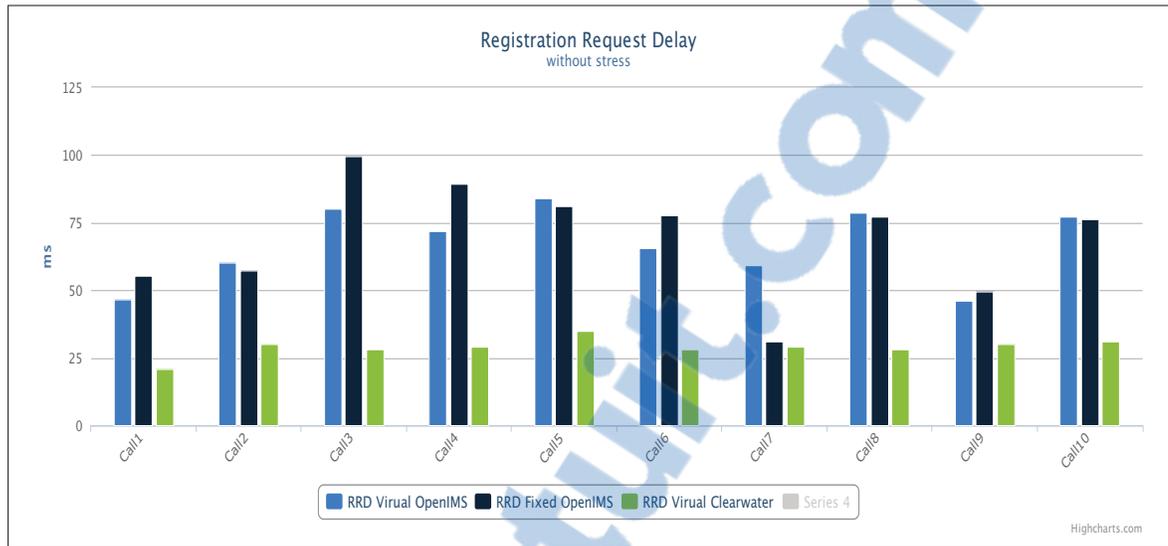


Figure 5.19 Variation du délai d'enregistrement (RRD) sans stress SIP

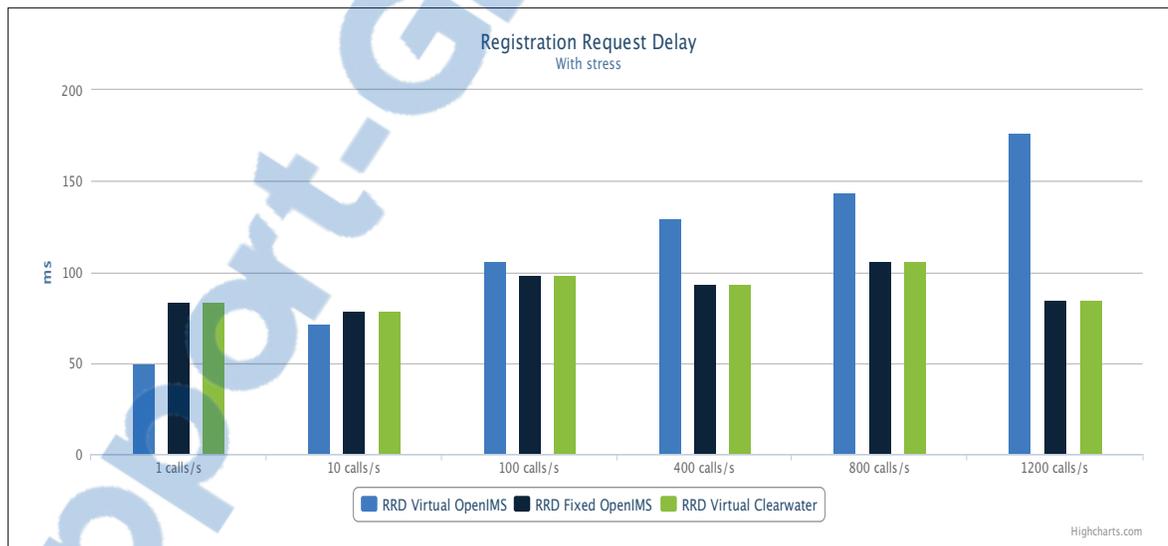


Figure 5.20 Variation de délai d'enregistrement (RRD) avec stress SIP

entre 10 appels/ s et 1200 appels/s nous avons remarqué que dans tous trois bancs d'essai il y a une augmentation dans le RRD et qui peut atteindre les 180 ms avec Cloud OpenIMS lorsque le PCSCF reçoit 1200 appels/s tels que le montre la Fi 5.20.

5.5.2 Analyse du délai d'une demande d'établissement d'une session vidéo (SRD)

La demande d'établissement d'une session audio est décrite dans la section 4.3.2. Le SRD est aussi calculé pour dix appels entre deux clients IMS enregistrés au sein du même réseau IMS pour les deux architectures déployées dans le nuage informatique.

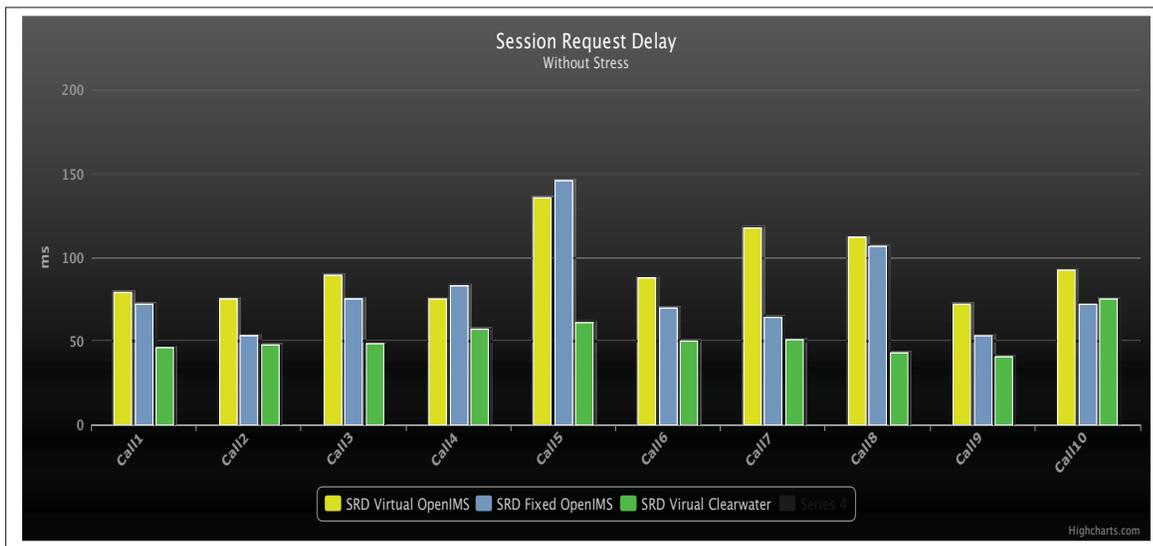


Figure 5.21 Variation du délai d'établissement d'une session audio (SRD) sans stress SIP

La variation de SRD est illustrée par la Fig.5.21. De même nous avons comparé le SRD d'une session audio au sein des deux architectures d'IMS déployés dans le nuage informatique avec celui d'IMS fixe. Les valeurs obtenues pour le SRD pour les architectures d'IMS dans le nuage en comparaison avec celui d'IM fixe sont proches. Le SRD maximale est 150 ms est obtenue avec Traditionnel OpenIMS. Cependant, Cloud Clearwater IMS a donné des meilleurs résultats où le SRD est toujours inférieur à 80 ms. En augmentant le trafic dans le réseau le SRD varie et atteint 300 ms lors d'établissements d'une session toujours avec l'architecture IMS fixe avec 1200 appels /s.

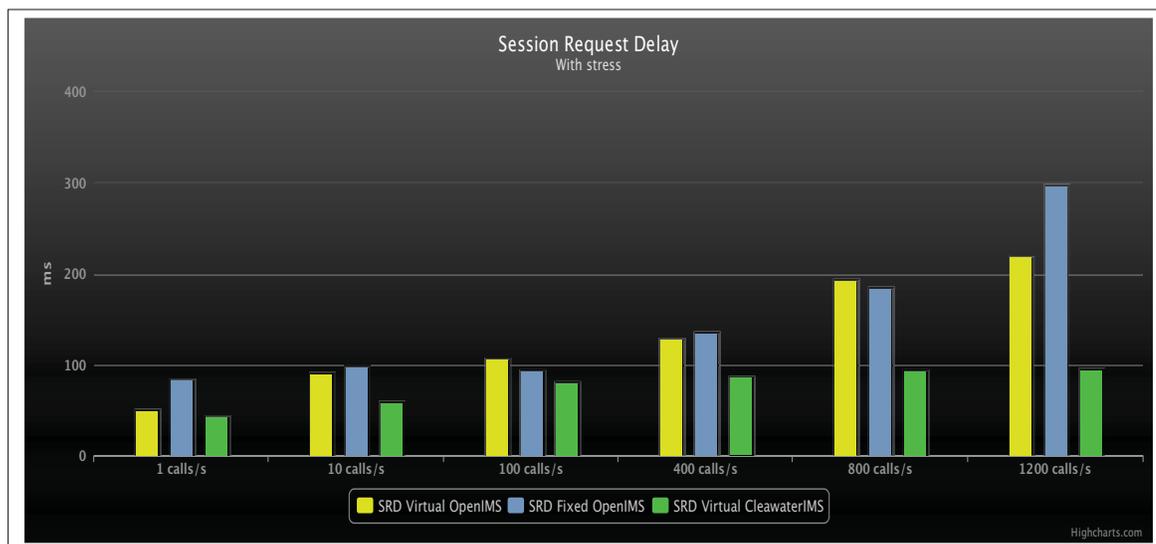


Figure 5.22 Variation de délai d'établissement d'une session audio (SRD) avec stress SIP

5.5.3 Analyse du délai d'une demande d'établissement d'une session donnée

Pour analyser le délai d'établissement d'une session donnée relatif au service de messagerie instantanée tel que décrite dans la section 4.3.3, nous avons analysé le délai de dix messages instantanés entre deux clients IMS enregistrés au sein du même réseau IMS pour les trois architectures déployées.

