



**HAL**  
open science

# Une architecture pour l'intégration du Cloud Computing dans les réseaux de capteurs sans fil (RCSF)

Hafnaoui Taleb

► **To cite this version:**

Hafnaoui Taleb. Une architecture pour l'intégration du Cloud Computing dans les réseaux de capteurs sans fil (RCSF). Réseaux et télécommunications [cs.NI]. Université de Haute Alsace - Mulhouse; Université Hadj Lakhdar (Batna, Algérie), 2019. Français. NNT: 2019MULH1078 . tel-03696889

**HAL Id: tel-03696889**

**<https://theses.hal.science/tel-03696889>**

Submitted on 16 Jun 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

## Thèse de Doctorat

---

En vue de l'obtention du grade de Docteur de l'Université de Haute-  
Alsace (France) et de l'Université Batna 2, (Algérie)

**Spécialité : Informatique**

---

**Intitulé de thèse : Une architecture pour  
l'intégration du Cloud Computing dans les réseaux de  
capteurs sans fil**

Par M. Hafnaoui TALEB

Soutenue le 25/06/2019 devant le jury composé de :

**Rapporteurs :**

M. Ahmed SERHROUCHNI Prof, Telecom-ParisTech, France

M. Hervé GUYENNET Prof, Université de Franche-Comté, France

**Examineurs :**

M. Jaafar GABER HDR, UTBM, France

**Encadrants de thèse:**

M. Azeddine BILAMI Prof, Université de Batna 2, Algérie

M. Pascal LORENZ Prof, Université de Haute-Alsace, France

## REMERCIEMENTS

*Je tiens à remercier mes directeurs de thèse le professeur Azeddine BILAMI et le professeur Pascal LORENZ de m'avoir apporté tout le soutien, les encouragements, les conseils et la liberté qu'on pouvait souhaiter pour mener ce travail à termes. Il m'a été un plaisir de travailler et d'acquérir une précieuse expérience avec eux. Je tiens aussi à remercier Mr Ayoub BENAYACHE pour son aide précieuse durant la thèse.*

*Je remercie les rapporteurs du manuscrit M. Ahmed SERHROUCHNI et M. Hervé GUYENNET pour le temps consacré à ma thèse. Merci également à M. Jaafar GABER qui a accepté d'évaluer mon travail.*

*Au cours de mon doctorat, j'ai eu la chance de travailler avec différents chercheurs, ingénieurs et collègues. Je remercie Dr Soufiane HAMRIOUI et Dr Hafidh ABOU AISSA pour leur volonté de partager leurs connaissances et leurs expériences avec moi, ainsi que pour leur soutien et leurs conseils tout au long de cette thèse. Mon profond respect et amour à mon père Lakhdar, mon frère Imad et ma mère qui m'ont toujours encouragé à atteindre les objectifs que je me suis fixé dans ma vie.*

*Enfin, je remercie tous ceux qui ont contribué de près ou de loin à l'aboutissement de ce travail de recherche.*

## *DÉDICACES*

À mes parents et mon frère.

## *LISTE DES ABRÉVIATIONS*

**AWS:** Amazon Web Services

**DV/DRP:** Distance Vector/

**EAR:** Eavesdrop and Register

**EGP:** Exterior Gateway Protocol

**EIGRP:** Enhanced Interior Gateway Routing Protocol

**ERP:** Enterprise Resource Planning

**GSN:** Global Sensor Network

**HPC:** High Performance System

**IAAS:** Infrastructure as a Service

**Intellisys:** The Intelligent Systems Centre

**IS-IS:** Intermediate System-Intermediate System

**IT:** Information and Technology

**MQTT:** MQ Telemetry Transport

**NIST:** National Institute of Standards and Technology

**NLSR:** Named-data Link State Routing Protocol

**OSI:** Open Systems Interconnection

**OSPF:** Open Shortest Path First

**PAAS :** Platform As A Service

**RCSF :** Réseau de Capteurs Sans Fil

**RCV :** Réseau de Capteurs Virtuels

**SAAS :** Software As A Service

**SMAC:** Self-organizing Medium Access Control for Sensor Networks

**TCP:** Transmission Control Protocol

**UDP:** Uniform Resource Locator

## RÉSUMÉ

*Le nombre croissant de réseaux de capteurs sans fil (RCSF) déployés pour diverses applications devient une tendance invincible dans la pluralité de domaines d'application. Ces réseaux appartiennent à une seule organisation dans la plupart des cas et sont conçus pour prendre en charge des applications spécifiques. Généralement, les RCSF génèrent une grande quantité de données et le nombre de nœuds de capteurs physiques utilisés augmente chaque jour. Pour partager, stocker et traiter les données brutes générées, il est essentiel d'utiliser des ressources informatiques performantes telles que les clusters distribués, les ordinateurs centraux ou le Cloud Computing. Dans ce travail, nous proposons une approche innovante pour exploiter la facilité d'utilisation et de partage des RCSF par plusieurs applications. D'autre part, le Cloud Computing est une technologie émergente qui offre de nombreux avantages comme la capacité de traitement et de stockage à différents niveaux. L'extension du Cloud Computing pour l'intégration des RCSF permettra le partage et le traitement des données détectées parmi de nombreuses communautés d'utilisateurs. La virtualisation est la technologie clé qui peut potentiellement permettre cette intégration. Nous discutons de l'idée d'utiliser des réseaux de capteurs virtuels (VSN) combinés avec le réseau de données nommées pour router les données détectées entre les capteurs virtuels. Le framework utilise l'infrastructure Cloud Computing pour traiter et stocker les données.*

*Les évaluations par simulation de la solution proposée montrent un bon niveau de performances*

**MOTS-CLÉS :** *Capteurs, réseaux de capteurs sans fil, Cloud Computing, réseaux de capteurs virtuels, Les réseaux de données nommées.*

## ABSTRACT

*The increasing number of wireless sensor Networks (WSNs) deployed for various applications is becoming an important trend in many domains. These Networks belong to a single organization in most cases and are designed to support specific applications. Typically, WSNs generate a large amount of data, and the number of used physical sensor nodes increases daily. To share, store and process the generated data, it is essential to use powerful computing resources such as distributed clusters, mainframes or Cloud Computing. In this work, we propose an innovative approach to exploit the ease of use and sharing of WSNs by several applications. On the other hand, Cloud Computing is an emerging technology that offers many advantages such as processing and storage capacity at different levels. The extension of Cloud Computing for the integration of WSNs will allow the sharing and processing of the sensed data among many user communities. Virtualization is the key technology that can potentially enable this integration. We discuss the idea of using virtual sensor Networks (VSNs) combined with the named data network to route collected data between virtual sensors. The framework uses Cloud Computing infrastructure to process and store data.*

*The performed evaluations by simulation of the proposed solution show a good level of performances.*

**KEYWORDS:** *Sensors, Wireless Sensor Networks, Cloud Computing, Virtual Sensor Networks, Named Data Networking*

## Table des matières

---

Remerciements .....	2
Dédicaces .....	3
Liste des abréviations .....	4
Résumé .....	V
Abstract .....	VI
Table des matières .....	VII
Liste de figures.....	XI
Liste de tableaux.....	XII
<b>1. Introduction.....</b>	<b>14</b>
1.1 La contribution .....	15
1.2 L'organisation du manuscrit .....	16
<b>I. Les réseaux de capteurs sans fil .....</b>	<b>17</b>
I.1 Introduction.....	18
I.2 Architecture de réseau de capteurs sans fil.....	19
I.2.1 L'architecture d'un micro-capteur.....	20
I.2.2 Les unités principales du capteur.....	20
I.3 Caractéristiques des RCSF.....	22
I.3.1 La consommation d'énergie.....	22
I.3.2 La topologie dynamique de réseau .....	22
I.3.3 La communication multi-sauts.....	23
I.3.4 Le déploiement des capteurs .....	24
I.4 Domaines d'application des RCSF .....	24
I.5 Paramètres influençant la conception d'un RCSF .....	27
I.6 Les différentes problématiques dans les RCSF .....	29
I.7 La pile protocolaire des capteurs.....	30
I.7.1 Les rôles des couches.....	31
I.8 Conclusion .....	33



<b>II Le Cloud Computing .....</b>	<b>31</b>
II.1 Introduction .....	32
II.2 Définition .....	32
II.3 Les formes de services .....	33
II.4 La technologie de virtualisation .....	34
II.5 Historique du Cloud Computing .....	36
II.6 Modèles du déploiement .....	37
II.7 Les caractéristiques du Cloud Computing .....	38
II.8 Avantages et inconvénients du Cloud Computing .....	40
II.8.1 Les avantages .....	40
II.8.2 Les inconvénients .....	41
II.9 Le cloud pour le RCSF .....	43
II.10 Conclusion .....	43
<b>III L'intégration des RCSF et le Cloud Computing .....</b>	<b>44</b>
III.1 Introduction .....	45
III.2 Motivation .....	45
III.3 Définition du nuage de capteurs (Sensor-Cloud) .....	46
III.4 Les avantages du nuage de capteurs .....	47
III.5 Les applications du Cloud de capteurs .....	49
III.6 Les architectures du Cloud de capteurs .....	51
III.7 Conclusion .....	62
<b>IV L'architecture Named Data Networking .....</b>	<b>63</b>
IV.1 Introduction .....	64
IV.2 Les composants d'un routeur NDN .....	66
IV.3 Le routage dans NDN .....	69
IV.3.1 Routage intra-domaine .....	72
IV.3.2 Routage inter-domaine .....	74
IV.4 Le forwarding .....	75
IV.5 Le système du cache .....	77

IV.6 Le système de nommage .....	78
IV.7 NDN pour les RCSF .....	80
IV.8 Conclusion .....	81
<b>V Une architecture basée sur le réseau de données nommées.....</b>	<b>82</b>
V.1 Introduction .....	83
V.2 Le réseau de capteurs virtuels.....	84
V.2.1 La structure et la configuration.....	85
V.2.2 Les types des réseaux de capteurs virtuels .....	89
V.3 L'architecture proposée.....	90
V.3.1 La couche physique.....	91
V.3.2 La couche infrastructure de Cloud Computing .....	92
V.3.3 La couche d'application .....	96
V.4 La structure de données.....	96
V.4.1 Le système de nommage .....	96
V.4.2 La gestion des paquets.....	98
V.5 Conclusion.....	104
<b>VI Simulation et évaluation.....</b>	<b>105</b>
VI.1 Introduction.....	106
VI.2 Environnement de simulation.....	106
VI.2.1 Le simulateur NDN-Sim .....	106
VI.2.2 La plateforme GSN.....	107
VI.3 Le scénario de la simulation .....	113
VI.4 Les métriques.....	114
VI.5 Les paramètres de simulation .....	116
VI.6 Résultats de simulation .....	118
VI.7 Conclusion .....	122
<b>VII Conclusion générale et perspectives.....</b>	<b>124</b>
VII.1 Conclusion générale.....	125
VII.2 Perspectives.....	126

Bibliographie.....	128
Mespublications .....	135

## Liste de figures

FIGURE 1: SCHÉMA GÉNÉRAL DE RÉSEAU DE CAPTEURS SANS FIL .....	18
FIGURE 2:EXEMPLES CAPTEURS SANS FIL .....	19
FIGURE 3: L'ARCHITECTURE D'UN CAPTEUR SANS FIL.....	20
FIGURE 4 : LES UNITÉS PRINCIPALES D'UN CAPTEUR SANS FIL .....	22
FIGURE 5: APPLICATIONS DES RÉSEAUX DE CAPTEURS .....	27
FIGURE 6: MODÈLE EN COUCHES POUR LA COMMUNICATION DANS LES RCSF.....	31
FIGURE 7: LES PRINCIPAUX FOURNISSEURS DE IAAS,PAAS ET SAAS .....	34
FIGURE 8: MODÈLE DE VIRUALISATION DE CLOUD COMPUTING.....	35
FIGURE 9: MODÈLE GÉNÉRAL D'UN CLOUD DE CAPTEURS .....	52
FIGURE 10: L'ARCHITECTURE MIREs [53].....	55
FIGURE 11: L'ARCHITECTURE MISSOURI S&T .....	56
FIGURE 12: L'ARCHITECTURE POUR INTÉGRER RCSF DANS LE CLOUD COMPUTING.....	57
FIGURE 13: ARCHITECTURE DE RÉSEAUX DE CAPTEURS VIRTUELS .....	59
FIGURE 14: LE PAQUET D'INTÉRÊT ET LE PAQUET DE DONNÉES .....	64
FIGURE 15: LA CLASSIFICATION DE NDN .....	65
FIGURE 16: LES TABLES D'UN ROUTEUR NDN.....	66
FIGURE 17: LE CONTENU DES ROUTEURS NDN .....	70
FIGURE 18: LES TYPES DE ROUTAGE DANS NDN.....	71
FIGURE 19: DIAGRAMME DE SÉQUENCE D'ÉCHANGE PAQUETS.....	76
FIGURE 20: LA RECHERCHE DU PRÉFIXE LE PLUS LONG.....	77
FIGURE 21: NOM DU CONTENU ET SA REPRÉSENTATION HIÉRARCHIQUE.....	79
FIGURE 22: EXEMPLE DE CAPTEURS VIRTUELS.....	84
FIGURE 23: LA STRUCTURE PLUSIEURS À UN .....	86
FIGURE 24: UN À PLUSIEURS .....	86
FIGURE 25: PLUSIEURS À PLUSIEURS .....	87
FIGURE 26: LA MODÉLISATION DE L'INTERACTION DES CAPTEURS VIRTUELS.....	88
FIGURE 27: LA VIRTUALISATION AU NIVEAU DU NŒUD.....	90
FIGURE 28: LA VIRTUALISATION AU NIVEAU DE RÉSEAU .....	90
FIGURE 29:L'ARCHITECTURE PROPOSÉE .....	91
FIGURE 30: LA STRUCTURE D'UN CAPTEUR VIRTUEL.....	94
FIGURE 31: ANATOMIE DE CAPTEUR VIRTUEL.....	95
FIGURE 32: EXEMPLE SIMPLIFIÉ DE TABLE FIB.....	98
FIGURE 33: ALGORITHME DE GESTION DE PAQUET D'INTÉRÊT .....	101
FIGURE 34: ALGORITHME DE LA GESTION DE PAQUET DE DONNÉES.....	102
FIGURE 35: SCHÉMA FONCTIONNEL DES COMPOSANTS DE NDN SIM[72] .....	107
FIGURE 36: LE MODÈLE GLOBAL DE GSN .....	110
FIGURE 37: UN CAPTEUR VIRTUEL COMPLEXE DANS GSN .....	112
FIGURE 38: SCÉNARIO DE SIMULATION.....	113
FIGURE 39: ÉVALUATION DU THROUGHPUT .....	118
FIGURE 40: LE POURCENTAGE DE PERTE DE PAQUETS .....	119
FIGURE 41: LA CONVERGENCE DES PROTOCOLES .....	119
FIGURE 42: LA SURCHARGE DE RÉSEAU.....	120
FIGURE 43: L'ÉVALUATION DU CACHE HIT RATIO.....	121

**FIGURE 44: L'ÉVALUATION DU DÉLAI..... 122**

## **Liste de tableaux**

**TABLEAU 1: CARACTÉRISTIQUES DES CAPTEURS POPULAIRES ..... 19**  
**TABLEAU 2: LA STRUCTURE DE LA TABLE DU CONTENU..... 67**  
**TABLEAU 3: LA STRUCTURE DE LA TABLE PIT..... 67**  
**TABLEAU 4: LA STRUCTURE DE LA TABLE FIB..... 68**  
**TABLEAU 5: LES COMPOSANTS DU GESTIONNAIRE DES CAPTEURS VIRTUELS..... 94**  
**TABLEAU 6: LES PARAMÈTRES DE LA SIMULATION..... 117**

# Introduction Générale

## 1. Introduction

Le besoin effréné d'informations et l'évolution rapide de la micro-électronique et des technologies sans fil, ont permis la création de petits appareils électroniques avec un coût très réduit (*ressources limitées*), capables de collecter et de traiter l'information d'une manière autonome et flexible. Ces appareils peuvent être interconnectés et déployés à grande échelle, donnant naissance à un nouveau type de réseaux nommé réseau de capteurs sans fil (*RCSF*)[1]. Le développement des RCSF était originalement motivé par les applications militaires (*surveillance des champs de bataille, localisation de l'ennemie...*). Néanmoins, leurs performances remarquables en matière de fiabilité et de faible coût ont permis de proliférer leur utilisation dans le domaine d'applications civiles et militaires (*surveillance d'environnement, l'industrie, la domotique, la santé...*).

Les réseaux de capteurs sans-fil sont conçus pour fonctionner en groupe et coopérer afin de transmettre les données collectées à un point central appelé station de base ou sink. Chaque nœud capteur est équipé d'un microprocesseur à faible puissance de calcul, d'une petite batterie, d'une antenne radio et d'un ou de plusieurs capteurs. Ainsi, les RCSF doivent opérer en prenant toujours en compte leur limitation de ressources. Ces derniers sont le plus souvent déployés aléatoirement dans des zones hostiles et inexplorées, et doivent s'auto-organiser à l'aide des communications sans fil. La station de base est le seul lien avec le monde extérieur et dispose de plus de ressources par rapport aux nœuds capteurs. Le réseau de capteurs joue le rôle d'un pont entre le monde physique et le système informatique, en fournissant des mesures et des propriétés physiques du monde réel.

La thèse porte sur l'intégration de Cloud Computing et les réseaux de capteurs sans fil, nous visons à développer une architecture pour combler les pénuries majeures des réseaux de capteurs à savoir la mémoire réduite, la faible puissance de calcul et le manque gestion de données capturées par l'intégration de la technologie de Cloud Computing [2] qui est une technologie émergente. L'objectif de la recherche est de concevoir une architecture qui doit être évolutive vis-à-vis la grande quantité de données générées par les capteurs et les requêtes des utilisateurs, aussi, implémenter des méthodes efficaces pour récupérer, traiter, gérer et stocker les données au niveau de Cloud Computing et enfin délivrer les données aux utilisateurs. Le deuxième objectif est d'implémenter un protocole pour effectuer le routage des données au niveau du Cloud Computing. Notre méthodologie de

recherche est : définir la problématique de la recherche, spécifier le paramètre à traiter et enfin valider la proposition.

### **1.1 La contribution**

D'abord, nous avons établi un vaste état de l'art concernant les réseaux de capteurs sans fil où nous avons abordé l'importance de ce type de réseau et les solutions proposées dans la littérature pour intégrer les RCSF dans le Cloud Computing. Les questions liées à ce thème de recherche ont été discutées en détail, on espère que cet état de l'art va enrichir et améliorer l'état de l'art actuel.

La deuxième contribution est une proposition d'une architecture basée sur les réseaux de capteurs virtuels. Un réseau de capteurs virtuels (VSN) est constitué d'un sous-ensemble de nœuds capteurs d'un réseau de capteurs sans fil (RCSF), le sous-ensemble étant dédié à une tâche ou à une application donnée à un moment donné. Les détails des VSNs ont été discutés avec les détails de chaque composant de réseau. Nous avons proposé aussi une nouvelle méthode de routage dans les VSN qui se base sur le paradigme NDN (Named Data Networking).

La troisième contribution consiste à l'intégration du paradigme de réseau basé sur le contenu NDN pour l'acheminement et le routage des données. Le NDN propose un modèle de communication simple basé sur l'échange de deux types de paquets : l'intérêt et la donnée qui portent le même nom. Nous avons aussi utilisé un schéma de nommage adapté à notre architecture qui a donné de bons résultats concernant la livraison des données entre les différents routeurs.

Enfin, nous avons évalué l'architecture proposée par simulation. L'évaluation des performances est conduite sous le Framework GSN et le simulateur NDNSIM. Nous avons alors évalué le protocole NLSR par rapport aux protocoles de routage comme RIP, EIGRIP et BGP en utilisant l'inondation de réseau ("flooding"). Cette méthode consiste à inonder le réseau de requêtes en les diffusant sur toutes les interfaces des nœuds afin d'atteindre le contenu désiré. Les résultats obtenus montrent le gain en performances et la justesse des choix adoptés quant à l'utilisation de l'approche NDN pour le routage dans l'intégration des RCSF avec capteurs virtuels et le Cloud Computing.



## **1.2 Organisation du manuscrit**

Le reste de la thèse est organisé comme suit :

### **Chapitre I : Les réseaux de capteurs sans fil**

Dans ce chapitre, nous présentons la technologie de RCSF, ainsi les contraintes liées à ce dernier comme la faible capacité de calcul, la mémoire réduite et la source énergétique qui réduit la durée de vie du capteur sans fil.

### **Chapitre II : Le Cloud Computing**

Dans ce chapitre, nous soulignons les avantages et les défis liés à l'utilisation du Cloud Computing pour les applications de RCSF. Ensuite, nous avons discuté les différentes caractéristiques comme la flexibilité, l'auto-guérison, La virtualisation et la multi-location. Enfin, nous avons cité des inconvénients et les avantages du Cloud Computing.

### **Chapitre III : Intégration des RCSF et le Cloud Computing**

Dans ce chapitre, Nous avons présenté la motivation et un résumé de l'état de l'art concernant l'intégration des RCSF et le Cloud Computing. Ensuite, nous avons étudié les travaux existants dans ce domaine et enfin nous avons justifié l'importance de l'utilisation des Réseaux de capteurs virtuels dans notre recherche.

### **Chapitre IV: Architecture Named Data Networking**

Ce chapitre fournit une description de l'architecture NDN (Named Data Networking). Nous avons présenté l'architecture basée sur les noms de données NDN pour le routage à la place du routage IP classique. Les principes fondamentaux de cette architecture ainsi que les justifications de son choix y sont présentés.

### **Chapitre V : Une architecture basée sur le réseau de données nommées**

Ce chapitre décrit l'architecture globale proposée ainsi que les détails de chaque composant. Un modèle analytique est présenté à la fin du chapitre.

### **Chapitre VI : Simulation et évaluation**

Ce chapitre porte sur l'évaluation de l'architecture proposée par simulation. Nous avons évalué le protocole NLSR par rapport aux protocoles de routage comme les protocoles RIP, EIGRIP et BGP.

### **Chapitre VII : Conclusion générale et perspective**

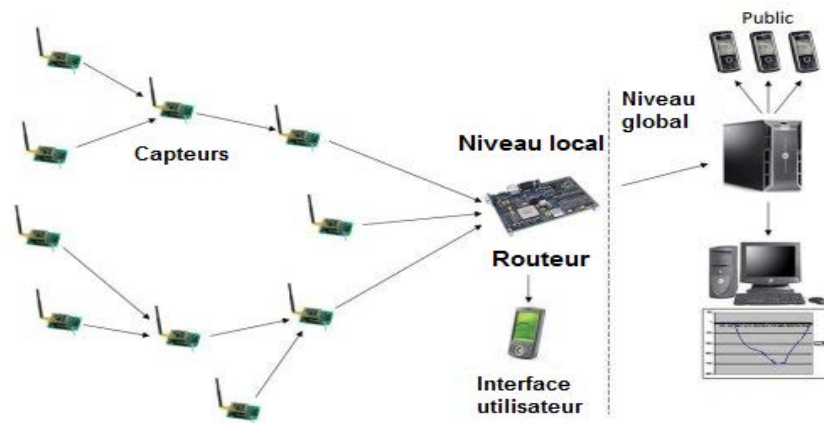
Nous terminons ce travail par quelques conclusions finales et nous soulignons les questions en suspens ainsi que les principales orientations de notre future recherche.

**Sommaire**

<b>I. Les réseaux de capteurs sans fil.....</b>	<b>17</b>
I.1 Introduction.....	18
I.2 Architecture de réseau de capteurs sans fil .....	19
I.2.1 L'architecture d'un micro-capteur.....	20
I.2.2 Les unités principales du capteur .....	20
I.3 Caractéristiques des RCSF.....	22
I.3.1 La consommation d'énergie.....	22
I.3.2 La topologie dynamique de réseau .....	22
I.3.3 La communication multi-sauts.....	23
I.3.4 Le déploiement des capteurs .....	24
I.4 Domaines d'application des RCSF .....	24
I.5 Paramètres influençant la conception d'un RCSF.....	27
I.6 Les différentes problématiques dans les RCSF .....	29
I.7 La pile protocolaire des capteurs.....	30
I.7.1 Les rôles des couches .....	31
I.8 Conclusion .....	33

## I.1 Introduction

Récemment, les réseaux de capteurs sans fil (*RCSF* ou *WSNs* : *Wireless Sensor Networks*)[1] sont devenus l'un des domaines de recherche qui se développent le plus. Un grand intérêt de la part de la communauté scientifique et de l'industrie entraîne un développement rapide de nouveaux types d'appareils, de technologies et de protocoles. En effet, la facilité de déploiement et la grande quantité d'utilisations possibles justifient un tel intérêt. Les nœuds peuvent avoir diverses caractéristiques. Au sein d'une même plateforme, ils peuvent intégrer de nombreux types de capteurs (température, lumière, humidité, caméras, accéléromètres, etc.) ou d'actionneurs (climatisation, contrôle de porte, alarmes, etc.).



**Figure 1:** Schéma général de réseau de capteurs sans fil

Un réseau de capteurs sans fil est un type spécial de réseaux Ad Hoc avec un grand nombre de nœuds qui sont des micro-capteurs capables de recevoir et de transmettre des données environnementales d'une manière autonome sans intervention humaine. Un nœud dans un RCSF (Réseau de capteurs sans fil) a généralement des ressources et des capacités très limitées. Le **Tableau 1** présente certaines caractéristiques des capteurs populaires. La quantité de mémoire ne dépasse généralement pas 32 Ko et peut même être plus petite. Les contraintes nécessitent un développement minutieux de protocoles, de systèmes et d'applications qui minimisent la consommation de ressources.

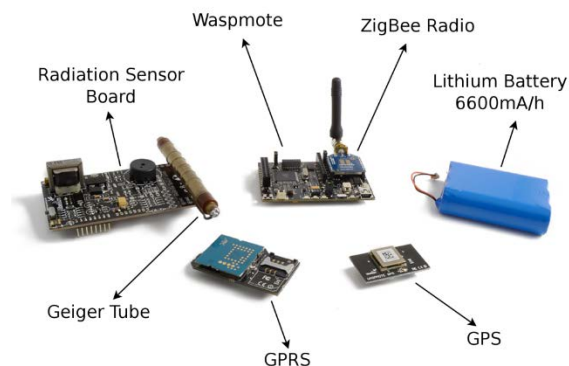
Capteur	Architecture (Bit)	RAM (KB)	Flash(K)	CPU
MicaZ	8	4	128	8 MHz
TelosB	16	16	48	16 MHz
OpenMote	32	32	512	32 MHz
GreenNet	32	32	512	32 MHz

**Tableau 1:** Caractéristiques des capteurs populaires

Les capteurs communiquent à travers un canal sans fil fourni par les différentes technologies. Dans certains scénarios, il peut être avantageux d'utiliser l'une des normes comme 802.11 qui sont également utilisées dans la plupart des ordinateurs portables modernes. La norme 802.11 peut fournir une bande passante significative et une portée d'environ cent mètres, cependant la consommation d'énergie est généralement été trop en utilisant cette technologie.

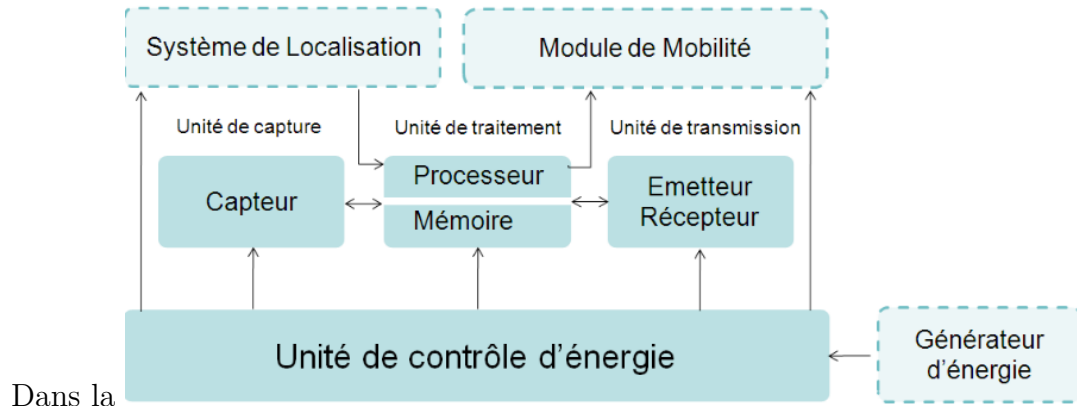
## I.2 Architecture de réseau de capteurs sans fil

Un réseau de capteurs sans fil peut être constitué d'un certain nombre de nœuds, ces nœuds sont généralement des capteurs distribués dans un champ de captage et peuvent surveiller en collaboration les conditions d'environnement comme le son, la température, la pression, le mouvement, la vibration, la pollution ... etc. Un capteur est un dispositif destiné à recevoir des informations à partir d'une grandeur de l'extérieur et de le transformer dans une autre dimension, généralement électrique que nous pouvons quantifier et manipuler (voir figure 2).



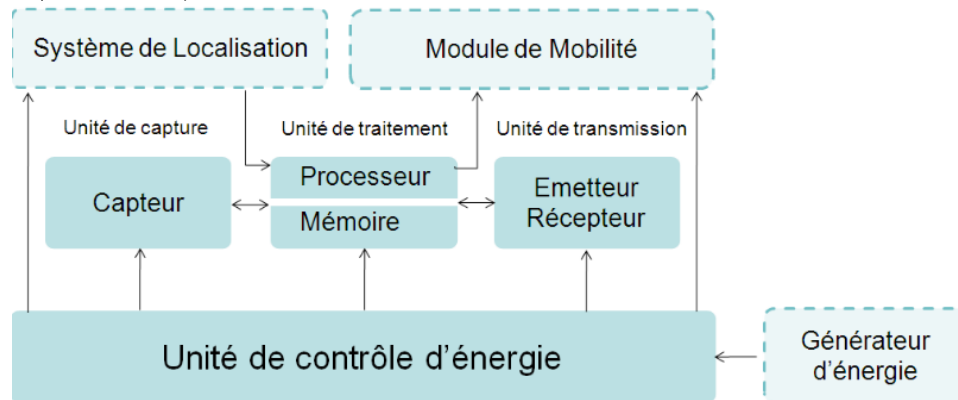
**Figure 2:** Exemples capteurs sans fil

### I.2.1 L'architecture d'un micro-capteur



Dans la

**Figure 3**, nous présentons une architecture générique d'un capteur sans fil. Un capteur sans fil permet de mesurer des grandeurs physiques (chaleur, humidité, vibrations, etc.). La plupart, de capteurs se compose d'une mémoire de stockage, de microcontrôleur capable d'effectuer les traitements nécessaires, d'une source d'énergie souvent une batterie ou cellule et d'un dispositif de transmission (émetteur/récepteur).



**Figure 3:** L'architecture d'un capteur sans fil

Les capteurs peuvent être équipés d'éléments supplémentaires tels qu'un système de localisation (GPS), des composants de récupération d'énergie ou des actionneurs. Le coût des nœuds doit être maintenu bas pour minimiser le coût global de réseau, et un bon compromis doit être trouvé entre la quantité de fonctionnalités fournies par le capteur et son coût.

### I.2.2 Les unités principales du capteur

Le capteur est le plus souvent composé de trois parties à savoir : une unité de captage, une unité de traitement et une unité de transmission (voir Figure 4).

- **L'unité de captage**

L'unité de captage est généralement composée de deux sous-unités : le capteur lui-même et un convertisseur Analogique/Numérique. Le capteur est responsable de fournir des signaux analogiques basés sur le phénomène observé, ce dernier transforme ces signaux en un signal numérique compréhensible par l'unité de traitement. Parmi les dispositifs les plus utilisés comme des nœuds capteurs on trouve le capteur d'image, le capteur accéléromètre, le capteur thermique et le capteur microphone.

- **L'unité de traitement**

L'unité de traitement comprend un processeur associé généralement à une petite unité de stockage et fonctionne à l'aide d'un système d'exploitation spécialement conçu pour les micro-capteurs (TinyOS par exemple). Cette unité est chargée d'exécuter les protocoles de communication qui permettent de faire collaborer le nœud avec les autres nœuds de réseau. Elle peut aussi analyser les données captées pour alléger la tâche du nœud puits (sink).

- **L'unité de transmission**

Cette unité est responsable des émissions et réceptions des données sur un médium sans fil. Elle peut être de type optique ou de type radiofréquence. Les communications de type optique sont robustes vis-à-vis des interférences électriques. Néanmoins, elles présentent l'inconvénient d'exiger une ligne de vue permanente entre les entités communicantes. Par conséquent, elles ne peuvent pas établir de liaisons à travers des obstacles. Les unités de transmission de type radiofréquences comprennent des circuits de modulation, démodulation, filtrage et multiplexage ; ce qui implique une augmentation de la complexité et du coût de production du micro-capteur.

Concevoir des unités de transmission de type radiofréquence avec une faible consommation d'énergie est un véritable défi. En effet, pour qu'un nœud ait une portée de communication suffisamment grande, il est nécessaire d'utiliser un signal assez puissant. Cependant, l'énergie consommée serait importante. L'autre alternative serait d'utiliser de longues antennes, mais ceci n'est pas possible à cause de la taille réduite des micro-capteurs

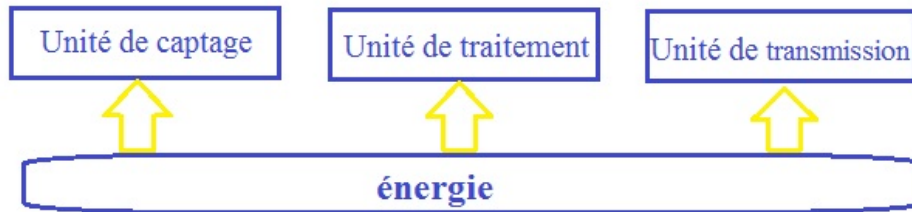


Figure 4 : Les unités principales d'un capteur sans fil

### I.3 Caractéristiques des RCSF

Dans cette partie, on présente les contraintes et les principaux facteurs de conception des réseaux de capteurs sans fil. La possibilité de déployer un nombre important de capteurs sans fil est une caractéristique importante des applications de RCSF. Cette fonctionnalité déclenche tous les défis de conception et de mise en œuvre liés aux applications de RCSF.

#### I.3.1 La consommation d'énergie

Les communications dans un réseau de capteurs sans fil se produisent de différentes manières en fonction de l'application. Dans tous les types de communication, la durée de vie d'un capteur et la durée de vie de réseau sont limitées par la quantité d'énergie de chaque capteur. La communication multi-sauts dans les réseaux de capteurs sans fil devrait consommer moins d'énergie que la communication traditionnelle à un seul saut. La puissance de transmission requise augmente avec la distance entre l'émetteur et le récepteur. Par conséquent, de nombreux bonds courts nécessitent moins d'énergie qu'un long saut. L'énergie consommée pour transmettre  $q$ -bit de données sur une distance  $d$  à un nœud capteur est égale à [3]:

$$E_{tx}(q, d) = q * E_{elec} + q * \epsilon_{fs} d^2 \quad (1)$$

L'énergie consommée pour recevoir  $q$ -bit de données est égale à :

$$E_{rx}(q) = q * E_{elec} \quad (2)$$

Avec :

$E_{elec}$  : l'énergie consommée par les circuits électroniques (énergie de calcul) pour la transmission/réception électronique ;

$d$  : distance entre l'émetteur et le récepteur ;

$\epsilon_{fs}$  : l'énergie perdue dans l'espace de transmission.

#### I.3.2 La topologie dynamique de réseau

Généralement, le RCSF est un réseau dynamique. Le nœud peut échouer à cause d'épuisement de la batterie ou à cause d'autres circonstances, le canal de

communication peut être perturbé ainsi que d'autres nœuds supplémentaires peuvent être ajoutés au réseau qui cause des changements fréquents dans la topologie de réseau, c'est pour cela les RCSF doivent être reconfigurables et auto-ajustables.

Les problèmes liés à la maintenance et à la modification de la topologie peuvent être classés en trois phases [1] :

- La phase de pré-déploiement et de déploiement : les nœuds de capteur peuvent être projetés en masse ou placés un par un dans le champ du captage. Ils peuvent être déployés en larguant un avion, livrés dans un obus d'artillerie, une roquette ou un missile ou placés l'un après l'autre par un humain ou un robot.
- La phase post-déploiement : après le déploiement, les changements de topologie sont dus à la modification de la position des nœuds de capteurs, de l'accessibilité (due au blocage, au bruit, aux obstacles mobiles, etc.), de l'énergie disponible, du dysfonctionnement et des détails des tâches.
- La phase de redéploiement de nœuds supplémentaires : des nœuds de capteurs supplémentaires peuvent être redéployés à tout moment pour remplacer des nœuds défaillants ou en raison de changements dans la dynamique des tâches.

### **I.3.3 La communication multi-sauts**

Un RCSF se compose d'un grand nombre de nœuds déployés dans une zone généralement très large. La communication multi-saut est un type de communication dans les réseaux sans fil dans lesquels la zone de couverture est plus grande que la portée radio des nœuds. La communication du nœud émetteur et la station de base se fait à l'aide des nœuds intermédiaires via un protocole de routage. Si nous avons besoin de communiquer avec un autre nœud ou station de base qui est au-delà de sa fréquence radio, il doit passer par nœud intermédiaires. Dans un réseau de capteurs multi-sauts, les nœuds en communication sont reliés par un support sans fil. Ces liens peuvent être formés par un média radio, infrarouge ou optique. Pour permettre le fonctionnement global de ces réseaux, le support de transmission choisi, doit être disponible partout. Une grande partie du matériel actuel des nœuds de capteurs repose sur la conception de circuits de radiofréquence (*RF*). Parmi les technologies utilisées pour la communication entre les capteurs on trouve la norme IEEE **802.15.4 (Zigbee)**, **802.15.1 (Bluetooth)** et la norme **802.11 (Wifi)** qui permettent l'échange bidirectionnel de données sur une bande de fréquence de 2,4 GHz. Un autre mode



possible de communication entre les nœuds dans les réseaux de capteurs est l'infrarouge.

