REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR

ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE SAAD DAHLAB DE BLIDA 1

FACULTE DES SCIENCES

**Laboratoire de Recherche pour le Développement des Systèmes Informatisés**

**PROJET DE FIN D'ETUDE POUR L'OBTENTION DU DIPLOME DE MASTER EN SECURITE DES SYSTEMES D'INFORMATION**

**Mémoire réalisé par :**

Ramla Mohamed

Walid Miloud Dahmane

# THEME :

# Etude et Implémentation des Mécanismes de Sécurité pour le Routage Centré Contenu

**Organisme d'accueil :** USDB

**Promotrice :** ARKAM Meriem

PRESIDENT JURY : BOUSTIA NARIMENE
EXAMINATEUR : BAOUA

**Année Universitaire :** 2017-2018

# Résumé


## Résumé

Les Réseaux Centrés Information (ICNs) ont introduit des nouveaux concepts et idées dans le domaine de recherche des protocoles de routage de prochaine génération, proposant une approche alternative à la suite de protocoles TCP/IP bien connue et consolidée. Un ICN envisage un réseau de périphériques de mise en cache intelligents qui transmettent non seulement des bits d'un endroit à l'autre, mais aussi un support du réseau pour fournir aux utilisateurs finaux ce qu'ils sont vraiment intéressés : les données nommées. Cependant, bien qu'une grande partie de la littérature existante souligne les avantages de ce nouveau paradigme de réseau, nous nous concentrons sur certains problèmes de sécurité liés au routage centré contenu comme l'opportunité de créer des dénis de service distribuées (DDoS), communément appelées les attaques d'inondations d'intérêt (Interest Flooding Attack, IFA). Dans notre travail, nous avons étudié ce type d'attaque ainsi que les solutions proposées pour lutter contre cette attaque, comme Traceback et Poséidon. Pour valider notre étude, nous avons simulé plusieurs scénarios d'attaque en utilisant le package CCN-lite sous le simulateur OMNeT++, nous avons aussi aborder des solutions pour d'autres types d'attaques liés au routage tel que Hijacking et l'attaque d'Interception.


**Mots clés :** réseaux centrés sur l'information (ICN), routage centré contenu, sécurité, DDoS, spoofing, Traceback, CCN-lite.



## Abstract

Information Centric Network (ICN) has introduced new concepts and ideas in the next generation routing protocols research area, proposing an alternative approach to the well-known and consolidated TCP/IP protocol suite. ICN envisions a network of smart caching devices that not only transport bits from one place to another but also support the network to provide end users with what they are really interested in: named data. However, while a large portion of the existing literature highlights the benefits of this new network paradigm, we focus on some specific security issues related to the opportunity of mounting distributed denial of service attacks (DDoS), commonly known as Interest Flooding Attack (IFA). In our work, we studied this type of attack of IFA as well as the solutions proposed to fight against this attack, like Traceback and Poseidon. To validate our study, we simulated several attack scenarios using the CCN-lite package under the OMNeT ++ simulator, we also discuss solutions for other types of routing-related attacks such as hijacking and interception attack.



**Keywords: Information Centric Network, Content Centric Network, Security, DDoS, Spoofing, Traceback, CCN-lite.**

# Remerciement

*Je remercie d'abord ALLAH le tout puissant qui m'a guidé et qui m'a donné la force et la volonté de réaliser ce travail.*

*Mes pensées vont vers mes parents, qui ont toujours cru en moi*

.

*C'est grâce à leur soutien que j'ai pu réaliser ce travail. Ils savent déjà combien je leur dois.*

*Comme on remercie notre promotrice Mme Meriem ARKAM de nous avoir pris en charge et aidé tout au long du projet.*

*Nos remerciements les plus sincères à toutes les personnes qui auront contribué de près ou de loin à l'élaboration de ce mémoire ainsi qu'à la réussite de cette formidable année universitaire.*

*Enfin, nous tenons aussi à remercier les jurés pour avoir accepté d'examiner et de juger notre travail.*

*Merci à tous et à toutes.*

# Sommaire

# Liste des figures

# Liste des tableaux

# Introduction Générale

## Introduction

Le succès d'Internet est largement dû à la quantité de contenus mis à disposition, Et l'accès à ces contenus est aujourd'hui le service dominant des usages de l'Internet. Cependant, l'architecture traditionnelle d'Internet, basée sur le modèle de communication hôte à hôte, ne permet pas la livraison de contenu à grande échelle. Pour remédier à ce problème, les chercheurs proposent un nouveau paradigme de communication centré sur le contenu où, plutôt que de connecter des hôtes distants comme le fait actuellement IP, le réseau gère directement les éléments d'information (contenus) que les utilisateurs veulent publier, récupérer ou échanger. Ce nouveau type de réseau est connu sous différentes appellations : réseau centré contenu (Content Centric Network, CCN) ou encore plus généralement : réseau centré information (Information Centric Networking, ICN).

## Problématique

De plus de la livraison des contenus, l'ICN permet aussi de traiter d'autres limitations dans l'architecture actuelle d'Internet notamment la gestion de la sécurité. En effet, ils basculent vers la sécurisation de contenus au lieu de la sécurisation du chemin (l'infrastructure et des nœuds) des contenus. En conséquence, de nouvelles attaques sont apparues avec ce nouveau modèle de sécurité, en plus des attaques héritées qui peuvent avoir un impact sur l'ICN. Par exemple, nous trouvons plusieurs attaques liées au routage centré contenu telles que les attaques d'infrastructure, de source, de Blocage mobile, de Flooding, de timing, de Jamming, de Hijacking et d'Interception.

## Objectifs :

Ce sujet consiste à étudier les problèmes de sécurité des ICNs, notamment les attaques potentielles liées au routage centré contenu, comme le déni de service distribué (DDoS), l'interception, le hijacking et l'inondation (Flooding), et ce dans le but de proposer des solutions.

## Organisation :

Notre mémoire sera organisé comme suite :

- **Chapitre I :** Nous parlerons dans ce chapitre de la sécurité dans les CCNs avec présentation d'une classification des attaques dans les ICNs.
- **Chapitre II :** Dans ce chapitre, nous présenterons et paramétrons les différentes solutions pour contre les IFA's.

- **Chapitre III :** Ce chapitre compare les performances des différentes solutions pour contrer les IFA's, sous le simulateur omnet++ avec le package CCN-lite.
- **Conclusion et Perspectives :** Nous terminerons avec une conclusion générale et perspectives

# Chapitre I : Réseaux Centrés sur l'Information et leurs sécurités

## I.1. Introduction :

Au cours des dernières décennies, de sérieux efforts ont été déployés pour diverses architectures pour le futur Internet. Chacune de ces architectures à une chose en commun, à savoir, se concentrer sur la livraison de contenu plutôt que sur des approches centrées sur l'hôte. Cependant, seuls quelques-uns d'entre eux ont gagné en popularité en raison de leurs applications possibles à l'étude.

Au cours de ce chapitre, nous allons aborder les réseaux centrés sur l'information. Ensuite, nous allons décrire et comparer les principaux projets ICN comme : DONA, NetInf, PSIRP et CCN ainsi qu'une description détaillée sur le routage dans les CCNs, avant de de présenter les problèmes de sécurité de cette nouvelle architecture. Ensuite, nous présentons les solutions de sécurités offertes par l'architecture CCN.

## I.2. Les Réseaux Centrés sur l'Information :

Internet a été créé dans les années 70s pour échanger ou communiquer des informations. La généralisation de l'utilisation d'Internet à l'échelle mondiale s'est opérée plus vite que quiconque aurait pu l'imaginer. L'évolution rapide de ce réseau mondial induit un bouleversement des interactions sociales, commerciales, politiques et même personnelles **[CCNA1]**.

Les utilisations d'Internet étaient relativement simples, comme rechercher des informations sur des sites web, parler avec des amis grâce à des logiciels de messagerie instantanée, envoyer/recevoir des mails, télécharger des fichiers depuis des serveurs FTP, etc. Pour ces usages, les communications dans les réseaux sont réalisées selon un modèle de bout-en-bout, en établissant des tunnels de communications, depuis un point de terminal (un utilisateur par exemple) vers un autre point (voir *Fig. I.1*).

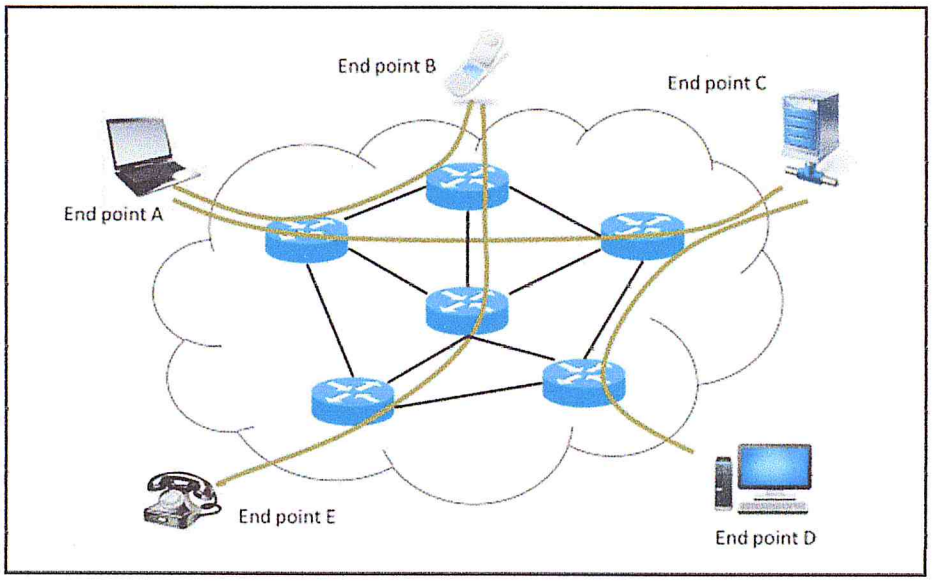


***Fig. I.1.*** *L'architecture de réseaux IP* **[WEI 14]**

Si on analyse bien, on peut trouver que ce qui a changé est le concept fondamental des usages Internet. Ce qui intéresse les clients n'est plus l'endroit où il peut trouver (localisations) les informations, mais plus précisément les informations (contenus) elles-mêmes. Néanmoins, le modèle original d'Internet de type bout-en-bout n'est pas efficace pour de tels services de distributions de contenus. C'est pour cette raison que des applications P2P (Peer to Peer) et CDN (Content Delivery Networks) sont appliqués dans les réseaux. La communauté scientifique a formulé des propositions connues sous le nom de Réseau Centré sur l'Information (Information Centric Networks, ICN) pour changer la pile de protocoles réseau afin de transformer Internet en une infrastructure de distribution de contenu **[SAN 15]**.