La variation de délai pour envoyer un message instantané sans aucun SIP stress dans le réseau est illustrée par la Fig.5.23. La Fig.5.24 illustre la variation d'IMD avec une charge de trafic variable dans le réseau. L'IMD varie entre 40 ms obtenues avec Cloud Clearwater IMS et 2.5 s obtenus avec Cloud OpenIMS lorsqu'il n'y a aucune charge sur le réseau. Lorsque la charge sur les réseaux IMS varie l'IMD peut atteindre 3.8 s avec un stress de 1200 appel/s. on peut dire que ces valeurs sont acceptables étant données que la messagerie instantanée est l'un des services les moins exigeants en matière de la QoS.

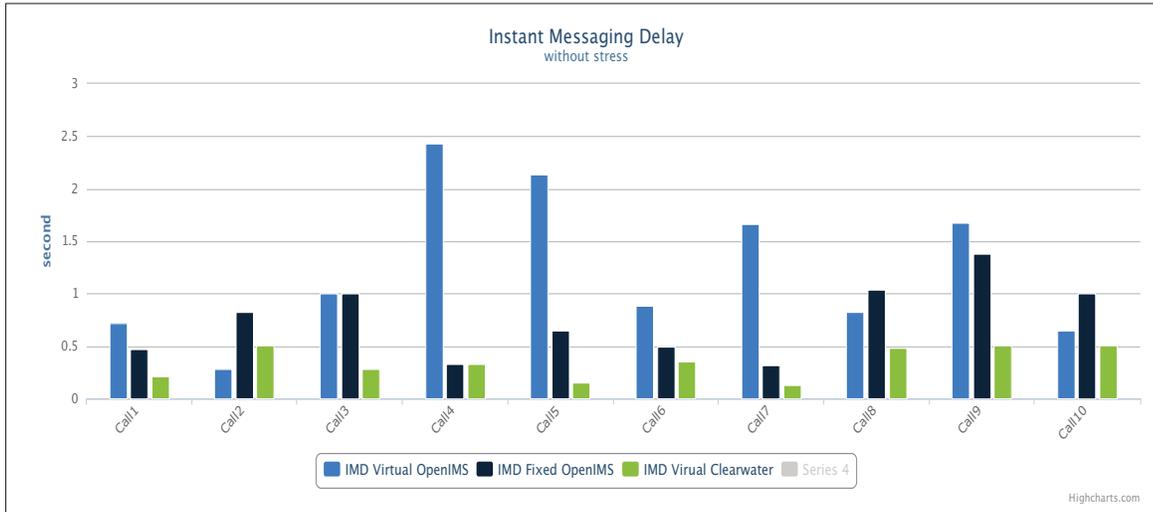


Figure 5.23 Variation du délai d'établissement d'une session donnée (IMD) sans stress SIP

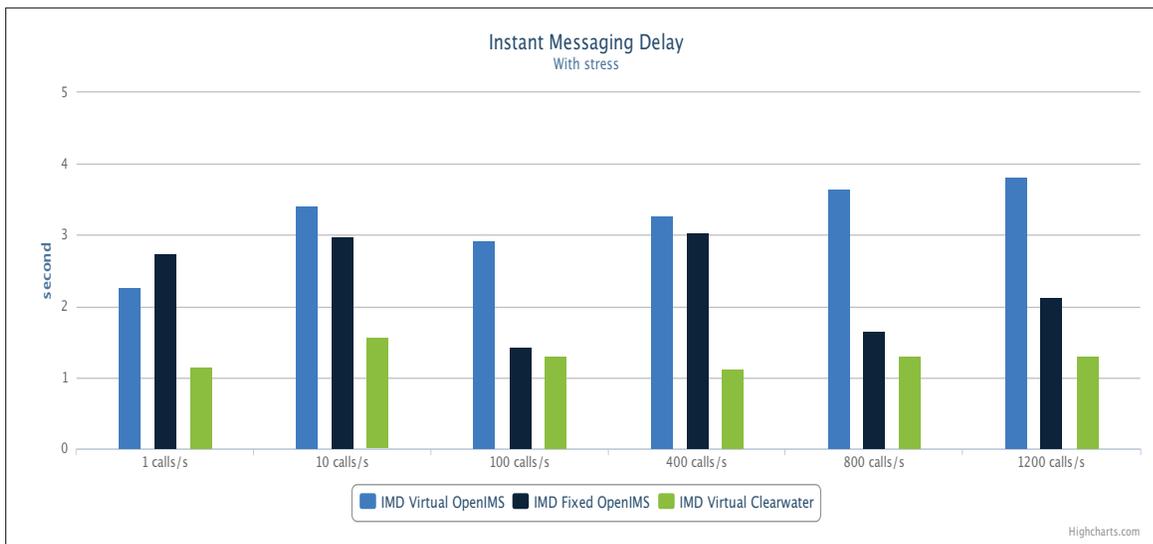


Figure 5.24 Variation du délai d'établissement d'une session donnée (IMD) avec stress SIP

5.5.4 Analyse de flux RTP d'une session audio

Nous avons analysé le flux RTP capturé pour les appels audio lors du test des différents systèmes. La figure Fig.5.25 illustre la fenêtre de l'analyse du flux RTP par wireshark capturé pour le banc d'essai "Cloud OpenIMS". Pour chaque flux audio de base nous avons capturés :

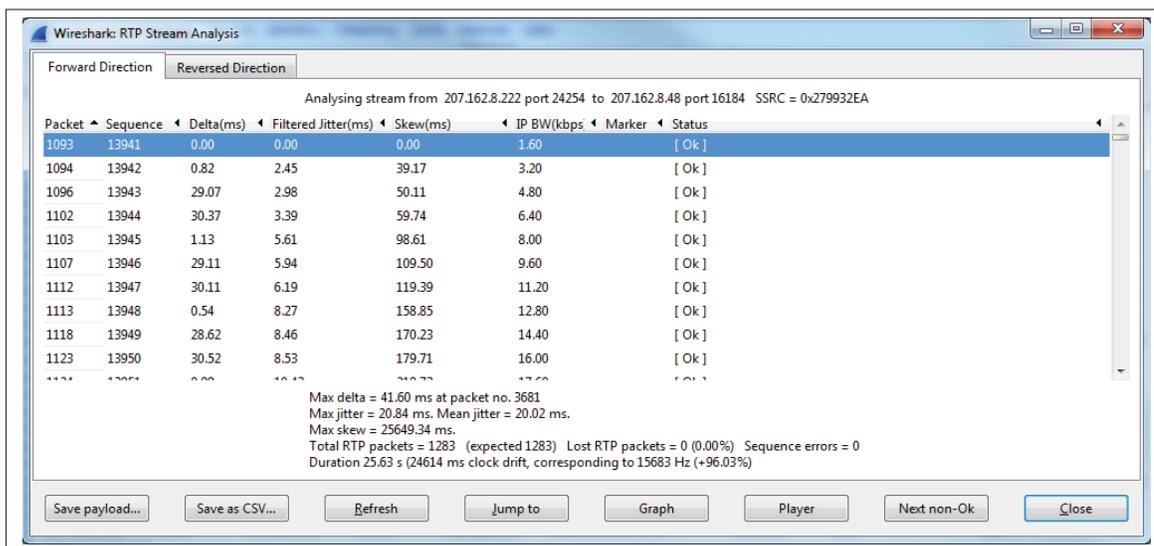


Figure 5.25 Analyse de flux RTP par wireshark

nombre de paquets et le numéro de séquence, l'heure d'arrivée, le délai, la gigue, la taille des paquets, etc.

En plus, des statistiques globales sont analysées comme le délai maximum, gigue maximum et moyenne et le nombre total de paquet RTP pour l'appel audio. Aussi une indication de paquets perdus et la durée de l'appel est incluse. Les résultats montrent un PLR inférieur à 0% avec un délai bouche-à-oreille inférieur à 41.60 ms et une gigue maximale de 20.84 ms avec un codec audio G.722. On peut conclure que la qualité de la voix est acceptable pour offrir une très bonne performance.

Analyse de délai pour le flux RTP d'un appel téléphonique

Lors de la transmission, plusieurs traitements qui sont effectués sur les paquets RTP, éventuellement la voix est codée et compressée, avant d'être transmise. Ce traitement engendre un délai supplémentaire pour chaque paquet. La variation de ce délai pour l'appel audio dans les trois bancs d'essai est illustré par la Fig.5.26. En comparaison avec le standard et avec les résultats obtenus dans système IMS fixe du banc d'essai 3, les résultats obtenus dans les deux bancs d'essais d'IMS dans le nuage sont adéquats.

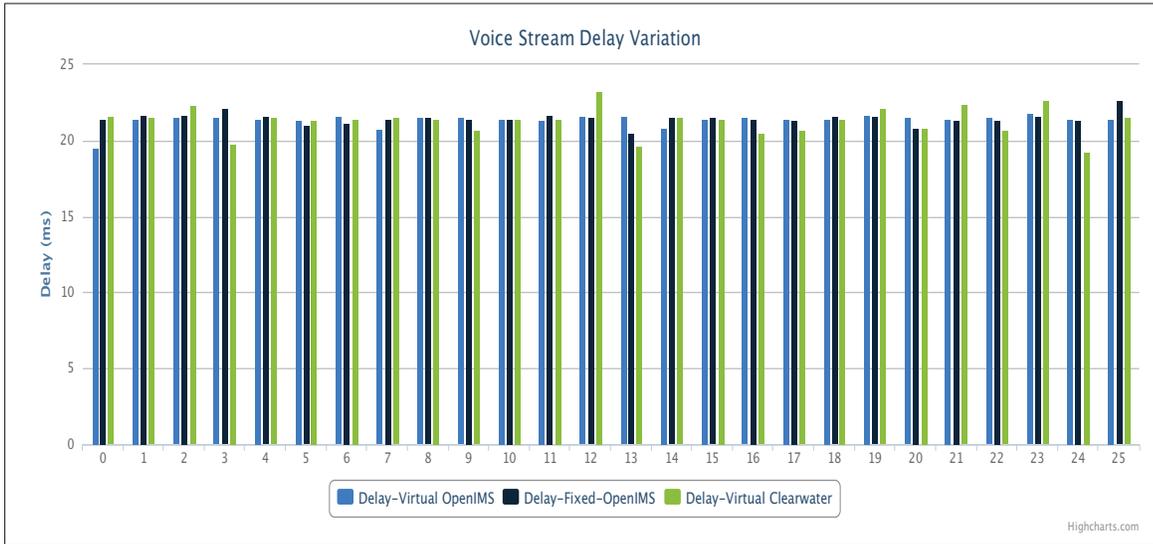


Figure 5.26 Variation de délaipour le flux audio

Analyse de Jitter pour le flux RTP d'un appel téléphonique

En comparant la commutation de paquet avec la commutation de circuit pour l'acheminement de la voix on remarque que la contrainte se situe au niveau transport ou le ratio réseau de la commutation de paquet introduit une grande gigue. L'inter arrivée de giges mesurée pour