### I.3.4 Le déploiement des capteurs

C'est la première opération (phase) dans le cycle de vie d'un réseau de capteurs. On peut envisager plusieurs formes de déploiements selon les besoins des applications. Les nœuds peuvent être déployés aléatoirement d'un avion ou d'une roquette par exemple, ou bien ils peuvent être placés un par un d'une manière déterministe par un humain ou un robot. Le déploiement peut être fait d'un seul coup ou bien peut être un processus continu en redéployant d'autres capteurs dans une même zone. Dans un grand nombre d'applications, le déploiement manuel est impossible. De plus, même lorsque l'application permet un déploiement déterministe, le déploiement aléatoire est adopté dans la majorité des scénarios à cause de raisons pratiques tels que le coût et le temps. Cependant, le déploiement aléatoire ne peut pas fournir une distribution uniforme sur la région d'intérêt, ce qui déclenche de nouveaux problèmes dans les réseaux de capteurs. Les principaux problèmes engendrés sont la localisation, la couverture de la zone, la connexité et la sécurité.

Le nombre de nœuds capteurs déployés pour étudier un phénomène peut être de l'ordre de centaines ou de milliers selon l'application. Les nouveaux schémas doivent pouvoir fonctionner avec ce nombre de nœuds. Ils doivent également utiliser la haute densité des réseaux de capteurs. La densité peut aller de quelques nœuds de capteurs à plusieurs centaines de nœuds de capteurs dans une région, dont le diamètre peut être inférieur à 10 m. La densité  $\mu$  peut être calculée selon [1] comme suit :

$$\mu(R) = (N \cdot \pi R^2)/A. \quad (3)$$

Où  $N$  est le nombre de nœuds de capteurs dispersés dans la région  $A$  et  $R$  est la plage de transmission radio. Fondamentalement,  $\mu(R)$  donne le nombre de nœuds dans le rayon de transmission de chaque nœud dans la région  $A$ .

## I.4 Domaines d'application des RCSF

Actuellement, les réseaux de capteurs sont utilisés dans plusieurs domaines, parmi ces domaines où ces réseaux se révèlent très utiles et peuvent offrir de meilleures contributions ; on peut noter le médical, le militaire, l'environnemental et les applications de la surveillance (voir **Figure 5**).

- **Le domaine médical**

La demande émergente des réseaux de capteurs sans fil implique leur utilisation dans les soins médicaux[4][5][6][7]. Les RCSF peuvent être utilisés pour assurer une surveillance permanente des organes vitaux de l'être humain grâce à des micro-capteurs qui pourront être avalés ou implantés sous la peau. Ils peuvent aussi faciliter le diagnostic de quelques maladies en effectuant des mesures physiologiques telles que : la tension artérielle, battements du cœur à l'aide des capteurs ayant chacun une tâche bien particulière.

- **Le domaine militaire**

Les applications militaires[8] sont très étroitement liées au concept des réseaux de capteurs sans fil. En fait, il est difficile de dire avec certitude si des nœuds sans fil ont été développés en raison des besoins militaires ou s'ils ont été inventés séparément et ensuite ils ont été appliqués aux services militaires. En ce qui concerne les applications militaires, la zone d'intérêt s'étend de la collecte de l'information au repérage de l'ennemi, la surveillance du champ de bataille et la classification cible

- **La surveillance**

L'application des réseaux de capteurs dans le domaine de la sécurité[9][10] peut diminuer considérablement les dépenses financières consacrées à la sécurisation des lieux et des êtres humains. Ainsi, l'intégration des capteurs dans de grandes structures telles que les ponts ou les bâtiments aidera à détecter les fissures et les altérations dans la structure suite à un séisme ou au vieillissement de la structure. Le déploiement d'un réseau de capteurs de mouvement peut constituer un système d'alarme qui servira à détecter les intrusions dans une zone de surveillance.

- **Les applications environnementales**

L'évolution de la technologie dans le domaine de communication sans fil a permis de déployer les réseaux de capteurs très proche ou à l'intérieur du phénomène à observer et a permis aux petits nœuds de capturer les informations environnementales telles que la pression barométrique, la température ambiante, l'humidité atmosphérique, la direction et la vitesse du vent, le niveau de l'eau souterraine et les précipitations météorologie[11]. La collaboration des nœuds de capteurs apporte plusieurs avantages par rapport à la détection traditionnelle, notamment une plus grande tolérance aux pannes et une meilleure précision.

- **La sécurité**

L'application des réseaux de capteurs dans le domaine de la sécurité[12] peut diminuer considérablement les dépenses financières consacrées à la sécurisation des lieux et des êtres humains. Ainsi, l'intégration des capteurs dans de grandes structures telles que les ponts ou les bâtiments aidera à détecter les fissures et les altérations dans la structure suite à un séisme ou au vieillissement de la structure. Le déploiement d'un réseau de capteurs pour la détection du mouvement peut constituer un système d'alarme qui servira à détecter les intrusions dans une zone de surveillance.

- **La domotique**

Avec le développement technologique, les capteurs sans fil peuvent être embarqués dans des appareils, tels que les aspirateurs, les fours à micro-ondes, les réfrigérateurs...etc. Ces capteurs embarqués peuvent interagir entre eux et avec un réseau externe via l'internet pour permettre à l'utilisateur de contrôler les appareils domestiques localement ou à distance. Le déploiement des capteurs pour détecter le mouvement et mesurer la température dans les futures maisons dites intelligentes permet d'automatiser plusieurs opérations domestiques telles que : la lumière s'éteint et la musique se met en état d'arrêt quand la chambre est vide, la climatisation et le chauffage s'ajustent selon les points multiples de mesure, le déclenchement d'une alarme par le capteur anti-intrusion quand un intrus veut accéder à la maison[13].

- **Les applications commerciales**

Des nœuds capteurs pourraient améliorer le processus de stockage et de livraison. Le réseau ainsi formé pourra être utilisé pour connaître la position, l'état et la direction d'un paquet ou d'une cargaison. Un client attendant un paquet peut alors avoir un avis de livraison en temps réel et connaître la position du paquet. Des entreprises manufacturières via des réseaux de capteurs pourraient suivre le procédé de production à partir des matières premières jusqu'au produit final livré. Grâce aux réseaux de capteurs, les entreprises pourraient offrir une meilleure qualité de service tout en réduisant leurs coûts. Les produits en fin de vie pourraient être mieux démontés et recyclés ou réutilisés si les micro-capteurs en garantissent le bon état. Dans les immeubles, le système de climatisation peut être conçu en intégrant plusieurs micro-capteurs dans les tuiles du plancher et les meubles. Ainsi, la climatisation pourra être déclenchée seulement aux endroits où il y a des personnes présentes et seulement si c'est nécessaire[14].



Figure 5: Applications des réseaux de capteurs

### I.5 Paramètres influençant la conception d'un RCSF

La conception d'un RCSF est influencée par plusieurs paramètres qu'une fois pris en compte, génèrent un guide de conception, ce guide peut être aussi utilisé pour comparer entre deux RCSF. Parmi ces paramètres, nous citons les suivants :

- **La tolérance aux pannes**

Les algorithmes et protocoles doivent tenir compte du fait qu'un nœud peut cesser de fonctionner à cause de manque d'énergie ou parce qu'il a été détruit. Ils devront adapter leur niveau de tolérance aux pannes en fonction de l'hostilité du milieu dans lequel est déployé le réseau.

- **Le facteur d'échelle**

Le nombre de nœuds capteurs déployés peut atteindre des centaines, des milliers pour certaines applications. Les protocoles et algorithmes devront pouvoir fonctionner correctement dans tous les cas. Classiquement, la densité des nœuds est de l'ordre de 300 nœuds pour 25m<sup>2</sup> pour la surveillance de machines et de 10 nœuds par région pour la surveillance de véhicule. Une densité de 20 nœuds/m<sup>2</sup> est très courante. Dans une maison, on estime à 2 douzaines le nombre d'appareils qui comporteront des capteurs. Pour la domotique, la densité devrait être comprise entre 20 et 100 nœuds par région. Les densités devraient être encore plus importantes si l'on insère des capteurs dans les lunettes, les vêtements, les chaussures, les montres, les bijoux ou encore le corps humain.

- **Les coûts de production**

Si le coût d'un nœud capteur est tel que le déploiement d'un réseau revient plus cher que les capteurs traditionnels, alors le RCSF n'est pas justifié économiquement. L'état actuel de la technologie nous permet de produire des capteurs Bluetooth pour dix dollars l'unité et des pico-capteurs pour moins d'un dollar. Le seuil à atteindre pour que la solution RCSF soit économiquement faisable est bien inférieur à un dollar.

- **Les contraintes matérielles**

Un nœud capteur est constitué de composants consommant de l'énergie électrique, comme il peut aussi avoir des modules dédiés à l'application ou destinés à la localisation ou au mouvement. Par conséquent, la capacité de la batterie d'alimentation du nœud forme une contrainte matérielle importante. Il est aussi connu que le volume du nœud capteur n'excède pas celui d'une boîte d'allumettes, ceci s'oppose à ce qui précède. Par ailleurs, les transmissions radio d'un nœud capteur sont plus complexes et consomment plus d'énergie que les liaisons optiques. Elles sont quand même préférées en raison de leur temps de fonctionnement très bref. Ensuite, bien que les progrès soient constants en micro-électronique, les ressources du microprocesseur sont limitées.

- **La topologie de réseau**

La forte probabilité de panne d'un nœud capteur ou la possibilité de rajouter des nœuds font que la topologie de réseau doit pouvoir changer dynamiquement. Il faut donc gérer avec précision la maintenance de cette topologie. On distingue trois phases : le déploiement, le post-déploiement et le redéploiement des nœuds capteurs. Les nœuds capteurs peuvent être disséminés à partir des hauteurs ou placés dans des localisations précises de la zone d'intérêt. Quelle que soit la méthode utilisée, le coût du déploiement doit rester le plus faible possible. À cause des déplacements des nœuds, d'obstacles mobiles ou de brouillages volontaires, des actions post déploiement peuvent être opérées sur la topologie du RCSF. Des nœuds capteurs peuvent aussi être déployés dans un deuxième temps pour remplacer ceux tombés en panne ou détruits. La topologie d'un RCSF doit s'adapter à toutes les situations, à savoir les pannes, la mobilité et le rajout des nœuds.

- **L'environnement de déploiement**

Les nœuds peuvent être déployés tout près de l'objet à surveiller. L'environnement de déploiement peut être à l'intérieur d'une grosse machine, au fond

d'un océan, dans un lieu contaminé biologiquement ou chimiquement, dans un champ de bataille, dans une maison ou un immeuble, sur un animal, sur un véhicule, etc. Ces situations très variées engendrent des contraintes très fortes de l'environnement sur les nœuds capteurs.

- **Le support de transmission**

Le support de transmission des nœuds doit être universel. Une possibilité de choisir les bandes de fréquences des domaines de l'industrie, les sciences et la santé qui ne nécessitent pas de licences sont disponibles dans tous les pays. Des contraintes de puissance et de consommation d'énergie font qu'en réalité seront les hautes fréquences qui sont avantageuses. Le choix s'est porté sur la bande des 433 Mhz en Europe et des 915 MHz aux USA. La transmission infrarouge aurait pu être intéressante, mais elle a été écartée parce qu'elle exige une vue directe entre l'émetteur et le récepteur.

- **Le traitement des données**

Sachant que le traitement d'un paquet est beaucoup moins coûteux que sa transmission par un nœud capteur, on privilégie alors systématiquement les traitements locaux dans chaque nœud capteur (filtrage, calcul de la moyenne, etc.) pour minimiser le volume des données circulant dans le RCSF.

- **La consommation d'énergie**

Les nœuds capteurs embarquent en général une quantité d'énergie très limitée. Les transmissions de données entre les nœuds consomment beaucoup d'énergie, surtout si des nœuds tombent en panne et qu'il faut réorganiser le RCSF. C'est pourquoi des algorithmes et des protocoles ont été développés ayant pour principal souci l'économie d'énergie, au détriment parfois de la qualité de service fournie par un RCSF.

## **I.6 Les différentes problématiques dans les RCSF**

Les recherches dans le domaine des réseaux de capteurs ont révélé plusieurs problématiques, parmi ces problématiques, nous citons :

- **Routage** : Concevoir un protocole de routage performant en termes de minimisation de la consommation de l'énergie, du choix des routes optimales pour l'acheminement de l'information d'un capteur à la station de base et vice versa, de réduction du délai de délivrance des paquets...Ainsi le réseau doit passer à l'échelle sans que ses performances se dégradent.

- **Couche MAC** : La spécificité des réseaux de capteurs sans fil mobiles nécessite le développement de nouveaux protocoles MAC qui s'adaptent aux contraintes imposées par ces réseaux. Ceci est dans le but d'améliorer le débit, minimiser la consommation d'énergie, optimiser le partage du médium ainsi que minimiser le délai de délivrance des paquets.

- **Qualité de service** : Des protocoles au niveau de la couche MAC devraient être capables d'établir des priorités entre les flux, limiter les pertes de paquets vitaux pour la gestion de réseau, ou du moins en restreindre l'impact.

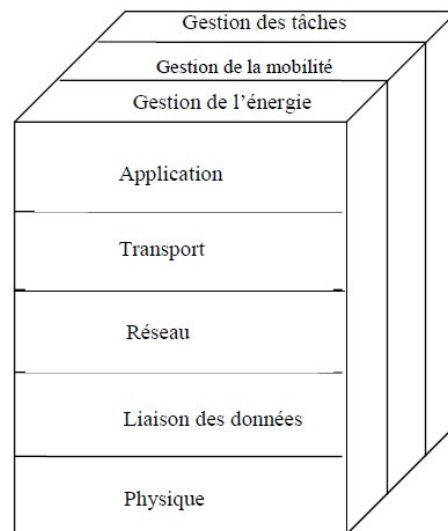
- **Cross-layer** : Dans les modèles classiques, les concepteurs essaient d'optimiser les performances au niveau d'une couche indépendamment des autres couches. Cependant, une telle indépendance est communément considérée comme trop onéreuse pour les réseaux de capteurs. Par conséquent, une coopération permettant un compromis entre performances, dépendance et flexibilité doit être proposée pour optimiser les capacités de réseau en général.

- **Diffusion de l'information** : les protocoles de diffusion conçus pour les réseaux de capteurs doivent tenir compte de leurs spécificités ainsi que de leurs contraintes intrinsèques imposées. Ainsi, pour concevoir un protocole efficace, il faudrait assurer une couverture maximale des capteurs composant le réseau (taux d'accessibilité supérieur 90%), minimiser le nombre des réémetteurs et des réceptions redondantes ainsi que la consommation d'énergie.

- **la sécurité** : pour les applications qui exigent un niveau de sécurité assez élevé tel que les applications militaires, des mécanismes d'authentification, de confidentialité, et d'intégrité doivent être mis en place au sein de leur communauté. Les algorithmes de cryptographie conçus pour les réseaux de capteurs doivent tenir compte des ressources limitées que présentent ces réseaux.

## **I.7 La pile protocolaire des capteurs**

Cette architecture est mise en place afin de structurer les protocoles de communication dans les RCSF. Ce modèle comprend huit couches comme présentées dans la **figure 6**, cinq d'entre elles ont les mêmes tâches que celles du modèle OSI (*Open System Interconnection*), et trois autres couches pour assurer la gestion d'énergie (*Power Management Plane*), la gestion de la mobilité (*Mobility Management Plane*), et la gestion des tâches (*Task Management Plane*).



**Figure 6** : Modèle en couches pour la communication dans les RCSF

### I.7.1 Les rôles des couches

Le rôle de chacune des cinq couches est :

- **La couche physique** : Cette couche se charge de tout ce qui est spécifications des caractéristiques matérielles, la génération des ondes porteuses, la modulation de données et leur injection sur le support de transmission tout en sélectionnant les bonnes fréquences.
- **La couche liaison de données** : dans cette couche on spécifie comment les données sont expédiées entre deux nœuds dans une distance d'un saut. Elle est responsable de l'accès au média physique, la gestion de la mobilité, la gestion de l'énergie, la gestion des tâches, le multiplexage des données et le contrôle d'erreurs. Elle assure la liaison point à point et multipoint dans un réseau de communication. Parmi les protocoles qui opèrent au niveau de cette couche on cite : SMAC (*Self-organizing Medium Access Control for Sensor Networks*) et EAR (*Eavesdrop And Register*).
- **La couche de réseau** : La couche réseau a pour but principal de baliser une route optimale en vue d'acheminer efficacement les données captées depuis leur source jusqu'au puits, tout en minimisant la dissipation énergétique des nœuds capteurs inclus dans le chemin. La tâche de routage au sein d'un réseau de capteurs est spécifique du fait que :



- L'acheminement de données récoltées à partir de multiples sources vers une seule destination (la station de base).
  - La forte redondance de données et l'exigence de l'agrégation.
  - La nécessité d'une gestion soigneuse des ressources (énergie, mémoire, bande passante).
- **La couche du transport** : Cette couche est chargée du transport de données, de la vérification de la qualité de la transmission et de la gestion des éventuelles erreurs. Dans le cas des réseaux de capteurs sans fil, la bonne qualité de transmission est souvent négligée, car d'une part les pertes sont très probables avec un support de transmission sans fil et d'autre part, les mécanismes de gestion de la fiabilité sont trop lourds (tout comme le protocole TCP (*Transmission Control Protocole*)). Ainsi, les pertes et les erreurs de transmission sont tolérables et peuvent même être camouflées par la redondance de données et l'agrégation. Le protocole UDP (*User Datagram Protocole*) qui fournit un service de transport en mode datagramme (sans connexion, sans gestion de congestion et sans fiabilité) est jugé être le protocole de transport le mieux adapté aux environnements capteurs en raison de sa faible empreinte mémoire et simplicité.
  - **La couche application** : La couche application présente le niveau le plus proche des utilisateurs. De nombreux profils d'applications peuvent être configurées et utilisées dans la couche application des réseaux de capteurs sans fil.
  - **Le plan de gestion d'énergie** : Les nœuds capteurs sont sévèrement limités en ressources d'énergie, qui influence directement sur la durée de vie du réseau. Ainsi, le plan de gestion d'énergie doit fournir des mécanismes de gestion efficaces pour réduire le degré de consommation d'énergie, et éliminer les sources de gaspillage de celle-ci.
  - **Le plan de gestion de mobilité** : Ce plan est responsable du contrôle du mouvement des nœuds capteurs dans le cas où ils sont mobiles. Il peut par exemple enregistrer les trajectoires d'un nœud capteur afin de l'aider à se localiser.
  - **Le plan de gestion de tâche** : Dans un réseau de capteurs, les nœuds peuvent effectuer des tâches qui se différencient en termes de consommation de ressources. Ainsi, un plan de gestion de tâche est souvent nécessaire afin de répartir d'une manière

équitable les tâches sur les nœuds capteurs, et offrir ainsi une gestion efficace des ressources disponibles.

## **I.8 Conclusion**

Dans ce chapitre, nous avons introduit les réseaux de capteurs sans fil, ou nous avons présenté tous les aspects liés à cette technologie et comment elle est utilisée dans tous les domaines d'application. Outre ce type de RCSF classique que nous avons présenté et qui était déployé pour des applications privées où les données étaient récupérables à partir des stations de base, la nouvelle génération des RCSF est désormais invitée à intégrer l'Internet. Dans ce cas, les rapports des capteurs intégrés à Internet sont accessibles de n'importe où et n'importe quand, à partir d'un autre bout connecté également à Internet. Donc, l'accès, le stockage des données et la récupération des données deviennent une question très importante à traiter.

Sommaire

<b>II Le Cloud Computing .....</b>	<b>31</b>
II.1 Introduction.....	32
II.2 Définition.....	32
II.3 Les formes de services .....	33
II.4 La technologie de virtualisation .....	34
II.5 Historique du Cloud Computing .....	36
II.6 Modèles du déploiement .....	37
II.7 Caractéristiques du Cloud Computing .....	38
II.8 Avantages et inconvénients du Cloud Computing .....	40
II.8.1 Les avantages .....	40
II.8.2 Les inconvénients .....	41
II.9 Le cloud pour le RCSF .....	43
II.10 Conclusion .....	43

## II.1 Introduction

Le Cloud Computing est une nouvelle technologie qui fournit des ressources, des logiciels et des données à la demande et permet de gérer et adapter rapidement toute infrastructure à des variations de charges de manière transparente. Dans cette section, nous présentons les concepts du Cloud Computing et son architecture. Premièrement, nous définissons le Cloud Computing et nous soulignons ses caractéristiques et ses opportunités. Ensuite, nous décrivons les modèles de service courants et les modèles de déploiement du Cloud Computing.

## II.2 Définition

Le Cloud Computing (l'informatique en nuage)[2] est apparu récemment comme un nouveau paradigme pour héberger et offrir des services sur le réseau Internet. La définition du Cloud Computing selon NIST (*National Institute of Standards and Technology*) [15]« *le Cloud Computing est l'accès via un réseau de télécommunications à la demande et en libre-service, à des ressources informatiques partagées et configurables. La plateforme de Cloud Computing fournit, configure et reconfigure les serveurs dynamiquement quand et tant que les utilisateurs l'ont besoin* ».

Le Cloud Computing permet d'offrir des capacités informatiques énormes et variées : capacité de calcul, espace de stockage, bande passante ou encore des logiciels, mais la forte demande de ressources soumise par les utilisateurs pose un problème lorsqu'elle dépasse les capacités du fournisseur de ressources. Les serveurs du Cloud Computing peuvent être sous la forme de machines virtuelles ou des machines physiques dans le Cloud Computing [16]. Ce nouveau concept a été initié par le géant Amazon en 2002, il consiste à fournir des capacités de traitement informatique évolutives, élastiques et mises à disposition comme un service pour les utilisateurs qui y accèdent via Internet sans aucune gestion de l'infrastructure sous-jacente.

Avec cette technologie, c'est possible pour les utilisateurs d'accéder via le réseau Internet à l'infrastructure matérielle ou logicielle et de déployer des applications indépendamment de leurs localisations, à des prix compétitifs et avec une bonne qualité de service. Dans le Cloud Computing, les services sont fournis dynamiquement à travers des centres de données (*Data Centers*) en hébergeant des machines virtuelles interconnectées. La notion d'évolutivité et d'élasticité du Cloud Computing s'explique

par le fait que l'utilisateur peut faire varier les ressources demandées à la hausse comme à la baisse et ce de manière très dynamique, avec une facturation des ressources à la consommation [12]. Amazon Web Services (AWS)[17] et OpenStack [18] sont des exemples des plateformes qui offrent des services aux utilisateurs dans le modèle 'payer à la demande' .

### II.3 Les formes de services

Le cloud computing est un paradigme orienté service. Conceptuellement, les services offerts par le Cloud Computing peuvent être regroupés en trois catégories en fonction du niveau d'abstraction[15] :

- **Le service IAAS (Infrastructure As A Service)**

L'infrastructure en tant que service ou en anglais Infrastructure as a service (IAAS) C'est le modèle de Cloud Computing le plus connu, ce modèle de service est basé sur le concept de payer au fur et à mesure (*pay-as-you-use*), il est distribué et largement automatisé et capable de répondre aux exigences de mise en production des applications de l'entreprise. Ce modèle de service fait référence à la provision d'une demande de ressources infrastructurelles tels que le traitement, les réseaux, le stockage et autres ressources informatiques livrés en tant que service aux clients, généralement ces services offerts par ce modèle sont en termes de machines virtuelles. Parmi les fournisseurs qui offrent des services de type IAAS on trouve Desktone[19], Proectio[20], DotRiver[21], Amazon EC2[22], Windows Azure[23], Rackspace[24] et OVH[25].

- **Le service PAAS (Plateforme As Service)**

La plate-forme en tant que service est une approche qui offre un niveau d'abstraction plus élevé que celle d'IAAS. Cette classe de service offre une plateforme sur laquelle les développeurs peuvent créer et déployer des applications sans se préoccuper du nombre de serveurs ou de la quantité de mémoire que les applications utiliseront[2]. Dans ce modèle de service, le fournisseur est responsable du système d'exploitation et des outils infrastructurels. Le client (ou le consommateur) peut contrôler des applications et peut ajouter aussi ses propres outils. Parmi les services de type PAAS on trouve Google AppEngine[26] et netsuit[27].

- **Le service SAAS (Software As Service)**

Le modèle du service logiciel en tant que service (Software as Service), Ce modèle fournit les logiciels aux utilisateurs sous la forme d'un service hébergé utilisable par le biais d'Internet et du Web. Du point de vue des utilisateurs, le SAAS peut réduire les dépenses de la configuration du serveur et l'achat de la licence des logiciels; du point de vue des fournisseurs, il peut réduire les coûts de la maintenance de leurs logiciels [28]. SalesForce[29] et G-suite[30] sont les fournisseurs les plus connus de ce type de service. voir la **Figure 7**.

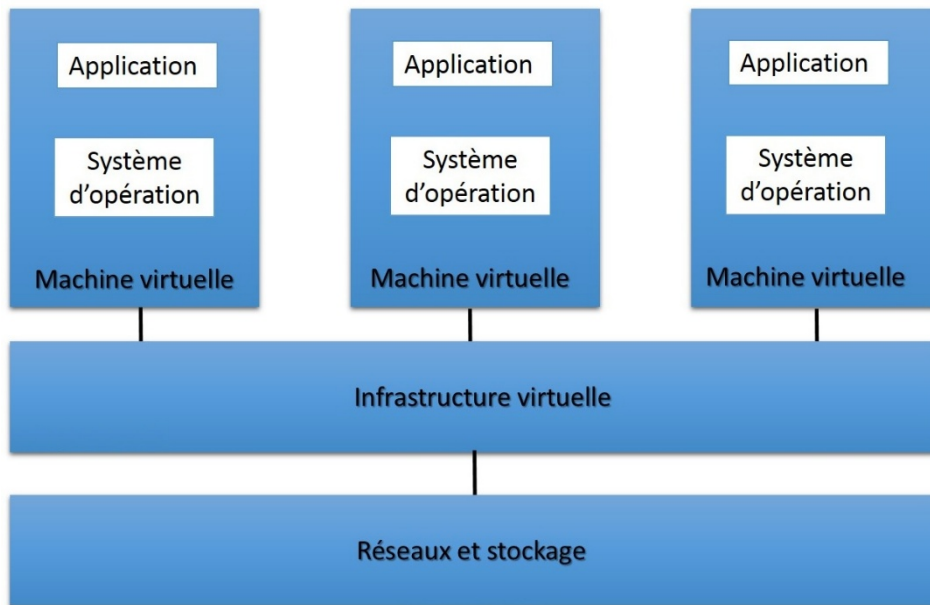


**Figure 7:** Les principaux fournisseurs de IAAS,PAAS et SAAS

## II.4 La technologie de virtualisation

Le Cloud Computing utilise la capacité de séparer et distribuer les ressources informatiques dynamiquement. Ce qui n'est pas une tâche facile. La technologie de virtualisation est la réponse à cette question. La virtualisation dans le Cloud Computing peut être obtenue au niveau du matériel et au niveau de logiciel. L'utilisation de cette technologie sépare la logique métier des ressources du calcul, ce qui améliore la capacité d'accès aux services [28], la **Figure 8** représente le modèle de la virtualisation dans le Cloud Computing le plus connu. Les ressources virtualisées comme le processeur et la mémoire peuvent être dimensionnés et

redimensionnés avec certaine flexibilité. Cette caractéristique rend la virtualisation de matériel ; la technologie idéale pour créer une infrastructure virtuelle qui partitionne un centre de données entre plusieurs locataires [2].



**Figure 8:** Modèle de virtualisation de Cloud Computing

Grâce à la virtualisation, l'infrastructure du centre de données peut être consolidée par de centaines de serveurs à seulement des dizaines de serveurs, tels que les serveurs Web, les serveurs de base de données et les serveurs de messagerie, qui s'exécutent en tant qu'instances virtualisées. Cela se traduit par une réduction du coût total de mis en œuvre des infrastructures du Cloud Computing [85].

La virtualisation est une technique qui permet de partager une seule instance physique d'une ressource ou d'une application entre plusieurs clients et organisations. La création d'une machine virtuelle sur le système d'exploitation et le matériel existants est appelée virtualisation matérielle. La virtualisation matérielle est principalement effectuée pour les plates-formes du serveur, car la gestion des machines virtuelles est beaucoup plus facile que la gestion d'un serveur physique. Une machine virtuelle fournit un environnement logiquement séparé du matériel sous-jacent. La machine sur laquelle la machine virtuelle est créée s'appelle 'la machine hôte' et cette machine virtuelle est appelée machine invitée. Après la virtualisation du système

matériel, on peut installer différents systèmes d'exploitation et exécuter différentes applications sur ces derniers.

La virtualisation joue un rôle très important dans la technologie de Cloud Computing. Les utilisateurs partagent les données et les applications. L'une des caractéristiques importantes de la virtualisation est qu'elle permet de partager des applications avec plusieurs clients et entreprises, La virtualisation dans le Cloud gère également la charge du travail en transformant l'informatique traditionnelle et en la rendant plus évolutive, économique et efficace., le client peut optimiser les ressources et réduire le système physique nécessaire.

## **II.5 Historique du Cloud Computing**

Le Cloud Computing [2] a émergé dans les dernières années comme une solution universelle utilisée par les différents types d'utilisateurs. Bien que l'idée de virtualisation est très ancienne, cette technologie offre des avantages tels que : un espace de stockage très important et un moyen de calcul puissant. Aussi, le concept du Cloud Computing est proposé comme une idée d'après John McCarthy<sup>1</sup> quand il déclare que : « dans le future l'informatique devient un outil publique et organisé »[31]. L'informatique classique a rencontré des limites d'utilisation telles que la faible qualité de la bande passante et le contrôle de données sont classés parmi les problèmes majeurs[32].

L'arrivé du Cloud Computing représente la solution à ces limites par offrir une très large bande passante et moyen de calcul très puissant. La première génération du Cloud Computing a été proposée pour le domaine e-commerce. D'après l'utilisation et le succès de ce paradigme dans les services e-commerce, les fournisseurs de Cloud Computing offrent des solutions et des opportunités afin de développer des services différents qui consistent en une liste de services internet consommés.

---

<sup>1</sup> John McCarthy (né le 3 septembre 1927, à Boston, dans le Massachusetts - mort le 24 octobre 2011) est le principal pionnier de l'intelligence artificielle ; il incarne le courant mettant l'accent sur la logique symbolique.



## II.6 Modèles du déploiement

Bien que le Cloud Computing soit émergé principalement dans un cadre d'utilité publique, d'autres modèles de déploiement ont été adoptés avec des variations de localisation physique et de distribution. Dans ce sens, quelle que soit sa classe de service, un Cloud peut être classé comme publique, privé ou hybride [2].

- **Le Cloud privé**

Appelé également Cloud interne, Le Cloud privé est une infrastructure de Cloud Computing exploitée uniquement par une seule organisation[15]. Mettre en place un cloud privé signifie restructurer une infrastructure existante en ajoutant la virtualisation et les interfaces de type cloud. Cela permet aux utilisateurs d'interagir avec le centre de données local tout en bénéficiant des mêmes avantages que les Clouds publiques, notamment l'interface libre-service, l'accès privilégié aux serveurs virtuels, ainsi que le contrôle et la facturation à la consommation. Le Cloud privé est construit, exploité et géré par une organisation pour son usage interne uniquement afin de prendre en charge ses opérations commerciales exclusivement. Les organisations publiques, privées et gouvernementales adoptent ce modèle pour exploiter les avantages du Cloud Computing, tels que la flexibilité, la réduction du coût et l'agilité [2].

- **Cloud publique**

Appelé également Cloud externe. Un Cloud Computing est dit publique lorsque les services sont rendus sur un réseau ouvert au public [14]. Les services de Cloud publique peuvent être gratuits. Techniquement, il peut y avoir peu ou pas de différence entre l'architecture Cloud publique et privée, cependant, la sécurité peut être considérablement différente pour les services (applications, stockage et autres ressources) mis à disposition par un fournisseur de service pour un public et lorsque la communication est effectuée sur un réseau non sécurisé.

Généralement, les fournisseurs de services de Cloud public comme Amazon Web Services (AWS[17]), Oracle[33], Microsoft et Google possèdent et exploitent l'infrastructure de leur centre de données et l'accès se fait généralement via Internet. AWS, Oracle, Microsoft et Google proposent également des services de connexion directe appelés respectivement «AWS Direct Connect», «Oracle FastConnect», «Azure ExpressRoute» et «Cloud Interconnect». Ces connexions nécessitent que les clients achètent ou louent une connexion privée à un réseau privé[34].

- **Cloud hybride**

Le Cloud hybride est une composition de deux ou plusieurs types de Cloud (privés, communautaires ou publics) qui restent des entités distinctes, mais sont liées entre elles, offrant les avantages de plusieurs modèles de déploiement. Le Cloud hybride peut également signifier la possibilité de connecter des services de colocalisation, gérés et / ou dédiés avec des ressources Cloud[15]. Gartner<sup>2</sup> définit un service de Cloud hybride comme un service composé d'une combinaison de services de Cloud privé, public et communautaire. Un service de Cloud hybride croise les limites d'isolation et de fournisseur de sorte qu'il ne peut pas être simplement mis dans une catégorie de service de Cloud privé, public ou communautaire. Il permet d'étendre la capacité d'un service Cloud, par agrégation, intégration ou personnalisation avec un autre service Cloud.

Un autre exemple de Cloud hybride est celui où les organisations informatiques utilisent les ressources de Cloud Computing public pour répondre aux besoins de capacité temporaire qui ne peuvent pas être satisfaits par le Cloud privé [15].

## **II.7 Caractéristiques du Cloud Computing**

Comme les services informatiques dans le Cloud se développent sur les plans commercial et technologique, il sera plus facile pour les entreprises de maximiser les avantages potentiels. Les services du Cloud Computing, en particulier les services du Cloud publique, partagent certaines caractéristiques attendues de tous les fournisseurs de services Cloud Computing. Parmi les caractéristiques du Cloud Computing on cite :

- **La flexibilité**

Les services Cloud sont destinés à supporter les lourdes comme les petites charges de travail en augmentant ou réduisant automatiquement les ressources utilisées selon la tailles des taches à traiter. Les ressources sont disponibles sur le réseau et sont accessibles via diverses plates-formes client. En d'autres termes, les services du Cloud sont disponibles sur un réseau - idéalement un lien de communication à large bande haut débit - comme Internet ou dans le cas d'un Cloud privé, il pourrait s'agir d'un réseau local.

---

<sup>1</sup>Gartner Inc. est une entreprise américaine et un fournisseur d'études de marché et d'analyse des développements informatiques.

- **L'auto-guérison**

Tout système Cloud Computing doit contenir une ou plusieurs copies de chaque application déployée, de telle façon qu'en cas de disfonctionnement de l'application en cours, l'application en copie peut la remplacer en prenant l'état actuel de l'application en échec. Les applications en copies doivent être maintenues et mises à jour à chaque fois que l'application en cours est modifiée [35].

- **La virtualisation**

La virtualisation est la création d'une version virtuelle (plutôt que réelle) de quelque chose similaire à la version réelle, par exemple un serveur, un périphérique de stockage, un ordinateur de bureau, un système d'exploitation ou des ressources réseau. En d'autres termes, la virtualisation est une technique qui permet de partager une seule instance physique d'une ressource ou d'une application entre plusieurs clients et organisations. Pour ce faire, il attribue un nom logique à une mémoire physique et fournit un pointeur sur cette ressource physique, le cas échéant. Un système basé Cloud Computing est un système complètement virtuel et indépendant de la couche physique sur laquelle les ressources et applications sont déployées.

- **L'élasticité**

L'un des points forts de Cloud Computing est la capacité de provisionner rapidement des ressources dans le Cloud selon les besoins des entreprises. Et ensuite pour les enlever quand ils n'en ont pas besoin. Les ressources informatiques dans le Cloud peuvent évoluer rapidement, voire automatiquement, en fonction des besoins de l'entreprise. C'est une caractéristique clé du Cloud Computing. L'élasticité est un élément clé de Cloud et implique que les entreprises de fabrication peuvent rapidement fournir et désapprovisionner toutes les ressources de l'informatique dans le Cloud Computing. Le provisionnement et le déblocage rapides peuvent s'appliquer au stockage, aux machines virtuelles ou aux applications client.

- **La multi-location**

Un cloud multi-locataire est une architecture de cloud computing qui permet aux clients de partager des ressources informatiques dans un cloud public ou privé. Les données de chaque locataire sont isolées et restent invisibles pour les autres locataires. Ce terme signifie que c'est essentiel d'avoir des modèles pour une politique d'exécution des applications, segmentation, isolation, gouvernance, niveau de service,

et facturation pour les différentes catégories de consommateurs. Dans un cloud à locataire unique, un seul client est hébergé sur un serveur et l'accès lui est accordé. Cependant, les architectures de multi-locataires hébergeant plusieurs clients sur les mêmes serveurs. Les clouds à locataire unique permettent aux clients de mieux contrôler la gestion des données, le stockage, la sécurité et la performance.

## **II.8 Avantages et inconvénients du Cloud Computing**

Beaucoup de travaux dans la littérature ont discuté les avantages et les inconvénients résultant de l'adoption de la technologie du Cloud Computing.