### I.2.1 Fonctions de base des ICNs :

Les ICNs proposent de changer l'Internet, qui est actuellement basé sur les localisations des serveurs avec des adresses bien définies, vers une architecture basée sur le nom des contenus (voir *Fig. I.2*), avec des fonctionnalités nativement intégrées comme le nommage indépendant de la localisation, le routage basé sur les noms de contenu, la faculté de cacher des contenus dans les réseaux, le multicast, la sécurisation des contenus, la mobilité etc. Grâce à ces fonctionnalités, les ICNs sont plus efficaces pour délivrer des contenus aux utilisateurs avec une meilleure qualité et permettent aussi d'améliorer la gestion des capacités réseaux des fournisseurs des réseaux **[WEI 14]**.

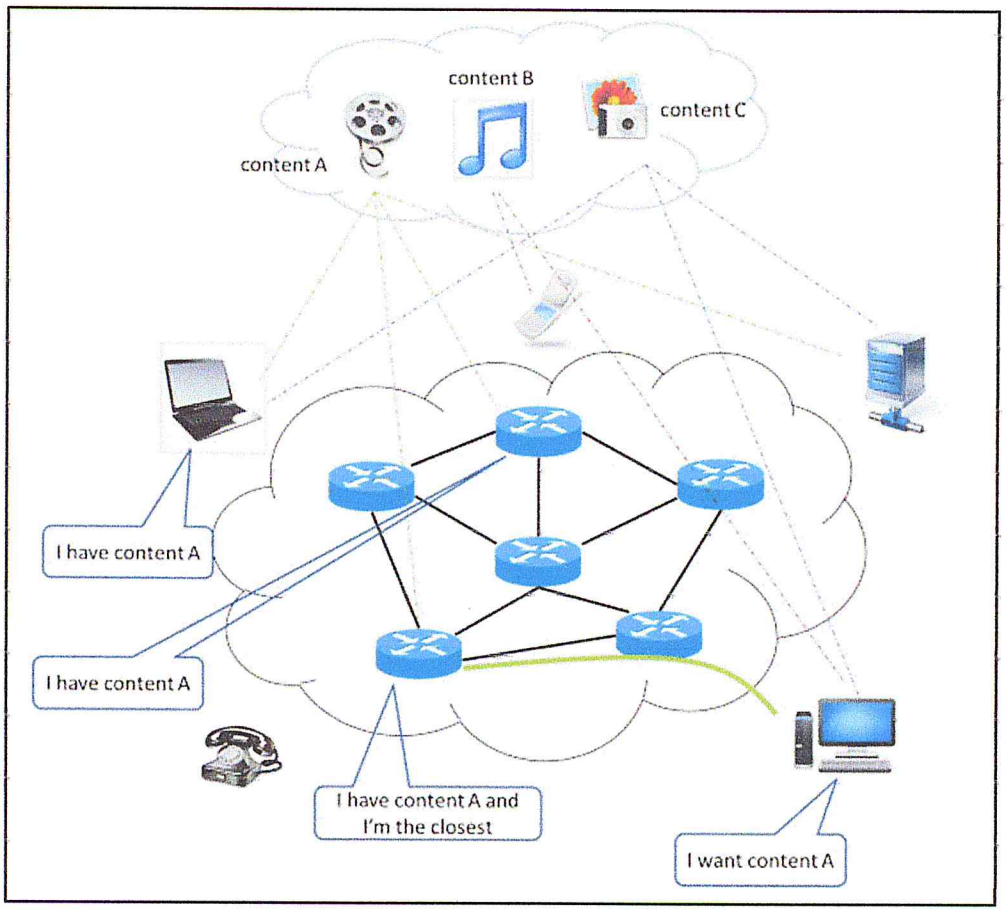


***Fig. I.2.*** *L'architecture des réseaux ICN* **[WEI 14]**

Les principales fonctionnalités intégrées dans un ICN sont :

**a. Nommage des contenus :**

Dans un ICN, l'unité de réseau de base est l'objet de contenu sur lequel toutes les activités de réseautage sont basées, Un objet de contenu est identifié par un nom globalement unique qui est composé d'un nombre variable de composants, organisés dans une structure plate ou hiérarchique. **[WEI 14]**.

**b. Routage basé sur le nom de contenu :**

Le réseau ICN est un modèle basé sur le récepteur (receiver-driven en anglais). C'est-à-dire qu'un utilisateur exprime seulement son intérêt sur des contenus au réseau. Ensuite, c'est le réseau qui a pour charge de trouver les bons contenus et les meilleures sources pour ces contenus, en se basant sur leurs noms. Quand l'intérêt du client arrive finalement à une source de contenu, le contenu est délivré en suivant le chemin inverse du message d'intérêt

jusqu'au client. Au final, le client est satisfait car il reçoit le contenu désiré, même s'il n'a pas connaissance de l'entité qui lui a fourni ce contenu. **[WEI 14]**

Dans notre travail, nous intéressons à cette fonction de l'ICN et en particulier à la sécurité de routage.

### c. Cache des contenus :

Une caractéristique importante des ICNs est la capacité de mise en cache de contenus directement dans les nœuds du réseau. Chaque morceau d'un contenu peut être mis en cache dans les nœuds de réseau se trouvant sur le chemin de la livraison du contenu, de sorte que les demandes ultérieures pourront être satisfaites plus rapidement, directement par les caches des nœuds ICN. Les paquets perdus pourront aussi être aussi récupérés plus rapidement par des retransmissions directes depuis les caches les plus proches.

### d. Sécurité du contenu :

ICN fournit une garantie de sécurité auto-protégée via des contenus cryptés et auto certifiés, et non pas via des connexions de communication sécurisées comme IP. Seuls les utilisateurs autorisés peuvent déchiffrer les contenus.

### e. Mobilité du contenu :

Étant donné qu'ICN fonctionne sur un modèle non-connecté (il n'y a pas de connexion établie dans un réseau ICN), la mobilité des utilisateurs ne modifie pas le comportement des réseaux ICN. Leurs demandes, issues de différents endroits à différents moments, sont traitées indépendamment par les réseaux ICN, chacune comme une requête unique.

## I.2.2 Principaux projets ICN :

Dans les dernières années, plusieurs projets ou solutions ont été mises en œuvre par des équipes de recherche différentes, avec l'objectif de réaliser et déployer un ICN ; Les plus connus sont DONA **[KOP 07]**, NetInf **[SAI 13]**, PSIRP **[LAG 12]** et CCN **[JAC 09]**. Dans ce qui suit, nous allons décrire les principales fonctionnalités (nommage et routage) de chacun de ces projets. Notre description s'achèvera avec une comparaison et une discussion pour justifier notre choix d'un projet dans notre travail.

### a. NetInf :

***Network of Information (NetInf)*** est une architecture de réseau proposée par le projet européen FP7 appelé 4WARD et son projet de suivi appelé SAIL **[SAI 13].**

La structure de réseau NetInf est basée sur un système de résolution de dénomination (NRS - *Naming Resolution System*) et un mécanisme de routage de table de hachage multiple (MDHT - *Multiple Distributed Hash Table*). Dans NetInf, les concepts de représentation de contenu et l'objet de données d'un contenu sont clairement distincts.

Un objet d'information (*IO - Information Object*) sert à référencer directement un objet de contenu et un objet de niveau de bit (*BO - Bit-level Object*) est la donnée de contenu elle-même. Un *IO* contient trois informations : un identifiant globalement unique du contenu, un ensemble de métadonnées et un objet de données (*DO - Data Object*) qui peut être considéré comme une référence de la charge utile du contenu réel - le BO. L'identifiant de contenu contient le type de contenu (comme le document texte, le fichier audio, l'image, la page Web ou les flux en direct) et le hash de la clé publique du propriétaire (ou éditeur). Le champ de métadonnées contient une méta-liste qui fournit des informations sémantiques sur le contenu qui peut être utile pour le service de recherche, comme le débit et le codec d'un enregistrement audio ou l'auteur et l'abstrait d'un document.

Les rôles des auteurs de contenu et des éditeurs dans NetInf sont différents. Les auteurs créent, signent et modifient les IO. Un même IO peut avoir différentes versions et différentes versions peuvent avoir différents auteurs et différentes signatures. Les éditeurs ne sont pas autorisés à modifier ou à signer les IO, mais ils peuvent redistribuer le contenu.

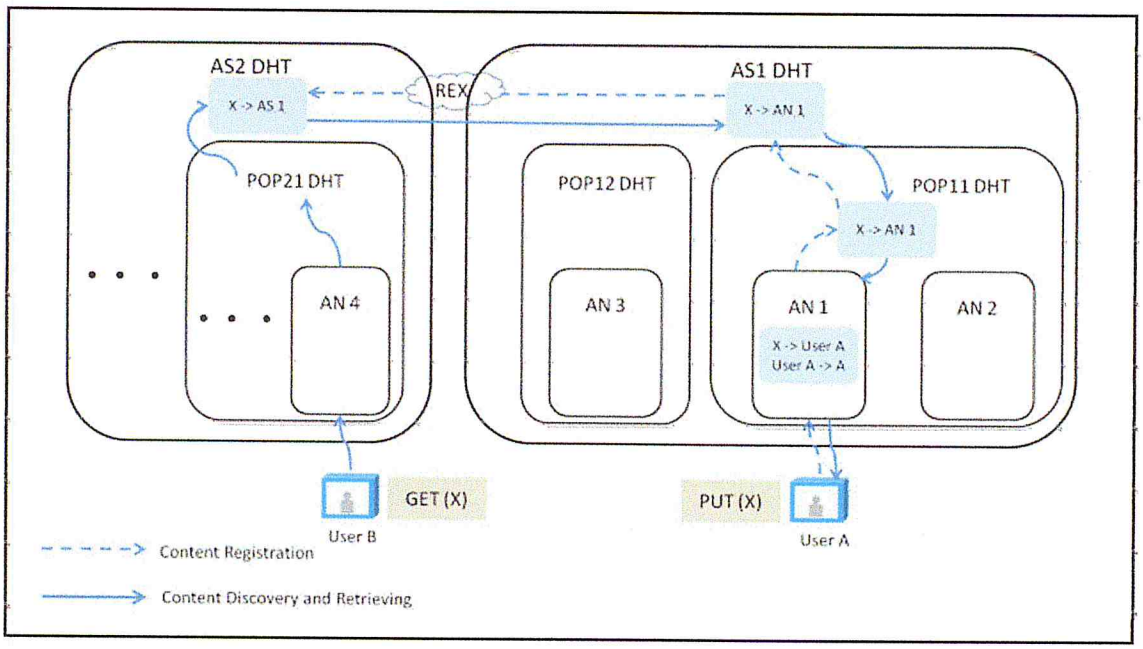


***Fig. I.3.*** *Schéma de routage de NetInf* **[ATH 15]**

Le système de résolution de noms NetInf (NRS) prend un identifiant de contenu ou un ensemble d'attributs qui décrivent l'objet recherché comme entrée et renvoie un ensemble d'enregistrements de liaison pour les IO qui correspond à l'entrée. Les IOs incluent une référence (DOs) qui, directement ou indirectement, peut être utilisée pour récupérer le BO. Cela signifie que dans le système NetInf, une résolution en deux étapes est possible. L'application ou l'utilisateur peut choisir dans la liste des OI renvoyés celui à sélectionner pour demander les BO correspondantes, en fonction de différents critères (coût, vitesse de téléchargement, définition, qualité, etc.).