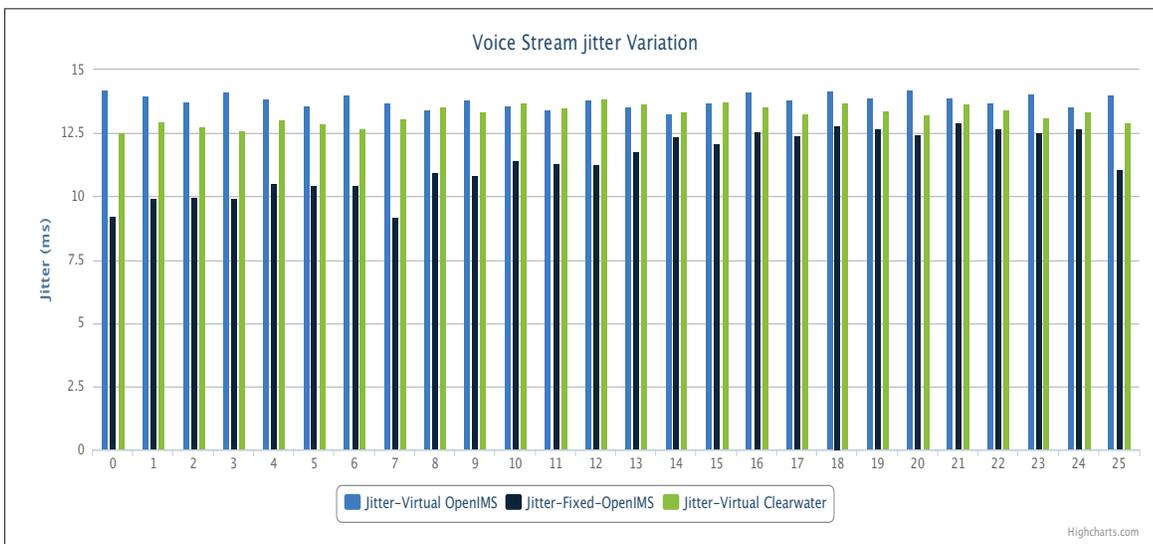


Figure 5.27 Variation de la Gigue pour le flux audio

l'appel voix et sa variation est illustrée par la Fig.5.27 est tout à fait adéquate, car elle est bien dans le délai spécifié comme idéal pour la qualité de la voix.

Analyse de la bande passante pour le flux RTP d'un appel téléphonique

La bande passante pour la voix sur IP dépend du codec utilisé durant la communication dans notre cas nous avons utilisé le code G722.

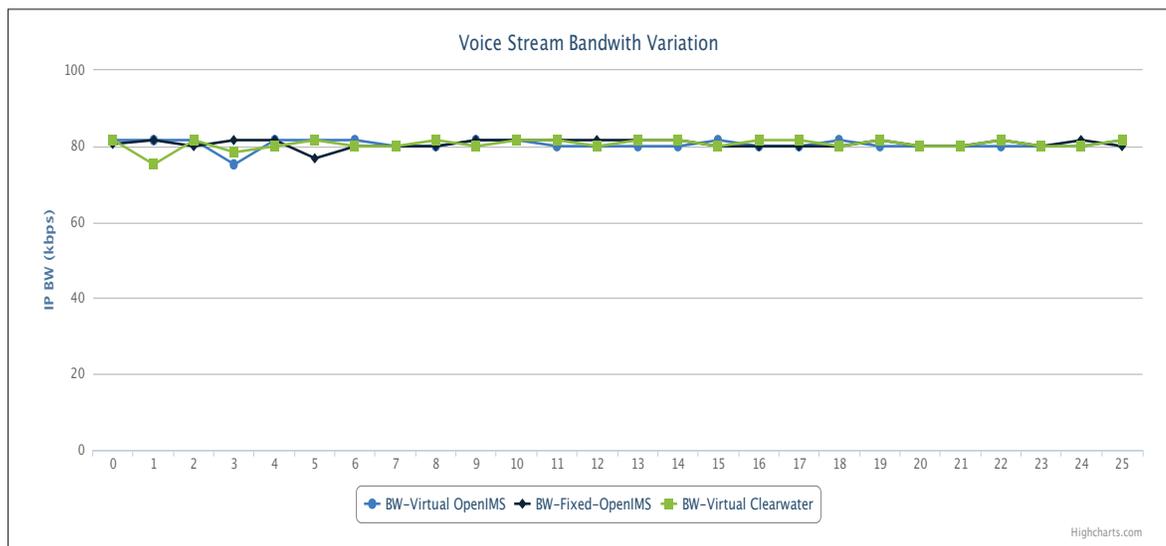


Figure 5.28 Variation de la bande passante pour le flux audio

Les résultats obtenus sont acceptables et respectent bien la norme la bande passante variée entre 75.2 kbps et 81.6 kbps comme le montre la Fig.5.28. Il faut noter que le codec G722 qui utilise deux fois plus de bande passante par rapport autre codec audio comme GSM était donnée qu'il offre une bonne qualité audio.

5.5.5 Analyse de l'utilisation de CPU et mémoire

Pour analyser le comportement de la consommation des ressources matérielles et virtuelles des différentes architectures. Nous avons effectué une série de tests en se basant sur une charge de trafic dynamique. Le trafic généré commence à partir de 10 appels/s à 1200 appels/s. Nous supervisons les deux ressources : le CPU et la mémoire de l'hôte physique pour les deux

architectures.

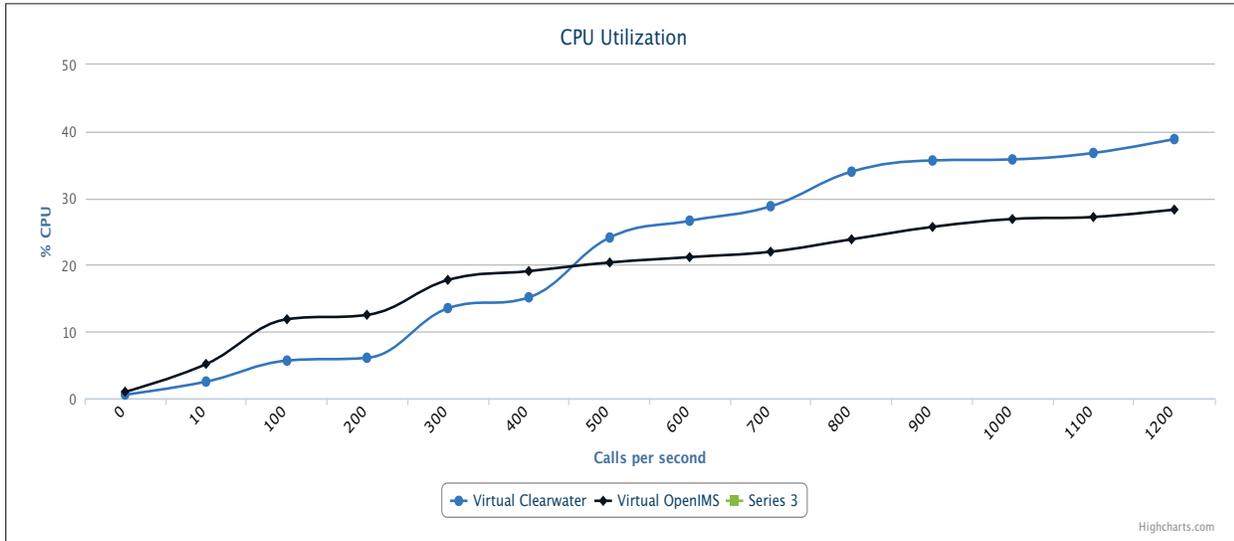


Figure 5.29 Utilisation de CPU

Fig.5.29 et Fig.5.30 illustrent respectivement le pourcentage de CPU et la mémoire utilisée par les serveurs physiques qui hébergent les machines virtuelles basées sur une charge de demande dynamique.

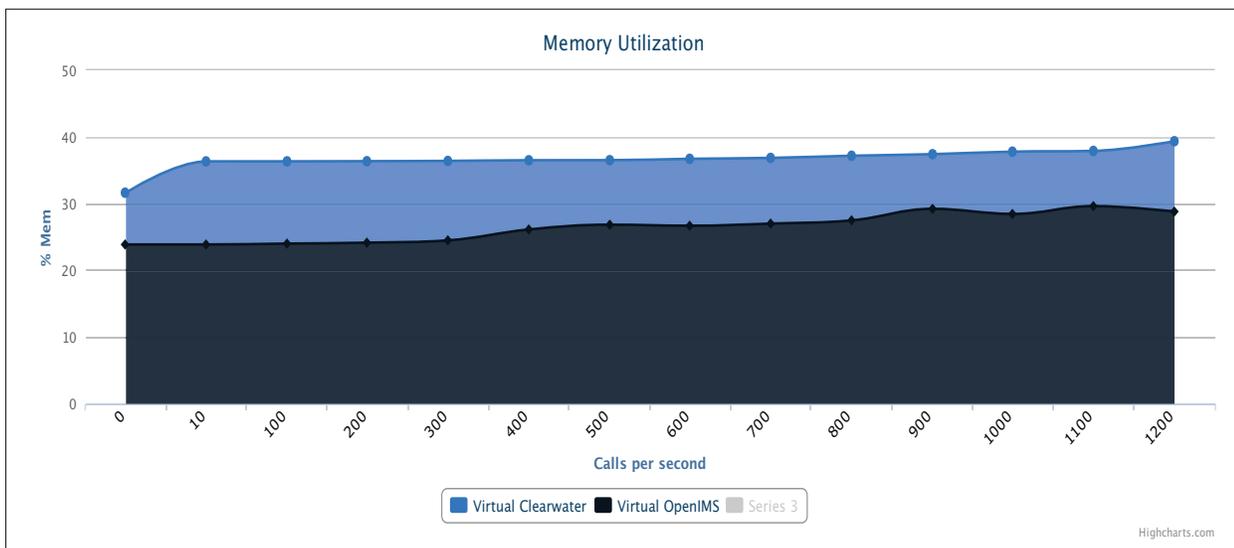


Figure 5.30 Utilisation de la mémoire physique

En général, le pourcentage d'utilisation des ressources varie en fonction du nombre d'utilisateurs connectés au système IMS. Ainsi une augmentation en conséquence du nombre d'utilisateurs enregistrés entraîne une augmentation des pourcentages des ressources pour satisfaire tous les clients. Par exemple, pour une faible charge de trafic sur le système IMS (10 appels / s) l'utilisation totale du CPU est inférieure à 10 % pour les 12 processeurs. Avec un trafic régulier de 450 appels / s le pourcentage de CPU utilisé est de 20 % pour les deux systèmes IMS.