### **II.8.1 Les avantages**

Le Cloud Computing existe depuis environ deux décennies. En plus, des données indiquant son efficacité, et ses avantages en termes de coûts et d'avantages concurrentiels dont il dispose par rapport à l'ancien mode de fonctionnement, le cloud computing présente d'autres points forts dont on cite :

- **La rentabilité**

Le Cloud Computing est probablement la méthode la plus rentable à utiliser, à maintenir et à mettre à niveau. Les logiciels de bureau traditionnels coûtent cher aux entreprises en termes de financement. En plus, des frais supplémentaires de licence des applications pour plusieurs utilisateurs peuvent devenir très coûteux pour un établissement quelconque.

- **le stockage illimité**

Stocker des informations dans le Cloud Computing donne une capacité de stockage presque illimitée. Par conséquent, on a plus besoin de se soucier du manque d'espace de stockage ou d'augmenter la disponibilité de l'espace de stockage actuel. Il existe des centaines de systèmes de stockage différents. Certains ont un objectif très spécifique, tel que le stockage de messages électroniques ou d'images numériques. D'autres sont disponibles pour stocker toutes les formes de données numériques.

- **Intégration automatique du logiciel**

Dans le Cloud Computing, l'intégration logicielle est généralement automatique. Cela signifie qu'on a pas besoin de faire des efforts supplémentaires pour personnaliser et intégrer les applications selon des préférences. Non seulement cela, le Cloud Computing permet aussi de personnaliser des options avec une grande facilité. Aussi,

on peut sélectionner uniquement les services et les applications logicielles qui conviennent le mieux à une organisation.

- **L'accès facile à l'information**

Une fois inscrit ou enregistré dans le Cloud Computing, on peut accéder à l'information de n'importe où, moyennant une connexion Internet. Cette fonction pratique permet d'aller au-delà des problèmes de localisation géographique.

### **II.8.2 Les inconvénients**

Les inconvénients du Cloud Computing discutés par la plupart des auteurs sont liés à la confidentialité et à la sécurité des données, parmi les inconvénients nous citons :

- **La sécurité et la confidentialité**

Parce que le Cloud Computing représente un nouveau modèle informatique, il existe une grande incertitude sur la sécurité à tous les niveaux (par exemple réseau, hôte, application et au niveau de données). Cette incertitude conduit les dirigeants des entreprises à déclarer constamment que la sécurité est leur préoccupation majeure. La capacité du Cloud Computing à répondre adéquatement aux règles de confidentialité a été remise en question[2]. Aujourd'hui, les organisations sont confrontées à de nombreuses exigences différentes visant à protéger la vie privée des individus. Il n'est pas encore établi que le modèle de Cloud Computing fournit la protection des informations.

- **La connectivité et libre accès**

Le plein succès du Cloud Computing dépend de la disponibilité de l'accès haute vitesse pour tous. La connectivité et l'accès ouvert à la puissance de calcul et la disponibilité de l'information à travers le Cloud favorisent une nouvelle ère de l'informatique. atique dite ubiquitaire.

- **La fiabilité**

Les applications d'entreprise sont maintenant si critiques qu'elles doivent être fiables et disponibles pour prendre en charge les opérations de manière continue. Dans le cas d'échecs d'accès ou de pannes, les plans d'urgence doivent prendre effet sans perturbation. Des coûts supplémentaires peuvent être associés aux niveaux de fiabilité requis. Cependant, l'entreprise doit tout prévoir pour atténuer les risques et le coût d'une panne [85].

- **L'interopérabilité**

L'interopérabilité et la portabilité de l'information entre les Clouds privés et les Clouds publiques sont des facteurs essentiels pour une large adoption du Cloud Computing par l'entreprise. De nombreuses entreprises ont fait des progrès considérables pour normaliser leurs processus, leurs données et leurs systèmes grâce à la mise en œuvre d'ERP (Enterprise Resource Planning). Ce processus est activé par des infrastructures évolutives avec des instances uniques, ou des connexions hautement intégrées entre instances, pour gérer et assurer les cohérences des bases de données et des transactions et produire des informations consolidées fiables. Même avec ces plateformes, la vitesse à laquelle les entreprises changent peut dépasser la capacité des organisations informatiques à répondre à ces changements[85]. Les applications SaaS fournies via le Cloud Computing offrent une option de déploiement rapide à faible coût. Parfois, il est essentiel d'intégrer des applications traditionnelles qui peuvent résider dans un Cloud séparé ou sur technologie traditionnelle.

- **La valeur économique**

La croissance du Cloud Computing repose sur le retour sur investissement qui en découle. Il semble intuitif que par le partage des ressources pour atténuer les coûts, et payer uniquement pour ce qui est utilisé, permet de réduire les investissements de capitaux pour le déploiement de solutions informatiques. Il faudra équilibrer soigneusement tous les coûts et avantages associés au Cloud, à court et à long termes. Les coûts cachés pourraient inclure un soutien ou une assistance conseil, des prestations de service pour la récupération après sinistre, la modification d'applications et les frais d'assurance après perte de données [86].

- **Changements dans l'organisation informatique**

L'organisation informatique sera affectée par le Cloud Computing , comme elle l'a été dans le cas d'autres changements technologiques, à l'exemple de l'évolution des langages de programmation, du Cobol jusqu'à l'avènement des langues de quatrième génération. Les rôles au sein des services informatiques, tels que l'analyste système et le programmeur, qui ont été fusionnés en analyste / programmeur. Les rôles seront changés encore une fois: la rapidité du changement aura un impact sur l'adoption des technologies du Cloud Computing. La nécessité de maintenir des contrôles pour gérer les risques informatiques dans l'entreprise va augmenter.

## **II.9 Le cloud pour le RCSF**

Bien que le cloud computing est une technologie récente, son potentiel a attiré l'attention de nombreuses recherches et travaux qui décrivent la collaboration entre le RCSF et le Cloud Computing, ce dernier est principalement conçu pour être orienté sur les centres de données, tandis que les RCSF sont conçus pour collecter des données dans le monde réel, les résultats sont généralement stockés en dehors du RCSF. Les données peuvent être conservées pour des raisons diverses, notamment pour les recherches historiques et futures et pour des analyses ultérieures à un moment donné[36].

L'intégration de ces deux technologies (Le cloud Computing et les RCSF) peut fournir une infrastructure robuste et évolutive pour plusieurs applications et permet de gérer efficacement les données collectées par les RCSF. Cette intégration fournit également les outils nécessaires à la collecte, à la diffusion et au stockage des données de capteurs en temps réel issues de capteurs. Les composants du Cloud fournissent une base pour la sécurisation et la fourniture d'un accès utilisateur aux ressources logicielles, middleware et matérielles.

## **II.10 Conclusion**

Comme toutes les technologies, le Cloud Computing peut offrir d'immenses opportunités comme il peut présenter des risques importants. La bonne compréhension de ses capacités et de ses challenges est la clé de la meilleure utilisation. La fourniture de ressources dans les Cloud Computing offre un gain énorme pour l'utilisateur et pour le fournisseur. Beaucoup de travaux ont été conduits sur ce sujet afin d'offrir le maximum de gain. Cependant ce problème est plus complexe dans les environnements d'intercloud. Nous avons exposé dans ce chapitre les différentes approches de fourniture de ressources dans le Cloud Computing. Le chapitre prochain sera consacré à l'intégration des réseaux de capteurs sans fil et le Cloud.

## Sommaire

---

III L'intégration des RCSF et le Cloud Computing.....	44
III.1 Introduction .....	45
III.2 Motivation.....	45
III.3 Définition du nuage de capteurs (Sensor-Cloud) .....	46
III.4 Les avantages du nuage de capteurs.....	47
III.5 Les applications du Cloud de capteurs .....	49
III.6 Les architectures du Cloud de capteurs.....	51
III.7 Conclusion.....	<b>62</b>

---



### III.1 Introduction

Au cours des dernières années, les réseaux de capteurs sans fil (RCSF) ont suscité une attention croissante. Les RCSF associent de manière transparente l'environnement physique au monde numérique. Les nœuds de capteurs surveillent de manière coopérative les conditions telles que la température, l'humidité, le mouvement des véhicules, la pression et la composition du sol. Ces dispositifs ont créé de nouveaux paradigmes informatiques grâce à la manière dont ils capturent ou génèrent des données, ceci a mené au développement de réseaux de capteurs comme un moyen d'intégrer les différents dispositifs de capteurs dans un réseau coopératif. De même, les différences entre les technologies informatiques et du stockage ont entraîné le développement du Cloud Computing comme un moyen de simplifier la gestion de la complexité et des ressources informatiques.

Pour la surveillance d'un environnement étendu, la communication entre les nœuds est limitée en raison d'obstructions dans l'environnement, qui affectent à leur tour la topologie (ou la connectivité) globale de réseau. Toutes ces limitations sur les réseaux de capteurs entravent probablement les performances et la qualité du service. Au milieu de ces problèmes, l'émergence de Cloud Computing est considérée comme un remède. Dans ce chapitre, on va entamer l'intégration des réseaux de capteurs et le Cloud Computing.

### III.2 Motivation

Un réseau de capteurs typique peut consister en un certain nombre de nœuds de capteurs agissant ensemble pour surveiller une région et extraire des données sur l'environnement. Un RCSF typique contient des capteurs autorégulés qui peuvent surveiller de manière coopérative les conditions environnementales, telles que le son, la température, la pression, le mouvement, les vibrations, la pollution, le feu et d'autres événements dépendants de l'application. Chaque nœud dans un réseau de capteurs est chargé avec un émetteur-récepteur radio ou un autre dispositif de communication sans fil, un petit microcontrôleur, et une source d'énergie le plus souvent des cellules / batterie. Les RCSF ont certaines limitations en termes de mémoire, d'énergie, de calcul, de communication et d'évolutivité, de gestion efficace du grand nombre de données.

Le Cloud Computing permet aux systèmes et aux utilisateurs d'utiliser la plateforme en tant que service (PaaS), par exemple, les systèmes d'exploitation (OS), Infrastructure en tant que service (IaaS), par exemple, des entrepôts et des serveurs et des logiciels en tant que service (SaaS), par exemple, des programmes au niveau de l'application, et ainsi de suite à un coût très faible qui sont fournis par plusieurs fournisseurs de Cloud (par exemple, Amazon, Google et Microsoft) sur la base des services (pay per use) [6]. Plateforme de Cloud Computing dynamiquement disponible, configure et met à jour les serveurs en fonction des besoins des utilisateurs finaux. Les limites des RCSF sont les points forts du Cloud Computing, c'est la raison pour laquelle les intégrations de Cloud Computing et de RCSF aboutiront à de meilleurs avantages et à une plus grande efficacité.

### III.3 Définition du nuage de capteurs (Sensor-Cloud)

Un nuage des capteurs ou 'Sensor-Cloud' est composé des capteurs virtuels construits sur des capteurs physiques sans fil. Selon IntelliSys<sup>1</sup>[37], le 'Sensor-Cloud' ou le nuage des capteurs peut être défini comme suit: *Une infrastructure qui permet un calcul véritablement omniprésent en utilisant des capteurs comme interface entre le monde physique et virtuel, les clusters de calcul de données en tant que réseau fédérateur et Internet en tant que moyen de communication* [38][39].

Un Sensor-Cloud collecte et traite les informations provenant de plusieurs réseaux de capteurs, permet le partage d'informations à grande échelle et collabore avec les applications sur le Cloud Computing parmi les utilisateurs. Il intègre plusieurs réseaux avec un certain nombre d'applications et de plateformes du Cloud Computing en permettant aux applications d'être interdisciplinaires pouvant s'étendre sur plusieurs organisations[39]. Le Sensor-Cloud permet aux utilisateurs de rassembler, accéder, traiter, visualiser, analyser, stocker, partager et rechercher une grande quantité de données des capteurs provenant de plusieurs types d'applications et en utilisant le calcul, ressources informatiques et du stockage du Cloud Computing.

Dans un réseau de capteurs, les capteurs sont utilisés par leurs applications spécifiques dans un but spécial et ces applications gèrent à la fois les données du capteur et le capteur lui-même de sorte que d'autres applications ne peuvent pas

---

<sup>1</sup> Intelisys : est un distributeur de services technologique américain qui vend des services commerciaux de voix, de données et du Cloud par l'intermédiaire d'un réseau de vente indépendant.

l'utiliser. Cela entraîne un gaspillage de ressources de capteurs précieuses qui peuvent être efficacement utilisées lors de l'intégration avec l'infrastructure d'autres applications. Pour réaliser ce scénario, l'infrastructure Sensor-Cloud permet d'utiliser les capteurs sur une infrastructure informatique en virtualisant le capteur physique sur une plateforme du Cloud Computing. Ces capteurs virtualisés sur une plateforme de Cloud Computing sont de nature dynamique et facilitent ainsi le provisionnement automatique de ses services en fonction des besoins des utilisateurs[40]. De plus, les utilisateurs n'ont pas besoin de s'inquiéter de l'emplacement physique de plusieurs capteurs et de l'écart entre les capteurs physiques, ils peuvent superviser ces capteurs virtuels en utilisant certaines fonctions standard[41].

### III.4 Les avantages du nuage de capteurs

Le Cloud Computing [2] est une solution pour l'infrastructure Sensor-Cloud pour plusieurs raisons telles que l'agilité, la fiabilité, la portabilité, le temps réel, la flexibilité,...etc. La surveillance structurelle de la santé et de l'environnement contient des données et des applications sensibles qui ne peuvent pas être traitées par des outils de données normales disponibles en termes d'évolutivité des données, de performance, de programmabilité ou d'accessibilité. Une meilleure infrastructure est donc nécessaire et peut contenir des outils pour faire face à ces applications sensibles en temps réel. Dans la suite, nous décrivons les nombreux avantages et bénéfices de l'infrastructure Sensor-Cloud qui pourrait être la cause de son succès et ceux sont les suivantes :

- **La gestion efficace :** Il permet une meilleure capacité de gestion des capteurs. Les utilisateurs peuvent utiliser et contrôler leur vue des RCSF avec des fonctions standard pour une variété de paramètres tels que la région d'intérêt, la fréquence d'échantillonnage, la latence et la sécurité.
- **Le partage de ressources :** Les données capturées par les RCSF peuvent être partagées entre plusieurs utilisateurs, ce qui réduit le coût global de la collecte des données du point de vue du système et du point de vue de l'utilisateur. La réutilisabilité des données dans les RCSF est transparente pour les utilisateurs du nuage de capteurs, ce qui augmente l'efficacité en réduisant la capture de données redondantes.

- **La transparence** : Le système est transparent pour le type de capteurs utilisés. L'utilisateur n'a pas à s'inquiéter des détails de bas niveau des nœuds utilisés et comment les configurer ; ces détails sont automatiquement gérés par le Sensor-Cloud.
- **L'analyse** : L'intégration d'énormes données de capteurs accumulées à partir de plusieurs réseaux de capteurs et du modèle de Cloud Computing le rend attrayant pour les différents types d'analyses requises par les utilisateurs grâce au provisionnement de la puissance de traitement évolutive[42].
- **Le passage à l'échelle** : Le Sensor-Cloud permet aux réseaux de capteurs d'évoluer sur une très grande taille en raison de l'architecture du routage de Sensor-Cloud[38]. Cela signifie que plus le besoin des ressources augmente, plus les organisations ne peuvent augmenter ou ajouter les services extra des fournisseurs de Cloud Computing sans avoir à investir massivement pour ces ressources matérielles supplémentaires.
- **La collaboration** : le Sensor-Cloud permet de partager les énormes données de capteurs entre différents groupes de consommateurs grâce à la collaboration de plusieurs réseaux de capteurs physiques. Il facilite la collaboration entre les utilisateurs et applications pour un grand partage de données sur le Cloud.
- **La visualisation** : La plateforme Sensor-Cloud fournit une visualisation pour représenter les diagrammes avec les données des capteurs stockées provenant de plusieurs équipements. Grâce aux outils de visualisation, les utilisateurs peuvent prédire les futures tendances possibles qui doivent être encourues [43].
- **L'approvisionnement dynamique de services** : Les utilisateurs de Sensor-Cloud peuvent accéder à leurs informations pertinentes où qu'ils le souhaitent et quand ils en ont besoin plutôt que de rester sur leur bureau [44]. Les applications orientées service sont principalement exécutées dans un environnement distribué. Le déplacement d'une application vers un environnement particulier nécessite une distribution des services avant le déploiement. Par conséquent, les fournisseurs de services doivent optimiser la distribution des services afin de gagner du temps et de l'argent. L'approvisionnement dynamique de services, signifie l'allocation dynamique des ressources et le déploiement dynamique des services dans un

environnement distribué ce qui permet d'une grande flexibilité et une utilisation optimisée de l'infrastructure.

- **L'agilité des services** : Le Sensor-Cloud fournit des services agiles et les utilisateurs peuvent provisionner les ressources d'infrastructure technologique coûteuses à moindre coût. L'agilité est définie comme «la capacité de se déplacer rapidement et facilement ». Cela signifie que lorsque les conditions du marché changent, les activités des utilisateurs doivent être suffisamment souples pour procéder aux ajustements nécessaires pour s'adapter rapidement et facilement à ces changements, l'évolutivité du sensor-cloud permet aux entreprises de contourner tout cela. Désormais, les utilisateurs peuvent travailler avec leur fournisseur de Cloud pour ajouter rapidement et facilement de nouveaux utilisateurs, applications et nœuds de calcul et de stockage en fonction des besoins. Cela simplifie le processus d'approvisionnement. L'intégration de réseaux de capteurs sans fil avec le Cloud Computing permet le traitement à grande vitesse des données en utilisant une immense capacité de traitement du Cloud Computing.
- **Optimisation des ressources** : L'infrastructure Sensor-Cloud permet l'optimisation des ressources en permettant le partage des ressources entre plusieurs applications [43]. L'intégration des capteurs et le Cloud Computing permet de réduire progressivement le coût des ressources et permet des gains des services plus importants. Avec le Sensor-Cloud les petites et moyennes organisations peuvent bénéficier d'une énorme infrastructure de ressources sans devoir l'impliquer et l'administrer directement.

### III.5 Les applications du Cloud de capteurs

L'intégration des réseaux de capteurs sans fil avec le Cloud Computing permet de partager et d'analyser en temps réel les données des capteurs à la volée. Les nœuds de capteurs inhérents à la propriété de détection en tant que service pour déclencher les événements sur Internet ajoutent des avantages au Cloud de capteurs. Certaines des applications en temps réel sont la surveillance des transports, la prévision météorologique, l'utilisation militaire, les soins de santé et ainsi de suite. Cependant, les implémentations naturelles utilisent le Cloud Computing pour consolider, contrôler et collaborer avec les capteurs. Certaines applications de réseau de capteurs utilisant le Cloud Computing sont expliquées ci-dessous :

- **Surveillance du transport**

Dans les systèmes de surveillance du transport, des capteurs sont utilisés pour détecter les véhicules et contrôler les feux de circulation. Les caméras vidéo sont également utilisées pour surveiller les segments de route à fort trafic et les vidéos sont envoyées à des opérateurs humains dans des lieux centraux. Des capteurs dotés d'une capacité de mise en réseau intégrée peuvent être déployés à chaque intersection routière pour détecter et compter le trafic du véhicule et estimer sa vitesse. Les capteurs communiquent avec les nœuds voisins pour éventuellement développer une « image de trafic global » qui peut être interrogée par les utilisateurs pour générer des signaux de contrôle. Les données provenant des capteurs sont acquises et transmises pour la fusion et le traitement centraux. Ces données peuvent être utilisées dans diverses applications telles que la classification des véhicules, le guidage du stationnement et le système d'information.

- **Prévisions météorologiques**

La prévision météorologique est l'application pour prédire l'état de l'atmosphère pour une heure future et un emplacement donné. Les systèmes de surveillance et de prévisions météorologiques comprennent généralement : la collecte des données, l'assimilation des données, la prédiction numérique et la présentation des prévisions. Chaque station météo est équipée de capteurs pour détecter les données. Les données recueillies à partir de ces capteurs sont énormes et difficiles à maintenir en utilisant les approches de base de données traditionnelles, et l'assimilation est faite. Les équations vérifiées gouvernent comment l'état de l'atmosphère change.

- **Utilisation militaire**

Les réseaux de capteurs sont utilisés dans l'armée pour surveiller les forces amies, l'équipement et les munitions, la reconnaissance des forces adverses. Les données collectées à partir de ces applications sont de la plus haute importance et nécessitent une sécurité de haut niveau qui peut ne pas être fournie en utilisant une connectivité Internet normale. Le Cloud Computing est l'une des solutions à ce problème en fournissant une infrastructure sécurisée exclusivement pour les applications militaires.

- **Soins de santé**

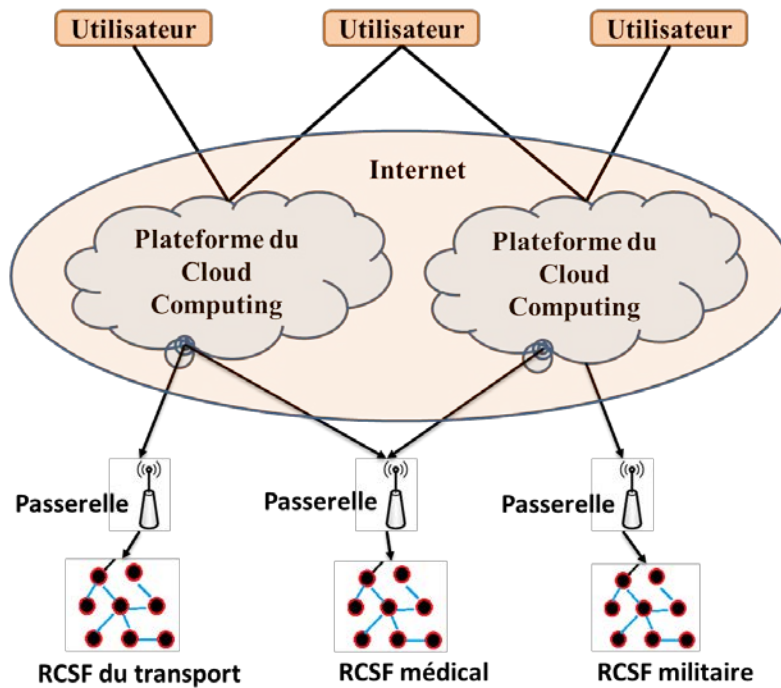
Les réseaux de capteurs sont également largement utilisés dans le domaine des soins de santé[45]. Les données collectées auprès des patients sont très sensibles et

doivent être correctement conservées, car les données collectées sont requises par les médecins pour leur diagnostic futur. Dans l'approche traditionnelle, la base de données d'antécédents du patient est conservée dans la maison de soins locale. Les médecins ne peuvent poser un diagnostic que lorsqu'ils visiteront la maison de retraite particulière. Ce problème peut être surmonté en formant un Cloud où les données critiques du patient peuvent être maintenues.

### **III.6 Les architectures du Cloud de capteurs**

L'infrastructure de services de Cloud Computing fournit les services de réseau partagé à travers lesquels les utilisateurs bénéficient des services et ils ne sont pas concernés par les détails d'implémentation des services qui leur sont fournis. Lorsqu'un utilisateur demande les instances de service (par exemple les capteurs virtuels) générées par les services de Cloud Computing sont automatiquement mises à leur disposition [12, 19]. La **Figure 9** présente une architecture générale d'un Sensor-Cloud.

Certaines études antérieures sur les capteurs sans fil ont porté sur le routage, la synchronisation d'horloge, le traitement des données, la gestion de l'alimentation, le système d'exploitation, la localisation et la programmation. Cependant, peu d'études se concentrent sur la gestion physique des capteurs, car ces capteurs physiques sont étroitement liés à leur application spécifique ainsi qu'à ses utilisateurs tangibles directement. Cependant, les utilisateurs, mis à part leurs propres services, ne peuvent pas utiliser ces capteurs physiques directement en cas de besoin. Par conséquent, ces capteurs physiques devraient être supervisés par des systèmes spéciaux de gestion des capteurs.



**Figure 9:** Modèle général d'un Cloud de capteurs

L'infrastructure Sensor-Cloud supporte la gestion du système de capteurs, ce qui assure que la gestion des données des ressources du capteur sera relativement améliorée. Il n'existe aucune application qui peut utiliser tous les types de capteurs physiques en même temps, mais chaque application nécessite des capteurs physiques pertinents pour son accomplissement. Pour réaliser ce concept, un mécanisme de publication/abonnement [46] est utilisé pour choisir le capteur physique approprié. Dans plusieurs réseaux de capteurs, chaque réseau de capteurs publie ses données de capteur et ses métadonnées. Les métadonnées comprennent les types, les emplacements, etc. pour les capteurs physiques.

L'application souscrit à un ou peut-être à plusieurs réseaux de capteurs pour récupérer des données en temps réel des capteurs physiques en permettant à chaque application d'opter pour le type de capteurs physiques appropriés. L'infrastructure Sensor-Cloud procède des capteurs virtuels à partir de multiples capteurs physiques, qui peuvent ensuite être utilisés par les utilisateurs. Cependant, avant d'utiliser l'installation de capteurs virtuels, les utilisateurs doivent d'abord vérifier la disponibilité des capteurs physiques et peuvent inspecter les défauts des capteurs physiques pour maintenir la qualité de données de ces capteurs physiques. De plus, sur



chaque nœud de capteur, le programme d'application détecte l'application et envoie les données du capteur à la passerelle dans le Cloud directement à travers la station de base ou de manière multi-chaînes via d'autres nœuds[47],[48].

L'infrastructure Sensor-Cloud fournit automatiquement des instances de service (capteurs virtuels) aux utilisateurs finaux au fur et à mesure des demandes, de sorte que ces capteurs virtuels font partie de leurs ressources informatiques (stockage sur disque, processeur, mémoire, etc.) [34]. Ces instances de service et leurs données associées peuvent être utilisées par les utilisateurs finaux via une interface utilisateur et les robots d'exploration. Cependant, pour la génération d'instance de service, les ressources informatiques (comme le processeur, les périphériques du stockage, etc.), des périphériques compatibles avec les capteurs et des modèles de service (qui sont utilisés pour créer des capteurs virtuels) doivent d'abord être préparés dans l'infrastructure Sensor-Cloud. Les utilisateurs effectuent la demande d'instances de service en fonction de leurs besoins en sélectionnant un modèle de service approprié de Sensor-Cloud, qui fournira ensuite les instances de service nécessaires librement et automatiquement en raison de l'intégration des services de Cloud Computing.

Les auteurs de [49] ont proposé une technique (FIND) pour localiser des capteurs physiques ayant un défaut en supposant un décalage entre le rang de distance et le rang des données du capteur. Une fois, les modèles préparés, les capteurs virtuels peuvent partager les capteurs physiques connexes et contigus pour recevoir les données de capteur de qualité. Les utilisateurs demandent ensuite ces capteurs virtuels en choisissant les modèles de service appropriés, utilisent leurs instances de service (capteurs virtuels) après avoir été approvisionnés et les déchargent lorsqu'ils deviennent inutiles [41].

Sensor-Cloud est un nouveau paradigme pour le Cloud Computing qui utilise les capteurs physiques pour accumuler ses données et transmettre toutes les données des capteurs dans une infrastructure de Cloud Computing. Sensor-Cloud gère efficacement les données des capteurs, qui sont utilisées pour de nombreuses applications de surveillance. À notre connaissance, la question de l'intégration de réseau de capteurs au Cloud Computing n'est pas encore abordée dans la littérature. L'architecture Sensor-Grid [50] existe déjà, mais n'est pas adaptée aux scénarios d'application actuels. Dans [4]

les applications sont aussi exécutées sur des grilles de calcul actuelles, mais elles sont différentes aux applications Cloud Computing. La grille est généralement orientée sur les applications HPC (High Performance Systems), tandis que le Cloud Computing se concentre sur les applications générales. Les utilisateurs passent maintenant des charges de travail HPC à des charges de travail plus génériques pour lesquelles le Cloud Computing est la solution la plus appropriée.

➤ **L'architecture TinySIP [51]:**

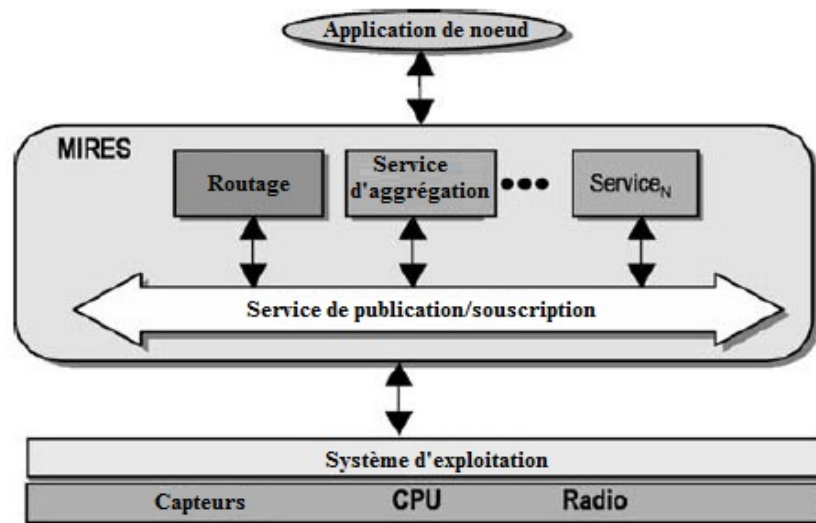
Cette architecture prend en charge la sémantique de session, les fonctions de la publication / abonnement et la messagerie instantanée. Elle offre un support pour plusieurs passerelles. La plupart des communications sont effectuées en adressant des appareils individuels. Comme les adresses des appareils sont liées à la passerelle utilisée, il est difficile de changer la passerelle à la volée.

➤ **L'architecture DV / DRP [52] :**

C'est une autre architecture de type publication / abonnement pour les RCSF, dans l'architecture DV/DRP (*distance vector/dynamic receiver partitioning*) les abonnements sont effectués en fonction du contenu des messages souhaités. Les abonnements sont inondés dans le réseau. Les nœuds intermédiaires regroupent les abonnements. Ils ne transmettent les publications que s'il y a un intérêt pour cette publication. En raison de la complexité de la correspondance des abonnements aux paquets de données arbitraires, il serait difficile d'implémenter ce protocole sur les périphériques que nous ciblons.

➤ **L'architecture Mires**

**Mires** [53] est une architecture de publication/abonnement pour les RCSF comme présenté dans **Figure 10**. Fondamentalement, les capteurs ne publient des lectures que si l'utilisateur a souscrit à la lecture spécifique du capteur. Les messages peuvent être agrégés dans les têtes de cluster. Les abonnements sont émis à partir du nœud récepteur (généralement directement connecté à un PC), qui reçoit ensuite toutes les publications.



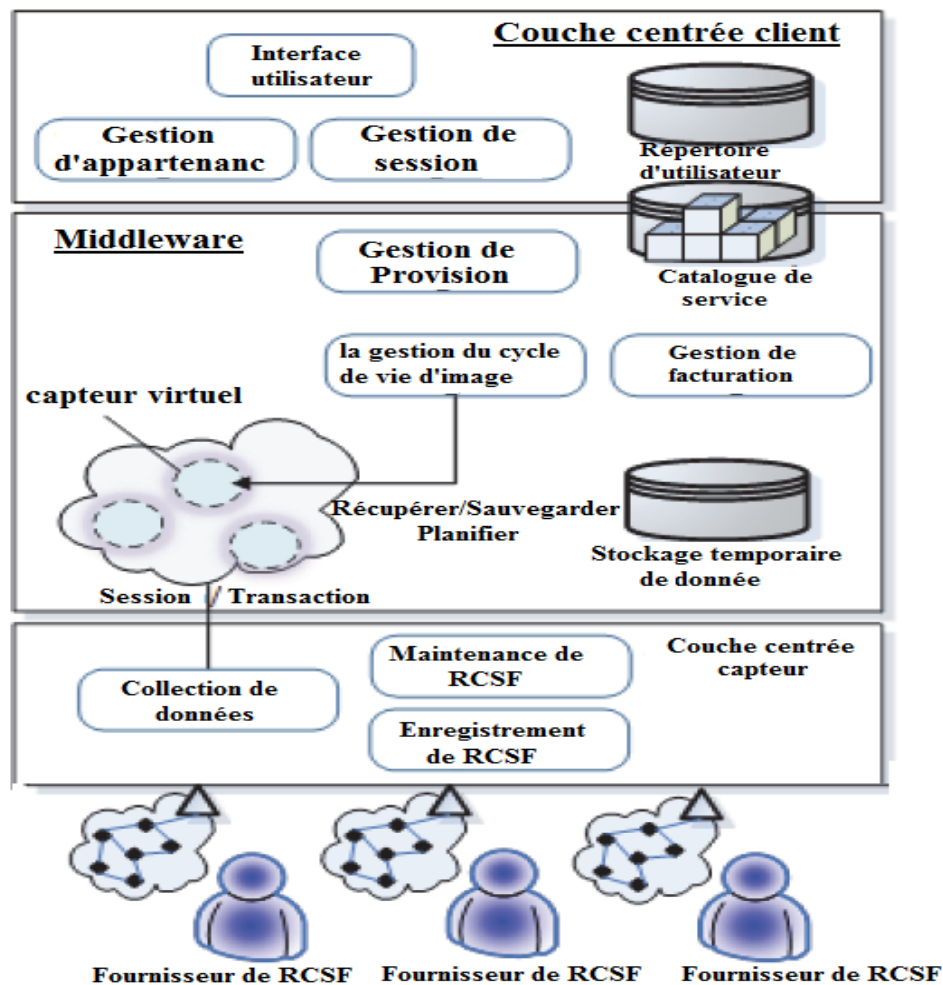
**Figure 10:** L'architecture Mires [53]

C'est une architecture de type publication/souscription conçue pour les réseaux de capteurs. Fondamentalement, le capteur ne publie seulement que s'il y'avait des utilisateurs qui sont souscrits à des lectures spécifiques de ce capteur. Les messages peuvent être regroupés dans les stations de base. Les souscriptions sont émises à partir du nœudrécepteur (généralement connecté directement à un PC) qui reçoit toutes les publications.

➤ **L'architecture Missouri S&T**

La **Figure 11** représente l'infrastructure de Sensor-Cloud de l'université du Missouri de la science et de la technologie (S&T) est divisée en trois couches importantes : couche centrée client, couche centrée middleware et la couche centrée capteur.

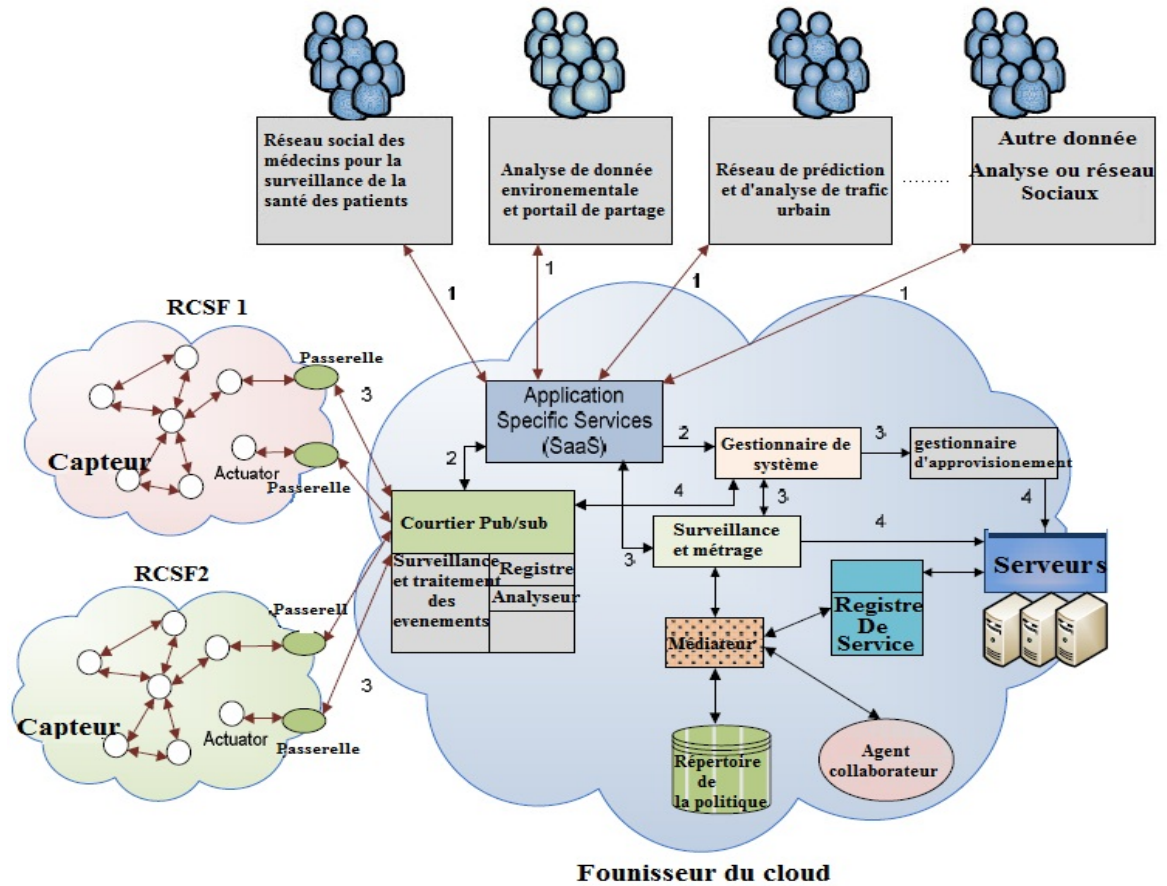
- **La couche centrée client** : qui connecte les utilisateurs au Sensor-Cloud.
- **La couche centrée middleware** : qui effectue la négociation de service, l'approvisionnement, la maintenance des capteurs virtuels et la communication des données entre la couche centrée capteur et la couche centrée client.
- **La couche centrée capteur** : ce module se charge des capteurs sans fil physiques et leur maintenance ainsi que l'acheminement des données et des commandes. [9]



**Figure 11:** L'architecture Missouri S&T

Le travail présenté dans [4], propose une plateforme qui combine WBAN et le Cloud Computing pour le partage de données. L'architecture repose sur l'intégration du protocole TCP/IP et du protocole Zigbee dans les dispositifs de coordination de sorte que les dispositifs WBAN peuvent utiliser un type de réseau différent, par exemple le Wifi et LTE pour soutenir la mobilité. De plus, l'architecture propose d'intégrer les réseaux CCN (*Content-Centric Networks*). L'architecture se compose de quatre couches, la couche Cloud est la plus importante et la partie essentielle de l'architecture. Les données collectées par la couche de perception sont transmises à la couche de Cloud Computing via différents réseaux locaux dans la couche de réseau. La couche de Cloud Computing permet ensuite de stocker et de gérer efficacement les données en vue d'une application diversifiée dans le domaine de la santé.