NetInf définit différentes zones de résolution de noms de niveau qui sont réalisées par un (MDHT). Chaque zone est responsable de stocker de manière persistante une BO avec l'IO d'identification correspondante. Lorsqu'un client demande un objet, une première recherche DHT est effectuée au premier niveau (par exemple, sa zone de réseaux d'accès). Si elle n'est pas trouvée, une autre recherche DHT est émise à un niveau supérieur (par exemple, zone POP). Si elle n'est toujours pas trouvée, une autre recherche DHT est effectuée à un niveau supérieur (par exemple, niveau de domaine), etc. Lorsque la recherche DHT est réussie à un niveau donné, le résultat est retourné au client. Il est à noter que, malgré le routage par hop-by-hop et la résolution locale, le niveau DHT supérieur doit contenir des liaisons pour toutes les données enregistrées dans le domaine, avec une évolutivité possible et éventuellement des problèmes de performance. La Figure I.3 illustre le schéma de routage de l'architecture NetInf.

## b. DONA :

***Data Oriented Network Architecture (DONA)*** une architecture orientée donnée qui a été proposée en 2007 par Koponen et al **[KOP 07]**. Le projet DONA se base sur l'utilisation de noms auto certifiés et incorporer des fonctionnalités de cache. Le noyau du routage dans DONA est un système hiérarchique de résolution de dénomination avec un nom de contenu plat.

Le problème de nomination dans DONA est similaire à celui de NetInf. DONA utilise une structure de nommage P : L. La partie P est le hash de la clé publique du propriétaire du contenu (le concept de principe dans DONA). Et le L est l'étiquette de contenu attribuée par le propriétaire. Les propriétaires de contenu ont la responsabilité d'assurer l'intégralité du nom P : L globalement unique.

Cependant, le routage dans DONA est différent par rapport au routage de NetInf. DONA applique un système hiérarchique de résolution de nom de contenu (*Fig. I.4*), les Poignées de Résolution (RHs - Resolution Handles). Chaque nœud RH a une base d'informations contenant trois tuples, le nom du contenu P : L, le prochain saut et la distance. Le prochain saut est d'où le nœud reçoit la publicité sur le nom du contenu.

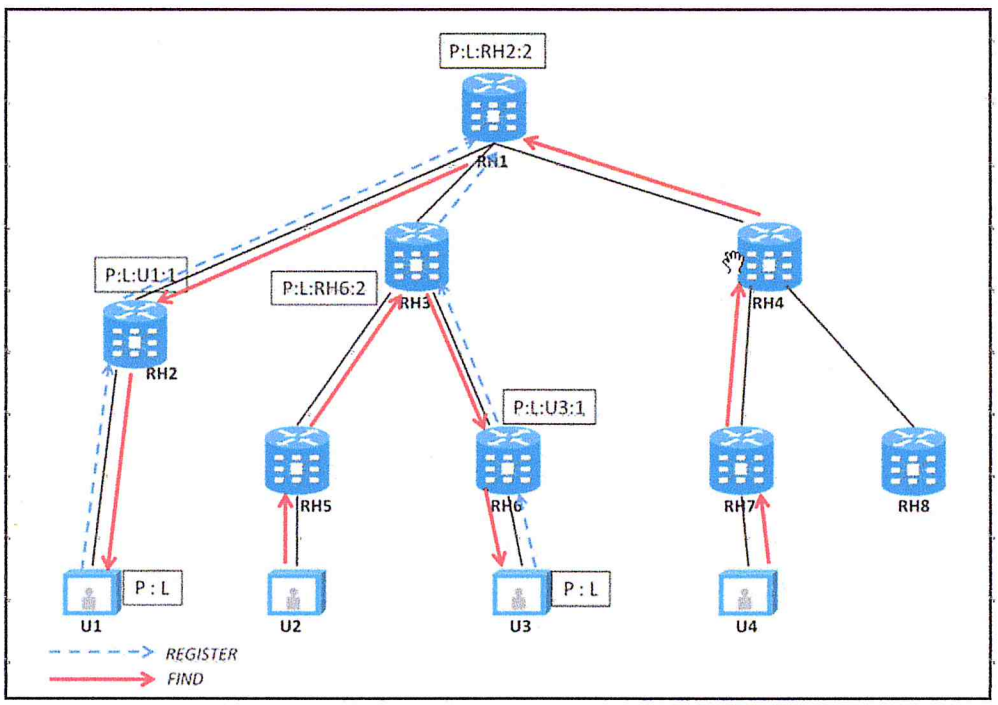


***Fig. I.4.*** *Schéma de routage de DONA* **[ATH 15]**

Le routage DONA contient deux messages : FIND et REGISTER. Tous les deux messages sont directement basés sur le nom de contenu plat. Lorsqu'un RH reçoit un message REGISTER, il ajoutera <P : L, prochain saut, distance> dans son tableau de registre pour un nouveau message d'arrivée, ou Mettre à jour la prochaine et la distance pour une entrée existante si le nouveau message arrivé à une distance plus courte. Ensuite, la RH transmet ce message à son parent (s). Enfin, l'inscription au contenu se terminera à la plus haute (s) RH (s). Lorsqu'une HR reçoit le message FIND, il examinera le nom du contenu dans sa table de registre locale. S'il trouve une entrée correspondante, le message FIND sera transmis à travers le prochain saut de l'entrée correspondante. Sinon, la RH transfère le message FIND à son (ses) RH (s) parent(s). Comme le chemin du message FIND, chaque HR qui est sur le chemin ajoute le FIND localement, donc une fois que le FIND arrive au conteneur de contenu le plus proche, l'objet de contenu sera renvoyé par le chemin inverse de chemin FIND.

### c. PSIRP :

***Publish Subscribe for Internet Routing Paradigm (PSIRP)*** **[LAG 12]** est un projet européen qui a débuté en 2008 et s'est terminé en 2010. PSIRP a proposé une ardoise propre d'architecture ICN basée sur une solution d'abonnement aux éditeurs.

PSIRP a appliqué la même structure de dénomination P : L que le DONA et NetInf. Le nom du contenu est appelé identificateurs de ressource (RIds). Le réseau PSIRP comprend un concept de base qui s'appelle Scopes, qui est identifié avec les identificateurs de scope (SIds). Les Scopes contrôler les caractéristiques d'un contenu, tel que l'accès droit, autorisations, disponibilité, accessibilité, réplication, persistance et les ressources en amont. La publication du contenu (publish) et la demande de contenu (subscribe) d'un contenu sont basé sur une paire de composition de <SId, RId>.

Le processus de routage PSIRP comprend quatre unités importantes :

Nœuds de Rendez-vous (RN), Nœuds de topologie (TN), Nœuds de Branchement (BN) et Nœuds de transfère (Forwarding Node, FN). L'ensemble des réseaux PSIRP est divisé en Domaines, qui est similaire au Système Autonome de l'Internet actuel (***Fig. I.5***). Chaque domaine contient un RN, un TN, un BN et plusieurs FN. Le RN de chaque domaine est en charge de la correspondance entre les éditeurs de contenu et les abonnés, la localisation des publications de contenu et des domaines. Chaque RN peut avoir son propre système de

résolution de noms. Toutes les RNs de chaque domaine sont interconnectés avec DHT dans une Interconnexion RendezVous (RI) globale qui fait les étendues de chaque domaine sont globalement accessibles. Le TN est chargé de la gestion de la topologie de réseau intra-domaine et de l'équilibrage de charge. Il échange également l'information vectorielle du chemin d'accès avec les autres TNs inter-domaines. Le BN construit une carte de routage pour acheminer les intérêts des abonnés vers les conteneurs de contenu inter-domaine ou intra-domaine en utilisant la topologie maintenue par le TN. Enfin, le rôle des FN est d'utiliser une implémentation de transmission basée sur le filtre Bloom pour réaliser le renvoi de contenu à partir du conteneur de contenu aux abonnés. Le filtre Bloom qui s'appelle identifiant de Forwarding (FId) est accumulé pendant la livraison de l'abonnement.