Lorsque le trafic est lourd, plus de 500 appels /s le pourcentage total de CPU utilisé par machine physique qui héberge OpenIMS est moins de celle qui héberge Clearwater IMS. Ce résultat est dû au nombre de machines virtuelles qui déploient Clearwater IMS et qui sont cinq machines (5VMs) et ce nombre est supérieur à celui utilisé pour déployer Open source IMS Core (3VMs). De même, en observant le pourcentage d'utilisation de la mémoire, Clearwater IMS utilise plus de ressources qu'OpenIMS comme le montre la Fig.5.30. Ceci est dû à la même raison, le nombre de machines virtuelles en cours d'exécution pour chaque système.

5.5.6 Analyse de l'utilisation de CPU et mémoire virtuelle

Étant donné, que chaque machine déploie un ou plusieurs composants d'IMS nous avons surveillé l'utilisation des ressources virtuelles de chaque VM séparément dans les deux architectures implémentées. Les résultats illustrés par les Fig.5.31 et Fig.5.32 représentent respectivement la VCPU et la mémoire virtuelle des OpenIMS alors que les figures illustrent respectivement Fig.5.34 et Fig.5.33 donne les résultats pour Clearwater IMS.

L'utilisation de la ressource virtuelle de chaque composant IMS augmenté simultanément avec l'augmentation de nombre d'abonné enregistrée dans le système IMS. Dans les deux, l'architecture d'IMS déployé dans le nuage l'entité qui utilise le plus de ressources en matière de VCPU et de la mémoire virtuelle est la base de données HSS. Ce composant représente l'élément centrale du système IMS, il contient toutes les informations relatives aux profils des abonnés, la localisation actuelle et les services offerts. Le HSS effectue aussi la génération des informations de sécurité pour chaque utilisateur connecté au système IMS.

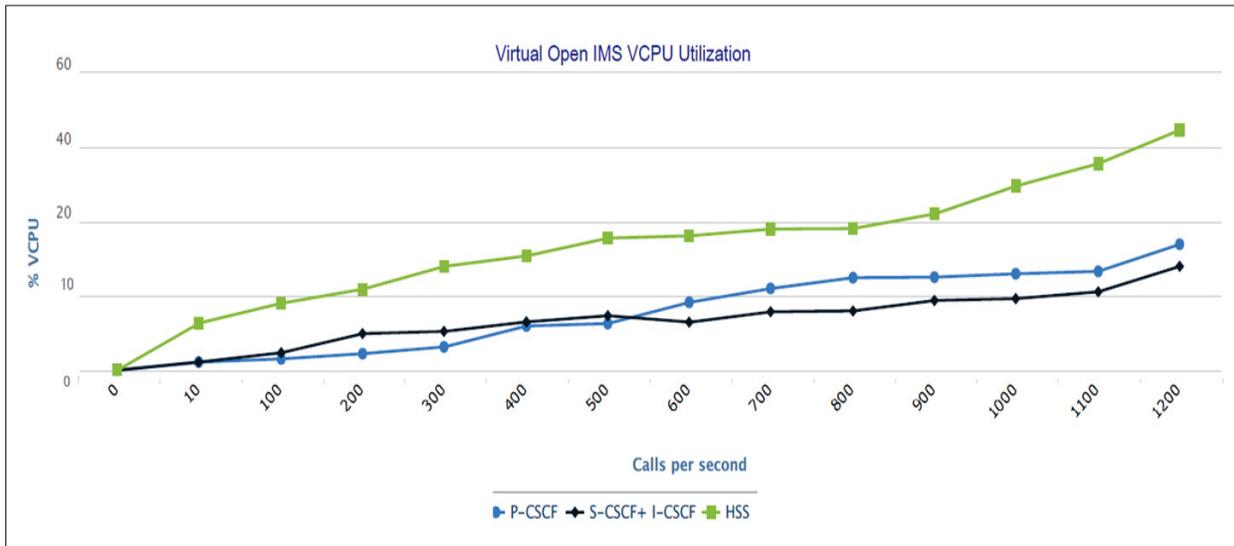


Figure 5.31 Utilisation de VCPU : OpenIMS

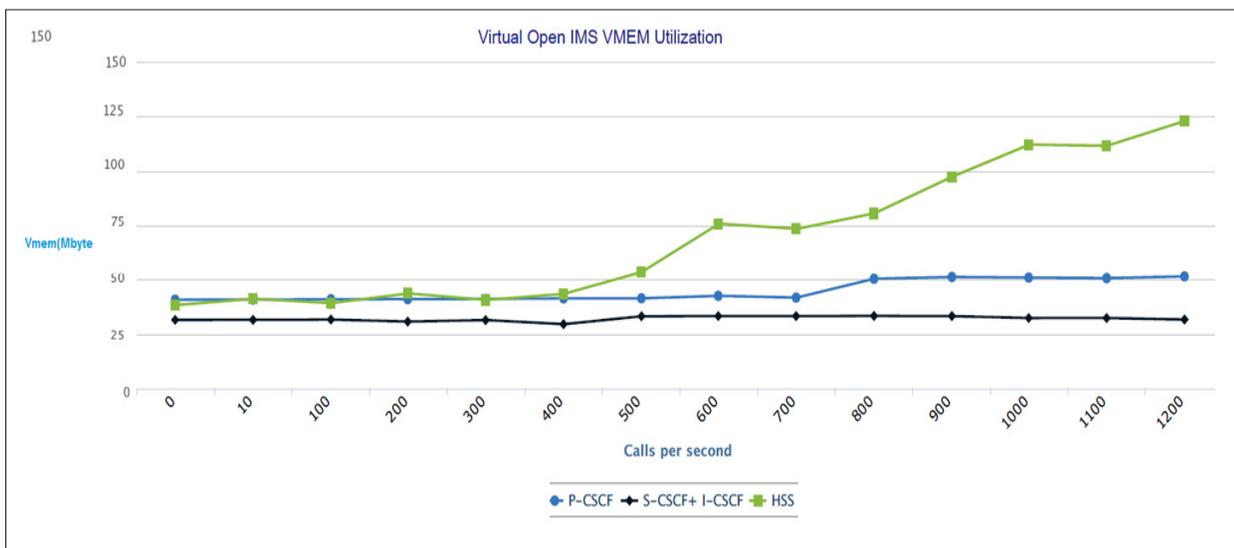


Figure 5.32 Utilisation de mémoire virtuelle : OpenIMS

À partir de ces résultats nous avons conclu que HSS a besoin de plus de ressources pour traiter ces nombreuses opérations et qu'il est absolument nécessaire de provisionner les ressources cette dernière.