Le même auteur a publié un travail antérieur qui propose une architecture [11] de type publication/souscription basé sur le contenu présenté dans la **Figure 12**.



**Figure 12:** L'architecture pour intégrer RCSF dans le Cloud Computing

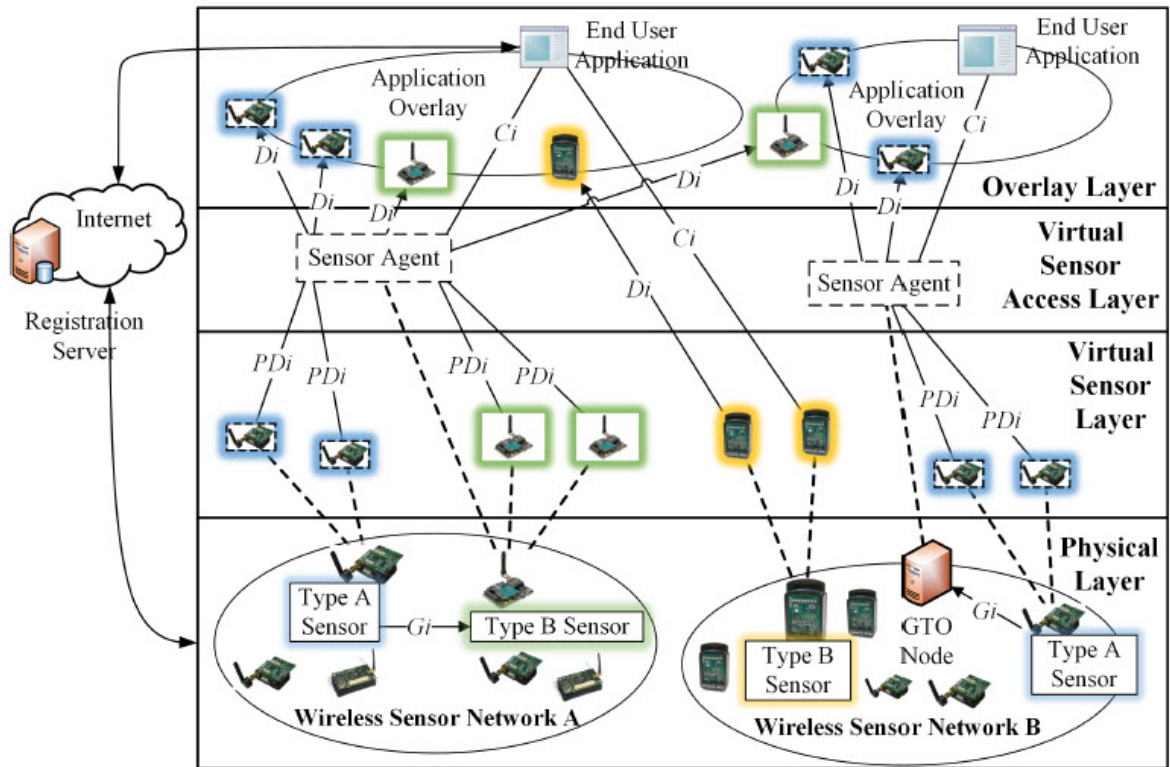
Dans ce scénario, les données des capteurs arrivent à travers les passerelles vers le courtier (Broker) qui est nécessaire dans ce système pour fournir des informations aux utilisateurs de l'application comme le réseau entier est dynamique. La recherche présentée dans [54] propose une architecture permettant d'offrir une interface RCSF IaaS, capable selon les auteurs d'interagir avec PaaS afin de permettre aux fournisseurs de services de fournir rapidement des applications et des services de RCSF.

Il s'agit d'une architecture multicouche qui exploite les capacités des nœuds de capteurs individuels pour exécuter des tâches d'application simultanées au niveau du nœud et de manière dynamique et rassemble ces nœuds au niveau de réseau pour le partage de données. Dans la couche physique, on trouve des capteurs avec des ressources limitées et des capteurs plus puissants. Étant donné que les capteurs sont limités en terme de ressources ne peuvent pas prendre en charge la virtualisation au

niveau réseau sans fil, ils s'appuient sur des nœuds les plus puissants. Ensuite, dans la couche des capteurs virtuels qui sont des abstractions des tâches de l'application exécutées par les capteurs physiques. Pour chaque application, un seul VS exécute sa tâche.

Les auteurs de [49] ont proposé une technique (FIND) pour localiser les capteurs physiques ayant des défauts de données en supposant une discordance entre le rang de distance et le rang de données du capteur. Cependant, l'étude menée par FIND vise à évaluer les défauts de capteurs, et il existe une relation étroite entre le virtuel et des capteurs physiques et donc un capteur virtuel fournira résultats incorrects si leurs capteurs physiques sont défectueux. Il conclut que les capteurs virtuels et physiques coopèrent tout en fournissant le rapport de données du capteur à ses applications, il permet de maintenir le meilleur rapport à la fois virtuel et physique des capteurs.

Les auteurs dans [54] proposent une architecture de réseaux de capteurs virtuels pour le modèle de service IaaS. La Figure 13 illustre une description détaillée de l'architecture. Il s'agit d'une architecture multicouche qui exploite les capacités des nœuds de capteurs individuels pour exécuter des tâches d'application concurrentes au niveau des nœuds et les assemble dynamiquement au niveau du réseau pour le partage de données. La couche physique contient des capteurs limités en termes de ressources et des capteurs ayant des ressources suffisantes. Étant donné que les capteurs limités en termes de ressources ne peuvent pas prendre en charge la virtualisation au niveau du réseau, ils s'appuient sur les nœuds des réseaux superposés (Overlay networks GTO) par exemple la station de base.



**Figure 13:** Architecture de réseaux de capteurs virtuels

Ensuite, dans la couche de capteurs virtuels se trouvent les capteurs virtuels (VS) qui sont des abstractions des tâches de l'application exécutées par les capteurs physiques. Pour chaque application, un seul VS exécute sa tâche. La troisième couche est la couche d'accès aux capteurs virtuels. Elle consiste en des agents détecteurs (SA) assurant l'indépendance de la plate-forme en utilisant des interfaces normalisées et des interfaces propriétaires. La dernière couche est la couche des réseaux superposés (Overlay layer), qui consiste en plusieurs applications utilisant les RCSF déployés. Il existe de différentes interfaces pour les messages de contrôle et de données [54].

La virtualisation est une sorte de technologie logicielle qui peut être appliquée dans nombreux domaines. En fait, le concept de « Cloud » est apparu très tôt, mais en raison du manque de bande passante, d'un espace de stockage suffisant et de la technologie de virtualisation prématurée, il était instable et elle n'était pas flexible. Cependant, avec le développement de la technologie de virtualisation, ce concept a été accéléré. Dans le passé, la puissance de calcul était basée localement sur des ordinateurs personnels, mais grâce à la technologie de virtualisation, il est désormais possible d'utiliser la puissance du Cloud Computing centralisée.

Le service du Cloud Computing fournit les services de réseau partagé par lequel les utilisateurs bénéficient des services, et ils ne sont pas concernés par les détails de mise en œuvre des services qui leur sont fournis. Certaines études antérieures sur les capteurs physiques ont porté sur le routage, la synchronisation d'horloge, le traitement de données, la gestion de l'alimentation, le système d'exploitation, la localisation et la programmation. Cependant, peu d'études se concentrent sur la gestion physique des capteurs, car ces capteurs physiques sont étroitement liés à leur application spécifique ainsi qu'à ses utilisateurs tangibles directement. Cependant, les utilisateurs, autres que leurs services de capteurs pertinents, ne peuvent pas utiliser ces capteurs physiques directement en cas de besoin. Par conséquent, ces capteurs physiques devraient être supervisés par une gestion spéciale des capteurs.

Selon notre connaissance, il n'existe aucune application qui peut utiliser toutes les sortes de capteurs physiques en tout temps ; à la place, chaque application requiert des capteurs physiques pertinents pour son exécution de tâches. Dans plusieurs réseaux de capteurs, chaque réseau de capteurs publie ses données de capteur et ses métadonnées, celles-ci comprennent des types, des emplacements, et ainsi de suite pour le capteur physique, l'application souscrit à un ou peut-être à plusieurs réseaux de capteurs pour récupérer des données en temps réel des capteurs physiques en permettant à chaque application d'opter pour le type des capteurs physiques.

L'infrastructure Sensor-Cloud procure des capteurs virtuels à partir de plusieurs capteurs physiques qui pourraient ensuite être utilisés par les utilisateurs. Cependant, avant de profiter de l'installation d'un capteur virtuel, les utilisateurs devraient d'abord vérifier la disponibilité des capteurs physiques et peuvent également inspecter les défauts des capteurs physiques pour maintenir la qualité des données émergentes de ces capteurs physiques. Aussi sur chaque nœud de capteur, un programme d'application détecte l'application et envoie les données du capteur à la passerelle dans le Cloud directement à travers la station de base ou de manière multi-sauts à travers d'autres nœuds.

Le Sensor-Cloud fournit des instances de service (capteurs virtuels) automatiquement aux utilisateurs finaux au fur et à mesure demandé, de telle sorte



que ces capteurs virtuels font partie de leurs ressources informatiques (comme le stockage sur disque, le processeur, la mémoire, etc.) .Ces instances de service et leurs procédures appropriées associées les données de capteur peuvent être utilisées par les utilisateurs finaux via une interface utilisateur à travers les robots d'exploration Web. Pour la génération d'instance de service, les ressources informatiques (comme le CPU, les périphériques de stockage, etc.) fourniront des appareils compatibles avec les capteurs et les modèles de services (qui est utilisé pour créer des capteurs virtuels) préparés d'abord dans l'infrastructure Sensor-Cloud. Les utilisateurs font la demande d'instances de service en fonction de leurs besoins pour sélectionner un modèle de service approprié de Sensor-Cloud, qui fournira ensuite les instances de service nécessaires librement et automatiquement à cause de l'intégration des services de Cloud Computing.

Une fois que les instances de service deviennent inutiles, elles peuvent alors être supprimées rapidement par les utilisateurs pour éviter les frais d'utilisation pour ces ressources. Le fournisseur de service de capteur gèrera des modèles de service (ST) et il peut ajouter ou supprimer le service de nouveaux modèles lorsque le modèle requis n'est plus nécessaire par les applications et les services. Cependant, le modèle du service de Cloud Computing réduit les coûts et les délais de livraison et également améliore l'efficacité et la flexibilité.

Le Cloud Computing permet aux capteurs physiques d'être virtualités, les utilisateurs de l'infrastructure Sensor-Cloud ne pas avoir à se soucier de l'état de leur capteur physique connectés (c.-à-d. s'il y a un défaut, le capteur disponible ou non). Cependant, ils ne devraient concerner qu'à l'état de leurs capteurs virtuels fournis uniquement lorsque les utilisateurs ne sont pas concernés par des résultats précis. Pour obtenir des résultats précis, les utilisateurs doivent être préoccupés par l'état des capteurs physiques aussi. Dans une infrastructure Sensor-Cloud, les propriétaires de capteurs sont libres d'enregistrer ou désinscrire leurs capteurs physiques et peuvent se joindre à cette infrastructure.

Les ressources informatiques (capteurs physiques, base de données, serveurs, processeurs, etc.) et les capteurs sont alors préparés pour devenir opérationnel, après cela, les modèles sont créés pour générer les instances de service (capteurs virtuels) et

ses groupes, une fois les modèles sont préparés, les capteurs virtuels sont en mesure de partager les connexes et les contigus avec les capteurs physiques pour recevoir des données de capteurs de qualité. Les utilisateurs ensuite demandent ces capteurs virtuels en choisissant le bon modèles de service , puis ils utilisent leurs instances de service (capteurs virtuels)après avoir été provisionné, et les éliminer quand est devenu inutile [41].

### **III.7 Conclusion**

Dans ce chapitre, Nous avons discuté l'utilisation de l'architecture Sensor-Cloud dans le cadre de plusieurs applications. L'architecture Sensor-Cloud permet aux données du capteur d'être classées, stockées et traitées de manière à devenir rentables, rapidement disponibles et facilement accessibles. Auparavant, la plupart des systèmes RCSF avaient plusieurs schémas de contrôle. La surveillance était par nature fermée avec une interopérabilité nulle, orientée vers des applications spécifiques et non extensibles. Ainsi, l'intégration des capteurs existants et le Cloud Computing permettra de construire un réseau de capteurs ouvert, extensible, évolutif, interopérable et facile à utiliser pour de nombreuses applications. Dans ce chapitre, nous avons discuté des possibilités offertes par la mise en œuvre de la technologie pour gérer des situations plus complexes d'un monde réel grâce à la capacité d'innovation de service de l'infrastructure Sensor-Cloud.

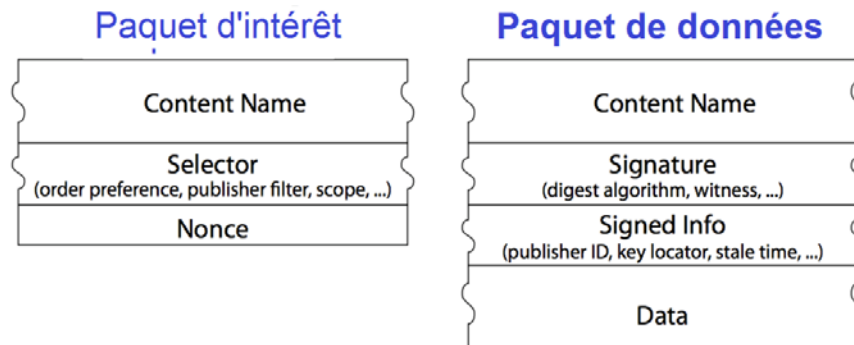
## Sommaire

<b>IV L'architecture Named Data Networking.....</b>	<b>63</b>
IV.1 Introduction .....	64
IV.2 Les composants d'un routeur NDN.....	66
IV.3 Le routage dans NDN .....	69
IV.3.1 Routage intra-domaine .....	72
IV.3.2 Routage inter-domaine .....	74
IV.4 Le forwarding.....	75
IV.5 Le système du cache .....	77
IV.6 Le système de nommage .....	78
IV.7 NDN pour les RCSF .....	80
IV.8 Conclusion.....	81

## IV.1 Introduction

Le NDN [55] a été proposé via le projet NDN[56]. C'est un nouveau paradigme qui tire son principe de réseau centré sur le contenu (*Content-Centric Networking* CCN[57]) introduit pour la première fois par Van Jacobson en 2006[57], qui présente une approche basée sur le contenu des données et considérée comme le nouvel avenir qui remplace l'internet actuel qui repose sur les adresses source et les adresses de destination, tandis que le NDN utilise des noms au lieu des adresses IP. Chaque nœud est identifié par un nom lisible hiérarchiquement. L'évolutivité et l'efficacité de la distribution du contenu constituent l'un des principaux avantages du NDN. En effet, le NDN prend en charge la mise en cache des données dans les routeurs et prend également en charge la multidiffusion dans le réseau.

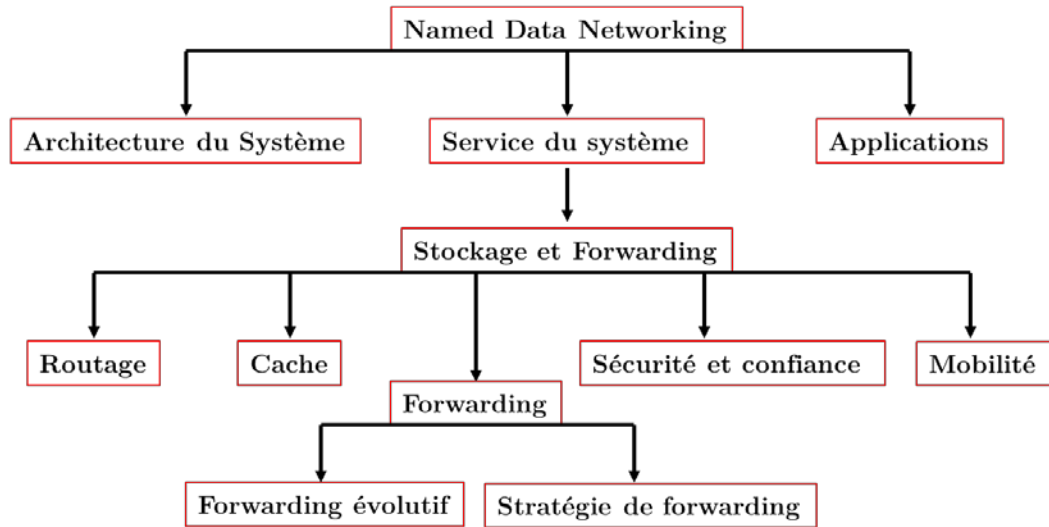
Les questions principales des réseaux centrés sur le contenu sont : comment nommer le contenu, comment localiser le contenu, comment diffuser le contenu, comment mettre en cache le contenu dans le réseau. Le principe de la communication dans le NDN est déterminé par le destinataire. Tous les points d'extrémité échangent deux types de paquets (paquet d'intérêt et le paquet de données) présenté dans **Figure 14**. Le paquet d'intérêt est envoyé par un utilisateur au producteur qui répond par un paquet de données.



**Figure 14:** Le paquet d'intérêt et le paquet de données

Comme pour tout réseau orienté sur le contenu, l'architecture NDN est basée sur l'échange des deux paquets, l'utilisateur envoie une requête sous forme de paquet d'intérêt qui exécute un nom qui identifie une partie des données souhaitées. Après cela, le paquet d'intérêt sera acheminé au producteur de données potentiel en utilisant une stratégie de forwarding. Une fois que le nœud qui possède les données demandées reçoit le paquet d'intérêt, il répond avec un paquet de données qui porte le même nom et suit la même route que le paquet d'intérêt jusqu'à ce qu'il atteigne le demandeur (l'expéditeur). En analysant la conception

architecturale, les caractéristiques fonctionnelles clés et le support auxiliaire important; la taxonomie du NDN peut être classé comme illustré dans la Figure 15.



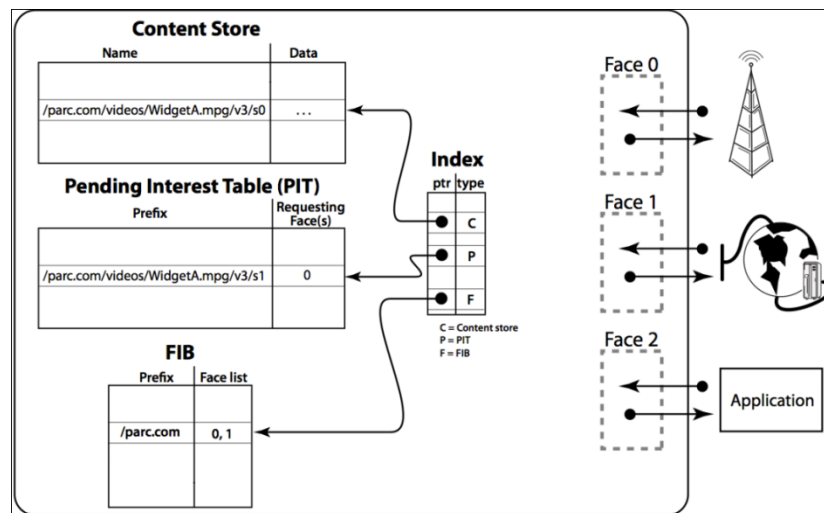
**Figure 15:** La classification de NDN

Le NDN étant un nouveau paradigme, les applications qui ont été conçues ou peuvent être développées en utilisant NDN aideront à convaincre les futurs chercheurs que le NDN a le potentiel d'assumer les responsabilités d'Internet à toutes fins utiles dans les années à venir. Voici une description des composants les plus importants :

- **Architecture du système.** Cette section traite de l'organisation des différents composants et leurs interactions au sein du système, ainsi que de ses différents principes de fonctionnement.
- **Services du système :** Il concerne les principales caractéristiques fonctionnelles du NDN, notamment le routage, le stockage et le forwarding, la sécurité, la confidentialité et la mobilité. Étant donné que le NDN prend en charge la mise en cache sur le réseau du contenu demandé par l'utilisateur tout en le diffusant. Stocker et transférer peuvent également être subdivisé en mise en cache et en transfert.
- **Les applications :** Il montre les avantages pratiques d'une architecture NDN bien conçue dans la vie réelle à travers les différentes applications traditionnelles. Basé sur la nature de la conception de l'architecture sous-jacente, le NDN prend en charge ces fonctionnalités pour le développement des applications et de réseaux robustes. Le système de nommage dans le NDN est purement spécifique à l'application

## IV.2 Les composants d'un routeur NDN

Pour transférer les paquets d'intérêt et les paquets de données, tous les routeurs NDN gèrent trois tables comme présenté dans la **Figure 16**:(i) Table de contenu ou *content Store* (CS) (ii) une table d'intérêt ou *Pending Interest Table* (PIT) et (iii) une table de transmission aussi appelée *Forwarding Information Base* (FIB).



**Figure 16:** Les tables d'un routeur NDN

- **La table de contenu (Content Store)**

La table de contenu a le même rôle que la mémoire tampon d'un routeur IP avec une différence significative et une politique de remplacement différente. Le **Tableau 2** présente la structure de la table de contenu. Le CS est une sorte de mémoire cache qui permet de stocker des copies locales des contenus. Les routeurs IP récupèrent les paquets une fois qu'ils les propagent en aval, tandis que les routeurs NDN conservent les paquets dans les routeurs pour une utilisation ultérieure (par exemple, de nombreux utilisateurs regardent la même vidéo). Par conséquent, les routeurs NDN ont besoin de plus de mémoire tampon que le routeur IP. Dans ce cas ; le Cloud Computing sera d'une grande utilité avec son énorme capacité de mémoire. L'utilisateur envoie une demande en diffusant son intérêt sur toutes les interfaces disponibles, une fois qu'un routeur reçoit le paquet d'intérêt qu'il recherche dans la table de contenu pour un paquet de données existant. Si la recherche aboutit à une correspondance, elle renvoie une réponse à l'utilisateur en envoyant un paquet de données à l'aide de l'interface d'entrée[58].

NOM	Contenu
HUM/ENDROIT	....
TEM/ENDROIT/VILLE	.....

**Tableau 2:** La structure de la table du contenu

- **La table des intérêts en attente (PIT)**

La table des requêtes ou (Pending Information Table) contient une liste des intérêts en attente de réponse, une liste des interfaces entrantes où les intérêts sont reçus, ainsi une liste des interfaces sortantes vers lesquelles les intérêts ont été transmis. Dans la table PIT, chaque interface entrante enregistre la durée de vie la plus longue d'intérêt qu'elle ait reçu ; à l'expiration de la durée de vie, l'interface entrante est supprimée de la table PIT. Chaque interface sortante enregistre l'heure à du transfert de l'intérêt via cette interface. Ainsi, lorsque le paquet de données est renvoyé, il est possible de calculer le temps d'aller-retour (RTT). La mesure RTT est ensuite utilisée pour mettre à jour l'estimation RTT pour le préfixe de nom correspondant déjà stocké dans le FIB, le **Tableau 3** illustre la structure de données d'entrée PIT qui maintient l'état de transmission du datagramme [22] [23].

Si aucune correspondance du paquet d'intérêt n'est trouvée dans la table de contenu, le routeur applique un algorithme de recherche de correspondance la plus longue dans une autre table. La table PIT garde la trace des paquets d'intérêts non satisfaits. Si un paquet correspondant est trouvé dans le PIT, cela signifie qu'un autre utilisateur a demandé les mêmes données. Par conséquent, le routeur ajoute l'interface entrante de l'intérêt dans PIT, sinon il ajoute une nouvelle entrée avec le nom de l'intérêt et l'interface entrante. Lorsque le paquet de données arrive, le routeur supprime le nom de l'intérêt et l'interface de PIT (les intérêts qui ont été satisfaits), sinon il supprime l'entrée après l'expiration du délai. L'avantage de cela est que PIT agrège les demandes qui réduisent les coûts généraux de réseau.

NOM	Interface d'entrée	Interface de sortie
HUM/ENDROIT	1,3	5
TEM/ENDROIT/VILLE	1,4,5	7

**Tableau 3:** La structure de la table PIT

Le PIT contient les interfaces d'arrivée des intérêts qui ont été transférés mais attendent toujours les données correspondantes. Cette information est nécessaire pour fournir des données à leurs consommateurs. Pour maximiser l'utilisation du PIT, les entrées PIT doivent être temporisées assez rapidement, quelque part autour du temps d'aller-retour des paquets. Toutefois, s'ils expirent prématurément, les données sont abandonnées et c'est la décision du consommateur de retransmettre ses intérêts. L'état PIT de chaque routeur a plusieurs fonctions critiques. Puisqu'il inclut l'ensemble d'interfaces par lesquelles les intérêts sont arrivés et il offre un support naturel pour la fonctionnalité de multidiffusion. Le routeur peut contrôler le débit des paquets de données entrants en contrôlant sa taille PIT. Avec la mise en cache des données, un réseau NDN supprime la dépendance à l'égard des protocoles de transport pour éviter la réduction des encombrements. Enfin, l'état PIT pourrait être exploité pour limiter les attaques par DDoS: le nombre d'entrées PIT est une indication explicite de la charge du routeur; une limite supérieure de ce nombre fixe le plafond aux effets d'une attaque par DDoS [59].

- **Table d'information de forwarding (FIB)**

Le **Tableau 4** présente la structure de la table FIB, Cette table est responsable du forwarding les paquets d'intérêt, elle se souvient de l'interface de transmissions du paquet. Une fois que l'intérêt atteint un nœud qui a les données demandées, un paquet de données est renvoyé qui porte à la fois le nom et le contenu des données, ainsi qu'une signature de la clé du producteur, ce paquet de données suit en sens inverse du chemin créé par le paquet d'intérêt initié par le consommateur.

Préfixe	Interface de sortie
HUM/ENDROIT	1,2
TEM/ENDROIT/VILLE	3,4

**Tableau 4:** La structure de la table FIB

Il faut signaler que ni les paquets d'intérêt ni les paquets de données ne portent d'adresses d'hôte ou d'interface (telles que les adresses IP) ; Les paquets d'intérêt sont acheminés vers les producteurs de données en fonction des noms et les paquets de données sont renvoyés en fonction des informations d'état configurées par les intérêts chaque saut de routeur.



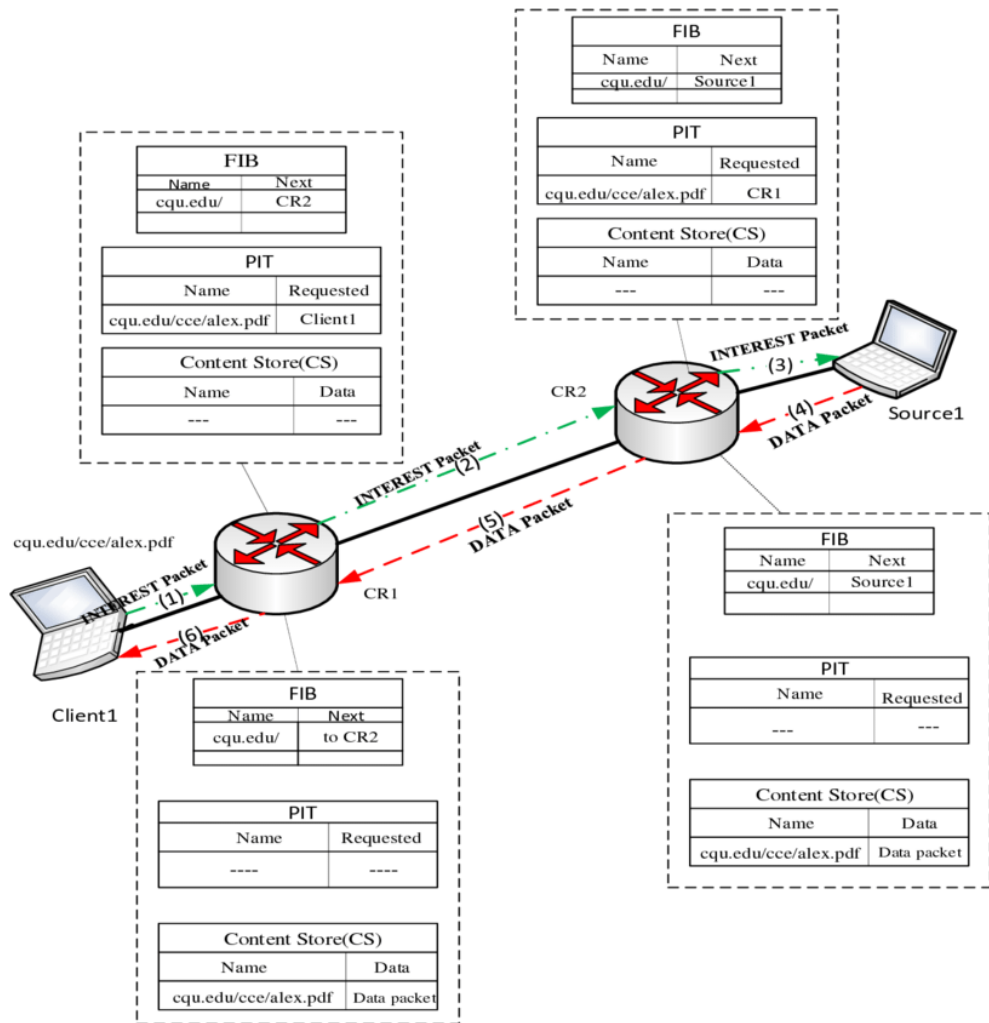
Le système de nommage étant implémenté pour faciliter la recherche et / ou la récupération du contenu. Les schémas de dénomination sont spécifiques à l'application et ils sont indépendants de réseau[55]. Les noms utilisés pour obtenir des données doivent être globalement uniques.

### IV.3 Le routage dans NDN

Les algorithmes de routage actuels ont des caractéristiques similaires aux mécanismes de routage de NDN. Comme le NDN récupère le contenu de réseau sans information de localisation, les protocoles actuels sont conçus pour atteindre la destination dans l'état de pré-topologie avec peu d'information en avance, par conséquent, les politiques de routage et de transmission du NDN peuvent accepter la plupart des protocoles de routage de l'Internet actuel. La différence majeure est la forme des informations d'accessibilité. Les protocoles de routage NDN doivent annoncer les préfixes de contenu basés sur des noms, tandis que les protocoles actuels utilisent des préfixes IP. Internet étant conçu pour atteindre facilement les destinations en utilisant des adresses IP, il est difficile d'appliquer simplement les protocoles actuels dans un NDN.

Le NDN utilise ses structures, à savoir PIT, FIB et CS, pour le routage et le forwarding des paquets. Dans le plan de données du NDN, afin de transmettre un paquet ; le tableau PIT utilise les paquets d'intérêts en attente et leurs interfaces d'entrée. Lors de la réception d'un paquet de données, le PIT envoie automatiquement le contenu via les interfaces d'où proviennent les intérêts correspondants et supprime les entrées de la table. Cela permet au plan de données du NDN de conserver l'état de routage, contrairement à celui du plan de données de l'architecture IP [60]. La table PIT peut également être utilisée pour empêcher les boucles, puisqu'un routeur abandonnera certains paquets d'intérêt lorsqu'il les aura dans la table.

Dans NDN, le routage est utilisé pour définir la topologie et les stratégies et pour gérer leurs modifications à long terme, ainsi que pour mettre à jour la table de forwarding. Le protocole de routage NDN se coordonne avec le plan de forwarding pour le classement et la vérification des interfaces [28] [29] comme présenté dans la **Figure 17**. Dans le NDN, la seule différence entre routage et forwarding réside dans le fait que le routage décide la disponibilité des chemins alors que le forwarding prend les décisions relatives à la préférence et à l'utilisation des chemins en fonction de leurs performances /statut.

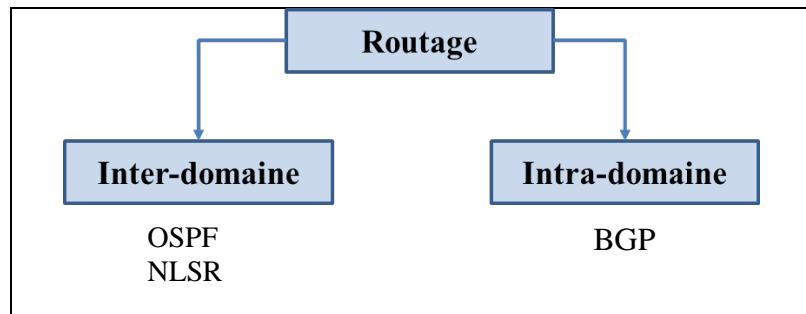


**Figure 17:** Le contenu des routeurs NDN

Les algorithmes de routage sont conçus pour le réseau internet, cela veut dire que les protocoles à l'état de liens et les protocoles à vecteur de distance peuvent être utilisés dans l'architecture NDN avec une légère modification. Les routeurs IP et les routeurs NDN utilisent le FIB pour stocker les informations relatives au routage. Le routeur IP recherche l'adresse de destination dans la FIB pour trouver le prochain saut et acheminer les paquets à l'adresse de destination, pas nécessairement via le meilleur chemin. Le routeur NDN cherche le préfixe du nom dans la table FIB pour chercher les prochains sauts et récupérer les données, pas nécessairement la copie la plus proche.

Les protocoles de routage Internet actuels peuvent être adaptés pour une utilisation dans le NDN en modifiant les types de message (paquet d'intérêt et paquet de données) et en ajoutant le forwarding par plusieurs chemins comme présenté dans **Figure 18**. Alors comme le NDN traite les contenus des noms, la table de routage de NDN peut utiliser plus d'espace mémoire que la table de routage

IP. Il existe trois principaux protocoles de routage [32] [33] dans le NDN et tous sont conçus pour effectuer uniquement le routage intra-domaine.



**Figure 18:** Les types de routage dans NDN

Les protocoles de routage Inter-domaines prennent des décisions de routage en fonction des chemins et des stratégies de réseau, tandis que les protocoles de routage intra-domaines tels que le routage à l'état de liens comme *Link State Routing* (LSR) et OSPF (*Open Shortest Path First*) calculent le chemin le plus court parmi les routeurs. Les opérateurs de réseau établissent et mettent en œuvre des politiques pour la sélection des chemins. Le NDN achemine et transmet les paquets en basant sur leurs noms, ce qui élimine quelques problèmes posés par les adresses IP. Le routage peut être effectué de la même manière que le routage IP actuel. Au lieu d'annoncer les préfixes IP, un routeur NDN annonce les préfixes des noms qui couvrent, les données que le routeur est prêt à servir. Cette annonce se propage sur le réseau via un protocole de routage et chaque routeur construit son FIB en fonction des annonces de routage reçues. Les protocoles de routage conventionnels, tels qu'OSPF et BGP, peuvent être adaptés pour router en se basant sur les préfixes de nom. Cependant, un espace de noms non limité pose la question de savoir comment garder les tailles de table de routage adaptables au nombre de noms de données.

Les routeurs traitent les noms comme une séquence de composants opaques et effectuent simplement les tâches les plus longues. L'architecture NDN prend en charge de manière inhérente le routage multi chemins. Le routage IP adopte un meilleur chemin pour éviter les boucles. Dans NDN, les intérêts ne peuvent pas être mis en boucle de manière persistante, car le nom est unique et donc on peut identifier les doublons à supprimer. Les données ne sont pas en boucle, car elles empruntent le chemin inverse des intérêts. Ainsi, un routeur NDN peut envoyer un paquet d'intérêt en utilisant plusieurs interfaces sans se soucier des boucles. Les premières données renvoyées satisfieront le paquet d'intérêt seront mises en cache

localement ; les copies arrivées plus tard seront rejetées. Cette fonctionnalité multi-accès intégrée prend en charge l'équilibrage de charge ainsi que la sélection de services.

Par exemple, un routeur peut transférer quelques premiers paquets d'intérêts via toutes les interfaces possibles, mesurer les performances en fonction du retour des données et sélectionner la ou les interfaces les plus performantes pour les prochains paquets d'intérêts. La sécurité du routage est améliorée dans NDN. Premièrement, la signature de toutes les données, y compris les messages de routage, les empêche d'être falsifiées ou falsifiées. Deuxièmement, le routage multi chemins atténue le détournement de préfixes, car les routeurs peuvent détecter l'anomalie provoquée par le détournement de préfixes et essayer d'autres chemins pour récupérer les données. Troisièmes, le fait que les messages NDN ne peuvent échanger que de données et ne peuvent tout simplement pas être adressés à des hôtes, le rend difficile d'envoyer des paquets malveillants à une cible particulière. Pour être efficaces, les attaques contre le NDN doivent être axées sur le déni de service[59].