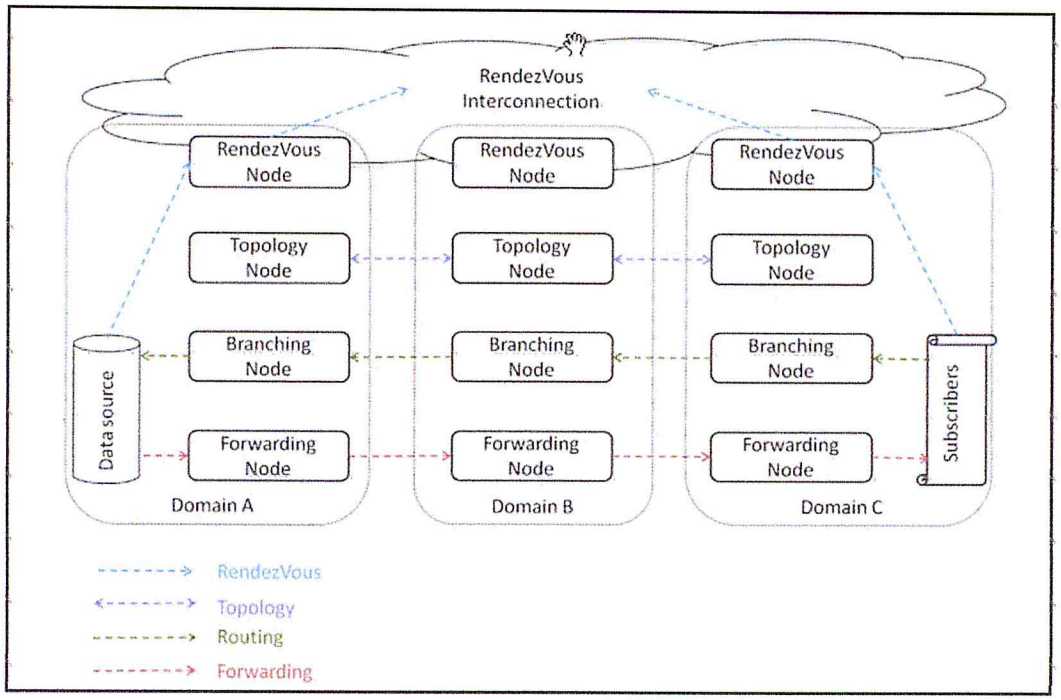


***Fig. I.5.*** *Schéma de routage de PSIRP* **[ATH 15]**

Pour résumer, dans PSIRP, un abonné exprime son abonnement d'un contenu à la RN locale de son domaine pour obtenir le contenu emplacement du conteneur. Le BN utilise la topologie de routage qui est obtenu par le TN pour transmettre l'abonnement au contenu récipient. Le paquet d'abonnement cumule le chemin de retour dans le filtre Bloom et construit la FId. Enfin, les FN utilisent les FId pour retourner le contenu requis à l'abonné.

**d. CCN :**

Van Jacobson propose la nouvelle architecture d'Internet orientée contenu nommée ***Content Centric Networking (CCN)*** **[JAC 09]**. Au lieu d'identifier les machines comme dans l'Internet d'aujourd'hui, l'architecture CCN (autant de propositions ICN), identifie les objets de données avec des noms. Les objets de données sont ensuite divisés en plusieurs paquets, chacun identifiant en utilisant un nom globalement unique. Les noms de CCN sont composés d'un nombre variable de composants, organisés dans une structure hiérarchique. Les noms structurés hiérarchisés, en ce qui concerne les noms à plat, ont l'avantage qu'ils peuvent être agrégés dans des préfixes, ce qui réduit considérablement la taille de la table de transfert sur le routeur d'un CCN, afin d'accélérer le processus d'acheminement (détaillé dans la section suivante). Comme présenté dans **[JAC 09]**, la figure I.6 montre un exemple du schéma de dénomination dans lequel un fichier vidéo est identifié par le nom ***/parc.com/ video/ WidgetA.mpg/ _v<timestamp>/*** et ses morceaux avec des noms du type **/parc.com/ video/ WidgetA.mpg/ _v<timestamp>/ _s_id** où **_s_id** représente l'identifiant du segment (paquet de données)

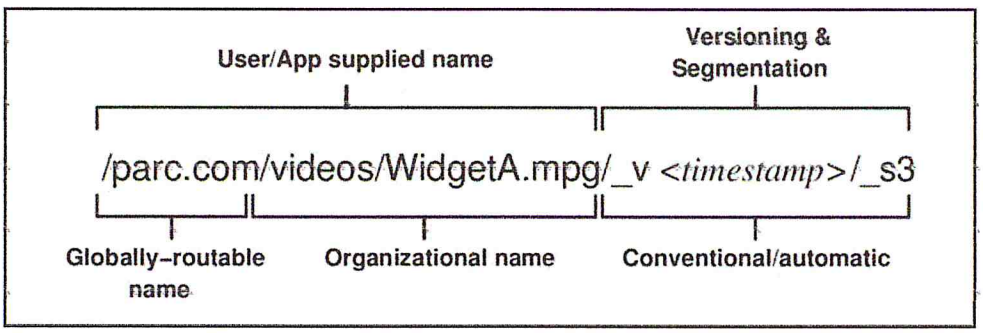


***Fig. I.6.*** *Exemple de nom hiérarchique de CCN.* **[JAC 09]**

De plus, l'architecture du CCN permettrait une recherche simple et directe d'un contenu sans avoir à identifier son détenteur, comme dans le cas des réseaux IP. Un utilisateur cherche une donnée à travers son nom. Dès lors que la requête est lancée, une demande sous forme d'un paquet dit "Intérêt" est envoyée à son routeur d'accès. Si la donnée n'est pas présente dans le Content Store (CS) de ce routeur, la requête se propage au fur et à mesure dans le réseau. Une fois la donnée trouvée, elle suit le chemin inverse de la requête de recherche jusqu'à l'utilisateur final et sera stockée dans un CS dans les routeurs CCN intermédiaires.

Cette architecture offre plusieurs possibilités de disponibilité indépendamment de l'adresse d'une machine. La sécurité est associée directement aux données et pas aux

"conteneurs" (liens, routeurs, serveurs, ...) ce qui permet d'ajuster de manière très flexible le niveau de sécurité à la nature du contenu en question. Plus intéressant encore, les contenus ne sont plus associés à des conteneurs précis mais peuvent être dupliqués à volonté et stockés notamment dans des mémoires caches au sein du réseau. On discutera en détail le fonctionnement et le routage de CCN.

### e. Comparaison :

Dans le tableau suivant, nous comparons les 4 projets selon les fonctions de base ci-dessus.

| | **DONA** | **NetInf** | **PSIRP** | **CCN** |
|---|---|---|---|---|
| **Début/fin de projet** | - 2007 | - 2010 / 2013 | - 2008 / 2010 | - 2009 |
| **Nommage** | - Plat | - Plat | - Plat | - Hiérarchique. |
| **Routage** | - Basé sur le nom. | - Basé sur le nom.<br>- Résolution des noms | - Basé sur le nom.<br>- Utilise modèle de résolution appelé « point de rendez-vous ». | - Basé sur le nom. |
| **Cache** | - Les RHs peuvent être activés avec un mécanisme de mise en cache universel. | - On-path caching<br>- Off-path caching.<br>- Peer caching. | - Les caches multiples d'un objet peuvent être maintenues en fonction de la portée du point de rendez-vous pour l'identifiant associé à l'objet. | - La mise en cache du contenu dans le CS d'un nœud.<br>- Le stockage des paquets est possible à chaque nœud NDN. |
| **Mobilité** | - L'architecture DONA s'appuie sur les protocoles de transport existants, c'est-à-dire TCP, pour fournir les mécanismes | - Différents mécanismes de transmission sont utilisés pour récupérer un objet de données, un localisateur ou des conseils de | - Le processus de transfert de base repose sur les filtres bloom.<br>- Chaque objet a un nom dérivé algorithmiquement unique à partir du | - L'architecture CCN ne fournit aucune fonctionnalité de couche de transport.<br>- Est fournie par l'application ou |

| | | | | |
|---|---|---|---|---|
| | d'acheminement et d'autres fonctionnalités de transport | redirection en utilisant des messages de demande / réponse. | nom d'origine, ce qui permet de gérer le contrôle de flux | certaines bibliothèques de support. |
| **Sécurité** | - L'espace de noms auto-certifié utilisé dans les attributs DONA à l'intégrité des noms-données | - L'espace de noms auto-certifié.<br>- la sécurité de l'objet est fournie par la cryptographie de clé publique | - L'espace de noms auto-certifié.<br>- cryptographie à courbe elliptique (ECC) pour vérifier la signature et l'authentification | - L'éditeur signe cryptographique-ment chaque paquet de données |
| **Code source de simulation** | | - openNetInf<br>- NetInf (nilib)<br>- GIN | - Blackadder | - CCN-lite.<br>- SCoNet.<br>- ccnSim<br>. CCNPL-Sim<br>- Mini-CCNx |

***Tab. I.1.*** *Comparaison entre des principaux projets ICN.*

Bien que ces architectures ICN (NetInf, DONA, PSIRP et CCN) diffèrent dans leurs détails, ils partagent plusieurs propriétés fondamentales : nom unique pour le contenu, routage basé sur le nom, mise en cache et assurance de l'intégrité du contenu.

En plus des autres initiatives prises par la communauté de recherche pour les futures architectures Internet, le CCN a récemment fait l'objet d'une attention particulière. Nous pouvons constater cela du nombre des codes sources disponibles qui permettent de simuler et d'implémenter les différents protocoles. Pour cette raison, nous nous intéressons à ce projet et à ce type de réseau CCN pour étudier les problèmes/solutions de sécurité du routage centré contenu. Par conséquent, nous détaillons le routage dans le CCN dans la section suivante et la sécurité dans le CCN dans le chapitre suivant.

### I.2.3 Routage dans CCN :

L'idée générale de CCN est d'adapter les messages envoyés sur Internet à ce qu'ils sont vraiment : le contenu. Au lieu de la restriction aux communications de bout en bout entre les paires d'utilisateurs, CCN permet un échange de messages beaucoup plus flexible et efficace.

Par rapport à l'Internet actuel, les routeurs sont très différents (voir *Fig I.7*). Ils contiennent essentiellement trois tables : Le stockage de contenu (Content Store), la table (PIT) et la table (FIB) **[WEI 14]**.

• ***Content Store (CS)*** **:** le CS est un cache (ou une mémoire tampon) installé dans les nœuds CCN. Lorsqu'un nœud reçoit des données (message Data), selon les stratégies de caches définies, le nœud peut sauvegarder une copie de ces données dans son CS pour répondre aux demandes ultérieures.

• ***Pending Interest Table (PIT)*** **:** la table PIT a deux fonctionnalités principales. La première est qu'elle mémorise temporairement des messages d'intérêt « Interest » que le nœud reçoit avant de les transmettre ensuite au nœud suivant. Grâce à cette table, en retour des données, le paquet Data peut suivre les chemins inverses et finalement arriver jusqu'aux clients demandeurs. Le deuxième rôle de PIT est d'éviter de multiple envoi des mêmes messages Interest. Lorsque plusieurs messages Interest qui demandent un même contenu arrivent sur un nœud, seul le premier est renvoyé pour chercher le contenu, les autres restent dans ce nœud et attentent la réception du contenu. Une fois reçues, les données seront retournées sur chacune des faces présentes dans la PIT.