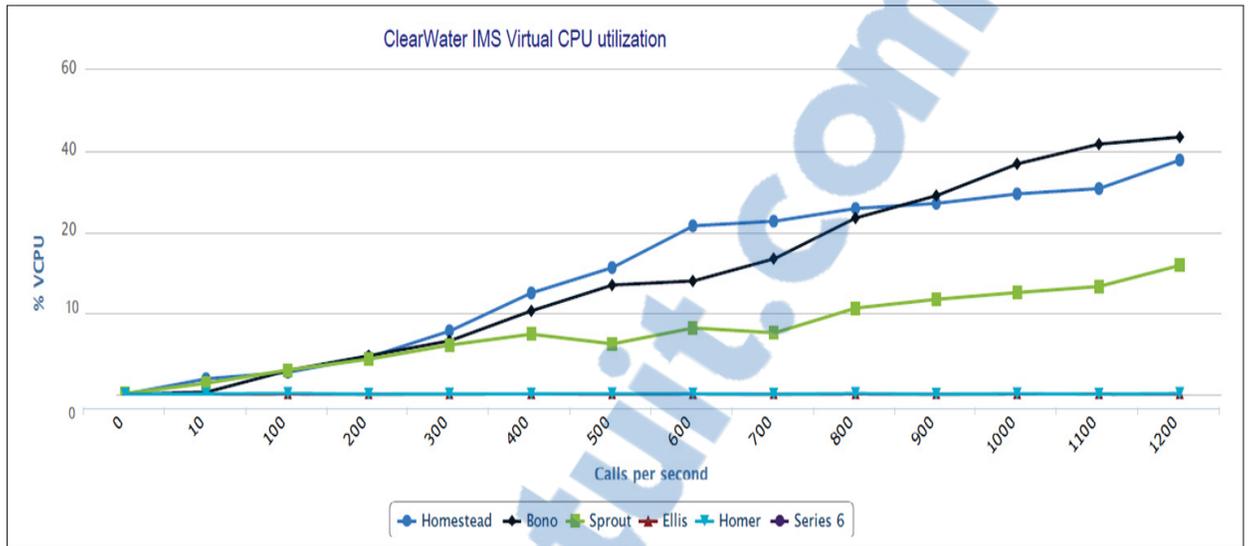


Figure 5.33 Utilisation du VCPU : Clearwater IMS

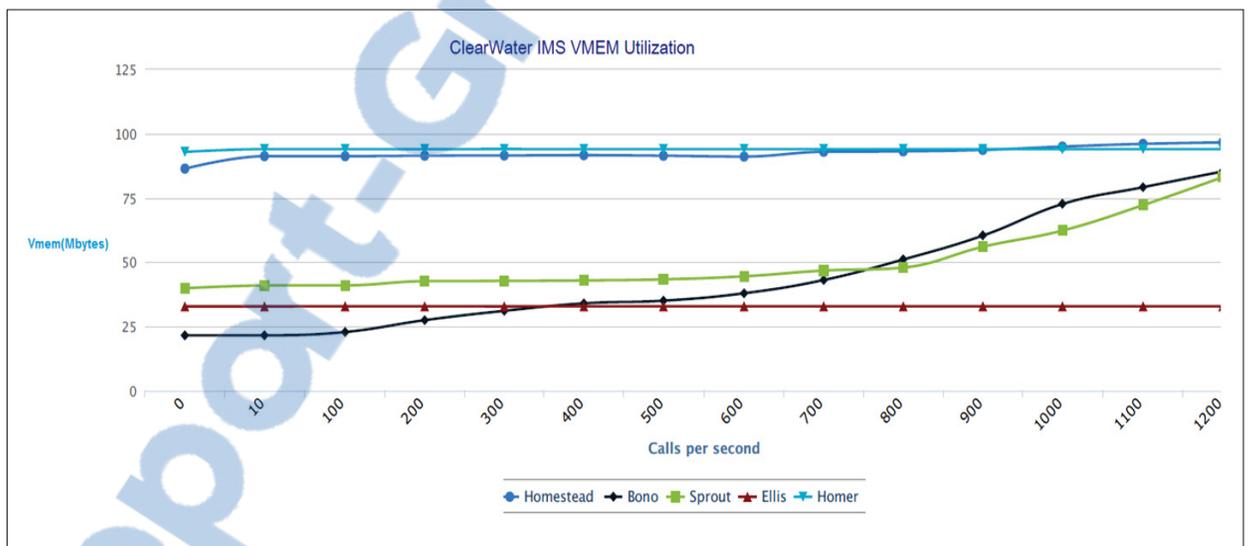


Figure 5.34 Utilisation de la mémoire virtuelle : Clearwater IMS

5.5.7 Comparaison des systèmes IMS déployés dans le nuage informatique

L'analyse de la QoS des deux systèmes IMS dans le nuage montre que la variation des métriques de la QoS délai d'enregistrement RRD et délai d'établissement d'une session SRD donnent des résultats proche à ceux du système déployé dans des serveurs physique. Ceci justifie la possibilité de la migration l'architecture vers le nuage informatique. Toutefois, Cloud

Clearwater IMS donne une meilleure performance que Cloud OpenIMS et Traditionnel OpenIMS. Ceci est dû à la nouveauté de la solution et aux différentes solutions technologiques utilisées par cette dernière comme la base de données Cassandra qui est plus performante que celle de MySQL utilisée par open source IMS.

Concernant la consommation des ressources physiques et virtuelles, plus le nombre des appels téléphoniques augmente plus le pourcentage total d'utilisation des ressources processeur et mémoire utilisée augmente en parallèle.

Globalement, les résultats montrent que Clearwater IMS utilise plus de ressources qu'OpenIMS vu qu'il est déployé sur plus de VM que open source IMS. Nous sommes certains que nous pouvons atteindre 100% des ressources utilisées par l'augmentation de la charge de plus de 1200 appels par seconde.

5.6 Conclusion

Dans ce chapitre nous avons présenté les différentes architectures d'IMS déployées, les deux premiers bancs d'essai sont deux cas d'études pour analyser et valider la possibilité de la cloudification du système Telco IMS dans le nuage. Le troisième banc du test est une implémentation d'IMS fixe. Le test de fonctionnement des architectures dans le nuage est effectué avec succès. Les résultats obtenus montrent que les performances des systèmes IMS dans le nuage sont similaires voire meilleurs que celui d'IMS fixe.

L'analyse des métriques de performances RRD et SRD montre que "Cloud Clearwater IMS" du second banc de test est plus performant que les deux autres architectures. Ce résultat est logique dû à la nouveauté de cette solution et à la performance des techniques utilisées dans ce système en comparaison avec l'ancien framework open source IMS publié en 2005 là où ils n'ont pas pris en compte les performances lors de leurs implémentations.

En ce qui concerne la variation des ressources requises pour chaque instance nous avons constaté que dans les trois bancs d'essais, l'élément qui nécessite plus de ressources virtuelles pour le traitement des différentes opérations est la base de données.

CHAPITRE 6

CONCLUSION

Les opérateurs de téléphonie ont souffert cette dernière décennie d'une diminution de leurs revenus et une augmentation des dépenses, d'une part parce qu'avec l'évolution de la téléphonie mobile a poussé les abonnés du téléphone fixe à abandonner complètement l'usage du téléphone fixe pour passer au téléphone mobile et d'autre part, l'absence de valeur ajoutée des réseaux qui attire la clientèle. Ajouter à tout cela la difficulté du processus d'intégration de nouveaux services dans les réseaux d'opérateurs qui peut prendre même des mois engendrant un coût considérable pour les opérateurs.

Dans ce mémoire, nous avons abordé les problèmes liés au provisionnement de la qualité de services et à la complexité de déploiement des applications et des services dans le système IMS au sein du projet Ecolotic d'Ericsson. Et pour résoudre ces problèmes nous avons choisi de migrer IMS dans l'environnement de nuage informatique.

Le passage à des services de nuages offre divers avantages non seulement pour l'opérateur, mais aussi pour les utilisateurs finaux. Ainsi, à travers la cloudification des services et des équipements, l'opérateur télécom va gérer mieux les ressources réseaux et intégrer rapidement de nouveaux services avec moins d'intervention humaine et avec un coût compétitif.

Nous avons commencé également par déployer trois bancs d'essai de l'architecture d'IMS sur l'infrastructure de production de la compagnie Ericsson. Pour cela nous avons utilisé des produits IMS open sources qui sont : open source IMS Core (FOKUS, 2006) et Clearwater IMS (Metaswitch, 2013). Les deux solutions choisies utilisent des technologies différentes tout en respectant les exigences du standard IMS. En tant qu'infrastructure de service (IaaS) nous avons aussi utilisé l'intergiciel libre OpenStack (Rackspace, 2013).

Dans les deux premiers bancs d'essai «Cloud OpenIMS» et «Cloud Clearwater IMS», nous avons implémenté IMS dans l'écosystème de nuage d'Ericsson. Le déploiement des deux sys-

tèmes dans le nuage a été effectué avec succès et le test de fonctionnement a été réalisé avec l'utilisation des produits logiciels libres en tant que client IMS.

Le banc d'essai 3 «Traditionnal OpenIMS» est une implémentation d'open source IMS core sur l'infrastructure physique. Son objectif principal est de construire un cas d'utilisation du système IMS fixe déployé sur des serveurs physiques, servi comme un banc d'essai de comparaison avec les deux architectures au sein de l'environnement de nuage informatique. Le test de fonctionnement de cette architecture d'IMS a été effectué avec l'utilisation des mêmes clients IMS.

Lors du déploiement des architectures d'IMS dans le nuage nous avons testé plusieurs scénarios de déploiement et nous avons conclu que les performances sont mieux, si les modules de chaque élément est installé sur une machine virtuelle séparée avec une connectivité réseau à haut débit.

La plupart des problèmes rencontrés durant le déploiement sont liés à la configuration des composants d'open source IMS core qui a pris beaucoup de temps. Cette solution est une ancienne implémentation publiée en 2005, elle est composée par CSCFs et HSS qui sont les composants de base de l'architecture d'IMS telle que définie dans NGN. Il est développé à l'aide de la technologie JAVA à des fins de recherche et de test d'interopérabilité pour les réseaux IMS fixes dont la performance n'est pas pris en considération.

En revanche, la configuration de Clearwater IMS à occupé moins de temps que celle d'open source IMS Core, ceci est fortement lié à l'automatisation de l'installation de la plupart des entités. Clearwater IMS est une nouvelle implémentation publiée en mai 2013 et qui contient les mêmes composants de base d'IMS le CSCFs et le HSS. Il a été développé dès le départ pour un déploiement massivement évolutif dans le nuage informatique afin de fournir la voix, la vidéo et les services de messagerie pour servir des millions d'utilisateurs. Il est conçu pour être bien adopté pour le déploiement dans un environnement réseau avec la fonction de virtualisation (NFV).

En général, les mesures ont montré que la variation des deux délais d'enregistrement (RRD) et le délai d'établissement d'une session (SRD) sont liés au nombre d'utilisateurs connectés aux réseaux d'opérateurs. De même pour les ressources virtuelles (CPU et mémoire) utilisées par les entités d'IMS qui sont hébergées sur des machines virtuelles. En augmentant le nombre d'appels téléphoniques sur le réseau IMS, nous aurons une augmentation des valeurs de métriques du plan de contrôle (RRD, SRD, IMD) et plan de données (RTP delay, RTP Inter arrival jitter). Cette augmentation du nombre d'utilisateurs procréé une croissance semi-linéaire des ressources requises par chaque machine virtuelle.

Travaux futurs

Le processus de déploiement d'IMS et de provisionnement de la QoS dans le nuage informatique est un processus long et complexe. Nos travaux futurs visent à :

- automatiser le processus de déploiement et de configuration des différents composants d'IMS. On a proposé dans (Ridha *et al.*, 2013) un modèle générique de déploiement automatique des applications dans le nuage OpenICRA qui a été testé avec l'intergiciel distribué OpenSAF et le système de travail collaboratif ICRA. Nous allons utiliser cette plateforme pour la migration d'IMS dans le nuage. Il pourrait être utile aussi d'utiliser software defined network (SDN) pour l'automatisation de la configuration ;
- améliorer notre modèle de provisionnement de la QoS dans IMS. Nous proposons d'ajouter d'autres règles dans le module de "Scaling" afin d'améliorer la méthode de gestion des ressources. D'ailleurs il sera utile si on prend considération la qualité de services perçue ;
- étudier le provisionnement de la QoS dans le plan de données, plusieurs techniques peuvent être utilisées y compris classificateur de paquets et la planification ;
- étudier l'équilibrage de la charge dans IMS dans le but d'offrir l'évolutivité au composant de base d'IMS. Il y a deux possibilités, la première en s'appuyant sur un équilibrage de charge SIP pour le P-CSCF et la seconde la mise en œuvre d'une solution d'équilibrage de charge pour les I-CSCF et S-CSCF en utilisant SDN. Les auteurs dans (Ito *et al.*,

2013) ont proposé un mécanisme d'équilibrage de charge basé sur OpenFlow qui est une implémentation open source de SDN ;

- étudier les solutions qui permettent d'offrir une haute disponibilité d'IMS afin d'éviter la dégradation des performances et réagir aux changements imprévisibles comme la croissance de la charge de trafic, l'augmentation des besoins en ressources et aux pannes des périphériques dans le réseau. Dans (Fakhfakh *et al.*, 2009) les auteurs proposent d'introduire la technique de la migration au sein de l'architecture IMS pour résoudre le problème de surcharge et de garantir la qualité de service.