### **IV.3.1 Routage intra-domaine**

On dit que le routage intra-domaine dans le réseau concentré sur le contenu est très compatible avec les schémas de routage connus dans les réseaux IP. Les deux s'appuient sur la recherche la plus longue afin d'agréger de manière hiérarchique des informations de connexion plus détaillées dans le processus de rapprochement du contenu. La similitude est respectée dans la relation étroite entre la FIB et la table de routage IP. La différence principale de NDN est que la FIB a plus d'une interface sortante. La raison est la différence sémantique des entrées. Alors qu'un routeur IP peut atteindre tous les hôtes commençant par un préfixe via l'interface, le routeur NDN peut atteindre une partie du contenu avec ce préfixe. En conséquence, le routeur NDN diffuse des informations sur toutes les interfaces de cette entrée afin de collecter des informations complètes.

Dans un réseau IP, cette opération entraînerait des boucles dans la topologie. En revanche, cela ne pose pas de problème pour le réseau NDN, car ni les paquets d'intérêt ni les paquets de données ne peuvent être mis en boucle. Ainsi, le NDN est libre de maintenir un arbre de recouvrement dans la topologie. Par exemple, si deux nœuds d'une topologie annoncent le même préfixe, cela signifie que tous les contenus avec un tel préfixe peuvent être atteints. Par conséquent, le mécanisme de forwarding doit choisir le meilleur d'entre eux. Dans le NDN, ces deux nœuds

n'annoncent pas tous les deux l'accessibilité au contenu entier sous ce préfixe, mais peut-être juste un sous-ensemble. Ainsi, les deux nœuds doivent être interrogés. Ce mécanisme de forwarding soulève des questions d'évolutivité [15].

Malgré ces différences, les protocoles de routage existants, à savoir IS-IS (*Intermediate-System-Intermediate System*) et le protocole OSPF, devraient être facilement adaptables aux besoins du NDN. Les protocoles de routage mentionnés gèrent essentiellement deux tâches distinctes. La première consiste à découvrir les connectivités locales vers les nœuds adjacents. La seconde consiste à annoncer les ressources disponibles sur l'ensemble de réseau, par ces moyens, un nœud compatible NDN peut diffuser des annonces d'état de lien contenant les préfixes attachés qu'il peut servir. Les auteurs de NDN proposent maintenant que cela soit déjà possible avec les déploiements existants de ces protocoles[61].

- **Le protocole NLSR (Named-data Link State Routing protocol)**

En tant que protocole d'état des liens, NLSR[62] diffuse les LSA (*Link State Advertisements*) pour créer une topologie de réseau et distribuer l'accessibilité du préfixe de nom. Au niveau de la NLSR, le routeur établit et maintient des relations de contiguïté avec les routeurs voisins. Chaque fois qu'il détecte la défaillance, il diffuse un nouveau LSA sur l'ensemble de réseau. De plus, il annonce les préfixes de la configuration statique et de l'enregistrement dynamique par les producteurs de contenu locaux. Chaque fois qu'un préfixe de nom est ajouté ou supprimé, il diffuse également un nouveau LSA.

La dernière version des LSA est stockée dans une base de données sur l'état des liens (LSDB) pour chaque nœud. Une telle diffusion de la topologie peut être implémentée dans des protocoles de routage IP. Cependant, étant donné que le protocole NLSR est implémenté en utilisant des paquets d'intérêt et de données de NDN, la conception doit changer à partir des concepts familiers d'adresses IP et de transmission de données IP (c'est-à-dire que tout nœud peut simplement envoyer n'importe quel paquet à un autre nœud). Au lieu de cela, nous devons penser en termes de noms de données et de récupération de données. Plus spécifiquement, nous avons besoin d'un schéma de nommage systématique pour les routeurs et les mises à jour de routage. Nous devons donc récupérer rapidement les mises à jour de routage sans savoir a priori quand mettre à jour peut être généré, puisqu'un changement de topologie ou de préfixe de nom peut avoir lieu à tout moment.

En termes de fonctionnalité de routage, le protocole NLSR se distingue des protocoles de routage à état de liens sous deux aspects : (i) il fournit plusieurs

chemins pour chaque préfixe du nom, plutôt qu'un seul chemin le plus court ; et (ii) il signe et vérifie tous les *LSA* pour s'assurer que chaque routeur ne peut générer que ses propres informations de préfixe et de connectivité. En tant qu'étape préliminaire dans le développement de protocoles de routage basés sur NDN, la conception de NLSR se situe dans le contexte d'un domaine de routage unique[63].

- **Le protocole OSPFN (Open Shortest Path First for NDN)**

Le protocole OSPFN [26] est le premier protocole de routage intra-domaine pour NDN, qui étend le protocole OSPF pour prendre en charge le routage basé sur le nom. Au lieu de porter les préfixes IP, il annonce les préfixes nommés et l'utilise pour déterminer le meilleur itinéraire. Afin de mettre en œuvre ce protocole ; un nouveau type de message, appelé OLSA (*Opaque Link State Advertisement*) a été créé qui inclut une seule information de routage. Un routeur OSPFN a trois processus différents : CCND (*Démon de réseau centré sur le contenu*), OSPFN et OSPFD (démon OSPF). Une fois que l'OSPFN a créé des OLSA contenant un préfixe de nom unique et les a envoyés à OSPFD, le démon les diffuse vers tous les routeurs de réseau.

Compte tenu des informations de routage, un OSPFD envoie les OLSA à son OSPFN. Il met à jour et installe les nouvelles informations de routage sur sa table FIB. En conséquence, le routeur détermine le meilleur prochain saut basé sur les informations dans la table FIB. Cependant, les propriétés inhérentes au protocole de routage basé sur IP présentent quelques inconvénients. Les adresses IP des routeurs sont toujours nécessaires pour identifier les routeurs voisins, car elles sont utilisées en tant qu'ID de routeur dans le réseau OSPFN. De plus, les conceptions non encore fonctionnelles du protocole permettent aux routeurs de ne transmettre les paquets qu'au meilleur saut suivant, ce qui ne prend pas en charge efficacement la fonctionnalité de trajet multi chemins, l'une des caractéristiques avantageuses du NDN.

### **IV.3.2 Routage inter-domaine**

Lorsque quelques clients d'un fournisseur d'accès Internet commencent à utiliser le NDN, il est préférable de déployer un ou plusieurs routeurs de contenu afin de réduire les coûts de peering (un seul exemplaire du contenu doit traverser un lien de peering inter-fournisseur, indépendant. Le nombre de clients qui le demandent) tout en réduisant la latence moyenne des clients. Il existe donc une structure incitative ascendante orientée sur la périphérie permettant de développer le réseau NDN dès qu'il atteint un seuil de déploiement[64].

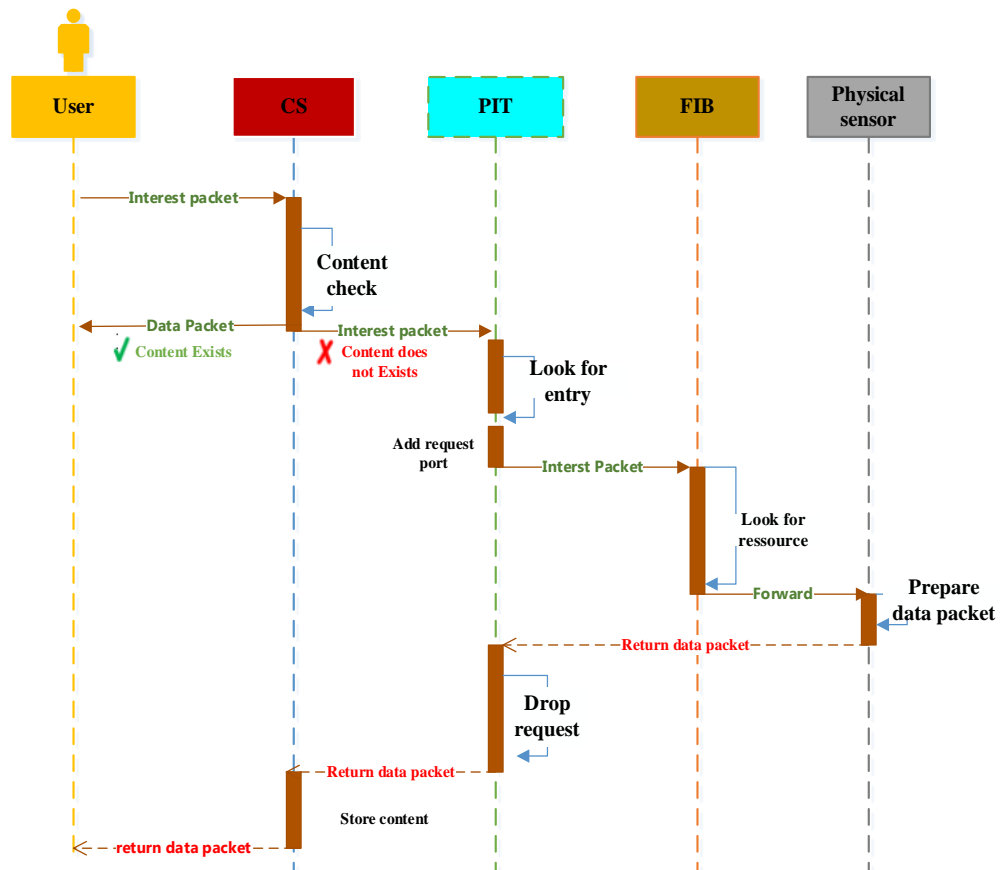
Le problème central de ce type de déploiement de bas en haut est de combler le fossé entre les domaines. Les routeurs ISPs bénéficient du contenu entrant (contenu demandé par leurs clients), mais non sortant (contenu créé par leurs clients). Cela annule partiellement un avantage majeur à long terme pour le NDN qui consiste à rendre le trafic proche de la racine d'un arbre de distribution de contenu, indépendamment de la popularité du contenu. Aujourd'hui, le trafic croît de manière linéaire avec la popularité. Ce problème peut être résolu en intégrant des préfixes de contenu au niveau du domaine dans BGP.

Le protocole inter domaine BGP actuel a l'équivalent du mécanisme TLV IGP qui permettrait aux domaines d'annoncer les préfixes de contenu de leurs clients. Les informations de chemin AS BGP permettent également à chaque domaine de construire une carte de topologie équivalente à celle construite dans le cas IGP, mais au niveau du système autonome (AS) plutôt qu'au niveau du préfixe de réseau. Cette carte est fonctionnellement équivalente au cas d'IGP (on découvre quels domaines servent des intérêts dans certains préfixes et quel est le domaine le plus proche capable de NDN sur les chemins d'accès à ces domaines) de sorte que les mêmes algorithmes s'appliquent [64].

#### IV.4 Le forwarding

Comme le plan de transfert de NDN est capable de détecter les défaillances (nœud, lien ou paquet) et d'effectuer des récupérations, le routage n'a pas besoin d'effectuer de mises à jour continues de la table FIB, ce qui améliore l'évolutivité et la stabilité du plan de routage NDN. Le forwarding peut être classé selon les stratégies de forwarding et du transfert évolutif, le transfert évolutif prend en charge le forwarding intelligent et dynamique. Grâce à cela, les routeurs NDN peuvent mesurer le RTT (*Round Trip Time*), le débit, les pertes de paquets et le chemin alternatif du paquet pendant la surcharge [65]. Le plan de forwarding dans le NDN agit également en tant que plan de contrôle, car les stratégies de forwarding exécutent toutes les décisions nécessaires au transfert des paquets d'intérêt et les paquets de données.

Le diagramme de séquence illustré dans la **Figure 19** présente le processus de forwarding d'un paquet d'intérêt et du paquet de données correspondant sur un routeur NDN. Lorsqu'un paquet d'intérêt d'un utilisateur arrive sur un routeur NDN, le nom du contenu associé est recherché dans le Content store pour trouver une correspondance

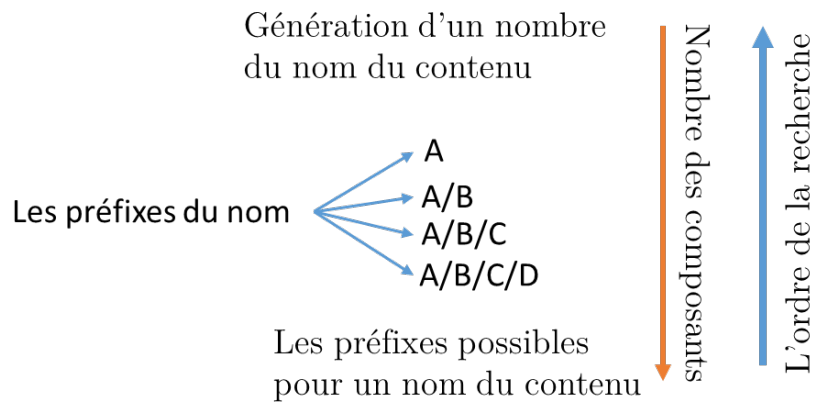


**Figure 19:** Diagramme de séquence d'échange paquets

Si le contenu correspondant est trouvé, le routeur NDN transmet le contenu à l'utilisateur via un paquet de données, signé par la clé du producteur [10]. Sinon, on cherche le nom du contenu dans le PIT. Si une entrée PIT correspondante est trouvée, l'interface d'entrée du paquet d'intérêt sera ajoutée à la liste des interfaces cette opération s'appelle *l'agrégation* d'intérêts. Ainsi, lorsque le paquet de données correspondant sera disponible, tous les utilisateurs intéressés recevront une copie de ce paquet de données.

Si aucune entrée PIT n'est trouvée pour un paquet d'intérêt, celui-ci est transmis au FIB du routeur qui effectue un l'algorithme de la recherche du préfixe le plus long (LPM) *Longest Prefix Matching*, comme indiqué sur la **Figure 20**. Lorsqu'une entrée FIB correspondante est trouvée, le paquet d'intérêt est transmis au prochain saut correspondant et une nouvelle entrée PIT sera créée avec son interface d'entrée pour la même chose. Sinon, le paquet d'intérêt sera inondé dans toutes les interfaces sortantes ou il sera supprimé en fonction de la stratégie de forwarding du routeur.





**Figure 20:** La recherche du préfixe le plus long

Chaque fois qu'un paquet de donnée revient à un routeur NDN, toutes les entrées PIT sont recherchées pour trouver un nom de contenu correspondant. Si une entrée PIT correspondante est disponible, le paquet de donnée est transmis à toutes les interfaces mentionnées dans la liste des interfaces entrantes. Après cela, l'entrée PIT est supprimée et le contenu est stocké dans le CS en fonction de la stratégie de mise en cache locale, afin de répondre aux futures demandes du même contenu. Si aucune entrée correspondante n'existe (peut-être parce que sa durée de vie est terminée), le paquet de données est supprimé.

La stratégie de forwarding décide comment utiliser efficacement les options de forwarding et choisit la meilleure interface pour transférer le paquet d'intérêt. La conception d'une stratégie de forwarding dépend de l'environnement de réseau et du contexte [29] [65]. Les principaux facteurs permettant de concevoir une stratégie de forwarding sont les suivants : chemin de travail (inondation ou meilleur chemin unique), une sélection basée sur le contexte, sélection à plusieurs chemins à la demande et la recherche efficace d'interface (s). Certaines stratégies de forwarding sont utilisées par le plan de forwarding NDN [29] [65].

#### IV.5 Le système du cache

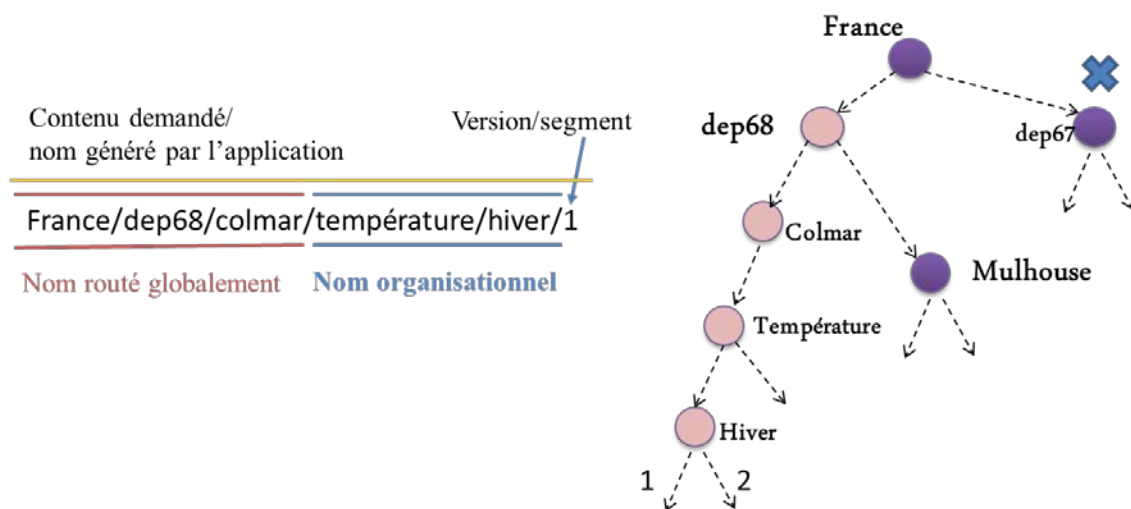
À la réception d'un intérêt, un routeur NDN vérifie d'abord la table de contenu. S'il existe des données dont le nom tombe sous le nom de l'intérêt, les données seront renvoyées en réponse. Le Content Store, dans sa forme de base, n'est que la mémoire tampon du routeur actuel. Les routeurs IP et les routeurs NDN mettent en mémoire tampon les paquets de données. La différence est que les routeurs IP ne peuvent pas réutiliser les données après les avoir transférés, alors que les routeurs NDN peuvent les réutiliser, car ils sont identifiés par des noms persistants. Pour les fichiers statiques, le NDN assure une livraison des données presque optimale. Même

le contenu dynamique peut tirer parti de la mise en cache dans le cas de multidiffusion (par exemple, téléconférence) ou retransmission de paquets après une perte de paquet.

La mise en cache de données nommées peut soulever des problèmes de confidentialité. Les réseaux IP actuels offrent une faible protection de la vie privée. On peut savoir ce qu'il y a dans un paquet IP en inspectant l'en-tête ou la charge utile, et qui a demandé les données en vérifiant l'adresse de destination. NDN nomme explicitement les données, ce qui permet sans doute plus facilement à un moniteur réseau de voir quelles données sont demandées. On peut également être en mesure de savoir quelles données sont demandées par le biais de schémas de sondage intelligents pour extraire ce qui se trouve dans le cache. Cependant, le NDN supprime entièrement les informations concernant qui demande les données. À moins d'être directement connecté à l'hôte demandeur par un lien point à point, un routeur saura uniquement que quelqu'un a demandé certaines données, mais ne saura pas qui est à l'origine de la demande. Ainsi, l'architecture NDN offre naturellement une protection de la vie privée à un niveau fondamentalement différent de celui d'Internet actuel[66].

#### **IV.6 Le schéma de nommage**

Les noms dans NDN sont opaques pour le réseau, c'est-à-dire que les routeurs ne connaissent pas la signification d'un nom, bien qu'ils connaissent les limites entre les composants d'un nom. Ce découplage permet à chaque application de choisir le schéma de dénomination qui lui convient et permet aux schémas de dénomination d'évoluer indépendamment de réseau. Cependant, la conception NDN implique que les noms doivent être structurés hiérarchiquement. Par exemple, une donnée produite par un capteur de température peut porter le nom /paris/temps/cs où le « / » délimite les composants de nom dans des représentations textuelles similaires aux URLs. Le schéma général de nom du contenu (*Content Name*) et sa structure hiérarchique sont illustrés dans **Figure 21**, cette structure hiérarchique permet aux applications de représenter les relations entre les éléments de données et permet aussi l'agrégation de noms. Les noms à plat peuvent être traités comme un cas spécial dans les environnements locaux. Cependant, les espaces de noms hiérarchiques sont importants pour la mise à l'échelle du système de routage. Les producteurs et les consommateurs des données doivent se mettre d'accord sur le schéma des noms.



**Figure 21:** Nom du contenu et sa représentation hiérarchique

Afin de récupérer des données générées dynamiquement, les consommateurs doivent être en mesure de construire de manière déterministe le nom d'un élément de données souhaité sans avoir préalablement vu le nom ou les données. Soit : (1) un algorithme déterministe permet au producteur et au consommateur d'obtenir le même nom sur la base des informations disponibles aux deux partenaires, soit (2) des sélecteurs d'intérêts associés à la recherche de la correspondance la plus longue récupèrent les données souhaitées au moyen d'une ou plusieurs itérations. Le consommateur peut alors demander des segments ultérieurs en utilisant une combinaison d'informations révélées par le premier paquet de données et la dénomination convention de l'application de publication pour demander des paquets ultérieurs.

Les NDN utilisent un schéma de nommage hiérarchique composé de plusieurs composants qui peuvent être une longueur de chaîne quelconque organisée en structure hiérarchique [67]. Les noms de l'architecture NDN sont similaires à l'URL (*Uniform Resource Locator*) classique. Cependant, ce n'est pas nécessairement que les noms peuvent être lisibles, seuls les noms utilisés pour extraire des données requièrent une unicité globale. En fait, les noms de données individuels peuvent avoir un sens dans divers domaines et contextes spécifiques. La gestion de l'espace de noms ne fait pas partie de l'architecture NDN, tout comme la gestion de l'espace d'adressage ne fait pas partie de l'architecture IP. Cependant, la dénomination est la partie la plus importante de la conception du NDN. Les noms de données permettent une prise en charge naturelle de fonctionnalités telles que la distribution de contenu, la multidiffusion, la mobilité et la mise en réseau.

Le système de nommage est l'élément le plus important de l'architecture NDN et fait toujours l'objet de recherches approfondies ; en particulier, comment définir et attribuer des noms de premier niveau reste un défi ouvert. Cependant, il n'est pas nécessaire de répondre immédiatement à toutes les questions de nommage ; l'opacité des noms sur le réseau - et la dépendance vis-à-vis des applications - signifie que la conception et le développement de l'architecture NDN peuvent, et doivent en fait, se poursuivre en parallèle avec nos recherches sur la structure des noms, la découverte de noms et la navigation dans les espaces de noms dans le contexte du développement des applications.

#### IV.7 NDN pour les RCSF

Un RCSF est composé des nœuds de capteurs déployés de manière dense et aléatoire, susceptible de tomber en panne et ils ont des ressources limitées. Le RCSF fonctionne de manière centrée sur les données, car les requêtes sont traitées avec les données requises. De plus, dans un RCSF ; l'adressage est unique et ce n'est pas possible, car un nœud de capteur ne possède pas d'identification globale (ID global) en raison d'une surcharge élevée et de la présence d'un grand nombre de nœuds de capteur. La nature centrée sur les données de RCSF et l'absence d'identificateur global (Global ID) montrent que le NDN et ses caractéristiques peuvent être facilement appliqués au RCSF pour améliorer les performances globales du système. Certains des avantages du NDN par rapport au RCSF sont la récupération rapide des données, car le NDN utilise une dénomination hiérarchique, une évolutivité, car le NDN prend en charge de manière inhérente la communication par diffusion, la mise en cache, le développement facile des applications et un déploiement flexible.

Le NDN peut être considéré comme une solution intéressante pour les défis de RCSF. En effet, ses caractéristiques et ses avantages s'adaptent aux applications des capteurs déployés et aux contraintes de leur RCSF. NDN peut offrir les services suivants :

- **La récupération facile de données.** : La dénomination hiérarchique des données rend les opérations de recherche de contenu et de récupération de données plus flexibles à grande échelle.
- **L'agrégation de données** : le schéma de nommage de l'approche NDN facilite l'agrégation de données.

- **L'évolutivité** : L'approche NDN est conçue pour la récupération de données de nombreux nœuds dans une zone surveillée. En exploitant le support sans fil de diffusion et en appliquant des stratégies de transmission légères.
- **Flexibilité de déploiement** : Le NDN peut être implémenté soit en tant que solution à remplacer, en remplaçant les protocoles TCP / IP au-dessus des technologies de la couche deux, soit en superposition au-dessus d'un réseau IP pour permettre la conformité en amont avec le réseau principal existant.
- **La mise en cache** : Le contenu mis en cache à différents nœuds (selon capacité de stockage) peut également être disponible avec une connectivité intermittente (par exemple, en raison d'un fonctionnement à faible consommation).
- **Le développement facile d'applications** : Les applications peuvent demander des données en fonction du contenu, quel que soit leur emplacement physique, qui est donc transparent pour les applications.

Le NDN empêche la dégradation de la communication de la zone surveillée, en exploitant les procédures de transfert de média sans fil de diffusion et il évite la dégradation de la qualité de la communication lorsque le nombre de nœuds impliqués augmente. Les approches hybrides peuvent également être déployées avec un segment câblé dans lequel IP est largement utilisé et un WSN centré sur les données dans le segment d'accès.

## IV.8 Conclusion

Dans ce chapitre, nous avons étudié le paradigme de 'Named Data Networking' et son avantage pour les réseaux de capteurs sans fil. Ce chapitre explore les potentialités de NDN dans les RCSF. Le NDN présente deux avantages majeurs pour les RCSF : une recherche facile des données et une acquisition facile des données sans connaître l'emplacement avec l'avantage du système de nommage hiérarchique dans les RCSF à grande échelle. De plus, utiliser le contenu mis en cache sur les routeurs intermédiaires est également un avantage important offert par le NDN. Dans le prochain chapitre, on décrit notre solution proposée pour intégrer les RCSF dans le Cloud Computing en utilisant l'approche NDN.

## V Une architecture basée sur le réseau de données nommées

### Sommaire

<b>V Une architecture basée sur le réseau de données nommées .....</b>	<b>82</b>
V.1 Introduction .....	83
V.2 Le réseau de capteurs virtuels .....	84
V.2.1 La structure et la configuration .....	85
V.2.2 Les types des réseaux de capteurs virtuels.....	89
V.3 L'architecture proposée .....	90
V.3.1 La couche physique.....	91
V.3.2 La couche infrastructure de Cloud Computing.....	92
V.3.3 La couche d'application .....	96
V.4 La structure de données .....	96
V.4.1 Le système de nommage .....	96
V.4.2 La gestion des paquets .....	98
V.5 Conclusion .....	104

## **V.1 Introduction**

Comme nous avons vu dans le chapitre précédent, les travaux de recherche ne traitent pas la problématique du routage dans les réseaux de capteurs virtuels ou Virtual Sensor Networks[68](VSN), dans notre étude on considère qu'il y a un nombre élevé de capteurs virtuels ou communément appelés «Cloud de capteurs». Le routage est très important et la question doit être abordée dans nos recherches. Nous nous concentrons sur la transmission des données dans VSN adapté réseau des données nommées(Named Data Networking) comme solution parce que son avantage est d'abord de nommer un réseau des données nommées dans un cadre indépendant qui peut être utilisé dans toute plate-forme et il est similaire au paradigme TCP/IP avec une énorme différence d'adresse hôte, comme mentionné dans NDN[55]. Dans cette approche nous utilisons le schéma de nom pour héberger l'hôte, formant un réseau de capteurs virtuels évolutif sous les hypothèses de conception suivantes :

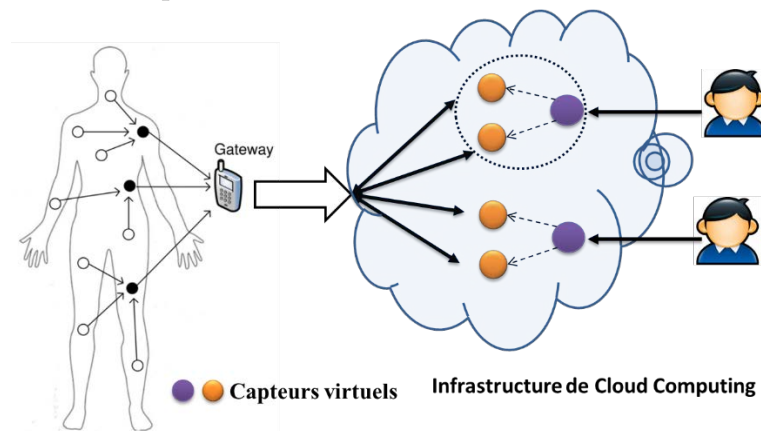
- Un groupe d'utilisateurs (privés, professionnels et / ou gouvernement) est prêt à déployer un nœud de capteur matériel pour couvrir les zones sous leur pouvoir.
- Un utilisateur typique appartenant au groupe emprunte souvent les capacités des nœuds de capteurs à partir de parties d'un autre capteur infrastructure de réseau pour former un réseau de capteurs virtuels sur une zone de couverture souhaitée.
- Les règles de partage sont dictées par le propriétaire de la ressource et à aucun moment la tâche de détection d'une personne locale ou d'un utilisateur distant ne viole l'autorité, la vie privée et la sécurité du fournisseur.
- éventuellement, un modèle de gestion associé à des comptes pour équilibrer le coût du partage des ressources (telles que coûts de matériel et de batterie) entre le privé et l'usage communautaire.

La virtualisation des nœuds physiques joue un rôle très important dans notre architecture, on suppose qu'il existe différents types de nœuds déployés. Le capteur virtuel permet l'utilisation des capteurs sans se soucier des emplacements et de la spécification des capteurs physiques. Chaque capteur virtuel est créé à partir d'un ou de plusieurs de capteurs physiques. Les utilisateurs peuvent créer des capteurs virtuels et les utiliser librement comme s'ils possédaient des capteurs. Par exemple, ils peuvent activer ou désactiver leur serveur virtuel, vérifier leur état et définir la

fréquence de la collecte de données. Si plusieurs utilisateurs librement contrôler les capteurs physiques, certaines commandes incohérentes peuvent être émises. Les utilisateurs peuvent contrôler librement leurs propres capteurs virtuels en vitalisant le capteur physiquement tant que capteur virtuel.

## V.2 Le réseau de capteurs virtuels

Un capteur virtuel est une émulation d'un capteur physique qui récupère les données à partir des capteurs physiques sous-jacents. Les capteurs virtuels fournissent une vue personnalisée aux utilisateurs à l'aide de la transparence de la distribution et de l'emplacement (voir **Figure 22**). Dans les capteurs sans fil, le matériel est à peine capable d'exécuter plusieurs tâches à la fois et la question de pouvoir exécuter plusieurs machines virtuelles, comme dans le Cloud Computing traditionnel, est hors de question.



**Figure 22:** Exemple de capteurs virtuels

Pour surmonter ce problème, nous utilisons des capteurs virtuels dans notre architecture comme image dans des capteurs physiques correspondants. Les capteurs virtuels contiennent des métadonnées sur les capteurs physiques et l'utilisateur. De plus, le capteur virtuel peut avoir un code de traitement des données, qui peut être utilisé pour traiter des données en réponse à des requêtes complexes de l'utilisateur. Nous avons implémenté des capteurs virtuels dans quatre configurations différentes : des configurations un-à-plusieurs, plusieurs-à-un, plusieurs-à-plusieurs et dérivées.

De nouveaux modèles d'architecture permettent d'intégrer des capteurs sans fil dans le Cloud Computing. Cette communication entre le capteur physique et le Cloud Computing se fait par une couche virtuelle qui implémente le fonctionnement des capteurs réels. Une telle couche donne naissance à de nouvelles architectures qui



s'appellent (Sensor-Cloud) ou Cloud de capteurs virtuels. Avec une telle architecture, il est possible de créer des services basés sur les capteurs virtuels, c'est-à-dire des environnements de capteurs distribués géographiquement et pouvant être utilisés à la demande par plusieurs utilisateurs. Dans les capteurs sans fil, le matériel est à peine capable d'exécuter des tâches multiples en même temps et la question d'être en mesure d'exécuter plusieurs machines virtuelles, comme dans le Cloud Computing traditionnel, est hors de question.

La puissance des capteurs virtuels réside dans le fait que les capteurs physiques utilisés par le capteur virtuel peuvent être hétérogènes. Les capteurs virtuels peuvent cacher la source des données provenant des réseaux de capteurs, ce qui va permettre aux utilisateurs d'utiliser les capteurs sans se soucier d'emplacement et des caractéristiques des capteurs physiques. Les capteurs physiques sont interconnectés par des liaisons sans fil pour effectuer des tâches de détection et ils sont déployés dans des environnements non surveillés. Tandis que les nœuds capteurs virtuels sont reliés entre eux par des liens virtuels et ils sont déployés dans des machines virtuelles (VM).

### **V.2.1 La structure et la configuration**

Les capteurs virtuels comportent des métadonnées sur les capteurs physiques et l'utilisateur qui détient actuellement ce capteur virtuel. De plus, le capteur virtuel peut avoir un code de traitement des données, qui peut être utilisé pour traiter des données en réponse à des requêtes complexes de l'utilisateur. Les capteurs virtuels peuvent être implémentés en quatre configurations :

#### **1) Un à plusieurs (one-to-many)**

Comme présenté dans la **Figure 23**, plusieurs capteurs virtuels reçoivent des flux de données provenant d'un seul capteur physique. Bien que les utilisateurs individuels possèdent l'image virtuelle, le capteur physique sous-jacent est partagé entre tous les capteurs virtuels qui y accèdent.

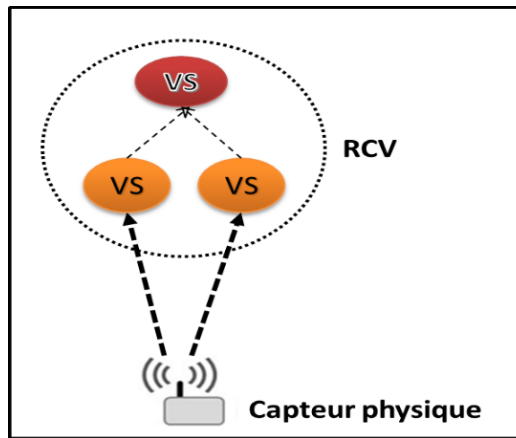


Figure 23: La structure plusieurs à un

### 1) Plusieurs à un (many-to-one)

Dans cette structure, la zone géographique est divisée en régions et chaque région peut comporter un ou plusieurs réseaux physiques (voir la Figure 24). Lorsqu'un utilisateur a besoin de données agrégées de phénomènes spécifiques d'une région, tous les RCSF sous-jacents sont activés avec les informations respectives au phénomène et l'utilisateur dans ce cas accède aux données partagées de ces réseaux.

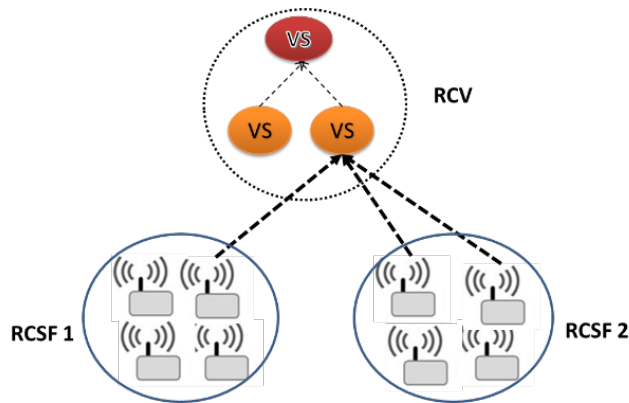


Figure 24: Un à plusieurs

### 2) Plusieurs à plusieurs (many-to-many)

Cette structure est une combinaison des deux structures, un capteur physique peut correspondre aux nombreux capteurs virtuels alors qu'il fait partie d'un seul réseau qui fournit des données pour un capteur virtuel unique comme représenté sur la Figure 25.

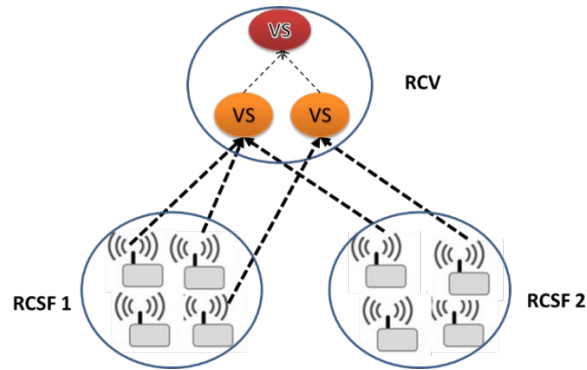


Figure 25: Plusieurs à plusieurs

### 3) Capteur combiné

Cette configuration peut être considérée comme une généralisation des trois autres configurations, mais la différence réside dans les types de capteurs physiques avec lesquels un capteur virtuel communique. Alors que dans la configuration dérivée, le capteur virtuel communique avec plusieurs types de capteurs. Dans les trois autres configurations, le capteur virtuel communique avec le même type de capteurs physiques.

La modélisation de l'interaction des capteurs virtuels est présentée dans **Figure 26** nous avons modélisé le capteur virtuel comme suit : un ensemble de capteurs physiques peut être représenté comme l'ensemble  $G\{gi\}$  qui comporte tous les capteurs physiques. Le capteur virtuel  $v_i$  peut être sous l'une des formes suivantes :

- $$v_i \left\{ \begin{array}{l} p_i \text{ Simple} \dots\dots\dots \text{①} \\ \{p_i | p_i \in B\} \text{ Sélectionneur} \dots\dots\dots \text{②} \\ B \text{ Accumulateur} \dots\dots\dots \text{③} \\ h(B) \text{ Agrégateur} \dots\dots\dots \text{④} \\ \{p_i : \forall p_i \in B \rightarrow f(p_i)\} \text{ qualificateur} \dots\dots\dots \text{⑤} \end{array} \right.$$

Vous trouverez ci-dessous une description ainsi que des cas d'utilisation pour chacun des types de capteurs virtuels ci-dessus.

**① Simple** : un capteur virtuel direct ou Simple (CS) correspond à un seul capteur physique (one-to-one). Donc, pour chaque capteur physique  $p_i \in B$  un capteur virtuel correspondant  $v_i$  est créé.