• ***Forwarding Information Base (FIB)*** **:** la table FIB de CCN est similaire à celle d'IP. Elle est utilisée pour gérer les informations de transfert des paquets Interest vers des sources qui ont les contenus demandés. La table FIB est remplie par des publications de contenus qui sont publiés par des fournisseurs de contenu.

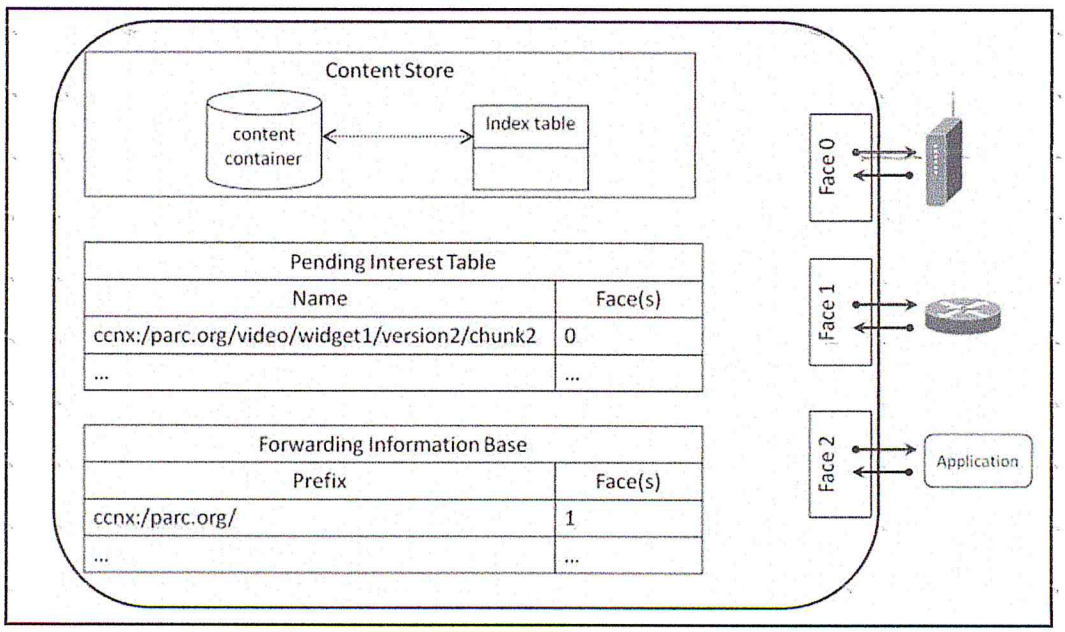


***Fig. I.7.*** *La structure d'un nœud CCN.* **[WEI 14]**

Dans un CCN, il n'y a nul besoin d'acheminer les adresses source et destination à travers le réseau pour récupérer la donnée. Le format des paquets Intérêts et Data est explicité à la *Fig. I.8*.

A la réception d'un Intérêt par un nœud, ce dernier vérifie si le chunk demandé existe dans son Content Store. Si c'est le cas, le paquet Data sera envoyé à l'interface demandeuse. Sinon le chunk demandé sera recherché dans le PIT. S'il est trouvé, l'interface demandeuse sera rajoutée au PIT. Si les deux bases de données ne fournissent aucune information, on cherchera dans le FIB si une entrée matche avec le chunk recherché. Alors le paquet Interest sera acheminé vers les interfaces conduisant à la donnée. La table PIT sera mise à jour avec une nouvelle entrée pour le chunk en question.

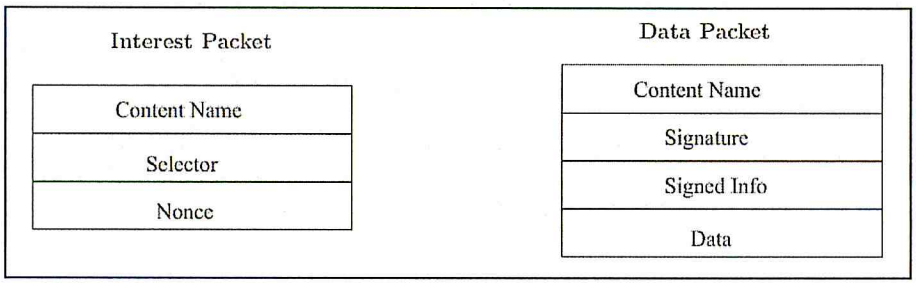


***Fig. I.8.*** *Format des paquets CCN*

Les contenus sont divisés en morceaux (chunks), chaque morceau a typiquement la taille d'un paquet IP. CCN respecte le déroulement logique d'une requête : un utilisateur demande une donnée en émettant des paquets de type "Interest" et reçoit en retour des paquets de données de type "Data". A chaque paquet Interest correspond un seul paquet Data et chaque paquet Data correspond à un Chunk.

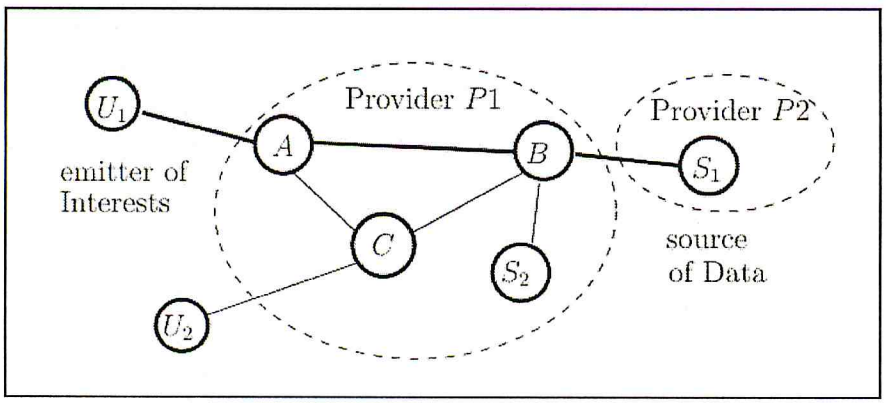


***Fig. I.9.*** *Un segment du réseau CCN reliant un utilisateur U1 à une source S* **[NAD 14]**

La *Fig. I.9* représente un segment d'un réseau CCN. Pour récupérer des données du fournisseur **P2**, l'usager **U1** envoie des paquets "Intérêt" pour le contenu demandé au travers

des routeurs **A** et **B**. Supposant que les Content Stores de **A** et **B** ne contiennent pas le document demandé, les paquets Data suivent le chemin inverse de **S1** vers **U1** en passant par **B** et **A**.

A la réception d'un paquet Data par un nœud, une recherche est effectuée dans le Content Store. Si une entrée correspond, alors le paquet reçu est supprimé, car ceci implique que le chunk est déjà livré à toutes les interfaces demandeuses. Sinon la donnée sera recherchée dans le PIT. Si une entrée matche avec la donnée reçue, elle sera acheminée vers les interfaces demandeuses. Le chunk sera typiquement stocké en même temps dans le Content Store.

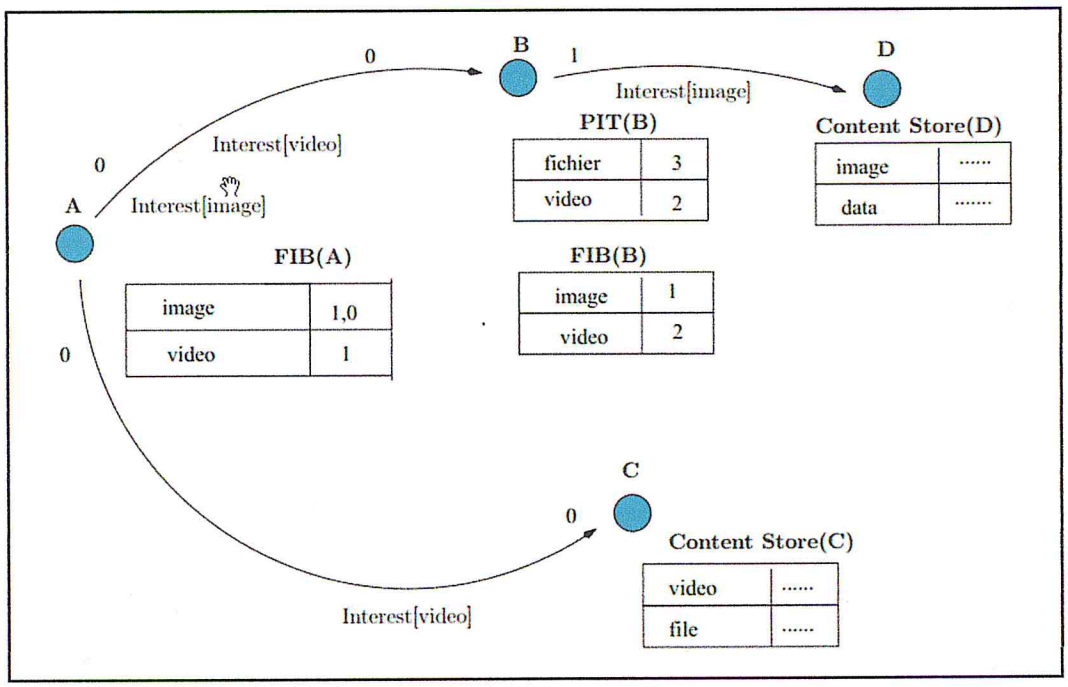


***Fig. I.10.*** *La recherche des données dans CCN* **[NAD 14]**

Dans l'exemple de la *Fig. I.10*, le nœud **A** cherche les chunks "vidéo" et "image". Le FIB du nœud **A** indique que les paquets Intérêts doivent être acheminés vers l'interface **0** et **1** pour l'image, et vers l'interface **1** pour la vidéo. A la réception de l'Intérêt demandant la vidéo par le nœud **B**, ce dernier ignore l'intérêt reçu car le PIT contient déjà une entrée.

Cette entrée est mise à jour. Cependant, quand le nœud **B** reçoit l'intérêt demandant l'image, il l'envoie à l'interface **1** indiquée par le FIB. Le nœud **D**, par la suite, achemine la donnée vers le nœud **A**. Cette donnée sera stockée dans tous les Content Stores des nœuds l'ayant reçu ou transité.

## I.3. Sécurité dans les CCNs :

ICN améliore plusieurs faces de l'expérience d'utilisateur mais comporte plusieurs défis, de la sécurité, de la confidentialité et des contrôles d'accès. Dans ce qui suit, nous allons présenter les problèmes de sécurité de cette nouvelle architecture. Ensuite, nous présentons les solutions de sécurités offertes par l'architecture CCN.