ANNEXE I

INSTALLATION D'OPEN SOURCE IMS CORE

1. Exigences de Open source IMS core

Il est préférable d'utiliser la dernière version de Linux server(Ubuntu 12.04)

Accès réseaux

- une adresse IP publique ;
- un serveur DNS exemple Bind.

Exigences logiciels

- GCC3/4, make, JDK1.5, ant ;
- mySQL server, bison, flex, libxml2, libmysql ;
- ipsec-tools, curl and libcurl4-gnutls-dev.

2. Téléchargement du code source

Créer une nouvelle répertoire **ser_ims** et télécharger le code de CSCFs :

- `mkdir ser_ims ;`
- `svn checkout https://svn.code.sf.net/p/openimscore/code/ser_ims/trunk ser_ims.`

Créer une nouvelle répertoire **FHoSS** et télécharger le code de HSS :

- `mkdir FHoSS ;`
- `svn checkout https://svn.code.sf.net/p/openimscore/code/FHoSS/trunk FHoSS.`

3. Compilation du Code

Pour compiler le code suivez les étapes suivantes :

- `cd ser_ims ;`
- `make install-libs all ;`
- `cd .. ;`
- `cd FHoSS ;`
- `ant compile ;`
- `ant deploy ;`
- `cd .. .`

4. Configuration de l'environnement

Configuration de DNS :

Par défaut open ims core est configuré pour être exécuté localement, un simple zone DNS est dans le fichier suivant *ser_ims/cfg/open-ims.dnszone*

Configuration de MySQL :

- `mysql -u root -p -h localhost < ser_ims/cfg/icscf.sql ;`
- `mysql -u root -p -h localhost < FHoSS/scripts/hss_.db.sql ;`
- `mysql -u root -p -h localhost < FHoSS/scripts/userdata.sql.`

5. Configuration de open IMS Core

Copier les fichiers suivant dans `/opt/OpenIMSCore`

- `cp ser_ims/cfg/*.cfg /opt/OpenIMSCore ;`
- `cp ser_ims/cfg/*.xml /opt/OpenIMSCore ;`
- `cp ser_ims/cfg/*.sh /opt/OpenIMSCore.`

6. Démarrer les composants

Démarrer les composants CSCFs :

- Les trois composants doivent être exécutés en parallèle ;
- Start pcsf.sh, icsf.sh and scscf.sh.

Démarrer le FHoSS :

- start FHoSS/deploy/startup.sh ;
- Vérifier que l'interface web de la base de données est en cours d'exécution. **http://localhost:8080/**

7. Configuration des abonnés

Par défaut existe deux utilisateurs (Alice et Bob) enregistrés dans la base de données FHoSS :

- alice@open-ims.test ;
- bob@open-ims.test.

Pour ajouter des nouveaux abonnés au HSS il faut suivre les étapes suivantes :

- création de l'abonné ;
- création de l'identité privé ;
- création de l'identité public.

8. Test

Utiliser Wireshark pour collecter le trafic :

- surveiller le trafic SIP des ports 4060, 5060 and 6060 ;

- surveiller le trafic Diameter ports 3868, 3869 and 3870.

9. Configuration de l'environnement

Configuration de DNS :

Par défaut open ims core est configuré pour être exécuté localement, un simple zone DNS est dans le fichier suivant *ser_ims/cfg/open-ims.dnszone*

Configuration de MySQL :

- `mysql -u root -p -h localhost < ser_ims/cfg/icscf.sql ;`
- `mysql -u root -p -h localhost < FHoSS/scripts/hss_.db.sql ;`
- `mysql -u root -p -h localhost < FHoSS/scripts/userdata.sql.`

10. Configuration de open IMS Core

Copier les fichiers suivant dans `/opt/OpenIMSCore`

- `cp ser_ims/cfg/*.cfg /opt/OpenIMSCore ;`
- `cp ser_ims/cfg/*.xml /opt/OpenIMSCore ;`
- `cp ser_ims/cfg/*.sh /opt/OpenIMSCore.`

11. Démarrer les composants

Démarrer les composants CSCFs :

- Les trois composant doivent être exécuté en parallèle ;
- Start `pcscf.sh`, `icscf.sh` and `sccsf.sh`.

Démarrer le FHoSS :

- Start `FHoSS/deploy/startup.sh ;`

- Verifier que l'interface web de la base de données est en cours d'exécution : **http ://localhost :8080/**.

12. Configuration des abonnés

Par défaut existe deux utilisateurs(Alice et Bob) enregistré dans la base de donnée FHoSS :

- alice@open-ims.test ;
- bob@open-ims.test.

Pour ajouter des nouveaux abonnés au HSS il faut suivre les étapes suivante :

- création de l'abonnée ;
- création de l'identité privé ;
- création de l'identité public.

13. Test

Utiliser Wireshark pour collecter le trafic :

- surveiller le trafic SIP des ports 4060, 5060 and 6060 ;
- surveiller le trafic Diameter ports 3868, 3869 and 3870.

ANNEXE II

INSTALLATION DE CLEARWATER IMS

Dans cet annexe nous allons élaborer les étapes nécessaire pour installer et configurer Clearwater IMS Avant de passer au processus d'installation nous devons s'assure que nous satisfaisons tout les pré-requit.

- avoir cinq machines avec Ubuntu 12.04, 64bit comme système d'exploitation ;
- chaque machine doit posséder deux adresses IP, une accessible au public et la deuxième accessible au réseau privé ;
- avoir le nom de domaine complet de la machine, qui se résout en adresse IP publique de la machine, sinon, utilisez l'adresse IP publique ;
- accès SSH aux machines à un utilisateur autorisé à utiliser sudo pour accéder a distance.

0.1 Amorcer les Machines

Il y'a quelques étapes qui sont communes à tous le noeud Clearwater a savoir la configuration des sources de logiciels APT, fournir les adresses IP ou noms d'hôte DNS à utiliser pour garantir la communication entre les hôtes. La configuration du APT : Les machines doivent être configurés de telle sorte que APT peut utiliser la dépendance du server Clearwater.

- Sous sudo, créer /etc/apt/sources.list.d/clearwater.list avec le contenu suivant : " deb http://repo.cw-ngv.com/stable binary/ " ;
- reindexé les packages du ubuntu par un update : " sudo apt-get update " .

Pour établir la commuication entre noeud nous devons creer un fichier de configurer le repertoire Clearwater "/etc/clearwater/config" et saisir la configuration suivante :

```
# Deployment definitions
```

```
home_domain=<zone>
```

```
sprout_hostname=sprout.<zone>
chronos_hostname=<privateIP> :7253
hs_hostname=hs.<zone> :8888
hs_provisioning_hostname=hs.<zone> :8889
ralf_hostname=ralf.<zone> :10888
xdms_hostname=homer.<zone> :7888
# Local IP configuration
local_ip=<privateIP>
public_ip=<publicIP>
public_hostname=<hostname>
# Email server configuration smtp_smarthost=<smtp server>
smtp_username=<username>
smtp_password=<password>
email_recovery_sender=clearwater@example.org
# Keys
signup_key=<secret>
turn_workaround=<secret>
ellis_api_key=<secret>
ellis_cookie_key=<secret>
# activer la fonction option I-CSCF et/ou external HSS lookups, ajouter également les éléments
suivants :

I-CSCF/S-CSCF configuration
icscf=5052
upstream_hostname=<sprout_hostname>
upstream_port=5052
# HSS configuration
hss_hostname=<address of your HSS>
hss_port=3868
```

Si vous voulez héberger plusieurs domaines du même déploiement Clearwater, ajouter ce qui suit (et configurer DNS pour acheminer tous les domaines aux mêmes serveurs) :

Additional domains

additional_home_domains=<domain 1>,<domain 2>,<domain 3>...

Si vous voulez que vos nœuds incluent les serveurs d'applications Gemini/Memento ajouter ce qui suit :

Application Servers

gemini_enabled='Y'

memento_enabled='Y'

0.2 Étape d'installation de nœud spécifique

À ce stade, vous devez décider qui des six machines aura quel rôle du Clearwater.

Les six rôles sont :

- ellis ;
- bono - This role also hosts a restund STUN server ;
- sprout ;
- homer ;
- homestead.

0.2.1 Ellis

Installer Ellis package : " sudo DEBIAN_FRONTEND=noninteractive apt-get install ellis -yes
"

" sudo bash -c "export PATH=/usr/share/clearwater/ellis/env/bin :\$PATH cd /usr/share/clearwater/ellis/src/metaswitch/ellis/tools/ python create_numbers.py -start 6505550000 -count 1000"

" Apres l'exécution de cette commande vous devez avoir ce resultat : "Created 1000 numbers, 0 already present in database"

0.2.2 bono

Installer bono package :

```
" sudo DEBIAN_FRONTEND=noninteractive apt-get install bono restund -yes "
```

0.2.3 sprout

Installer sprout package :

```
" sudo DEBIAN_FRONTEND=noninteractive apt-get install sprout -yes "
```

si vous voulez activer Gemini server : " sudo DEBIAN_FRONTEND=noninteractive apt-get install clearwater-cassandra -yes "

```
"sudo DEBIAN_FRONTEND=noninteractive apt-get install memento memento-nginx -yes"
```

0.2.4 homer

Installer homer package :

```
" sudo DEBIAN_FRONTEND=noninteractive apt-get install clearwater-cassandra -yes "
```

```
"sudo DEBIAN_FRONTEND=noninteractive apt-get install homer -yes "
```

0.2.5 homestead

Installer homestead package :

```
" sudo DEBIAN_FRONTEND=noninteractive apt-get install clearwater-cassandra -yes "
```

```
"sudo DEBIAN_FRONTEND=noninteractive apt-get install homestead homestead-prov -yes "
```

```
"
```

%subsectionMaking calls through Clearwater

ANNEXE III

CONFIGURATION DE SIPP TRAFFIC GENERATOR

SIPP est un outil de test gratuit et open source, générateur de trafic pour le protocole SIP. Il comprend quelques scénarios de base de l'agent utilisateur (UAC et UAS) et établit et libère de multiples appels avec les méthodes INVITE et BYE.

1. Installation de SIPP

SIPP est fourni sous la forme de code source. À fin de l'utiliser il est nécessaire de le compiler SIPP. D'abord il faut installer les librairie suivant :

- compilateur C++ ;
- librairie curses ou ncurses ;
- support TLS : OpenSSL >= 0.9.8 ;
- support pcap : libpcap et libnet ;
- support SCTP : lksctp-tools.