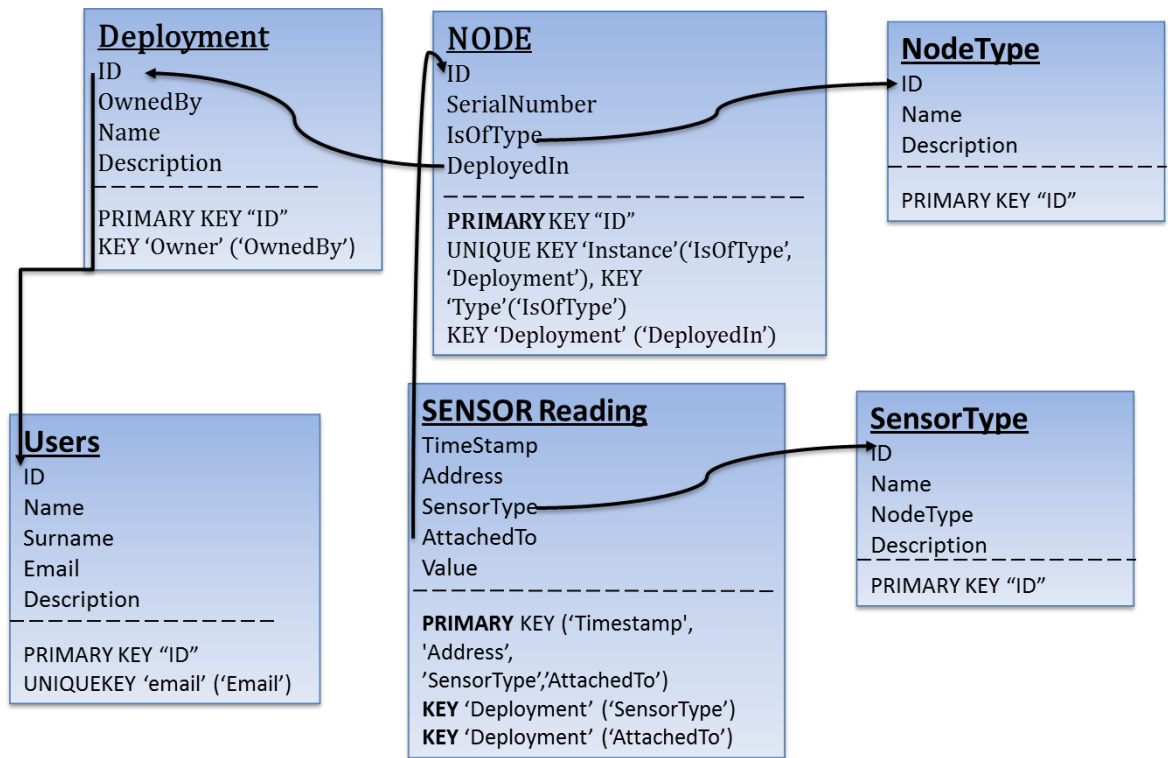
**② Sélectionneur** : Un groupe de capteurs physiques homogènes est représenté par un capteur un seul capteur virtuel  $v_i$ . ce type correspond à (Plusieurs à un) ou many-to-one.

③ **L'accumulateur** d'un faisceau  $B$  comprenant des capteurs homogènes ou hétérogènes peut être représenté sous la forme d'un capteur virtuel.

$v_i$ . Par exemple, dans le cas d'un réseau de capteurs corporels, tous les capteurs physiques peuvent être considérés comme le groupe  $B$  et peuvent être représentés comme un seul VS.

④ Une fonction générique définie pour fonctionner sur des types de données d'entrée spécifiques (éventuellement hétérogènes) afin de calculer la mesure souhaitée. La fonction  $h(B)$  est appliquée pour les données provenant de tous les capteurs physiques :  $\forall p_i \in B$ , et le résultat est représenté en tant que capteur virtuel  $v_i$ .

⑤ Un **capteur virtuel**  $v_i$  est créé et affecté en fonction d'un capteur physique  $p_i$ , mais ce capteur virtuel ne devient actif que lorsque la valeur détectée provenant du capteur physique  $p_i$  qualifie une fonction qualificative  $f$ .



**Figure 26:** La modélisation de l'interaction des capteurs virtuels

Parmi les avantages de la virtualisation des réseaux de capteurs, on cite :

- **Le partage de ressources :** La virtualisation permet de représenter les ressources informatiques par l'abstraction en unités logiques, ce qui permet de les utiliser efficacement par plusieurs utilisateurs indépendants, y compris

plusieurs applications simultanées. Elle permet aussi de créer de nouvelles applications qui n'étaient pas envisagées au moment de la conception.

- **L'agrégation de données** : Plusieurs types de données peuvent être agrégés dans un seul capteur virtuel.
- **L'anonymat de source** : Les utilisateurs ne soucient pas l'endroit des capteurs.
- **Création à la demande** : La création et la destruction capteurs virtuels selon les demandes des utilisateurs.

### V.2.2 Les types des réseaux de capteurs virtuels

Dans la littérature il y a deux approches pour la virtualisation des RCSF, dans la première approche[69]: les applications partagent les données capturées dans un RCSF, dans ce cas la passerelle collecte toutes les données de RCSF et le faire partager parmi plusieurs utilisateurs. Les réseaux de capteurs sont immergés dans le Cloud Computing en envoyant toutes les données capturées à travers un gestionnaire de l'hôte (host manager) qui se trouve en dehors de RCSF, le gestionnaire de l'hôte collecte, agrège et permet aux plusieurs applications de les utiliser pour leurs propres besoins.

#### 1) La virtualisation au niveau du nœud

Comme présenté dans **Figure 27**, dans cette approche : on utilise la capacité individuelle des capteurs pour exécuter des tâches en concurrence et permettre aux applications de grouper les nœuds selon leurs besoins. La différence entre ces deux approches est que la première approche permet de partager les données des réseaux de capteurs entre plusieurs utilisateurs et la deuxième approche permet de partager plusieurs nœuds entre plusieurs applications.

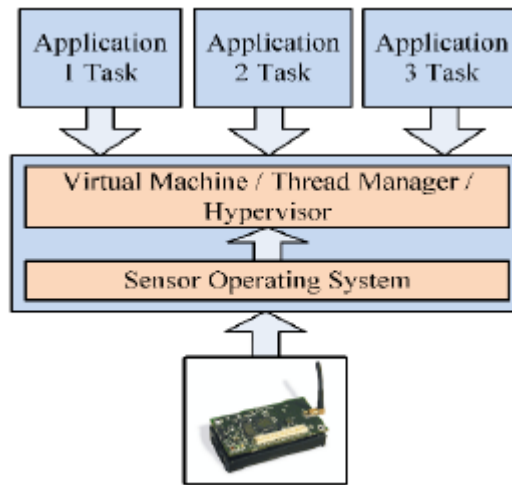


Figure 27 : La virtualisation au niveau du nœud

## 2) La virtualisation au niveau de réseau

Un sous-ensemble d'un RCSF forme un réseau virtuel pour exécuter une tâche d'une application donnée dans un temps spécifique. Tant que les autres nœuds restent disponibles pour les autres applications comme présenté dans **Figure 28**. La virtualisation au niveau de RCSF peut être réalisée en deux catégories : soit plusieurs RCV peuvent être créés à partir d'un seul RCSF ou un seul RCV (Réseau de capteur virtuel) est créé à partir de plusieurs RCSF.

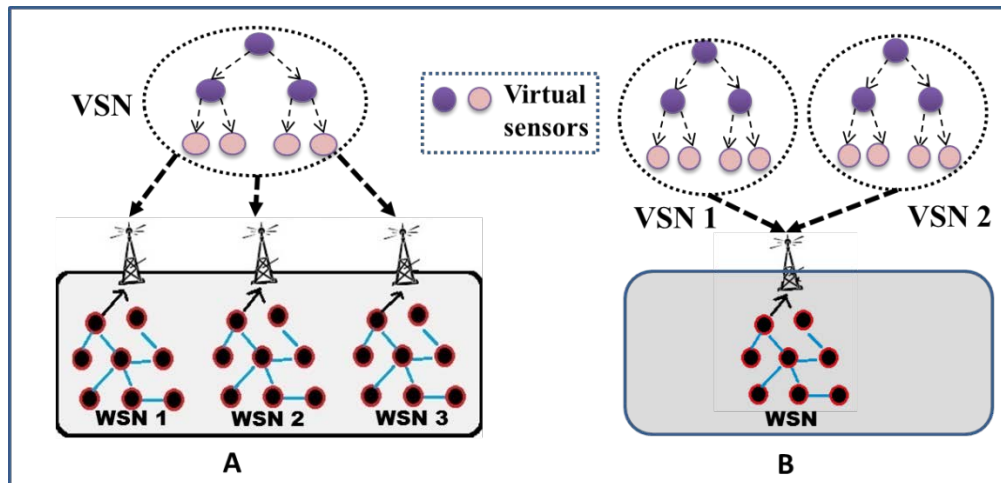


Figure 28 : La virtualisation au niveau de réseau

## V.3 L'architecture proposée

La **Figure 29** représente l'architecture proposée, cette architecture multicouche est conçue pour un réseau à grande échelle et toute nouvelle application ou service est déployé sur un réseau de capteurs physique sans fil, connecté à l'internet. Chaque

capteur virtuel exécute localement une approche de routage NDN[55] et interagit avec d'autres capteurs virtuels pour une application donnée et que le capteur virtuel peut (ou pas) appartenir à la même organisation du Cloud Computing différente. L'architecture se compose de quatre couches (Physique, Infrastructure de Cloud, Application) et huit modules (Répertoire de services, gestionnaire d'approvisionnement, gestionnaire de virtualisation, registre de service, gestionnaire du système, gestionnaire de ressources, gestionnaire de registre, gestionnaire de capteurs virtuels)

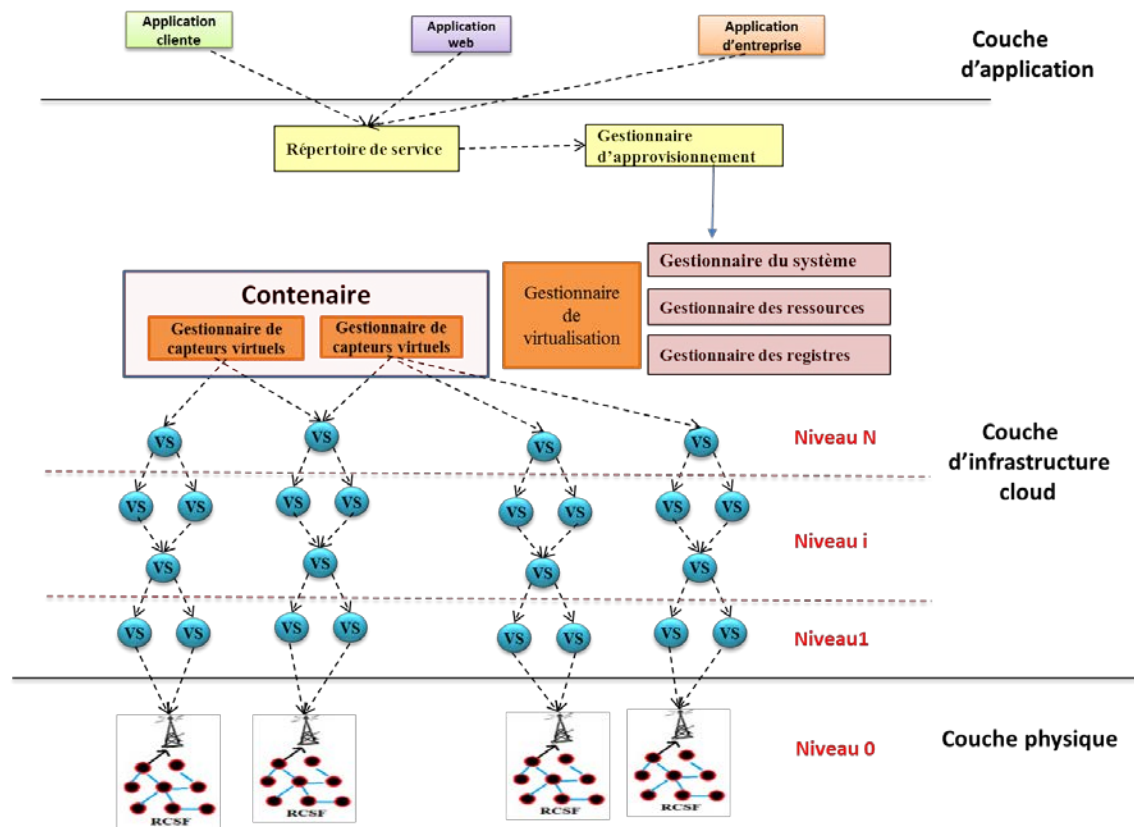


Figure 29 : L'architecture proposée

### V.3.1 La couche physique

Dans cette couche, nous avons des RCSF indépendants, les capteurs peuvent être de plusieurs types donc hétérogènes. Cependant, il y a des nœuds spéciaux dans chaque RCSF appelés 'station de base'. Le rôle de la station de base désignée est limité à la collecte des données et la distribution de requêtes, et ne comprend pas toute forme de surveillance. Généralement, l'internet lie l'utilisateur de réseau et la station de base, la communication peut être établie aussi par un satellite.

L'utilisateur peut envoyer des requêtes aux nœuds de réseau en fournissant le type de données requises et en récoltant les données environnementales captées. La couche physique représente le niveau zéro dans l'architecture.

### V.3.2 La couche infrastructure de Cloud Computing

Cette couche est constituée de la représentation logique de chaque capteur physique exécutant plusieurs tâches d'application simultanément. Chaque représentation logique est appelée '*un capteur virtuel*' dans notre architecture. Le capteur virtuel qui est une abstraction d'une tâche exécutée par un capteur physique. Il existe un mappage un-à-un entre une tâche d'application et l'application d'utilisateur final, ce qui signifie qu'une application ne peut pas avoir deux capteurs virtuels (deux tâches d'application différentes) sur un nœud physique.

Les capteurs virtuels peuvent être créés pour plusieurs fins. Comme mentionné précédemment, les capteurs physiques déployés ont des capacités limitées en termes de ressources de stockage et d'énergie. Ainsi, en créant des abstractions virtuelles de capteurs physiques dans un environnement de Cloud Computing et en leur permettant d'utiliser des ressources supplémentaires, il est fort possible d'améliorer les capacités de ces dispositifs par la combinaison de plusieurs flux de données provenant des nœuds physiques.

Cette couche est supposée comporter des milliers de capteur virtuels organisés en plusieurs niveaux, les capteurs virtuels communiquent entre eux à travers des liens logiques en exécutant l'approche NDN, chaque capteur virtuel contient trois tables consécutives. La première table c'est le dépôt de données (Content store) cette table contient toutes données sollicitées par l'utilisateur. Chaque fois le capteur virtuel reçoit un paquet de données il fait une mise à jour de la table de dépôt de données.

La deuxième table est la table de la file d'attente des requêtes qui ne sont pas encore satisfaites comme nous avons expliqué dans la section précédente. La troisième table est la table d'information de forwarding FIB (Forwarding Information Base) responsable de transférer la requête vers le meilleur saut. Généralement, il y a plusieurs stratégies de forwarding qui peuvent être implémentées dans cette table, comme la stratégie de meilleur chemin, la stratégie de l'inondation (flooding), des



stratégies de Forwarding basées sur la qualité de service, la stratégie de forwarding ou basé sur le cache de donnée...ETC. cette couche se compose des modules suivants :

- ***Le gestionnaire du système***

Ce module est responsable du traitement et de l'archivage des données reçues et gère également les ressources du système. Les cycles de calcul sont utilisés en interne pour traiter les données émanant des capteurs. Stocker les données du capteur aidera à analyser les modèles dans les données recueillies sur une période de temps.

- ***Le gestionnaire de registres***

Il conserve les informations d'identification des différentes applications des clients les différentes données de capteur requises. Pour chaque application, le composant de registre stocke les abonnements des utilisateurs, les données de capteur et les types d'événements de capteur qui intéressent l'application. Chaque application est associée à un ID d'application unique avec le contrat de niveau de service (SLA) ce dernier fournit une base pour le comptage et la comptabilité des services à utiliser, en couvrant tous les attributs du service.

- ***Gestionnaire de ressource***

Ce composant responsable de l'allocation de ressources informatiques pour gérer et maintenir les capteurs virtuels. Il alloue et libère les ressources selon les besoins des capteurs.

- ***Gestionnaire de virtualisation***

Ce module sert comme une interface entre l'environnement virtuel et le matériel physique sous-jacent, dans le but de simplifier l'administration des ressources, d'optimiser l'analyse des données et de rationaliser les opérations. De nombreux éditeurs de logiciel et fabricants de matériels prennent en charge ce type de logiciel, ce qui permet aux entreprises de déployer la solution de gestion qui leur convient le mieux.

- ***La structure d'un capteur virtuel***

La conception générale de la structure de capteur virtuelle est représentée sur la **Figure 30**. Les deux composants principaux sont le gestionnaire de capteur virtuel et le composant NDN.

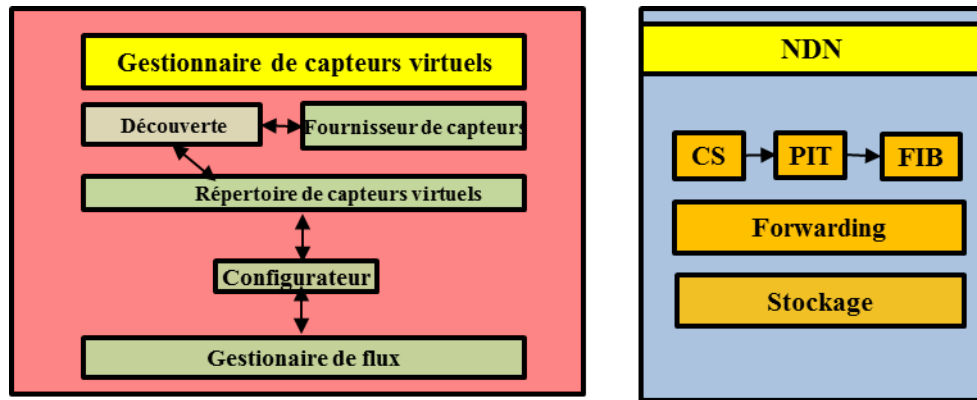


Figure 30: La structure d'un capteur virtuel

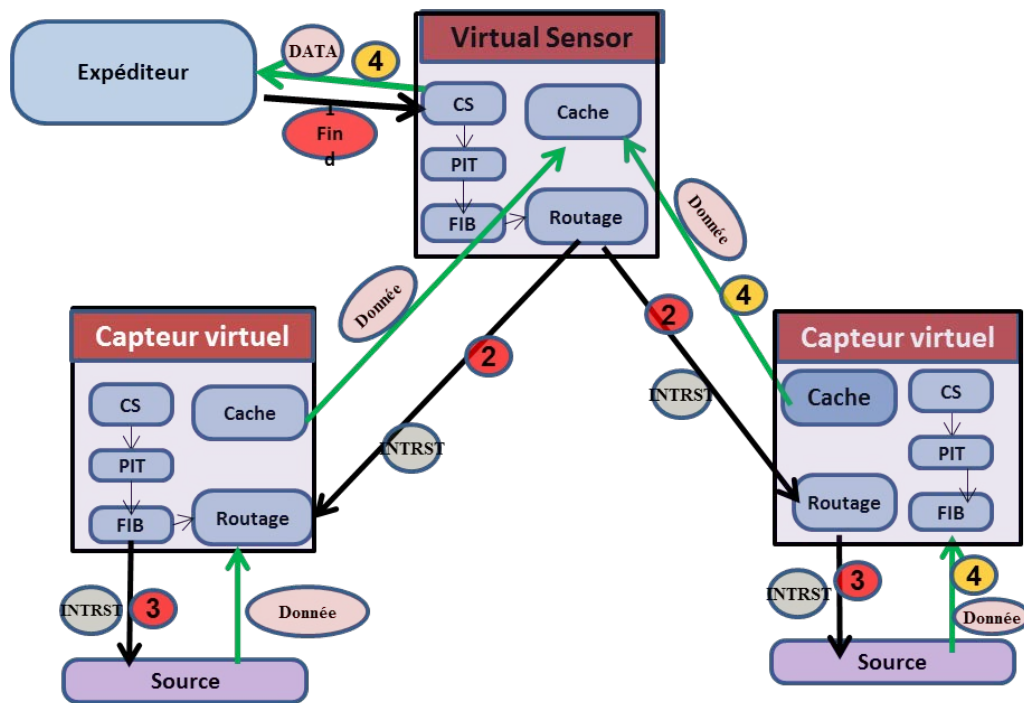
- **Le gestionnaire des capteurs virtuels** : Il est responsable de la gestion de tous les capteurs virtuels, Il reçoit les requêtes pour instancier des opérations telles que : démarrer, arrêter, supprimer et migrer des capteurs virtuels, toutes les missions à exécuter par les capteurs virtuels sont également diffusées via le gestionnaire des capteurs virtuels. Les différents composants du gestionnaire des capteurs virtuels sont des modules présentés dans le **Tableau 5**.

<b>Approvisionnement des capteurs virtuels</b>	Il décide quand créer, démarrer, arrêter ou supprimer les capteurs virtuels. Il communique avec le gestionnaire des capteurs virtuels
<b>Découverte des capteurs</b>	interagit avec le référentiel de capteurs virtuels pour rechercher le virtuel requis
<b>Répertoire des capteurs virtuels</b>	Contient toute la description virtuelle avec ses critères comme le type de capteur, l'emplacement et sa mesure
<b>Configurateur</b>	Il prépare la tâche pour vs en fonction de la demande reçue du fournisseur VS
<b>Gestionnaire des flux</b>	Il est responsable de la gestion des flux, de l'allocation des ressources et du partage des ressources entre eux.

Tableau 5: Les composants du gestionnaire des capteurs virtuels

Tous les capteurs virtuels utilisent le schéma de routage réseau de données nommées NDN pour récupérer et fournir des données, chaque vs est équipé de trois tables (CS, PIT, FIB) comme mentionné dans la section précédente. Notre architecture est basée sur deux technologies (la virtualisation et NDN), comme

présenté dans la **Figure 31**, d'abord la passerelle collecte les données provenant des capteurs physiques puis on les encapsule et on le transfert vers les capteurs virtuels. Les clients émettent des paquets d'intérêt contenant le nom du contenu souhaité. Le nom des paquets d'intérêt est organisé hiérarchiquement. Les paquets d'intérêt sont acheminés en fonction de leurs noms vers des sources de données potentielles en réponse, la source renvoie le paquet de données contenant et les données associées requises et a un nom de paquet d'intérêt identique. Puis, suivant l'approche NDN, le capteur virtuel mémorise les données localement pour une utilisation ultérieure si le même intérêt est demandé.



**Figure 31:** Anatomie de capteur virtuel

Le routage dans NDN se diffère du routage IP traditionnel de deux manières : (a) routage sur les noms : les producteurs de données enregistrent le nom des préfixes et non pas les préfixes d'adresse ; (b) Multi-chemins : le protocole de routage devrait fournir plusieurs chemins à chaque préfixe de nom (si de tels chemins existent). Les capteurs virtuels utilisent l'algorithme suivant pour acheminer le paquet d'intérêt et le paquet de donnée.

### V.3.3 La couche d'application

Cette couche traite des utilisateurs et de leurs applications pertinentes. Plusieurs utilisateurs veulent accéder aux précieuses données de capteurs provenant de la plate-forme indépendante pour une variété d'applications. Cette structure permet aux utilisateurs de différentes plates-formes d'accéder et d'utiliser les données du capteur sans rencontrer de problème en raison de la haute disponibilité de l'infrastructure et du stockage dans le Cloud Computing. La couche application consiste en une application de WSN. Bien qu'il existe plusieurs types de nœuds, chaque application n'a pas besoin de tous les utiliser. Chaque application utilise certains types de capteurs.

### V.4 La structure de données

Nous nous sommes focalisés sur les applications de RCSF basée sur un NDN dans laquelle plusieurs récepteurs agissent en tant que consommateurs de contenu qui envoient des intérêts périodiques pour interroger les informations connexes (par exemple, température et humidité) détectées par les capteurs déployés dans une zone géographique agissant en tant que fournisseurs de contenu dans des zones spécifiques. Les passerelles sont des périphériques puissants, mais les nœuds capteurs sont des dispositifs limités en termes de ressources, notamment l'énergie, le calcul et la mémoire.

#### V.4.1 Le système de nommage

Le système de nommage des données est l'une des technologies les plus importantes de l'architecture NDN, susceptible d'affecter la conception du forwarding et de routage des intérêts. Dans l'application concernée, il existe des trois attributs permettant de décrire un nom de contenu : le temps, l'espace et type.

Par conséquent, sur la base des principes de l'architecture NDN, nous proposons le système du nommage suivant pour que les données brutes, où nous avons attribué un préfix pour identifier des données provenant des RCSF :

**/RÉSEAU/TYPE/.../SITE/.../TEMPS/...**

(i) **TYPE** : type des données détectées par les nœuds de capteur déployés, par exemple, '/ TYPE / température', '/ TYPE / humidité'

(ii) **SITE** : Cet attribut sert à localiser le contenu souhaité.

Il peut être exprimé comme un numéro d'immeuble ou numéro d'une chambre par exemple ou on peut le configurer manuellement pour un objectif plus général, par exemple, nous pouvons diviser la zone en plusieurs sous-zones en fonction des

exigences de l'application. Et nous utilisons l'ID de sous-zone pour décrire l'attribut de l'emplacement, par exemple '/SITE/ZONE1'.

(iii) **TEMPS** : cet attribut décrit le temps de création des données sollicitées, on peut étendre cet attribut pour comprendre d'autres attributs pour plus de précision, par exemple : /TEMPS/HEURE/MINUTE/...

Dans notre conception, chaque routeur est nommé en fonction de réseau dans lequel il réside, du site auquel il appartient, ainsi que du nom attribué. Nous pouvons savoir par l'exemple présenté dans la **Figure 32** que si deux routeurs ont le même préfixe de nom / <réseau> dans le d'exemple FR/, ils appartiennent au même réseau et s'ils partagent le même préfixe /FR/HUM/ALS/, ils appartiennent au même site. Ce schéma de nommage facilite le filtrage des messages de routage erronés. Comme nous allons détailler par la suite, le processus NLSR sur un routeur est nommé d'après le nom du routeur : le nom du routeur est utilisé comme préfixe, suivi du nom du processus NLSR, à savoir, / <réseau> / <site> / <routeur> / NLSR. Ce nom de démon de routage NLSR est utilisé dans les messages d'information périodiques entre les routeurs NLSR adjacents pour détecter la défaillance des liaisons ou des processus de routage eux-mêmes/

Idéalement, toute mise à jour de routage créée par un processus NLSR devrait avoir le nom du processus comme préfixe pour identifier facilement l'expéditeur des messages. En d'autres termes, le nom d'un LSA doit commencer par / <réseau> / <site> / <routeur> / NLSR / LSA pour indiquer qu'il est généré par le processus NLSR.

En général, l'approche NDN utilise des différentes interfaces sur un nœud pour distinguer les flux de données des voisins en amont et les voisins en aval (upstream et downstream). Dans le NDN, un nœud peut transférer l'intérêt entrant aux voisins en amont en envoyant le paquet sur toutes les interfaces réseau, à l'exception de celle d'où l'intérêt arrive. Dans l'environnement de communication multi-sauts sans fil ; il existe une seule interface réseau sans fil au niveau d'un nœud, ce qui pose un défi pour la conception de la stratégie de transfert d'intérêt pour les NDN sans fil. C'est pour cela que nous avons adapté d'utiliser l'architecture NDN dans le niveau du middleware.

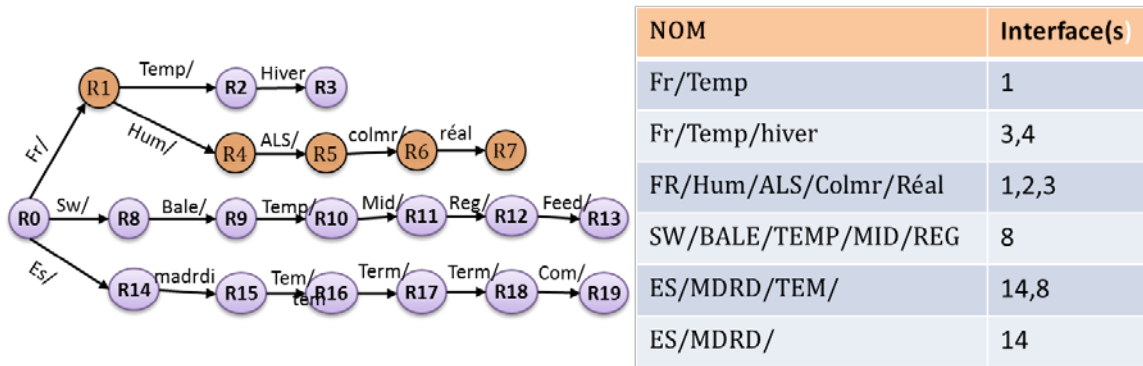


Figure 32 : Exemple simplifié de table FIB

Le schéma de nommage est l'un des éléments le plus important dans l'approche NDN, le schéma hiérarchique semble le mieux pour lier les relations entre les divers composants du système, et facilite ainsi l'identification des routeurs appartenant au même réseau, ainsi que des messages générés par le processus de routage. Le protocole de routage, route les messages à travers les nœuds d'un réseau en fonction du contenu des messages. Chaque nœud de réseau joue le rôle de routeur basé sur le contenu. Chaque consommateur et producteur de données est connecté à un nœud de réseau ; nous appelons de tels nœuds des nœuds consommateurs et producteurs. Pour simplifier la présentation, nous supposons que les nœuds de consommateur et de producteur sont distincts, c'est-à-dire qu'il est impossible de connecter directement un producteur et un consommateur au même nœud de routeur. Les nœuds qui n'ont ni consommateur ni producteur sont des nœuds internes.

Nous supposons que tous les routeurs connaissent leurs voisins, ainsi que les meilleurs chemins menant à chaque producteur de donnée. Nous supposons également que le nombre et l'emplacement des nœuds producteurs sont connus. En revanche, il n'est pas nécessaire que la population de consommateurs soit connue. Les nœuds communiquent avec leurs voisins via les protocoles de routage basé sur le contenu et nous supposons que les nœuds et les liaisons n'échouent pas. Chaque nœud a un ensemble de liens, ou interfaces, qui le connectent à ses voisins directs. Nous supposons qu'un nœud nommé  $S1$  couvre un autre nœud  $S2$  noté  $S2 \subseteq S1$  cela signifie que  $S2 \subseteq S1$  signifie  $\forall x \in S2 (x \in S1)$ . où  $x$  représente toutes les données nommées correspondant à  $S2$  et correspond également à  $S1$ .

#### V.4.2 La gestion des paquets

Le récepteur récupère périodiquement les données des capteurs déployés. La région surveillée peut être un bâtiment unique ou un quartier résidentiel d'une ville

intelligente où un système d'automatisation contrôle certains paramètres (par exemple, la température, l'humidité, la pollution, la consommation d'énergie) en diffusant périodiquement des informations. La collecte de données peut servir à établir des statistiques et à prédire des événements et laisser le système prendre la décision. La communication est basée sur la diffusion des intérêts et des données qui suit la structure NDN avec des schémas de nommage et de transfert.

Pour garantir la mise à jour des paquets d'intérêts nous avons suivi le modèle présenté dans [70] où nous considérons un réseau NDN qui consiste d'un ensemble de  $(N)$  nœuds ; où  $N = \{1, 2, 3, \dots, n, \dots, N\}$ , pour  $n \in N$  avec un taux de transmission des intérêts égal à l'instant  $(t)$  par seconde. De plus, nous considérons que ce réseau contient un autre ensemble de liens  $L$  où  $L = \{1, 2, 3, \dots, l, \dots, L\}$  pour  $\forall l \in L$ . Soit  $(P)$  le paquet d'intérêt, chaque paquet  $(P)$  a une durée de Vie  $(LT)$  définie par le consommateur qui est noté  $C_{LT}$ . Soit  $m_n = [m_1(t), m_1(t), \dots, m_n(t)]^t$  un vecteur de taux de tous les intérêts des nœuds. Dans ce cas, chaque  $n$  utilise un ensemble  $L_n \subseteq L$  de tous les liens. Par conséquent, l'ensemble  $N$  qui utilise le lien  $l$  est représenté par  $N_l = \{n \in N, l \in L_n\}$ . En outre, nous considérons  $PIT_{Size}$  autant que la capacité maximale de PIT qui est comme suit :  $LT_{Size} = \{1, 2, 3, \dots, P_{LT}, \dots, LT_{Size}\}$ .

Dans le cas où le PIT ne débordera pas, soit :  $LT_{threshold}$  peut être représenté comme une durée de vie d'un intérêt dans une table PIT, qui se présente comme suit:

$$P_{LT} = MAX(LT_{Threshold}, P_{LT}) \quad (4)$$

où  $P_{LT}$  est la durée de vie d'un paquet d'intérêt entrant, Puisque  $LT_{threshold}$  est inférieur à la durée de vie  $P_{LT}$ , alors  $LT_{threshold}$  sera mis à jour dynamiquement. Dans le cas où la table PIT débordera, on peut calculer la durée de vie moyenne de tous les intérêts pour l'interface  $(f)$  (lorsque le taux d'intérêt des liens varie) ou de tous les intérêts dans la table PIT lorsque les taux d'intérêt des liens sont similaires, et on soustrait l'ancien seuil basé sur les équations suivantes :

En cas de taux d'intérêt différents sur les liens, pour déterminer les interfaces  $(f)$  ayant le taux d'intérêt maximal :

$$f_{max} = MAX(m_i(t), m_{i+n}(t)) \quad (5)$$

Soit  $LT_{Threshold}(f)$  la durée de vie moyenne que le PIT a reçue sur l'interface  $(f)$  :

$$LT_{Threshold}(f) = \frac{\sum_P P_{LT}(t)}{W} \quad (6)$$

où ( $W$ ) représente le nombre total de paquets d'intérêt de l'interface( $f$ ) ;  
 et  $P = 1, 2, \dots, W$ . Alors, dans le cas où le taux d'intérêt est similaire sur tous liens,  
 soit  $LT_{Threshold}(PIT)$  la durée de vie moyenne reçue par le PIT qui peut être  
 calculé comme suit :

$$LT_{Threshold}(PIT) = \frac{\sum_P P_{LT}(t)}{W} \quad (7)$$

Où ( $W$ ) est le nombre total de paquets d'intérêt reçus par PIT alors que  
 $P = 1, 2, \dots, W$

Par conséquent, si et seulement si l'une de ces équations est vraie (c'est-à-dire que  
 6 et 7 s'excluent mutuellement). En conséquence, la valeur moyenne de la durée de  
 vie est la valeur attendue de la variable aléatoire de la durée de vie est calculée  
 comme suit :

$$LT_{Threshold} = \left[ \frac{\sum_P P_{LT}(t)}{W} \right]^{f \text{ ou } PIT} \quad (8)$$

Où  $W$  est le nombre total de paquets d'intérêts ( $f/PIT$ ).  $[X]^{f \text{ ou } PIT}$  Représente la  
 mise à jour de la durée de vie du paquet d'intérêts entrant pour l'interface  $f$  ou  
 PIT. En cas de surcharge de la table (PIT), la valeur de la durée de vie d'un  
 paquet d'intérêts entrant est mise à jour en fonction de l'équation (9).

$$P_{LT} = MAX(LT_{Threshold}(f|PIT), P_{LT}) \quad (9)$$

Dans notre scénario, les différents capteurs sans fil sont déployés dans une zone  
 spatiale. Les utilisateurs envoient les requêtes vers la plateforme sans se préoccuper  
 de l'emplacement ou la spécification des capteurs physiques. Chaque capteur virtuel  
 est créé à partir d'un ou plusieurs capteurs physiques qui dépendent de la zone  
 d'application de l'utilisateur. Le groupe de capteurs est créé à partir d'un ou de  
 plusieurs capteurs virtuels. Les utilisateurs peuvent créer des groupes de capteurs  
 virtuels et utiliser librement les capteurs virtuels inclus dans les groupes comme s'ils  
 possédaient des capteurs. Par exemple, ils peuvent activer ou désactiver leurs  
 capteurs virtuels, vérifier leur statut et définir la fréquence de collecte des données.  
 Si plusieurs utilisateurs contrôlent librement les capteurs physiques, certaines  
 commandes incohérentes peuvent être émises. Les utilisateurs peuvent contrôler  
 librement leurs propres capteurs virtuels en virtualisant les capteurs physiques en  
 tant que capteurs virtuels



Comme nous avons déjà détaillé le capteur virtuel dans notre architecture se compose des modules de NDN alors quand le capteur virtuel reçoit un paquet d'intérêt et il exécute l'algorithme présenté dans la **Figure 33**.