### I.3.1. Attaques dans les ICNs :

#### a. Classification :

Dans la littérature, plusieurs travaux (**[ESL 15] ,[REZ 16]**) ont essayé de recenser les différentes attaques ICN et de proposer une classification. Parmi ces travaux, nous nous sommes intéressés à l'article **[ESL 15]** qui classe les attaques ICN (nouvelles et anciennes) en quatre catégories, comme le montre dans la *Fig. I.11* : nommage, routage, mise en cache et autres attaques diverses. Cette classification dépend de la cible principale de l'attaquant. Bien que chaque attaque soit incluse dans une seule catégorie, elle peut également influer sur d'autres catégories. Par exemple, flooding et les attaques unpopular requests (requêtes impopulaires) affectent le routage et la mise en cache ICN. Dans une attaque flooding, L'objectif principal de l'attaquant est de surcharger et d'échapper les ressources de routage et par conséquent, elle affecte le système de mise en cache. Dans unpopular requests, la cible principale de l'attaquant est de violer la pertinence du cache et par conséquent, elle affecte le système de routage.

Les catégories proposées sont brièvement présentées dans les quatre paragraphes suivants :

- ***Les attaques liées au nommage*** : Les architectures ICN sont plus menacées par rapport à la vie privée vu que les demandes de contenu sont visibles pour le réseau. De nombreux attaquants tentent de surveiller l'utilisation d'Internet. Une architecture ICN offre plus d'accès aux demandes des utilisateurs ce qui augmenterait le contrôle des attaquants sur le flux d'informations et rendrait les informations de blocage beaucoup plus faciles. Dans les attaques liées à la nomination dans ICN, un attaquant tente d'empêcher la distribution d'un contenu spécifique en bloquant la livraison de ce contenu et / ou en détectant qui demande ce contenu.

- ***Les attaques liées au routage :*** La distribution de contenu ICN dépend d'une publication et d'un abonnement asynchrone, ce qui ajoute des efforts supplémentaires pour

assurer la cohérence entre les états de données distribuées. Certaines attaques comme jamming et la synchronisation visent à échouer cette cohérence de l'état, ce qui peut entraîner des flux de trafic indésirables et / ou un déni de service. D'autres attaques, comme l'infrastructure et les attaques par inondation (flooding), essayent d'épuiser les ressources comme la mémoire et le pouvoir de traitement qui sont utilisés pour soutenir, maintenir et échanger des états de contenu. En outre, l'infrastructure de l'ICN repose sur l'intégrité et l'exactitude du routage du contenu, et est donc menacée par injections toxiques de chemins et de noms.

- ***Les attaques liées à la mise en cache :*** Le cache est l'un des composants importants dans ICN car la performance de son infrastructure est basée sur la mise en cache du récepteur qui vise à fournir la copie la plus proche disponible à un utilisateur. Par conséquent, ICN est vulnérable à toutes les opérations qui polluent ou corrompent le système de mise en cache.

- ***Des Attaques diverses*** : Les menaces dans cette catégorie visent à dégrader certains services ICN et à permettre à un attaquant de faire un accès non autorisé. Ces attaques entraînent une distribution de données insuffisante ou erronée.

Il y'a plusieurs attaques dans cette catégorie comme : Maltraitance des paquets (Packet Mistreatment), violation de clé de signature (Breaching Signer's Key) et L'accès non autorisé.

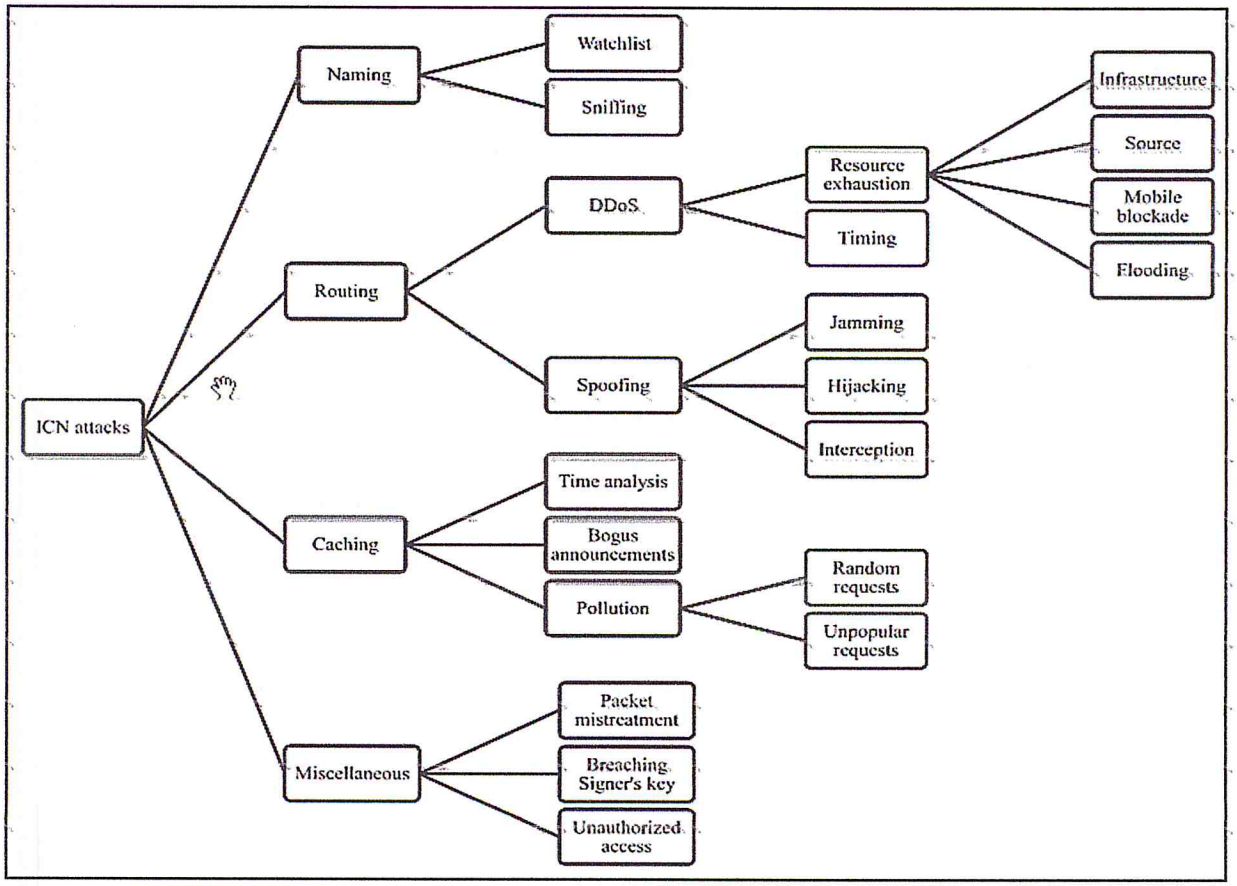


***Fig. I.11.*** *Classification des attaques dans le réseau ICN.* **[ESL 15]**

### b. Attaques liées au routage dans le réseau ICN

Les attaques liées au routage ont un impact sur les éléments suivants :

• ***Déni de service*** **:** Les DoS peuvent survenir en raison de nombreuses attaques dans cette catégorie, telles que l'envoi de nombreuses demandes de contenu indisponible ou d'une seule source, de blocage mobile, d'inondation (flooding), de Hijacking et de Timing. En conséquence, les temporisations intermédiaires suppriment les requêtes avec les délais expirés, ce qui peut entraîner des DoS ou au moins de longs délais.

• ***L'exténuation des ressources*** **:** Il existe de nombreuses sources d'exténuation des ressources dans l'infrastructure ICN qui proviennent d'un mauvais usage ou d'un trafic incontrôlé, comme l'envoi d'un grand nombre de requêtes et d'inondations.

• ***L'infiltration du chemin*** **:** Dans ICN, des copies de contenu sont généralement distribuées à de nombreux sites non approuvés, et il est donc difficile d'authentifier des origines valides pour le contenu. Le Hijacking et l'interception sont les principales sources d'infiltration de chemin dans le ICN, car les attaquants peuvent annoncer des routes non valides et les réclamer comme faisant confiance.

• ***Intimité*** **:** La violation de la vie privée (privacy) dans l'attaque d'interception donne à l'attaquant un accès non autorisé aux demandes de l'utilisateur, surtout lorsque l'attaquant est topologiquement proche ou sur la route vers l'utilisateur.

De ce fait, les attaques dans cette catégorie peuvent être classées dans les attaques de déni de service distribué (DDoS) et les attaques de Spoofing (attaques de falsification). Les attaques DDoS peuvent être classées dans l'épuisement des ressources et les attaques temporelles. L'épuisement des ressources peut être classé en blocs d'infrastructure, de source, de blocage mobile et les attaques de flooding. Les attaques par flooding peuvent être divisées en attaques de Jamming, de Hijacking et d'interception.

- ***Infrastructure***

Un attaquant envoie un grand nombre de demandes de contenu disponible/ indisponible. Lorsque les architectures ICN tentent de trouver la copie la plus proche du meilleur emplacement disponible, ces requêtes empruntent différentes routes vers la source provoquant des conditions de surcharge. Si le nombre de ces demandes est significativement

élevé, cela entraîne un déni de service. Cette attaque peut être amplifiée, car les utilisateurs réguliers envoient des demandes de retransmission après un temps spécifié. Cette menace peut être atténuée car les mécanismes de routage dans ICN tentent de se diriger vers plusieurs emplacements. Comme l'illustre la *Fig. I.12*, l'attaquant, qui contrôle de nombreux systèmes finaux, envoie un grand nombre de requêtes à un ou plusieurs routeurs ICN pour remplir la table de routage et les ressources de traitement et de mémoire d'échappement. En conséquence, les routeurs attaqués transmettent ces requêtes aux nœuds voisins, qui, à leur tour, les transmettent aux prochains nœuds voisins et ainsi de suite. Si le nombre de demandes invalides est élevé, toute demande légitime prend un temps de réponse plus long. Par conséquent, si le temps de réponse dépasse la période d'expiration de la demande, la demande peut ne pas être répondu. Ce scénario peut entraîner un déni de service ou au moins de longs délais.