Compillation de SIPP :

```
# tar -xvzf sipp-xxx.tar
# cd sipp
# ./configure
# make
```

2. Utilisation de SIPP

Exemple 1 : Exécutez Sipp avec 100 calls chaque 1 seconds

```
# cd /opt/sipp-3.3beta1/
# ./sipp -sf bob-invites-alice-client.xml -r 100 -rp 1000 172.16.0.207 :4060
```

Exemple 2 : Utilisation des fichiers xml par défaut pour le client et les serveurs

Exécutez Sipp avec le serveur (UAS) scénario : # ./sipp UAS -sn

Sur la machine cliente, exécuter Sipp avec le client intégré (UAC) scénario

./sipp -sn UAC 172.16.0.15

Fichier XML : Enregistrement d'Alice

```

1
2 <?xml version="1.0" encoding="ISO-8859-1" ?>
3 <!DOCTYPE scenario SYSTEM "sipp.dtd">
4
5 <scenario name="registration">
6
7 <send retrans="500">
8 <![CDATA[
9 REGISTER sip:open-ims.test SIP/2.0
10 Via: SIP/2.0/[transport] [local_ip]:[local_port];branch=[branch]
11 Max-Forwards: 20
12 From: "alice" <sip:alice@open-ims.test>;tag=[call_number]
13 To: "alice" <sip:alice@open-ims.test>
14 P-Access-Network-Info:
    3GPP-UTRAN-TDD;utran-cell-id-3gpp=C359A3913B20E
15 Call-ID: reg//[call_id]
16 CSeq: 1 REGISTER
17 Contact: <sip:alice@[local_ip]:[local_port]>
18 Expires: 300
19 Content-Length: 0
20 User-Agent: Sipp v1.1-TLS, version 20061124
21 Authorization: Digest username="alice@open-ims.test",
    realm="open-ims.test"
22 Supported: path
23 ]]>

```

```
24 </send>
25
26 <recv response="401" auth="true" rtd="true">
27 <action>
28 <ereg regexp=".*" search_in="hdr" header="Service-Route"
    assign_to="1" />
29 </action>
30 </recv>
31
32 <send retrans="500">
33 <![CDATA[
34 REGISTER sip:open-ims.test SIP/2.0
35 Via: SIP/2.0/[transport] [local_ip]:[local_port];branch=[branch]
36 Route: [$1]
37 Max-Forwards: 20
38 From: "alice" <sip:alice@open-ims.test>;tag=[call_number]
39 To: "alice" <sip:alice@open-ims.test>
40 P-Access-Network-Info:
    3GPP-UTRAN-TDD;utran-cell-id-3gpp=C359A3913B20E
41 Call-ID: reg///[call_id]
42 CSeq: 2 REGISTER
43 Contact: <sip:alice@[local_ip]:[local_port]>
44 Expires: 300
45 Content-Length: 0
46 User-Agent: Sipp v1.1-TLS, version 20061124
47 [authentication username=alice@open-ims.test password=alice]
48 Supported: path
49 ]]>
50 </send>
51
52 <recv response="200">
```

```

53 </recv>
54
55 <ResponseTimeRepartition value="10, 20"/>
56 <CallLengthRepartition value="10"/>
57
58 </scenario>
59
60 %\lstinputlisting{bob-invites-alice-client.xml}

```

Fichier XML : Enregistrement de Bob

```

1 <?xml version="1.0" encoding="ISO-8859-1" ?>
2 <!DOCTYPE scenario SYSTEM "sipp.dtd">
3
4 <scenario name="registration">
5
6 <send retrans="500">
7 <![CDATA[
8 REGISTER sip:open-ims.test SIP/2.0
9 Via: SIP/2.0/[transport] [local_ip]:[local_port];branch=[branch]
10 Max-Forwards: 20
11 From: "bob" <sip:bob@open-ims.test>;tag=[call_number]
12 To: "bob" <sip:bob@open-ims.test>
13 P-Access-Network-Info:
14   3GPP-UTRAN-TDD;utran-cell-id-3gpp=C359A3913B20E
15 Call-ID: reg//[call_id]
16 CSeq: 1 REGISTER
17 Contact: <sip:bob@[local_ip]:[local_port]>
18 Expires: 300
19 Content-Length: 0
20 User-Agent: Sipp v3.0, version 20061124

```

```
20 Authorization: Digest username="bob@open-ims.test",
    realm="open-ims.test"
21 Supported: path
22 ]]>
23 </send>
24
25 <recv response="401" auth="true" rtd="true">
26 <action>
27 <ereg regexp=".*" search_in="hdr" header="Service-Route:"
    assign_to="1" />
28 </action>
29 </recv>
30
31 <send retrans="500">
32 <![CDATA[
33 REGISTER sip:open-ims.test SIP/2.0
34 Via: SIP/2.0/[transport] [local_ip]:[local_port];branch=[branch]
35 Route: [$1]
36 Max-Forwards: 20
37 From: "bob" <sip:bob@open-ims.test>;tag=[call_number]
38 To: "bob" <sip:bob@open-ims.test>
39 P-Access-Network-Info:
    3GPP-UTRAN-TDD;utran-cell-id-3gpp=C359A3913B20E
40 Call-ID: reg//[call_id]
41 CSeq: 2 REGISTER
42 Contact: <sip:bob@[local_ip]:[local_port]>
43 Expires: 300
44 Content-Length: 0
45 User-Agent: Sipp v3.0, version 20061124
46 [authentication username=bob@open-ims.test password=bob]
47 Supported: path
```

```

48 ]]>
49 </send>
50
51 <recv response="200">
52 </recv>
53
54 <ResponseTimeRepartition value="10, 20"/>
55 <CallLengthRepartition value="10"/>
56
57 </scenario>

```

Fichier XML : Bob INVITE Alice

```

1  <?xml version="1.0" encoding="ISO-8859-1" ?>
2  <!DOCTYPE scenario SYSTEM "sipp.dtd">
3
4  <scenario name="Simple IMS Session Setup, client-side">
5
6  <send retrans="500">
7  <![CDATA[
8  INVITE sip:alice@open-ims.test SIP/2.0
9  Via: SIP/2.0/[transport] [local_ip]:[local_port];branch=[branch]
10 Max-Forwards: 20
11 Route: <sip:orig@scscf.open-ims.test:6060;lr>
12 P-Preferred-Identity: <sip:bob@open-ims.test>
13 Privacy: none
14 P-Access-Network-Info:
    3GPP-UTRAN-TDD;utran-cell-id-3gpp=C359A3913B20E
15 From: <sip:bob@open-ims.test>;tag=[call_number]
16 To: <alice@open-ims.test>
17 Call-ID: [call_id]

```

```
18 CSeq: 10 INVITE
19 Supported: 100rel
20 Contact: <sip:bob@[local_ip]:[local_port]>
21 User-Agent: Sipp v1.1-TLS, version 20061124
22 Allow: ACK, BYE, CANCEL, INVITE, REFER, OPTIONS, INFO,
    REGISTER, NOTIFY, UPDATE, SUBSCRIBE, PRACK
23 Content-Type: application/sdp
24 Content-Length: [len]
25
26 v=0
27 o=user1 53655765 2353687637 IN IP4 [local_ip]
28 s=-
29 c=IN IP4 [local_ip]
30 t=0 0
31 m=audio 30000 RTP/AVP 0 8
32 a=rtpmap:0 PCMU/8000
33 a=sendrecv
34 ]]>
35 </send>
36
37 <recv response="100" optional="true">
38 </recv>
39
40 <recv response="180" optional="true">
41 </recv>
42
43 <recv response="403" optional="true" next="1">
44 </recv>
45
46 <recv response="404" optional="true" next="1">
47 </recv>
```

```
48
49 <recv response="408" optional="true" next="1">
50 </recv>
51
52 <recv response="200" rrs="true">
53 </recv>
54
55 <send crlf="true">
56 <![CDATA[
57 ACK [next_url] SIP/2.0
58 [last_Via:]
59 Max-Forwards: 20
60 [routes]
61 From: <sip:bob@open-ims.test>;tag=[call_number]
62 [last_To:]
63 Call-ID: [call_id]
64 CSeq: 10 ACK
65 Content-Length: 0
66 ]]>
67 </send>
68
69 <pause milliseconds="5000" crlf="true" />
70
71 <send retrans="500">
72 <![CDATA[
73 BYE sip:[next_url] SIP/2.0
74 [last_Via:]
75 Max-Forwards: 20
76 [routes]
77 From: <sip:bob@open-ims.test>;tag=[call_number]
78 [last_To:]
```

```
79 Call-ID: [call_id]
80 CSeq: 11 BYE
81 Contact: <sip:bob@[local_ip]:[local_port]>
82 Content-Length: 0
83 ]]>
84 </send>
85
86 <recv response="200" crlf="true" next="2">
87 </recv>
88
89 <label id="1"/>
90
91 <send crlf="true">
92 <![CDATA[
93 ACK sip:alice@open-ims.test SIP/2.0
94 [last_Via:]
95 Max-Forwards: 20
96 From: <sip:bob@open-ims.test>;tag=[call_number]
97 [last_To:]
98 Call-ID: [call_id]
99 CSeq: 10 ACK
100 Content-Length: 0
101 ]]>
102 </send>
103
104 <label id="2"/>
105
106 <!-- definition of the response time repartition table (unit is
ms) -->
107 <ResponseTimeRepartition value="10, 20"/>
108
```

```
109 <!-- definition of the call length repartition table (unit is  
ms) -->  
110 <CallLengthRepartition value="10"/>  
111  
112 </scenario>
```