---

Algorithme : Gestion du paquet d'intérêt

---

**Input :**

InterstPKT.Name: interestname; //Le nom du paquet d'intérêt

CS: Content Store; // La table de contenu

PIT: Pending Interest Table; // La table d'intérêts en attente

FIB: Forwarding Interest Table; // La table de forwarding

data lifetime: Content lifetime ; // la durée de vie du paquet

$LT_{Threshold}=0$ ; // La seuil de la durée de vie

$P_{LT}=VALUE$ ; // La durée de vie de paquet

$F=VALUE$ ;

$NAME=$  InterstPKT.Name;

**Begin**

1: Lookup(NAME) in CS

2: **IF** Lookup(NAME) = TRUE **AND**  $P_{LT} \leq LT_{Threshold}$

3: | Return(DataPKT(DATA))

4: **ELSE**{

5:     **IF**  $P_{LT} > LT_{Threshold}$  **T HEN REMOVE** DATAPKT **FROM** CS

6:     LOOKUP(NAME) in PIT

5: **IF**{LOOKUP(NAME) =TRUE

7:    ADD InterstPKT.Interface in PIT } // Ajouter l'interface du paquet à la table PIT

8:    **ELSE**{

9:    CREATEInterstPKT(Name, Interface) in PIT // Ajouter le nom et l'interface du paquet au PIT

10:     LOOKUP(InterstPKT(Name) in FIB

11:     **IF** {LOOKUP(InterstPKT(Name))=TRUE

12:         FORWARD((InterstPKT(Name))}

13:     **ELSE** BROADCAST ((InterstPKT(name))}

14:     **ENDIF**

15:     **ENDIF**

16: **ENDIF**

17: **End Fetch()**

---

**Figure 33:** Algorithme de gestion de paquet d'intérêt

Quand le capteur virtuel reçoit le paquet de donnée ; le capteur virtuel exécute l'algorithme présenté dans la **Figure 34**. Dans cette architecture, les capteurs physiques sont connectés aux passerelles via une interface commune de différentes manières (série, USB et Ethernet), les nœuds acheminent les données collectées jusqu'à la passerelle, puis la passerelle convertit les données brutes en paquet ta et les pousse vers les capteurs virtuels, la communication se fait de deux manières : en

aval (i.e. : du capteur virtuel vers les passerelles) et en amont (i.e. : de la passerelle vers les capteurs virtuels).

Lorsque des paquets d'intérêt arrivent au capteur virtuel, ce dernier vérifie s'il existe déjà un paquet de données dans la table de contenu (*content store*) qui correspond à l'intérêt et il sera renvoyé sur l'interface où l'intérêt est arrivé et l'intérêt sera supprimé. Donc, une fois que le paquet d'intérêt arrive à l'interface du capteur virtuel ; Une recherche de correspondance (*longest match*) est exécutée sur le nom du contenu (*ContentName*) dans le magasin du contenu. Si une correspondance est trouvée alors les données seront renvoyées au demandeur, les données '*satisfont*' un intérêt si le nom du contenu (*ContentName*) dans le paquet d'intérêt est un préfixe du nom du contenu (*ContentName*) dans le paquet de données.

Sinon, une autre recherche est effectuée dans PIT s'il y a une entrée de correspondance exacte ; l'interface d'arrivée de l'intérêt sera ajoutée à PIT et les intérêts seront mis de côté. S'il n'y a pas de correspondance dans les entrées PIT, le paquet d'intérêt sera ajouté avec son interface d'arrivée. Et l'intérêt sera transmis à la source de données potentielle. Lorsque les paquets de données arrivent au capteur virtuel, ils seront simplement ajoutés au contenu du magasin pour plus d'intérêt et l'intérêt sera écarté du PIT. Si une correspondance est trouvée dans Content Store, les données seront renvoyées au demandeur, les données "satisfont" un intérêt si le nom du contenu dans le paquet d'intérêt est un préfixe du nom du contenu dans le paquet data. Sinon, une autre recherche est effectuée dans PIT s'il y a une entrée de correspondance exacte ; l'interface d'arrivée de l'intérêt sera ajoutée au PIT et les intérêts seront mis de côté.

---

**ALGORITHM 2: fetch(DataPKT)**

---

```

1: BEGIN
2: Lookup(DataPKT(Name)) in PIT
3:   If LOOKUP(DataPKT(Name)) =TRUE
4:     {FORWARD(DataPKT(Name, Interface))
5:     ADD DataPKT(Name, Data) in CS
6:     REMOVE InterstPKT(Name) in PIT}
7:   ELSE ADD (DataPKT(Name) in CS
8:   END
9: END

```

---

**Figure 34:** Algorithmme de la gestion de paquet de données

S'il n'y a pas de correspondance dans les entrées PIT, le paquet d'intérêt sera ajouté avec son interface d'arrivée, et l'intérêt sera transmis à la source de données potentielle. Lorsque les paquets de données arrivent au capteur virtuel, ils seront simplement ajoutés au contenu du magasin pour plus d'intérêt et l'intérêt sera écrasé du PIT. Les algorithmes dans la **Figure 33** et la **Figure 34** sont différents de l'algorithme original présenté dans [64], dans notre algorithme mémorise les données même si aucune correspondance trouvée dans la table PIT pour l'utilisation ultérieure des données, ce qui est différent de l'approche NDN qui élimine le paquet de données si aucune correspondance n'a été trouvée dans la table PIT.

Les différents types de capteurs physiques ont des fonctions différentes en termes de détection de l'environnement. Chaque capteur physique fournit ses propres fonctions de contrôle et de collecte de données. L'infrastructure traduit la fonction standard des capteurs virtuels en fonctions spécifiques pour les différents types de capteurs physiques. L'automatisation (en termes de réponse des données) améliore le délai de livraison du service et réduit le coût. L'infrastructure prépare des modèles pour les spécifications de différents types de capteurs physiques. Lorsque les utilisateurs sélectionnent le modèle d'un capteur virtuel ou d'un groupe de capteurs virtuels, l'infrastructure Sensor-Cloud fournit de manière dynamique et automatique les capteurs virtuels dans ce groupe de capteurs virtuels à partir des modèles. L'infrastructure Sensor-Cloud est une prestation de services à la demande et prend en charge le cycle de vie complet de la prestation de services depuis l'enregistrement des capteurs physiques

L'utilisateur peut contrôler ses capteurs virtuels directement ou via un navigateur Web. L'utilisateur peut surveiller l'état des capteurs virtuels. Quand ils deviennent inutiles, l'utilisateur peut les libérer. Les utilisateurs finaux peuvent utiliser les capteurs virtuels en payant pour l'utilisation et sans connaissances détaillées sur les capteurs physiques. Un utilisateur final est un acteur avec une ou plusieurs applications ou services qui utilisent les données du capteur. Un utilisateur final demande l'utilisation de capteurs virtuels ou de groupes de capteurs virtuels répondant aux exigences des modèles. Ces modèles sont facilement configurables, partageables et amovibles et peuvent facilement être créés.

## **V.5 Conclusion**

Dans ce chapitre, nous avons proposé une nouvelle architecture pour l'intégration des RCSF avec le Cloud Computing en utilisant l'architecture Named Data Networking et les réseaux de capteurs virtuels. Ainsi, nous avons détaillé les différents composants de l'architecture proposée. Le prochain chapitre fera l'objet d'une évaluation de performance de la proposition.

## Sommaire

---

<b>VI Simulation et évaluation .....</b>	<b>105</b>
VI.1 Introduction .....	106
VI.2 Environnement de simulation .....	106
VI.2.1 Le simulateur NDN-Sim .....	106
VI.2.2 La plateforme GSN .....	107
VI.3 Le scénario de la simulation.....	113
VI.4 Les métriques .....	114
VI.5 Les paramètres de simulation .....	116
VI.6 Résultats de simulation.....	118
VI.7 Conclusion.....	122

---

## VI.1 Introduction

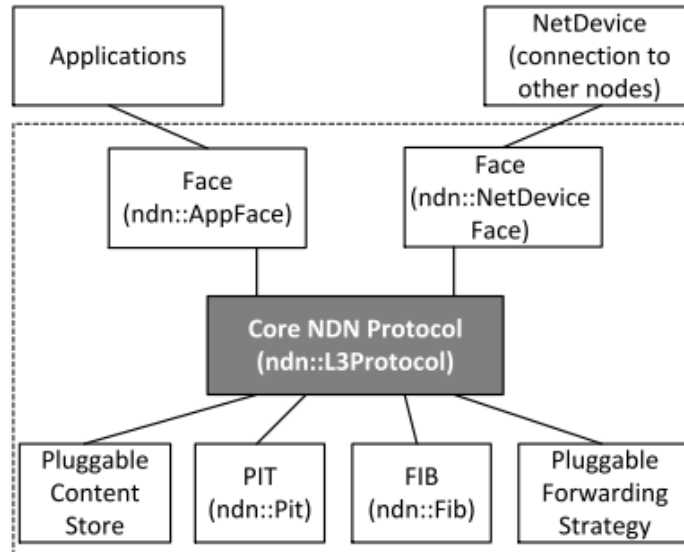
Cette section est dédiée à la discussion des différents résultats obtenus après la simulation de notre architecture proposée. L'évaluation des performances sera conduite à travers le Framework GSN[71] qui est l'un des logiciels leaders dans le domaine des réseaux de capteurs virtuels et le simulateur à évènement discret ndnSIM [72]. Nous divisons notre étude de performance en deux parties, dont la première est destinée à évaluer les réseaux capteurs virtuels (RCV) tandis que la deuxième se focalise sur l'évaluation des réseaux de capteurs virtuels en utilisant l'architecture NDN.

## VI.2 Environnement de simulation

Afin de mesurer les performances de l'architecture proposée, nous avons simulé les mesures des deux grandeurs physiques : la température et l'humidité. Dans cette structure, les propriétaires des réseaux déploient des détecteurs pour mesurer la température et l'humidité en même temps. Ces capteurs exécutent plusieurs tâches simultanément, en utilisant des capteurs virtuels. Les données brutes sont envoyées aux capteurs virtuels à travers une passerelle, le rôle principal des passerelles est de collecter, traiter et transmettre au serveur connecté à l'internet les données collectées des nœuds capteurs. Ils sont équipés d'une interface de communication sur le terrain pour les nœuds et d'un canal de communication externe pour les serveurs. Nous avons installé le système d'exploitation Ubuntu[73] dans trois machines. Ensuite, nous avons installé 50 machines virtuelles (VMs) pour implémenter une plateforme du Cloud privé en utilisant le framework vagrant[74] version 2.2.3

### VI.2.1 Le simulateur NDN-Sim

Le simulateur ndnSIM [75] est un simulateur de NDN libre basé sur le simulateur NS-3[76]. Le ndnSIM est devenu une plate-forme populaire utilisée par des centaines de chercheurs du monde entier. Le simulateur est implémenté de manière modulaire en langage C++ pour modéliser le comportement de chaque entité de réseau dans le NDN. Cette structure modulaire permet à tout composant d'être facilement modifié ou remplacé avec un impact nul ou minime sur les autres composants. Le simulateur ndnSIM inclut un certain nombre d'applications pour générer du trafic pour simplifier la création des scénarios de simulation et des outils permettant de rassembler des statistiques de simulation pour évaluer la performance de réseau. La **Figure 35** récapitule les abstractions des composants qui ont été implémentés dans ndnSIM et les interactions entre ces composants.



**Figure 35:** Schéma fonctionnel des composants de ndnSIM[72]

Chaque composant, à l'exception du noyau 'ndn::L3Protocol' a un certain nombre d'implémentations alternatives qui peuvent être arbitrairement choisies par le scénario de simulation à l'aide de classes auxiliaires.

### VI.2.2 La plateforme GSN

c Ensuite nous avons implémenté le schéma de routage proposé par le framework NDN pour acheminer les requêtes et les données dans le réseau virtuel. Nous l'avons comparé avec la technique par défaut utilisée dans NDN qui repose sur l'inondation de réseau ("flooding"). Cette méthode consiste à inonder le réseau des requêtes Interest en les diffusant sur toutes les interfaces des nœuds afin d'atteindre le contenu désiré.

GSN (Global Sensor Networks ) [71]est un middleware conçu pour faciliter le déploiement et la programmation des réseaux de capteurs qui fournit une infrastructure évolutive et flexible pour intégrer des technologies de réseau de capteurs hétérogènes en utilisant un ensemble d'abstractions puissantes[77]. La plateforme GSN est conçue pour traiter le problème principal de la technologie des réseaux de capteurs actuels qui est l'hétérogénéité des logiciels et des matérielles disponibles qui rend le déploiement et le développement d'applications fastidieux. Pour minimiser l'implémentation inutile et répétitive de fonctionnalités identiques pour différentes les plates-formes, la plateforme GSN prend en charge l'intégration et la découverte les réseaux de capteurs et les données des capteurs, et elle permet de déployer rapidement et d'ajouter de nouvelles plates-formes, filtrer et combiner les données des capteurs, et prend en charge

l'adaptation dynamique de la configuration du système pendant le fonctionnement. La plateforme GSN est conçue en se basant sur les quatre objectifs fondamentaux :

- **La simplicité.** Le but était de concevoir un système sur la base d'un ensemble d'abstractions puissantes, facilement configurables et adaptées aux besoins de l'utilisateur. Il est possible de définir les réseaux de capteurs et les flux de données de manière déclarative en utilisant SQL comme langage de manipulation de données, et le langage XML en tant que framework syntaxique pour la configuration du système.
- **L'adaptabilité :** L'ajout de nouveaux types de réseaux de capteurs et la reconfiguration dynamique des sources de données peuvent être pris en charge pendant l'exécution sans avoir besoin d'interrompre le fonctionnement du système en cours.
- **Le passage à l'échelle :** dans le cas d'un très grand nombre de producteurs de données et de consommateurs avec de différentes applications, la plateforme GSN tient en compte des problèmes d'évolutivité spécifiques au traitement des requêtes distribuées et à la découverte distribuée des réseaux de capteurs.

La plateforme GSN fournit une vision logique sur les réseaux de capteurs basés sur l'abstraction du capteur virtuel. Les capteurs virtuels s'abstiennent des détails d'implémentation pour accéder aux données de capteur et aux données de capteur de modèle sous forme de flux temporels de données relationnelles. Les capteurs virtuels peuvent également représenter des vues dérivées sur les flux de données du capteur, résultant du post-traitement et de la combinaison de données provenant de différentes sources. Ceci est indiqué de manière informelle par le flux de données conceptuel de la Figure 2, qui illustre comment les flux de données provenant des réseaux de capteurs peuvent être connectés dans un réseau superposé homologue à homologue.

Les capteurs virtuels sont liés aux nœuds physiques de réseau en les déployant sur des conteneurs GSN hébergeant des capteurs virtuels. Cette approche soutient notre objectif de simplicité, car les utilisateurs possédant une seule abstraction peuvent travailler avec des données de capteurs provenant de sources de capteurs hétérogènes et de toute donnée dérivée de données de capteur brutes. Sur le plan architectural, GSN adopte une vue orientée service sur les réseaux de capteurs, comme le montre **la**



**Figure 36.** Ce paradigme architectural a déjà prouvé son applicabilité dans plusieurs autres domaines, par exemple les services Web.

Dans ce schéma, les réseaux de capteurs sont considérés comme des types abstraits de services effectuant une tâche de détection et fournissant un type spécifique de données. Les services de détection sont publiés via un répertoire peer-to-peer et recherchés en fonction de leurs propriétés. Les applications peuvent découvrir des réseaux de capteurs à l'aide du registre et accéder ensuite à des réseaux de capteurs en utilisant une interface d'accès aux données standard. L'utilisation du paradigme orienté service facilite l'objectif de conception adaptatif de GSN, car il permet l'utilisation à la demande et la combinaison de réseaux de capteurs

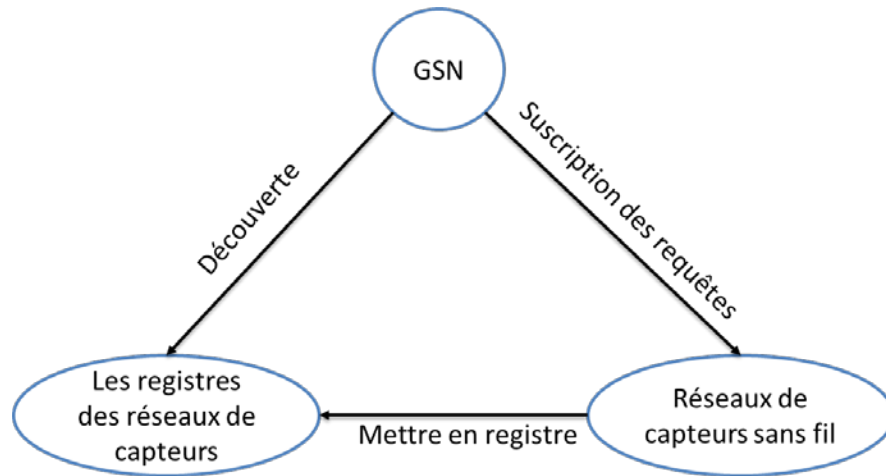
- **Modèles de réseau de capteurs**

Les architectures actuelles des réseaux de capteurs sans fil sont basées sur une approche physique. Des capteurs uniques ou des réseaux de capteurs entiers sont connectés via des connexions sans fil à un point d'accès. Les protocoles de routage ad hoc acheminent les données vers les puits qui communiquent avec un point d'accès connecté à l'internet et pouvant traiter et mettre à disposition les données reçues des capteurs. Les applications désirant utiliser des données des capteurs doivent d'abord identifier les points d'accès, éventuellement en passant par plusieurs couches, protocolaires puis suivre les protocoles d'accès spécifiques.

Le problème majeur de ce modèle est la dépendance de l'accès des données du capteur. Les capteurs pouvant être considérés comme des sources de données, GSN adopte comme principe fondamental le principe de l'indépendance des données. Cette abstraction est bien acceptée et a fait ses preuves dans la conception de systèmes de base de données. Il permet aux utilisateurs de GSN d'abrégier des infrastructures physiques hétérogènes avec les plates-formes actuelles de réseaux de capteurs, GSN fournit une vue logique des réseaux de capteurs basée sur une abstraction des capteurs virtuels.

Les capteurs virtuels résument les détails de la mise en œuvre pour accéder aux données des capteurs. Les capteurs virtuels peuvent représenter également des vues dérivées des flux de données de capteurs résultant du post-traitement et combinaison les données de capteurs provenant de différentes sources. Les capteurs virtuels sont liés aux nœuds physiques de réseau en les déployant dans des conteneurs GSN hébergeant des serveurs virtuels. Cette approche répond à notre objectif de simplicité, en tant

qu'utilisateur disposant d'un serveur unique l'abstraction peut fonctionner avec les données de capteurs de sources de capteurs hétérogènes et avec toutes les données dérivées des données brutes du capteur.



**Figure 36:** Le modèle global de GSN

Sur le plan architectural, GSN adopte une vue des réseaux de capteurs orientés service. Ce paradigme architectural a déjà prouvé son applicabilité dans plusieurs autres domaines, par exemple les services Web. Dans cette perspective, les réseaux de capteurs sont considérés comme des types abstraits de services effectuant une tâche de détection et fournissant un type spécifique de données. Les services de capteur sont publiés via un répertoire d'égal à égal et analysés en fonction de leurs propriétés. Les applications peuvent découvrir des réseaux de capteurs à l'aide du registre et accéder ensuite à des réseaux de capteurs à l'aide d'une interface d'accès aux données standard. L'utilisation du paradigme axé sur les services facilite la conception de GSN en matière d'adaptivité, car elle permet une utilisation à la demande et une combinaison de réseaux de capteurs.

- **La spécification des capteurs virtuels**

L'abstraction de clé dans GSN est le capteur virtuel (VS) (voir **Figure 37**). Les capteurs virtuels extraient des détails de mise en œuvre de l'accès aux données des capteurs. Ce sont les services fournis et gérés par GSN. Un capteur virtuel correspond soit à un flux de données reçu directement de capteurs, soit à un flux de données dérivé d'autres capteurs virtuels. La spécification d'un capteur virtuel fournit toutes les informations nécessaires au déploiement et à l'utilisation, notamment :

- Des métadonnées utilisées pour l'identification et la découverte
- La structure des flux de données que le capteur virtuel consomme et produit

- Une spécification déclarative basée sur SQL du traitement de flux de données effectué dans un capteur virtuel
- Les propriétés fonctionnelles liées à la persistance, au traitement des erreurs, à la gestion du cycle de vie et déploiement physique

Pour prendre en charge le déploiement rapide, ces propriétés de capteurs virtuels sont fournies dans un descripteur de déploiement déclaratif, comme illustré à la **Figure 37**. Pour utiliser les flux de données produits par un capteur virtuel, nous utilisons un adressage logique. Chaque capteur virtuel peut être équipé d'un ensemble de paires clé-valeur pouvant être enregistrées et découvertes dans GSN. L'exemple de spécification de **Figure 37** montre un capteur virtuel avec trois flux d'entrée. Les capteurs virtuels générant les flux d'entrée sont identifiés par leurs métadonnées (lignes 16 à 17, 23 à 24 et 29 à 30) et les métadonnées permettant d'identifier le capteur virtuel produisant le flux de sortie sont indiquées aux lignes 2 à 5. Cet exemple montre également la capacité de GSN d'accéder simultanément à plusieurs producteurs de flux.

Dans GSN, les flux de données sont des séquences temporelles de N uplets relationnels. Ceci est conforme au modèle utilisé dans la plupart des systèmes de traitement de flux. La structure du flux de données produit par un capteur virtuel est codée en XML, comme indiqué à la ligne 7 -10. La structure des flux d'entrée est reconnue à partir des spécifications respectives de leurs définitions de capteurs virtuels. Le traitement de flux de données est divisé en deux étapes : (1) traitement appliqué aux flux d'entrée (lignes 19, 25 et 32) et (2) traitement permettant de combiner les données des différents flux d'entrée et de produire le flux de sortie (lignes 34 et 37).

Pour spécifier le traitement des flux d'entrée, nous utilisons des requêtes SQL faisant référence aux flux d'entrée à l'aide du mot clé réservé WRAPPER. L'attribut wrapper = "remote" indique que le flux de données est obtenu à partir d'Internet via GSN. Ainsi, l'adressage logique pourrait être utilisé pour adresser le capteur virtuel. Dans l'exemple donné, le flux de sortie joint les données reçues de deux capteurs de température et renvoie une image de caméra si certaines conditions de température sont satisfaites. La spécification de la structure du flux de sortie concerne directement le traitement du flux de données effectué par le capteur virtuel et doit être compatible avec elle. Lors de la conception des spécifications GSN, nous avons décidé de séparer

les aspects temporels du traitement de données relationnel à l'aide de SQL. Ainsi, le SQL standard est utilisé pour les différentes étapes du traitement.

Le traitement temporel est contrôlé par divers attributs fournis dans les flux d'entrée et de sortie, par exemple l'attribut storage-size. En raison de son importance spécifique, nous traiterons du traitement temporel en détail dans une section ultérieure.

```

1 <virtual-sensor name="room-monitor" priority="11">
2   <addressing>
3     <predicate key="geographical">BC143</predicate>
4     <predicate key="usage">room monitoring</predicate>
5   </addressing>
6   <life-cycle pool-size="10" />
7   <output-structure>
8     <field name="image" type="binary:jpeg" />
9     <field name="temp" type="int" />
10  </output-structure>
11  <storage permanent="true" history-size="10h" />
12  <streams>
13    <stream name="cam">
14      <source alias="cam" storage-size="1" disconnect-buffer-size="10">
15        <address wrapper="remote">
16          <predicate key="geographical">BC143</predicate>
17          <predicate key="type">Camera</predicate>
18        </address>
19        <query>select * from WRAPPER</query>
20      </source>
21      <source alias="temperature1" storage-size="1m" disconnect-buffer-size="10">
22        <address wrapper="remote">
23          <predicate key="type">temperature</predicate>
24          <predicate key="geographical">BC143-N</predicate>
25        </address> <query>select AVG(temp1) as T1 from WRAPPER</query>
26      </source>
27      <source alias="temperature2" storage-size="1m" disconnect-buffer-size="10">
28        <address wrapper="remote">
29          <predicate key="type">temperature</predicate>
30          <predicate key="geographical">BC143-S</predicate>
31        </address>
32        <query>select AVG(temp2) as T2 from WRAPPER</query>
33      </source>
34      <query>
35        select cam.picture as image, temperature.T1 as temp from cam, temperature1
36        where temperature1.T1 > 30 AND temperature1.T1 = temperature2.T2
37      </query>
38    </stream>
39  </streams>
40 </virtual-sensor>

```

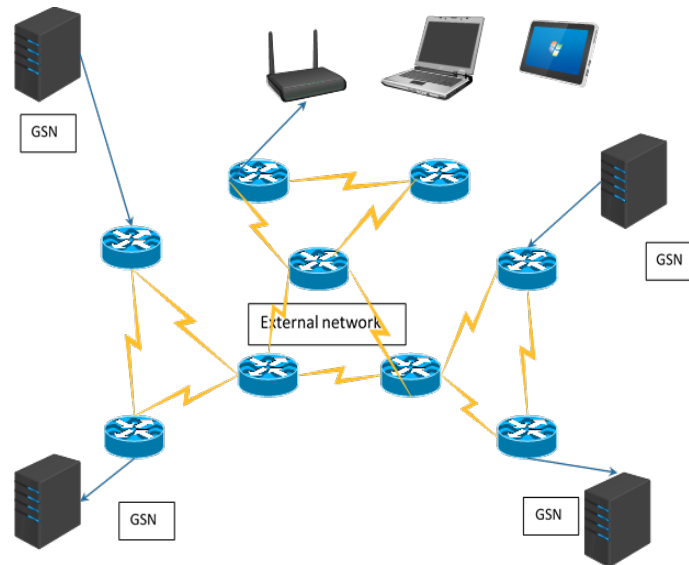
**Figure 37:** Un capteur virtuel complexe dans GSN

Pour évaluer notre proposition en utilisant une architecture NDN pour les RCSFs, nous avons utilisé le simulateur de réseau 3 (GSN) [78], le module open source ndnSIM[75] conçu pour prendre en charge les réseaux NDN. Nous avons utilisé une zone de 400 x 400 m avec 80 capteurs déployés dans la topologie de réseau et du nœud collecteur au milieu de la topologie, chaque nœud déployé utilise l'interface IEEE 802.11g. La couverture de chaque nœud est d'environ 80 m et la distance entre les

capteurs adjacents est de 50 m. dans ce cas, le nombre de liens établis entre les nœuds peut être de huit. Nous considérons que les intérêts sont diffusés par le récepteur pour une tâche spécifique (afin que les paquets de données puissent être extraits) à des créneaux horaires fixes. Dans la simulation, nous supposons un intervalle de surveillance de 60 secondes. Nous considérons également que le collecteur envoie 4 intérêts différents par tâche de surveillance, dont le nombre varie de 1 à 16. Dans ce cas, si les données surveillées sont uniques (par exemple température / région / 143), le nœud du récepteur envoie le message. Le premier intérêt à  $t = 0$ , puis deuxième intérêt à  $t = 60$ , et ainsi de suite

### VI.3 Scénario de la simulation

Nous avons comparé notre architecture proposée avec différents protocoles Internet internes et externes basés sur IP. La **Figure 38** présente notre scénario de la simulation pour le réseau IP et le réseau NDN. Les deux exploitent le middleware GSN pour gérer le réseau de capteurs virtuels. Dans ce scénario, des routeurs internes exécutant l'approche NDN. Nous installons un protocole interne pour connecter les nœuds GSN aux routeurs externes situés dans le réseau Internet pour le réseau IP. Concernant la plateforme NDN ; tous les routeurs jouent le même rôle.



**Figure 38:** Scénario de simulation

Nous avons utilisé le protocole OSPF [79] et le protocole RIP [80] pour le routage interne et le protocole BGP [81] pour le routage externe. L'évaluation est basée sur la simulation à l'aide du simulateur NS3[76]. Pour notre architecture proposée, nous avons utilisé le protocole NLSR[82]. En suivant la même stratégie de combinaison des

protocoles réseau internes avec les protocoles externes mentionnés dans [82], un protocole de routage utilisé pour définir le système contenu dans AS (*Système autonome*). Ce protocole qui interagit librement avec son système et échange mutuellement les informations de routage au sein d'un AS (*système autonome*). En ce qui concerne le protocole de routage utilisé pour connecter l'AS dans les grands réseaux, appelé EGP (*Exterior Gateway Protocol*) [83]. Ce protocole reconnaît l'autre AS en tant que AS voisin et n'est que le minimum d'échanges nécessaire pour suivre la capacité d'information [84].

Certains protocoles de routage dynamiques peuvent être utilisés dans un réseau, tels que les protocoles OSPF (*Open Short Path First*), le protocole RIP (*Routing Information Protocol*), le protocole inter-système inter-système (IS-IS) et le protocole de routage de passerelle intérieure améliorée (EIGRP).). Chaque protocole de routage a ses propres avantages et inconvénients. La détermination et la sélection des protocoles de routage dépendent de plusieurs paramètres qui affectent la qualité d'un réseau [75]. Le scénario consiste en deux protocoles de routage combinés, à savoir RIPv2, OSPF, EIGRP, RIPv2-BGP, OSPF-BGP et EIGRP-BGP.

#### VI.4 Métriques

Nous évaluons l'architecture proposée en utilisant la technique de flooding implémenté dans NDN avec les métriques suivantes :

- **Le throughput** : Généralement, le débit correspond à la production maximale d'un élément. Le throughput est le taux de remise réussie des messages sur un canal de communication. Les données auxquelles ces messages appartiennent peuvent être distribuées sur un lien physique ou logique ou peuvent transiter par un certain nœud de réseau. Dans les communications réseau, le throughput est généralement mesuré en bits par seconde (bit / s ou bps), et parfois en paquets de données par seconde (p / s ou pps) ou en paquets de données par créneau horaire. Le throughput peut être calculé par la formule suivante :

$$TP \left( \frac{\text{bit}}{\text{s}} \right) = \frac{\text{le nombre de caractères (octets) reçues par la destination} * 8}{T_G - T_F} \quad (10)$$

Où:  $TP$  est le throughput exprimé en bit/s et  $T_G$  : est le temps de fin de la simulation et  $T_F$  : est le temps de réception du premier paquet de données par la destination.

- **La perte des paquets (Packet loss)**

Il reflète le nombre de paquets perdus pour 100 paquets envoyés. La perte de paquets se produit lorsqu'un ou plusieurs paquets de données transitant sur un réseau informatique ne parviennent pas à atteindre leur destination. La perte de paquets est causée par les erreurs de transmission de données, sur un réseau ou à cause de la congestion de réseau. La perte de paquets est mesurée en pourcentage de paquets perdus par rapport aux paquets envoyés.

- **La convergence des protocoles**

La convergence est un terme utilisé pour décrire un réseau dans lequel tous les routeurs de réseau sont conscients de tous les autres réseaux et disposent de routes vers chaque réseau. Le temps de convergence est le temps nécessaire à tous les routeurs pour apprendre l'intégralité de réseau ou pour apprendre un changement sur le réseau. La durée de la convergence est très importante pour les ingénieurs réseau, car elle décrit le temps nécessaire à un réseau pour se remettre d'un incident. Un état de convergence est atteint une fois que toutes les informations spécifiques au protocole de routage ont été distribuées à tous les routeurs participant au processus de protocole de routage. La convergence est une notion importante pour un ensemble de routeurs qui effectuent un routage dynamique. Tous les protocoles de passerelle intérieure reposent sur la convergence pour fonctionner correctement. Le faire converger est l'état normal d'un système autonome opérationnel.

La convergence peut être calculé comme suit :

$$Tc = tc - t0 \quad (11)$$

à un moment donné  $Ta_n$ , le nième paquet est renvoyé à l'expéditeur. Le temps de convergence peut être calculé par l'équation (11) :

$$Tc - t0 < tan - t0 \quad (12)$$

- **La surcharge du réseau (Overhead)**

Le surcharge de réseau de capteurs sans fil dépend de différents protocoles de réseau, tels que le routage basé sur plusieurs chemins, L'établissement de la communication entre la source et le collecteur devient difficile lorsque le collecteur effectue un déplacement sur une longueur de réseau fixe en raison du plus court chemin emprunté par les routeurs pour maintenir un chemin et une communication cohérents. Comme la taille de réseau augmente, il faudra plus de paquets de contrôle pour trouver et conserver les chemins de routage et les liaisons de communication soient rompues

dans les chemins de communication, ce qui aboutira à la création de plus de paquets de contrôle en augmentant la surcharge.

- **Le Cache Hit Ratio**

Le Cache Hit Ratio est considéré comme un paramètre important pour évaluer les performances des architectures orientées contenu car elle indique la capacité à trouver un contenu sollicité dans le cache d'un routeur et indique l'efficacité de réseau et indique la proportion de requêtes ayant été satisfaites par les *Content Stores* des nœuds. Le nombre de messages Interest et Data envoyés, représentant le plan de données. Le nombre de messages de contrôle envoyés, représentant le plan de contrôle. Ce dernier est constitué des communications entre les nœuds et le contrôleur, Le CHR (Cache Hit Ratio) correspond au surcoût en communication généré les protocoles de routage. Le *Cache Hit* ( $c_i$ ) représente le pourcentage de requêtes satisfaites par le Content Store de  $n_i$  nœuds et se calcule ainsi :  $C_i = \frac{D_i}{I_i}$  (13)

Le Cache Hit Ratio (CHR) total de réseau équivaut au Cache Hit de manière globale :  $CHR = \frac{\sum_{i=1}^n D_i}{\sum_{i=1}^n I_i}$  [85] (14)

où  $n_i$  représente les nœuds de réseau et  $I_i$  représente le nombre de requête d'intérêt,  $D_i$  est le nombre de paquet de données renvoyés par la table de contenu (CS) des nœuds  $ni$ .

- **Le délai**

Le délai [86] de bout en bout dans un RCSF est défini comme le délai entre le moment où un nœud source a un paquet de données prêt à être envoyé et le moment où ce paquet atteint le nœud de destination. En général, le retard est influencé par un ensemble de paramètres tels que le nombre de sauts vers le nœud de destination, la densité des nœuds, le débit de données, les ressources du nœud et les protocoles de routage. Du point de vue de routage, le délai de bout en bout peut être calculé

comme suit :  $E_{toE_{delay}} = \left( \sum_{i=1}^k OneHop_{delay}(i) + C \right)$  (14)

Où  $i \in \{1 \dots K\}$  est l'ensemble des nœuds sur le chemin sélectionné de la source à la destination et  $C$  est une valeur bornée représente le délai.

## VI.5 Paramètres de simulation

Nous avons utilisé la configuration de simulation suivante pour évaluer notre architecture proposée.



Paramètre	Valeur
<b>Les paramètres de NDN</b>	
DW	15, 31, 127, 255, 511
Bandwidth	1 Mbits/s
DeferSlotTime	28 $\mu$ sec (802.11g DIFS)
Interest retransmission timeout	1s
Number of nodes	7
<b>Les paramètres d'application</b>	
Monitoring tasks	1, 4, 8, 12, 16
Monitoring interval	60s
Interest per tasks	4
Data payload	100 bytes
<b>Paramètre de la simulation</b>	
Simulation area	400m x 400m
Topology	Lattice with step 50m
Sink/Sensors	1/80

**Tableau 6:** Les paramètres de la simulation

Voici les détails pour les paramètres de simulation choisis :

- **Bande passante** : La bande passante est une mesure qui indique la quantité de données pouvant être transmise via un réseau. La quantité de bande passante utilisée sur cette simulation est de 1 Mbits / s.
- **Taille de la fenêtre** : Il s'agit de la valeur ou de la taille maximale des données pouvant être envoyées sans accusé de réception de paquet (confirmation). Plus la taille de la fenêtre est petite, plus le transfert sera ralenti, car le nombre de paquets de données nécessitant un accusé de réception.
- **Nœud** : il s'agit d'une jonction réseau ou d'un point de connexion. Chaque terminal, ordinateur, routeur et autre numéro est ajusté en fonction du nombre de routeurs utilisés.
- **Nombre de nœuds** : La simulation a utilisé sept routeurs avec deux machines connectées aux numéros de route neuf et sept.

La simulation utilise le simulateur Network Simulator 3 (NS3) version 3.31 avec l'outil NDN-Sim[72]. Le système d'exploitation était Ubuntu Server 16.04 Trusty. Les

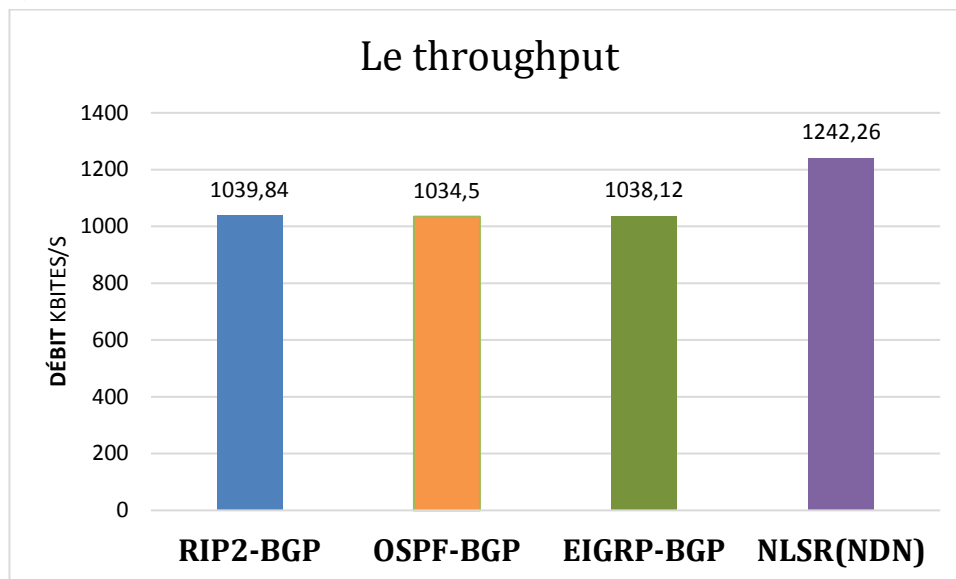
expériences ont été exécutées sur un serveur doté d'un processeur Intel i7 à 2,1 GHz et d'une mémoire vive de 16 Giga-octets.

## VI.6 Résultats de simulation

Nous évaluons notre architecture en utilisant le flooding (inondation) comme méthode de routage par défaut utilisée dans NDN, avec les paramètres de simulation présentés dans le **Tableau 6**, Nous mesurons les métriques : le throughput, le taux de perte de paquets, [a] convergence des protocoles, [a] surcharge de réseau, le cache hit ratio et le délai décrites dans la section 5 de ce chapitre où nous faisons varier le nombre de nœuds disposant de Content Stores, c'est à dire de nœuds pouvant stocker les contenus.