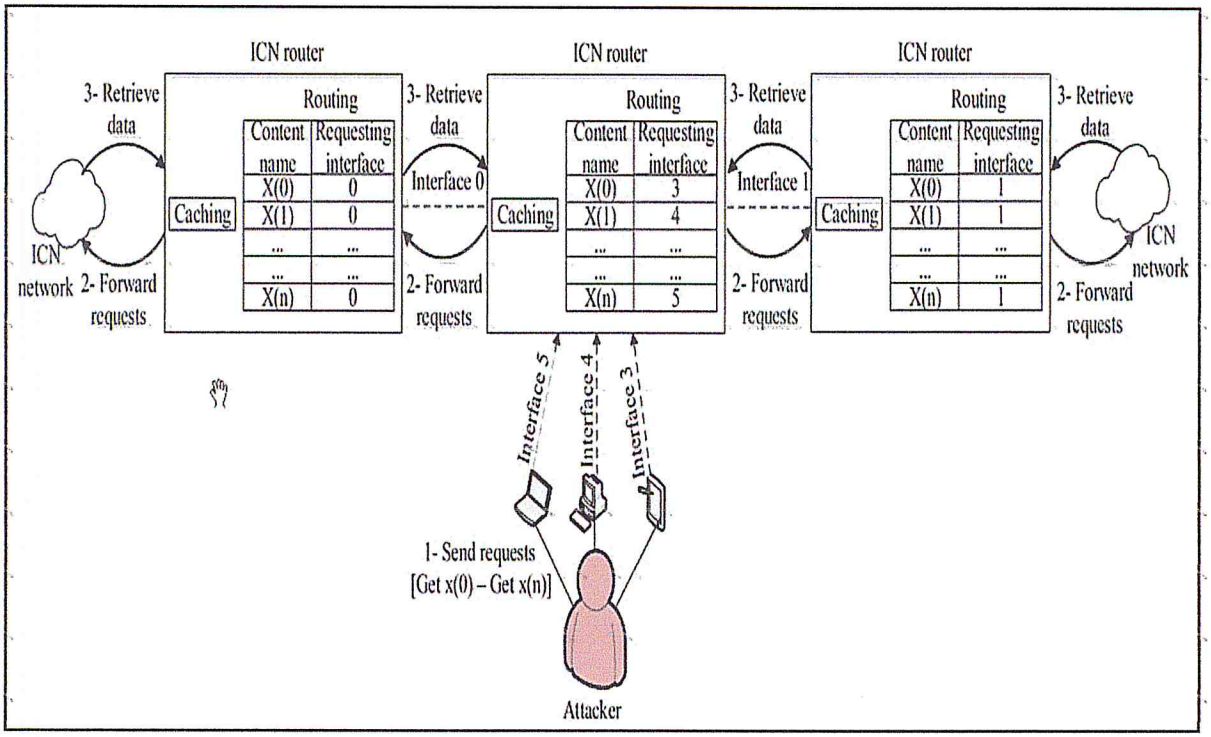


***Fig. I.12.*** *Attaque d'infrastructure.* **[ESL 15]**

Scénario de l'attaque :

1- Un attaquant, qui contrôle de nombreux systèmes finaux, envoie un grand nombre de requêtes aux routeurs ICN.
2- Les routeurs attaqués transmettent ces demandes aux routeurs voisins et, à leur tour, les envoient à leurs routeurs voisins, etc.
3- ICN commence à récupérer ces grandes quantités de données de différents chemins et l'envoie aux emplacements demandés.

- ***Source***

Dans ICN, l'attaque d'une seule source peut également entraîner des conditions de surcharge pour l'infrastructure de routage. Un attaquant envoie un grand nombre de requêtes à une source de contenu spécifique pour dégrader ses performances. En conséquence, cette attaque augmente le temps de réponse de la livraison de contenu pour cette source de contenu ou son routeur d'accès. En plus de cet effet, l'attaque peut réduire le taux de retour des données et affecter les demandes de tous les nœuds dans les chemins vers les récepteurs. Le scénario d'attaque est similaire au scénario d'attaque d'infrastructure. Cette attaque affecte non seulement la source attaquée, mais affecte également le réseau global.

- ***Blocage mobile***

Un attaquant mobile peut surcharger une région en traversant des réseaux voisins sur des chemins circulaires tout en envoyant un nombre important de demandes de contenu. L'attaquant vise à surcharger les routeurs d'accès mobiles pour qu'il dépasse le délai d'exécution de l'état qui entraîne un blocage des réseaux régionaux. La retransmission des demandes fait partie de l'aspect de la mobilité dans un environnement ICN qui ajoute de la difficulté à détecter cette attaque.

Le scénario d'attaque présenté à la *Fig. I.13* est similaire au scénario d'attaque d'infrastructure. La différence est que l'attaquant mobile envoie un nombre élevé de requêtes aux réseaux voisins, alors que l'attaquant se déplace entre les réseaux de façon circulaire et continue.

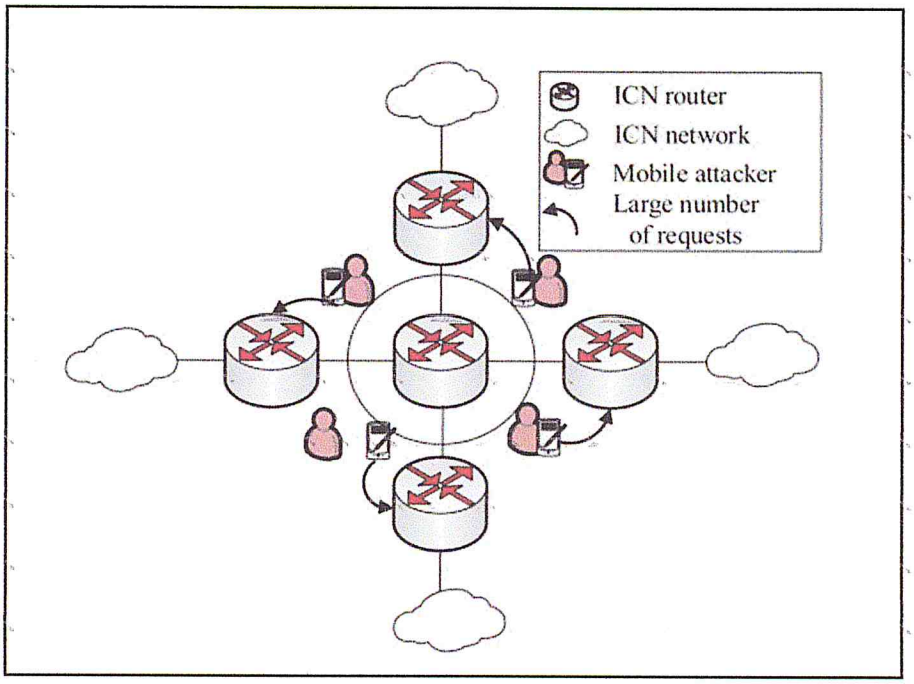

***Fig. I.13.*** *Attaque de blocage mobile : un attaquant mobile envoie un grand nombre de requêtes, alors qu'il parcourt les réseaux voisins ICN.* **[ESL 15]**

- ***Flooding***

Les solutions existantes pour les attaques d'inondation dans ICN sont conçues pour limiter le nombre de requêtes, ce qui n'est pas approprié pour ICN. Un attaquant peut envoyer un grand nombre de demandes qui dépassent cette limite. Le nœud attaqué accepte un certain nombre de demandes et ignore les demandes restantes. En conséquence, l'attaquant réussit à surcharger l'infrastructure globale et nuit à tous les utilisateurs à proximité. En outre, comme ICN est une architecture centrée sur le contenu, il est difficile d'appliquer des limites pour le taux de demande par utilisateur final car il n'y a pas d'identifiant hôte. Le scénario d'attaque est également similaire au scénario d'attaque d'infrastructure. La différence est que l'attaquant envoie un certain nombre de demandes qui dépassent les limites des nœuds ICN et, par conséquent, ICN néglige les demandes légitimes dirigées vers les nœuds attaqués.

- ***Timing***

Il s'agit d'augmenter le délai d'attente de la demande pour certains nœuds ICN pour violer la cohérence entre la publication asynchrone ICN et le processus d'abonnement. Un attaquant envoie un grand nombre de demandes pour dégrader la performance de certains routeurs, de sorte que le routage des demandes et le transfert de données présentent des retards plus longs. Le scénario d'attaque est également similaire au scénario d'attaque d'infrastructure. La différence est que l'attaquant envoie un grand nombre de requêtes à travers une ou plusieurs routes pour augmenter le délai d'attente de la demande pour les demandes des utilisateurs légitimes.

- ***Jamming***

Un nœud sur un lien partagé envoie un grand nombre de demandes de contenu inutiles malveillantes. L'attaquant qui se masque en tant qu'abonné approuvé envoie les demandes malveillantes pour perturber le flux d'informations dans le système. Le réseau ICN répond et le contenu est envoyé à la destination sans récepteur. Ce scénario d'attaque est similaire au scénario d'attaque d'infrastructure. La différence, telle que présentée à la *Fig. I.14*, c'est que l'attaquant envoie des requêtes à un nœud partagé, qui l'envoie aux nœuds voisins.

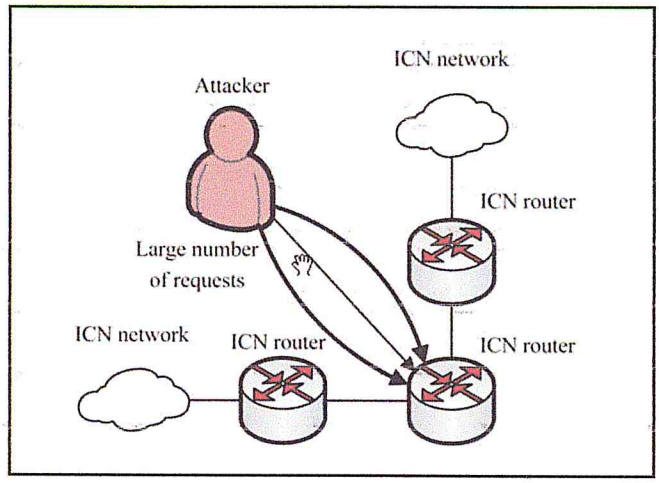


***Fig. I.14.*** *Attaque de Jamming* **[ESL 15]**

- ***Hijacking :***

Contrairement aux architectures centrées sur l'hôte, tout nœud dans ICN peut mettre en cache et publier/souscrire des contenus. Un attaquant qui se masque en tant qu'éditeur de confiance peut annoncer des routes non valides pour tout contenu. Les demandes de contenu d'utilisateurs à proximité de l'attaquant sont orientées vers ces routes non valides. Par conséquent, ces demandes seront sans réponse, ce qui entraîne une DoS. L'effet de cette attaque peut être exacerbé, si l'attaquant a la possibilité de détourner des routes invalides à grande échelle. L'effet de cette attaque diminue parce que les mécanismes de routage dans ICN tentent de se diriger vers plusieurs emplacements.