BIBLIOGRAPHIE

- 3GPP. Juin 2010. « Digital cellular telecommunications system (Phase 2+) ; Universal Mobile Telecommunications System (UMTS) ; LTE ; IP Multimedia Subsystem (IMS) ; Stage 2 (3GPP TS 23.228 version 7.16.0 Release 7) ». <http://www.etsi.org/deliver/etsi_ts/123200_123299/123228/07.16.00_60/ts_123228v071600p.pdf>.
- Bellavista, Paolo, Antonio Corradi, et Luca Foschini. 2009. « Understanding and enhancing the scalability of IMS-based services for Wireless Local Networks ». In *Local Computer Networks, 2009. LCN 2009. IEEE 34th Conference on*. p. 1033–1039. IEEE.
- Bertrand, Gilles. 2007. « The IP Multimedia Subsystem in Next Generation Networks ». *Network, Multimedia and Security department (RSM)-GET/ENST Bretagne*.
- Brocade. 2012. « Toward Cloud Computing ». En Ligne. <<http://www.brocade.com/company/contacting-brocade/Worldwide+Offices.page>>. Consulté le 12 février 2013.
- Calheiros, R.N., R. Ranjan, et R. Buyya. Sept 2011. « Virtual Machine Provisioning Based on Analytical Performance and QoS in Cloud Computing Environments ». In *Parallel Processing (ICPP), 2011 International Conference on*. p. 295-304.
- Cerqueira, Eduardo, Tiago Sousa, Augusto Neto, Luis Veloso, Marília Curado, Edmundo Monteiro, et Paulo Mendes. 2007. « QoS Mapping and adaptation in next generation networks ». In *Applications and the Internet Workshops, 2007. SAINT Workshops 2007. International Symposium on*. p. 11–11. IEEE.
- Cisco. 2013. « MPLS VPN QoS Design ». En Ligne. <http://www.cisco.com/c/en/us/td/docs/solutions/Enterprise/WAN_and_MAN/QoS_SRND/QoS-SRND-Book/VPNQoS.html>. Consulté juin 2013.
- Dominique Revuz, Etienne Duris. 2008. « La virtualisation de serveurs ». En Ligne. <<http://www-igm.univ-mlv.fr/~dr/XPOSE2008/virtualisation/definitions.html>>. Consulté le 15 juin 2012.
- E.800, ITU-T-Rec. 1993. « Terms and Definitions Related to Quality of Service and Network Performance Including Dependability », Technical Report, International Telecommunication Union ». En Ligne. <http://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-E.800-199408-S!!PDF-E&type=items>. Consulté le 12 juin 2012.
- El Barachi, May, Roch Glitho, et Rachida Dssouli. 2011. « Control-level call differentiation in IMS-based 3G core networks ». *Network, IEEE*, vol. 25, n° 1, p. 20–28.
- Ericsson. 2014. « Ericsson Mobility Report ». En Ligne. <<http://www.ericsson.com/res/docs/2014/ericsson-mobility-report-june-2014.pdf>>. Consulté septembre 2014.
- ETSI. « Quality of Service ». En Ligne. <<http://www.etsi.org/technologies-clusters/technologies/quality-of-service>>. Consulté le 12 juin 2012.

- Fakhfakh, Malek, Omar Cherkaoui, I Limam Bedhiaf, et Mounir Frikha. 2009. « High availability in IMS virtualized network ». In *Communications and Networking, 2009. ComNet 2009. First International Conference on*. p. 1–6. IEEE.
- FOKUS. 2006. « Open IMS Core ». En Ligne. <<http://www.openimscore.org>>. Consulté le 12 février 2013.
- FOKUS. 2013. « fraunhofer ». En Ligne. <http://www.fokus.fraunhofer.de/en/fokus_testbeds/open_ims_playground/index.html>. Consulté le 12 février 2013.
- FOKUS, Peter Weik. 2009. « Open IMS Core Project Overview ». En Ligne. <http://www.openimscore.org/docs/MySQL_cs_IMS_9_8.pdf>. Consulté le 12 février 2013.
- Gartner. 2013. « Gartner Says Worldwide Public Cloud Services Market to Total ». En Ligne. <www.gartner.com/technology/home.jsp>. Consulté le Mai 2013.
- H3C. 2008. « QoS Technology ». En Ligne. <http://www.h3c.com/portal/Products___Solutions/Technology/QoS/Technology_White_Paper/200812/623011_57_0.htm>. Consulté le 12 juin 2012.
- Hardy, W. C., 2001. *QoS Measurements and Evaluation of Telecommunication Quality of Service*, p. 1299. éd. 9.
- He, Sijin, Li Guo, Yike Guo, Chao Wu, Moustafa Ghanem, et Rui Han. 2012. « Elastic application container : A lightweight approach for cloud resource provisioning ». In *Advanced information networking and applications (aina), 2012 ieee 26th international conference on*. p. 15–22. IEEE.
- IBM. 2005. « IBM Systems Virtualization ». En Ligne. <<http://publib.boulder.ibm.com/infocenter/eserver/v1r2/topic/eicay/eicay.pdf>>. Consulté le 12 Mai 2012.
- Iqbal, Waheed, Matthew N Dailey, David Carrera, et Paul Janecek. 2011. « Adaptive resource provisioning for read intensive multi-tier applications in the cloud ». *Future Generation Computer Systems*, vol. 27, n° 6, p. 871–879.
- ISO. 2000a. « Quality Management and Quality Assurance Vocabulary. Technical Report, International Organization for Standardization. ». En Ligne. Consulté le 12 june 2012.
- ISO. 2000b. « Quality Management and Quality Assurance Vocabulary. Technical Report, International Organization for Standardization. ». En Ligne. <http://www.iso.org/iso/catalogue_detail?csnumber=29280>. Consulté le 12 june 2012.
- Ito, Manabu, Satoshi Komorita, YOSHINORI Kitatsuji, et HIDETOSHI Yokota. 2013. « OpenFlow-based Routing Mechanism for Call Session State Migration in the IMS ». In *WSEAS International Conference. Proceedings. Recent Advances in Computer Engineering Series*. WSEAS.

- ITU. 2009. « Réseaux de prochaine génération ». En Ligne. <<http://www.itu.int/itu-news/manager/display.asp?lang=fr&year=2009&issue=03&ipage=24&ext=html>>. Consulté le 10 juin 2012.
- Khelifi, Hechmi et J-C Gregoire. 2008. « IMS application servers : roles, requirements, and implementation technologies ». *Internet Computing, IEEE*, vol. 12, n° 3, p. 40–51.
- Letaifa, Asma Ben, Ael Haji, Maha Jebalia, et Sami Tabbane. 2010. « State of the Art and Research Challenges of new services architecture technologies : Virtualization, SOA and Cloud Computing. ». *International Journal of Grid & Distributed Computing*, vol. 3, n° 4.
- Mell et Grance. 2011. « The NIST Definition of Cloud Computing (Draft).[Online] <http://csrc.nist.gov/publications/nistpubs/800-145> ». *SP800-145.pdf*.
- Metaswitch. 2013. « Clearwater Project ». En Ligne. <<http://www.projectclearwater.org/about-clearwater/>>. Consulté en Mai 2013.
- Microsoft. 2011. « Cloud Computing témoignages entreprises ». En Ligne. <<http://www.microsoft.com/france/entreprises/decideur-it/cloud/cloud-computing-temoignages-entreprises.aspx>>. Consulté le 05 juillet 2012.
- MROUEH Lina, LABAKY Elie. 2006. « MEtude des procédures d'enregistrement et d'établissement de session en IMS ». En Ligne. <<http://perso.isep.fr/mroueh/docs/IMS.pdf>>. Consulté juin 2013.
- Nagesh, Bhavani B et al. 2014. « Resource Provisioning Techniques in Cloud Computing Environment-A Survey ». *IJRCCT*, vol. 3, n° 3, p. 395–401.
- Nelson, V et V Uma. 2012. « Semantic based resource provisioning and scheduling in inter-cloud environment ». In *Recent Trends in Information Technology (ICRTIT), 2012 International Conference on*. p. 250–254. IEEE.
- Networks, Metaswitch. 2014. « Metaswitch Networks ». En Ligne. <<http://www.metaswitch.com/>>. Consulté en Mai 2013.
- Poikselkä, Miikka et Georg Mayer, 2009. *The IMS : IP multimedia concepts and services*. John Wiley & Sons.
- Rackspace, NASA. 2013. « Open IMS Core ». En Ligne. <<http://www.opensatck.org>>. Consulté le 12 November 2013.
- Raouyane, B., M. Bellafkih, et D. Ranc. Sept 2009. « QoS Management in IMS : DiffServ Model ». In *Next Generation Mobile Applications, Services and Technologies, 2009. NGMAST '09. Third International Conference on*. p. 39-43.
- rfc2205. 1997. « Resource ReSerVation Protocol (RSVP) ». ietf.

- Ridha, Gadhgadhi, Khazri Saida, et Cheriet Mohamed. 2013. « OPENICRA : Towards A Generic Model for Automatic Deployment of Applications in the Cloud Computing ». *International Journal of Cloud Computing and Services Science (IJ-CLOSER)*, vol. 2, n° 4, p. 249–275.
- Rosenberg, Jonathan, Henning Schulzrinne, Columbia U, Gonzalo Camarillo, Alan Johnston, Jon Peterson, Robert Sparks, et Mark Handley. 2002. « SIP : session initiation protocol ». In *IETF RFC 3261*.
- Ryu, Mi Sun, Hong-Shik Park, et Sang-Chul Shin. 2006. « QoS class mapping over heterogeneous networks using Application Service Map ». In *Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies, 2006. ICN/ICONS/MCL 2006. International Conference on*. p. 13–13. IEEE.
- Siliconangle. 2014. « 20 cloud computing statistics every CIO should know ». En Ligne. <<http://siliconangle.com/blog/2014/01/27/20-cloud-computing-statistics-tc0114>>. Consulté le juin février 2014.
- Skorin-Kapov, Lea, Miran Mosmondor, Ognien Dobrijevic, et Maja Matijasevic. 2007. « Application-level QoS negotiation and signaling for advanced multimedia services in the IMS ». *Communications Magazine, IEEE*, vol. 45, n° 7, p. 108–116.
- Varia, J. Octobre 2012. « Amazon Web Services - Architecting for the cloud : Best practices ». <http://media.amazonwebservices.com/AWS_Cloud_Best_Practices.pdf>.
- Wireshark, Projet. 2004. *Wireshark*, version 1.12.0. Logiciel. Lieu de publication : Nom de l'éditeur.
- WMWare. « Virtualization Overview, year=2011 ». *White Paper*.
- Y.1540, ITU-Recommendation. 2011. « IP packet transfer and availability performance parameters ». En Ligne. <<http://www.itu.int/rec/T-REC-Y.1540-201103-I/en>>. Consulté le 12 juin 2012.
- Zhang, Lixia, Steve Berson, Shai Herzog, et Sugih Jamin. 1997. « Resource reservation protocol (RSVP)–Version 1 functional specification ». *Resource*.
- Zhang, Qi, Lu Cheng, et Raouf Boutaba. 2010. « Cloud computing : state-of-the-art and research challenges ». *Journal of internet services and applications*, vol. 1, n° 1, p. 7–18.