### ✓ Le throughput

C'est le taux moyen de remise réussie des messages sur un canal de communication. Dans la figure suivante, le débit est comparé entre les protocoles RIP2-BGP, OSPF-BGP, EIGRP-BGP et NLSR-NDN.



**Figure 39:** évaluation du throughput

Le graphique présenté dans **Figure 39:** évaluation du ci-dessus montre la comparaison du throughput moyen pour chaque combinaison de protocole de routage. La combinaison du protocole de routage NLSR-BGP a la valeur moyenne du meilleur débit. Mais la différence de valeur de throughput entre les protocoles de routage OSPF-BGP, EIGRP-BGP, RIPv2-BGP n'est pas trop grande. La valeur moyenne du plus petit throughput appartenant au protocole de routage OSPF-BGP.

- Le taux de perte de paquets

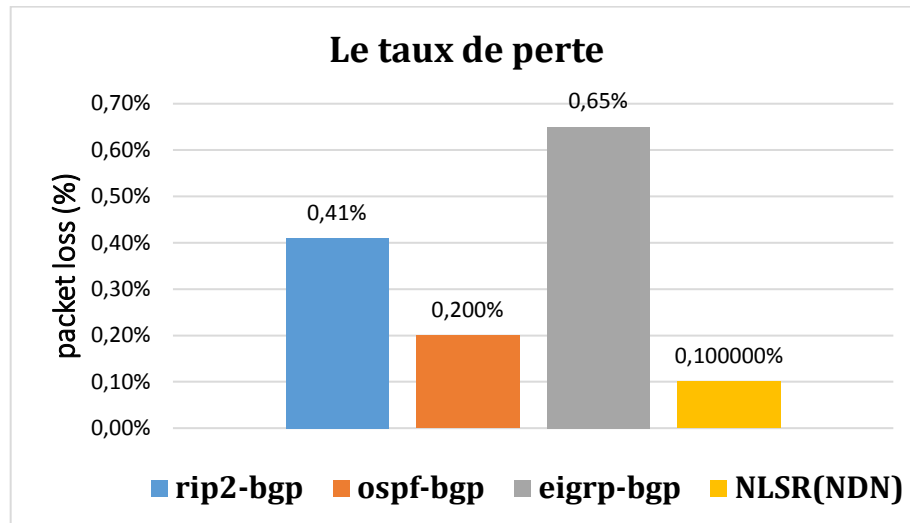


Figure 40: Le pourcentage de perte de paquets

Le graphique ci-dessus montre la comparaison de la valeur de la perte de paquets moyenne pour chaque combinaison de protocole de routage. On peut dire qu'un réseau est bon s'il a une valeur de perte de paquets est faible. Plus la perte de paquets est faible, plus les données perdues au cours du processus de transmission de données sont également de moins en moins nombreuses. Le protocole de routage NLSR-BGP et OSPF-BGP présente la meilleure valeur moyenne de perte de paquets. Alors que le protocole de routage EIGRP-BGP a la plus grande valeur de perte de paquets.

- ✓ La convergence des protocoles

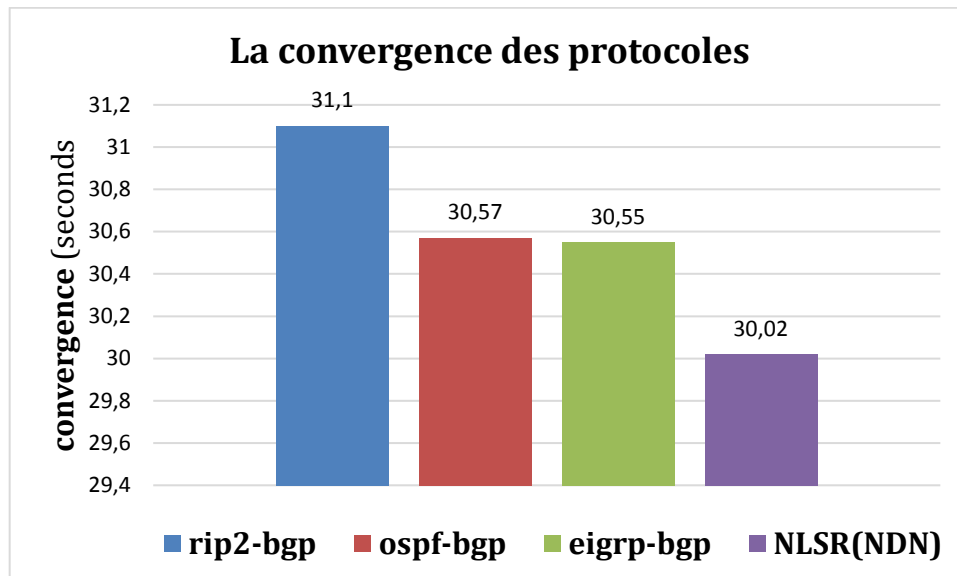
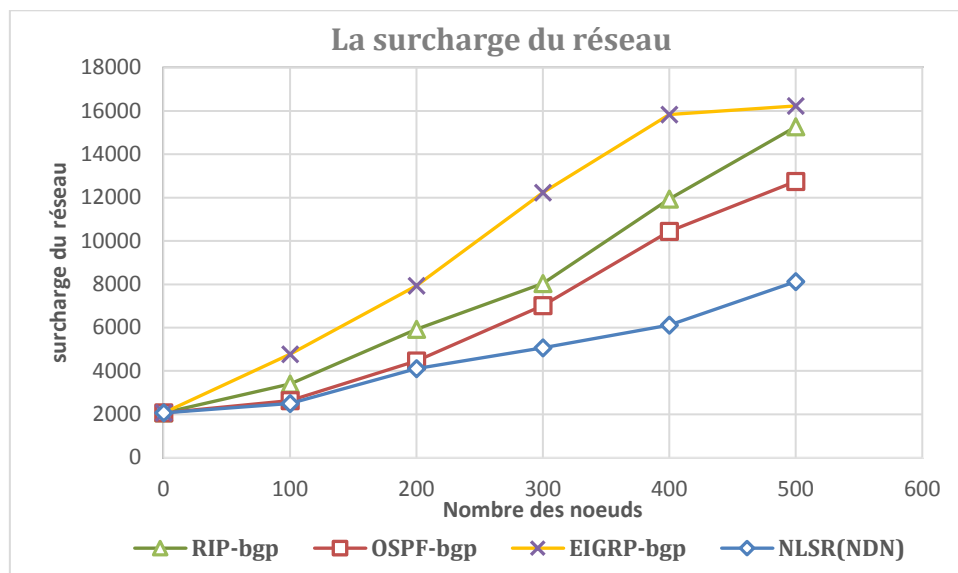


Figure 41: La convergence des protocoles

Le graphique ci-dessus montre la comparaison de la valeur moyenne de convergence pour chaque combinaison de protocole de routage. Plus la valeur est faible, plus la convergence de réseau de protocole de routage est rapide pour assurer la cohérence et mettre à jour la table de routage. Le protocole de routage NLSR-NDN a la valeur moyenne de la meilleure convergence suivie de la valeur de convergence EIGRP légèrement en contradiction avec la valeur de la convergence du protocole de routage OSPF, alors que le protocole de routage RIPv2 a la plus grande valeur de convergence.

✓ **La surcharge de réseau**



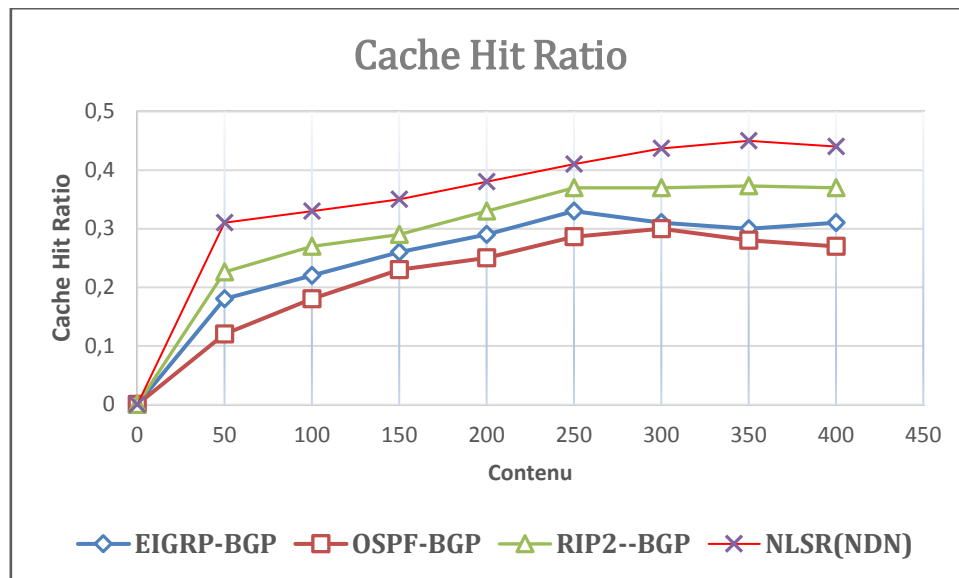
**Figure 42:** La surcharge de réseau

Dans la **Figure 42**, on remarque que la surcharge de réseau augmente considérablement lorsque d'une façon linéaire par rapport le nombre des capteurs virtuels. Lorsque le nombre de nœuds est égal à 100, la surcharge augmente presque linéairement. Lorsque les nœuds sont utilisés presque le temps système de communication devient très rapide. Lorsque le nombre de nœuds est proche de 200, la surcharge de communication diminue considérablement avec l'augmentation du temps de mise à jour.

- **Le cache hit Ratio**

Le Cache Hit Ratio (CHR) ou le taux d'accès au cache dépend de l'efficacité du système de mise en cache et est influencé par des facteurs tels que la stratégie de cache,

le nombre d'objets pouvant être mis en cache, la taille de la mémoire cache et le délai d'expiration de l'objet.



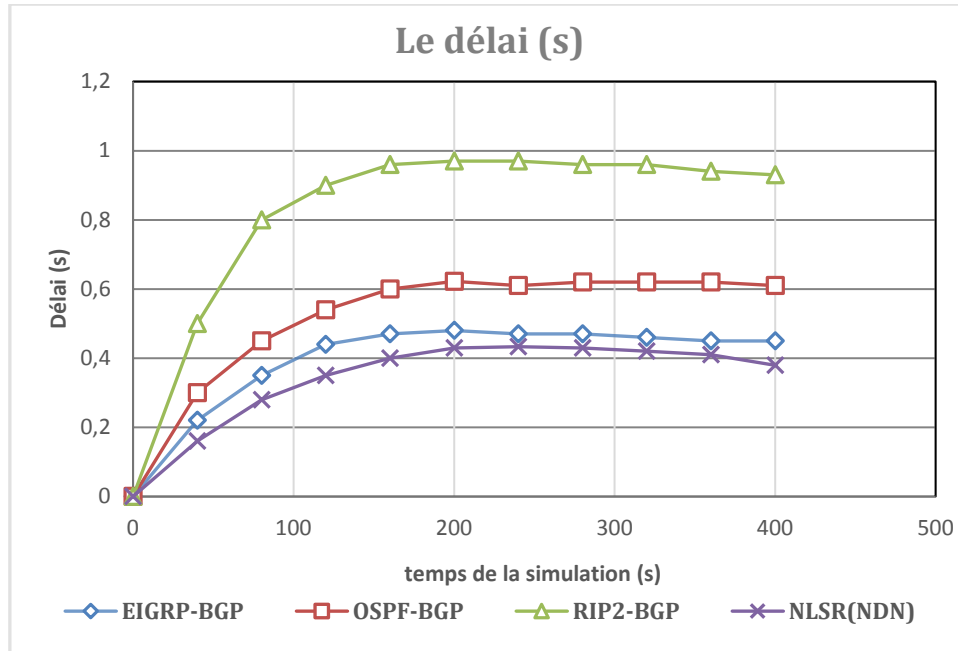
**Figure 43:** L'évaluation du cache hit ratio

Depuis la **Figure 43** Nous remarquons que le protocole de routage NLSR(NDN) présente un meilleur taux d'accès au cache (Cache Hit Ratio), mais également une meilleure utilisation des ressources de mise en cache des routeurs de réseau. Il y a cependant des différences de performances en fonction du protocole de routage, dans le moment  $t=0$ , on remarque le taux d'accès au cache = 0, car les tables de contenu sont encore vides au début de la simulation.

Il est également intéressant de noter qu'au cours de la simulation, le protocole EIGRP et OSPF, RIP donne des taux d'accès au cache amélioré à travers le temps, cependant le protocole OSPF donne un CHR le plus bas entre les autres protocoles, à partir de moment  $t= 250$  s, le taux d'accès au cache commence à diminuer et c'est à cause de de l'augmentation de contenu au niveau des routeurs ce qui augmente le délai de acheminement des paquets et l'élimination des paquets au niveau des tables de contenu selon la politique de gestion de cache. De cette manière, on déduit que dans le cas du protocole NLSR-NDN la plupart des requêtes sont satisfaites par rapport aux autres protocoles utilisés dans la simulation.

- **Le délai**

Le délai est une métrique très importante dans l'évaluation des réseaux basés sur le contenu.



**Figure 44:** L'évaluation du délai

La **figure 44** montre la comparaison des performances du délai des protocoles EIGRP, OSPF, RIPV2 et NLSR, comme on peut le voir, Le délai le plus stable est celui du protocole NLSR-NDN avec moins de 0,41S. Dans la  $T=0s$  le délai est égal à zéro parce qu'aucun paquet n'est mis en réseau au début de la simulation, ensuite remarque que le délai augmente pour tous les protocoles jusque  $t= 200s$ , là le délai est stable pour tous les protocoles à cause du remplissage des tables de contenu à cause des réceptions des paquets des données. Le protocole RIPV2 présente le délai le plus élevé parmi les protocoles utilisés dans la simulation, l'évaluation des performances du délai a montré un avantage substantiel du délai qui se produit pendant la simulation, le délai est minimisé sur en utilisant le protocole NLSR-NDN.

## VI.7 Conclusion

Dans ce chapitre, nous avons décrit et discuté l'évaluation de l'architecture que nous avons proposée pour intégrer les réseaux de capteurs sans fil et le Cloud Computing ainsi que les paramètres utilisés dans l'évaluation des performances. Dans notre proposition, nous avons intégré les RCSF et le Cloud Computing en utilisant les

réseaux de capteurs virtuels et l'architecture Named Data Networking. Les résultats obtenus confirment que le NDN est une technologie candidate prometteuse pour les RCSF et en se basant sur les résultats de la simulation présentés dans le chapitre précédent, on remarque que le protocole NLSR donne de bons résultats dans les expériences par rapport aux autres protocoles de routage interne et surtout quand on intègre les réseaux de capteurs virtuels dans la plateforme ce qui améliore plusieurs paramètres dans le réseau global.

Sommaire

---

VII Conclusion générale et perspectives .....	124
VII.1 Conclusion générale.....	125
VII.2 Perspectives.....	126

---

Ce chapitre fournit un résumé de la thèse ainsi que quelques éléments de travail pour les travaux futurs qui aideront à prolonger le travail effectué dans cette thèse.



## VII.1 Conclusion générale

Depuis leur apparition, les déploiements des RCSF ont été utilisés comme moyen de combler le fossé entre le monde physique et le monde virtuel. Grâce à leur capacité à détecter, calculer et communiquer, les RCSF fournissent à leurs utilisateurs la possibilité de réagir aux divers phénomènes physiques et de prendre les mesures qui s'imposent. L'un des inconvénients des déploiements des RCSF actuels est qu'ils sont spécifiques à une application, et orientés aussi sur les tâches particulières avec peu ou pas de possibilité de les réutiliser pour des applications plus récentes. Cette stratégie est inefficace et conduit à des déploiements redondants lorsque de nouvelles applications sont envisagées. Les RCSF sont considérés comme des éléments de base du paradigme dans les nouvelles technologies de l'information. Ainsi que la multitude d'objets du quotidien, communiquent, interagissent et partagent des données à grande échelle et doivent prendre en charge plusieurs applications simultanément.

Le Cloud Computing est apparu récemment comme un nouveau paradigme dans lequel les ressources de l'infrastructure informatique sont fournies sous forme de services sur l'Internet. Ce paradigme apporte également de nombreux nouveaux avantages en termes d'espace de stockage et de la capacité de calcul. Par conséquent, l'intégration des RCSF dans le Cloud Computing est une solution pour gérer les données provenant des RCSF et envisager les futurs déploiements des RCSF qui devront prendre en charge plusieurs applications simultanées.

Cette thèse propose une solution pour intégrer les RCSF et le Cloud Computing en utilisant les réseaux de capteurs virtuels qui sont considérés comme un nouvel axe de recherche très important avec l'architecture de réseau basé sur le contenu (*Named Data Networking*) pour gérer efficacement les applications et les services simultanément. Nos expériences de simulation ont montré qu'il est possible d'intégrer les RCSF et le Cloud Computing en utilisant NDN. Après une recherche bibliographique importante, nous avons adapté une approche basée sur la notion des réseaux de capteurs virtuels qui utilise le paradigme NDN (*Named Data Networking*), ce dernier, permet à acheminer les données en basant sur les noms des destinataires au lieu des adresses IP.

Nous avons étudié le routage en utilisant les réseaux de capteurs virtuels où chacun d'entre eux représente une abstraction du capteur physique et imite ses

fonctionnalités comme la récupération des données de traitement et enfin le transfert des données vers les autres couches, des capteurs virtuels gère les trois tables de NDN à savoir ; la table de contenu ( *Content store* ) , la table des intérêts PIT ( *Pending Information Base* ) et la table des informations de forwarding ( *Forwarding Information Base* ). Chaque table est responsable d'une tâche bien définie pour assurer l'acheminement des données échangées par le consommateur et le producteur. L'analyse de performance montre que l'utilisation du paradigme NDN peut être utilisé pour le routage des données générées par les RCSF au niveau du Cloud Computing en basant sur les noms des nœuds au lieu des adresses IP.

Nous avons évalué notre architecture en utilisant le simulateur de réseau de données nommé (ndnSIM) basé sur le simulateur NS3 et le framework GSN pour la création et la gestion des capteurs virtuels. Nous avons étudié à travers cette simulation l'efficacité des protocoles basés sur les réseaux de données nommées (dans notre cas NLSR) en le comparant avec d'autres protocoles de routage IP (EIGRP, RIP et OSPF). Les résultats obtenus montrent que le protocole NLSR peut être utilisé comme un protocole de routage pour l'intégration du Cloud Computing et les réseaux de capteurs sans fil.

## VII.2 Perspectives

La recherche sur l'intégration du Cloud Computing et les réseaux de capteurs sans fil est encore à ses débuts. Nous avons proposé à travers cette thèse, une architecture basée sur le paradigme NDN. Néanmoins, il reste encore quelques problèmes non résolus comme le nommage, la sécurité, la mise en cache, le contrôle du forwarding et le mode de déploiement, car originalement le NDN est conçu pour une architecture Internet complète plutôt que pour les RCSF. Les travaux futurs seront consacrés à :

- **Évaluer les performances à plus grande échelle** : Dans différents scénarios et applications ; notre prochain objectif est d'étudier en détail le comportement de NDN dans différents scénarios avancés possibles. Nous envisageons d'étudier des situations telles que le multicast, la répartition du trafic et l'influence de la mise en cache par rapport aux nombres des utilisateurs.
- **Amélioration de schéma de noms** : Nous sommes en train d'expérimenter comment les applications doivent choisir des noms qui facilitent à la fois le

développement d'applications et la diffusion réseau en mettant en œuvre dans des bibliothèques afin de simplifier le développement futur d'applications.

- **Amélioration du modèle analytique** : Le modèle analytique présenté dans cette thèse peut être développé en utilisant d'autres méthodes et algorithmes d'optimisation pour les utiliser dans la gestion de l'acheminement intelligent des paquets
- **Application de l'architecture dans autres domaines** : Comme le paradigme NDN est une plateforme indépendante de toutes plateformes, nous voulons étudier son comportement dans autres domaines comme le domaine de Healthcare.
- **NDN pour IoT** : En raison de ses caractéristiques intrinsèques, nous pensons que le NDN peut répondre à plusieurs besoins dans l'IoT en gérant directement plusieurs fonctionnalités (sécurité, nommage, agrégation de données, etc.) au niveau de la couche réseau. Grâce à l'utilisation d'espaces de noms hiérarchiques spécifiques aux applications et à la structure de transfert intelligent basée sur les structures PIT et FIB, NDN peut offrir une extraction de données simple, robuste et évolutive.

## Bibliographie

- [1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless sensor networks: a survey,” *Comput. Networks*, vol. 38, no. 4, pp. 393–422, 2002.
- [2] R. Buyya, J. Broberg, and A. Goscinski, *Cloud Computing: Principles and Paradigms (Google eBook)*. 2010.
- [3] B. D. Eddine, “Protocole de routage pour les réseaux de capteurs sans fils,” *Univ. Hadj-Lakhdar Batna*, no. Thèse de magistère, p. 81, 2008.
- [4] M. M. Hassan, K. Lin, X. Yue, and J. Wan, “A multimedia healthcare data sharing approach through cloud-based body area network,” *Futur. Gener. Comput. Syst.*, 2016.
- [5] S. Tennina *et al.*, “WSN4QoL: WSNs for remote patient monitoring in e-Health applications,” *2016 IEEE Int. Conf. Commun. ICC 2016*, 2016.
- [6] A. Lounis, A. Hadjidj, A. Bouabdallah, and Y. Challal, “Healing on the cloud: Secure cloud architecture for medical wireless sensor networks,” *Futur. Gener. Comput. Syst.*, vol. 55, pp. 266–277, 2016.
- [7] R. Gravina *et al.*, “Enabling multiple BSN applications using the SPINE framework,” *2010 Int. Conf. Body Sens. Networks, BSN 2010*, pp. 228–233, 2010.
- [8] I. Ahmad, K. Shah, and S. Ullah, “Military Applications using Wireless Sensor Networks: A survey,” *Int. J. Eng. Sci. Comput.*, vol. 6, no. 6, pp. 7039–7043, 2016.
- [9] F. Viani, G. Oliveri, M. Donelli, L. Lizzi, P. Rocca, and A. Massa, “WSN-based solutions for security and surveillance,” *Microw. Conf. (EuMC), 2010 Eur.*, no. September, pp. 1762–1765, 2010.
- [10] R. Dautov, I. Paraskakis, and M. Stannett, “Towards a framework for monitoring cloud application platforms as sensor networks,” *Cluster Comput.*, vol. 17, no. 4, pp. 1203–1213, 2014.

- [11] K. Lu, Y. Qian, D. Rodriguez, W. Rivera, and M. Rodriguez, “Wireless sensor networks for environmental monitoring applications: A design framework,” *GLOBECOM - IEEE Glob. Telecommun. Conf.*, pp. 1108–1112, 2007.
- [12] P. Chapin and C. Skalka, “SpartanRPC: Secure WSN middleware for cooperating domains,” *2010 IEEE 7th Int. Conf. Mob. Adhoc Sens. Syst. MASS 2010*, pp. 61–70, 2010.
- [13] A. Belghith and M. S. Obaidat, *Wireless sensor networks applications to smart homes and cities*. Elsevier Inc., 2016.
- [14] A. Wheeler, “Commercial applications of wireless sensor networks using ZigBee,” *IEEE Commun. Mag.*, vol. 45, no. 4, pp. 70–77, 2007.
- [15] B. Snaith, M. Hardy, and A. Walker, “Draft nist working definition of cloud computing,” *Emerg. Med. J.*, vol. 28, no. 12, pp. 1063–1065, 2011.
- [16] V. Rajesh, J. M. Gnanasekar, R. S. Ponmagal, and P. Anbalagan, “Integration of wireless sensor network with cloud,” *ITC 2010 - 2010 Int. Conf. Recent Trends Information, Telecommun. Comput.*, pp. 321–323, 2010.
- [17] “AWS.” [Online]. Available: <https://aws.amazon.com>. [Accessed: 25-Feb-2019].
- [18] “OpenStack.” [Online]. Available: <https://www.openstack.org/>. [Accessed: 25-Feb-2019].
- [19] “Desktone.” [Online]. Available: <https://www.desktone.com/>. [Accessed: 25-Feb-2019].
- [20] “PROVECTIO.” [Online]. Available: <https://www.provectio.fr/>. [Accessed: 25-Feb-2019].
- [21] “DotRiver - Environnement Virtuel de Travail Universel |.” [Online]. Available: <https://www.dotriver.eu/>. [Accessed: 25-Feb-2019].
- [22] “AWS | Amazon EC2 – Service d’hébergement cloud évolutif.” [Online]. Available: <https://aws.amazon.com/fr/ec2/>. [Accessed: 25-Feb-2019].
- [23] “Microsoft Azure.” [Online]. Available: <https://azure.microsoft.com>. [Accessed: 25-Feb-2019].

- [24] “Rackspace: Managed Dedicated & Cloud Computing Services.” [Online]. Available: <https://www.rackspace.com/>. [Accessed: 25-Feb-2019].
- [25] “OVH.” [Online]. Available: <https://www.ovh.com/fr/>. [Accessed: 25-Feb-2019].
- [26] “App Engine.” [Online]. Available: <https://cloud.google.com/appengine/?hl=fr>. [Accessed: 25-Feb-2019].
- [27] “NetSuite.” [Online]. Available: <https://forms.netsuite.com>. [Accessed: 25-Feb-2019].
- [28] G. Lu and W. H. Zeng, “Cloud Computing Survey,” *Appl. Mech. Mater.*, vol. 530–531, pp. 650–661, 2014.
- [29] “Salesforce.” [Online]. Available: <https://www.salesforce.com/fr/>. [Accessed: 25-Feb-2019].
- [30] “G Suite : Applications de collaboration et de productivité pour les entreprises.” [Online]. Available: <https://gsuite.google.fr/intl/fr/>. [Accessed: 25-Feb-2019].
- [31] B. P. Rimal, E. Choi, and I. Lumb, “A taxonomy and survey of cloud computing systems,” *NCM 2009 - 5th Int. Jt. Conf. INC, IMS, IDC*, pp. 44–51, 2009.
- [32] W. Lizhe, R. Rajiv, C. Jinjun, and F. G. Boualem Benatallah, “Cloud Computing : methodology , system , and,” *CRC Press. Llc*, p. 3, 2012.
- [33] “Oracle.” [Online]. Available: <https://www.oracle.com/index.html>. [Accessed: 25-Feb-2019].
- [34] D. Sarna, *Implementing and Developing Cloud Computing Applications*. Auerbach Publications, 2010.
- [35] Y. Dai, Y. Xiang, and G. Zhang, “Self-healing and hybrid diagnosis in cloud computing,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 5931 LNCS, no. 0831609, pp. 45–56, 2009.
- [36] K. Ahmed and M. Gregory, “Integrating Wireless Sensor Networks with Cloud

- Computing,” *2011 Seventh Int. Conf. Mob. Ad-hoc Sens. Networks*, pp. 364–366, 2011.
- [37] A. Alelaiwi, W. S. Ansari, M. S. Hossain, M. M. Hassan, M. A. Hossain, and A. Alamri, “A Survey on Sensor-Cloud: Architecture, Applications, and Approaches,” *Int. J. Distrib. Sens. Networks*, vol. 9, no. 2, p. 917923, 2013.
- [38] “Intelligent Systems Centre.” [Online]. Available: <http://www3.ntu.edu.sg/intellisys/>. [Accessed: 25-Feb-2019].
- [39] K. L. Tan, “What s NExT ? Sensor + Cloud ?,” p. 4503, 2010.
- [40] C. O. Rolim, F. L. Koch, C. B. Westphall, J. Werner, A. Fracalossi, and G. S. Salvador, “A cloud computing solution for patient’s data collection in health care institutions,” *2nd Int. Conf. eHealth, Telemedicine, Soc. Med. eTELEMED 2010, Incl. MLMB 2010; BUSMMed 2010*, no. i, pp. 95–99, 2010.
- [41] M. Yuriyama and T. Kushida, “Sensor-cloud infrastructure physical sensor management with virtualized sensors on cloud computing,” *Proc. - 13th Int. Conf. Network-Based Inf. Syst. NBiS 2010*, pp. 1–8, 2010.
- [42] A. M. Riad, H. M. El-Bakry, and G. H. El-Adl, “An extensible cloud architecture model for heterogeneous sensor services,” *Int. J. Comput. Sci. Inf. Secur.*, vol. 9, no. 7, 2011.
- [43] C. Doukas and I. Maglogiannis, “Managing wearable sensor data through Cloud Computing,” *Proc. - 2011 3rd IEEE Int. Conf. Cloud Comput. Technol. Sci. CloudCom 2011*, pp. 440–445, 2011.
- [44] X. H. Le *et al.*, “Secured WSN-integrated cloud computing for u-Life Care,” *2010 7th IEEE Consum. Commun. Netw. Conf. CCNC 2010*, pp. 1–2, 2010.
- [45] M. M. Hassan, K. Lin, X. Yue, and J. Wan, “A multimedia healthcare data sharing approach through cloud-based body area network,” *Futur. Gener. Comput. Syst.*, vol. 66, pp. 48–58, 2017.
- [46] P. T. Eugster, P. A. Felber, R. Guerraoui, and A.-M. Kermarrec, “The many faces of publish/subscribe,” *ACM Comput. Surv.*, vol. 35, no. 2, pp. 114–131, 2003.

- [47] S. K. Dash, J. P. Sahoo, and S. Mohapatra, "Sensor-Cloud : Assimilation of Wireless Sensor Network and the Cloud Cloud : Overview," pp. 455–464, 2012.
- [48] L. P. Dinesh Kumar, S. Shakena Grace, A. Krishnan, V. M. Manikandan, R. Chinraj, and M. R. Sumalatha, "Data filtering in wireless sensor networks using neural networks for storage in cloud," *Int. Conf. Recent Trends Inf. Technol. ICRTIT 2012*, pp. 202–205, 2012.
- [49] S. Guo, Z. Zhong, and T. He, "FIND: faulty node detection for wireless sensor networks," *Proc. 7th ACM Conf. Embed. Networked Sens. Syst. - SenSys '09*, pp. 253–266, 2009.
- [50] B. L. Hock, M. T. Yong, P. Mukherjee, T. L. Vinh, F. W. Weng, and S. Simon, "Sensor grid: Integration of wireless sensor networks and the grid," vol. 2005, pp. 91–769524214, 2005.
- [51] D. T. Laboratories, "TinySIP : Providing Seamless Access to Sensor-based Services," 2006.
- [52] C. P. Hall and A. L. Wolf, "DV/DRP:A Content-Based Networking Protocol For Sensor Networks," 2004.
- [53] E. Souto *et al.*, "Mires: A publish/subscribe middleware for sensor networks," *Pers. Ubiquitous Comput.*, vol. 10, no. 1, pp. 37–44, 2006.
- [54] I. Khan, F. Z. Errounda, S. Yangui, R. Glitho, and N. Crespi, "Getting virtualized wireless sensor networks' IaaS ready for PaaS," *Proc. - IEEE Int. Conf. Distrib. Comput. Sens. Syst. DCOSS 2015*, pp. 224–229, 2015.
- [55] L. Zhang *et al.*, "Named Data Networking ( NDN ) Project Named Data Networking ( NDN ) Project," 2010.
- [56] R. L. Van Jacobson Diana K.smetters, James D. Thornton Michael F. Plass Nicholas H.Braynard, "Networking Named Content," *Prog. Chem.*, vol. 24, no. 11, pp. 1–12, 2012.
- [57] V. Jacobson, M. Mosko, D. Smetters, and J. J. Garcia-Luna-Aceves, "Content-centric networking," *Response to DARPA RFI SN07-12*, no. June, 2007.
- [58] W. K. Chai, H. Diliang, I. Psaras, and G. Pavlou, "Cache 'Less for More' in



- Information-Centric Networks,” *Proc. 11th Int. IFIP TC 6 Conf. Netw.*, vol. Volume Par, pp. 27–40, 2012.
- [59] L. Zhang, D. Estrin, J. Burke, and V. Jacobson, “Named Data Networking (NDN) Project 2011 - 2012 Annual Report,” *Named-Data.Net*, 2012.
- [60] S. Kim, “SIDR: Scalable Inter-Domain Routing Protocol for Information Centric Networking,” *Florida State Univ. Libr.*, 2015.
- [61] F. Oehlmann, “Content-Centric Networking,” *Semin. Futur. Internet WS2012*, pp. 91–97, 2013.
- [62] a K. M. M. Hoque, S. O. Amin, A. Alyyan, B. Zhang, L. Zhang, and L. Wang, “NLSR: Named-data Link State Routing Protocol,” *Icn*, p. 15, 2013.
- [63] A. K. M. M. Hoque, S. O. Amin, A. Alyyan, B. Zhang, L. Zhang, and L. Wang, “NLSR: Named-data Link State,” *Proc. 3rd ACM SIGCOMM Work. Information-centric Netw. - ICN '13*, p. 15, 2013.
- [64] V. Jacobson, D. K. Smetters, J. D. Thornton, M. Plass, N. Briggs, and R. Braynard, “Networking named content,” *Commun. ACM*, vol. 55, no. 1, p. 117, 2012.
- [65] H. Yuan, T. Song, and P. Crowley, “Scalable NDN Forwarding - Concepts, Issues and Principles.,” *Icccn*, 2012.
- [66] M. N. O. Sadiku, A. E. Shadare, and S. M. Musa, “Named data networking,” *Int. J. Eng. Res.*, vol. 6, no. 7, p. 371, 2017.
- [67] D. Saxena, V. Raychoudhury, N. Suri, C. Becker, and J. Cao, “Named Data Networking: A survey,” *Comput. Sci. Rev.*, vol. 19, pp. 15–55, 2016.
- [68] S. Kabadayi, A. Pridgen, and C. Julien, “Virtual sensors: Abstracting data from physical sensors,” *Proc. - WoWMoM 2006 2006 Int. Symp. a World Wireless, Mob. Multimed. Networks*, vol. 2006, no. May, pp. 587–592, 2006.
- [69] I. Khan *et al.*, “Wireless Sensor Network Virtualization : A Survey.”
- [70] R. Alubady, S. Hassan, and A. Habbal, “Adaptive interest packet lifetime due to pending interest table overflow,” *Adv. Sci. Lett.*, vol. 23, no. 6, pp. 5573–5577, 2017.

- [71] K. Aberer, M. Hauswirth, and A. Salehi, “A Middleware For Fast and Flexible Sensor Network Deployment,” *Vldb*, pp. 1199–1202, 2006.
- [72] S. Mastorakis, A. Afanasyev, I. Moiseenko, and L. Zhang, “ndnSIM 2 . 0 : A new version of the NDN Simulator for NS-3,” *NDN, Tech. Rep. NDN-0028.*, pp. 1–8, 2015.
- [73] “Ubuntu-fr | Communauté francophone d’utilisateurs d’Ubuntu.” [Online]. Available: <https://ubuntu-fr.org/>. [Accessed: 27-Feb-2019].
- [74] “Vagrant by HashiCorp.” [Online]. Available: <https://www.vagrantup.com/>. [Accessed: 27-Feb-2019].
- [75] A. Afanasyev, I. Moiseenko, and L. Zhang, “ndnSIM: NDN simulator for NS-3,” *NDN Tech. Rep.*, pp. 1–7, 2012.
- [76] G Carneiro, “Network simulator 3,” *UTM Lab*, vol. 91, pp. 399–404, 2017.
- [77] K. Aberer, M. Hauswirth, and A. Salehi, “Infrastructure for data processing in large-scale interconnected sensor networks,” *8th Int. Conf. Mob. Data Manag.*, pp. 198–205, 2007.
- [78] A. Aberer, Karl; Hauswirth, Manfred; Salehi, “Global Sensors Networks,” no. 5005, 2014.
- [79] J. Moy, “Traffic Engineering (TE) Extensions to OSPF Version 2 Status,” *Internet Eng. Task Force*, no. RFC 2328, p. 3, 2000.
- [80] C. Hedrick, “Routing Information Protocol,” *RFC1058*, no. June, p. 278, 1988.
- [81] A. Y. Rekhter and T. Li, “A Border Gateway Protocol 4 (BGP-4),” *RFC1771 (March 1995).*, p. 259, 1995.
- [82] D. Xu and L. Trajković, “Performance Analysis of RIP , EIGRP , and OSPF using OPNET Roadmap,” pp. 1–16, 2011.
- [83] E. C. Rosen and B. B. and Newman, “Exterior Gateway Protocol (EGP),” *IEEE RFC 827*, vol. 84, no. October 1982, pp. 487–492, 2013.
- [84] M. Al Ameen and J. Liu, “Security and Privacy Issues in Wireless Sensor Networks for Healthcare Applications,” pp. 93–101, 2012.

- [85] E. Aubry, "Protocole de routage pour l'architecture NDN," *Univ. Lorraine*, vol. tel-01699, no. thèse de doctorat, p. 92, 2017.
- [86] F. Despaux, "Modeling and evaluation of the end-to-end delay in wireless sensor networks," *thèse Dr. l'Université Lorraine*, no. <https://hal.univ-lorraine.fr/tel-01751796>, p. 131, 2018.

## Mes publications

---

H. Taleb, S. Hamrioui, P. Lorenz, and A. Bilami, "Integration of energy aware WSNs in cloud computing using NDN approach," *Int. Conf. Ubiquitous Futur. Networks, ICUFN*, pp. 188–192, 2017

A. Benayache, A. Bilami, S. Barkat, P. Lorenz, H. Taleb, "MsM: A Microservice Middleware for Smart WSN-based IoT application", *Journal of Network and Computer Applications, JNCA, Elsevier*, (en cours de révision)