Comme le montre la *Fig. I.15*, l'attaquant annonce des itinéraires non valides pour certains contenus pour attirer les demandes des utilisateurs. Lorsque les utilisateurs légitimes envoient des demandes pour l'une de ces routes malveillantes, les nœuds ICN transmettent ces requêtes aux nœuds malveillants. En conséquence, l'utilisateur légitime ne reçoit pas de réponse.

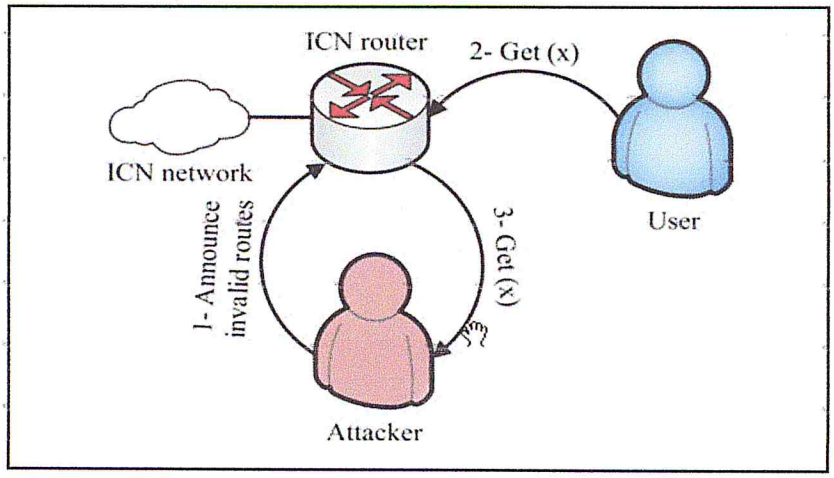


***Fig. I.15.*** *Attaque de Hijacking* **[ESL 15]**

Scénario de l'attaque :

1- Un attaquant annonce des routes non valides pour certains contenus, y compris (x).
2- Une demande d'utilisateur pour le contenu ICN nommé (x).
3- Le routeur ICN redirige les requêtes de l'utilisateur vers les routes malveillantes de l'attaquant et, à la suite, l'utilisateur ne reçoit aucune réponse.

- ***Interception :***

Cette attaque est semblable à l'attaque habituelle "l'homme du milieu ". Contrairement à une attaque de détournement (Hijacking), un attaquant qui se masque en tant qu'éditeur de confiance annonce des routes invalides, tout en conservant un registre des routes valides du contenu. Les demandes de contenu peuvent ensuite être capturées et envoyées à l'emplacement approprié. Bien que le récepteur reçoive le contenu normalement, l'attaquant obtient une connaissance du contenu demandé.

Comme le montre la *Fig. I.16*, l'attaquant annonce des itinéraires non valides pour certains contenus pour attirer les demandes de l'utilisateur. Lorsque les utilisateurs légitimes envoient des demandes pour l'une des routes malveillantes, les nœuds ICN transmettent ces requêtes au nœud malveillant de l'attaquant. L'attaquant enregistre qui a demandé ce contenu puis l'envoie pour obtenir les données réelles. Lorsque les données réelles arrivent au nœud de l'attaquant, l'attaquant l'envoie de nouveau au nœud ICN demandé, qui l'envoie à son tour à l'utilisateur légitime. Pour l'utilisateur, le scénario semble être normal, mais en fait, l'attaquant viole intimité et la vie privée de l'utilisateur.

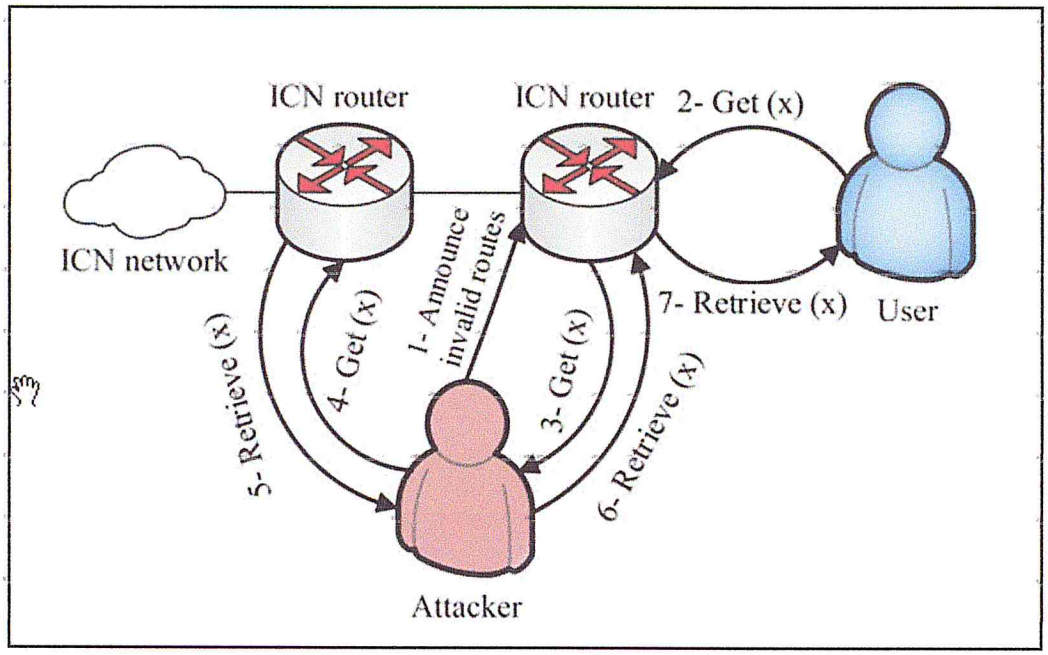


***Fig. I.16.*** *Attaque de Interception* **[ESL 15]**

Scénario de l'attaque :

1- Un attaquant annonce des routes non valides pour certains contenus, y compris (x).
2- Une demande d'utilisateur pour le contenu ICN nommé (x).
3- Le routeur ICN redirige la demande de l'utilisateur vers les routes malveillantes de l'attaquant.
4- L'attaquant transmet la demande pour obtenir le contenu réel.
5- L'attaquant récupère le contenu (x).
6- L'attaquant transmet le contenu à l'utilisateur demandé.
7- L'utilisateur récupère le contenu (x)...

### I.3.2 Solutions de Sécurité offertes par le CCN :

Dans ce qui suit, nous décrivons comment la sécurité et la vie privée des utilisateurs bénéficient de l'architecture de l'CCN. Cela comprend l'intégrité vérifiable avec le contenu signé, l'absence d'adresses dans les intérêts, l'abandon de la résolution des noms en faveur de noms hiérarchiques et un certain degré de protection contre les attaques de déni de service (DoS).

#### a. L'intégrité vérifiable :

Dans les CCNs, il est obligatoire pour les fournisseurs de contenu de signer le contenu et le nom avec leur clé privée. Par conséquent, le destinataire peut vérifier l'intégrité des

données et de la source. Cela empêche le spoofing comme c'est possible dans l'Internet actuel. Un effet secondaire positif est la sécurité de routage. Comme tous les autres contenus, la communication entre les routeurs doit être signée. Cela empêche les adversaires d'influencer les tables de routage, à savoir la FIB dans CCN.

### b. L'absence de l'adresse :

CCN abandonne totalement les adresses. Ni les intérêts ni les messages de livraison de contenu ne contiennent des adresses. Les intérêts dans CCN's contiennent uniquement le nom du contenu demandé, mais pas celui qui l'a demandé. Seul le premier routeur d'acheminement connaît l'interface à partir de laquelle le contenu a été demandé. Tous les autres routeurs connaissent uniquement le routeur précédent sur le chemin de transfert. Lorsque le fournisseur de contenu renvoie le contenu, son message comprend également le nom du contenu, sa signature, l'ID de l'éditeur et des informations sur l'endroit où récupérer la clé publique de l'éditeur. Par conséquent, il n'est pas nécessaire d'adresser des adresses, bien que l'éditeur puisse être déduit de l'ID et de la clé. Du point de vue de la vie privée, ce design améliore l'anonymat car la source d'un intérêt est inconnue ou au moins difficile à découvrir.

Mais ce différent paradigme de communication a encore plus d'avantages. Dans l'Internet actuel, de nombreuses attaques exigent que l'attaquant envoie directement des messages à la victime. Dans CCN, la victime n'a pas d'adresse, donc cette menace est complètement atténuée car aucun contenu ne peut arriver à un hôte sans que l'hôte ne demande le contenu à l'avance. Il est légèrement différent pour les éditeurs de contenu.

### c. La Protection contre le déni de service :

Outre la protection contre les attaques directes sur les hôtes, CCN offre même une protection contre les attaques de déni de service (DoS) sur les fournisseurs de contenu. Supposons qu'un attaquant lance une attaque DoS sur un éditeur en envoyant beaucoup d'intérêts. Ceux-ci sont simplement collectés au premier routeur et un seul intérêt est transmis. Même si l'attaquant contrôle un grand botnet qui est distribué sur de nombreux endroits qui ne partagent pas les routeurs sur le chemin de routage vers l'éditeur, l'agrégation des intérêts atténue la plupart des attaques. Ce qui s'est réellement passé est un changement dans le point d'attaque du fournisseur de contenu aux routeurs. Au lieu d'inonder l'éditeur, l'attaque remplit les tables d'intérêt en attente des routeurs. Certains chercheurs ont suggéré d'ajouter un drapeau à des intérêts qui indiquent que l'intérêt devrait être acheminé vers le fournisseur de

contenu sans renvoyer de contenu mis en cache. Un tel drapeau permet aux attaquants d'inonder directement les éditeurs à nouveau.

### d. La résolution du nom :

CCNs est basé sur des noms hiérarchiques et ne nécessite donc pas de définition de nom explicite. Dans l'Internet actuel, la résolution des noms est fournie par DNS et offre une multitude de possibilités pour les attaquants à exploiter, par ex. Le détournement de DNS. Si les noms forfaitaires sont utilisés à la place, l'avantage est perdu.

## I.3.3. Sécurité CCN versus Sécurité d'Internet :

Même si l'architecture est différente par rapport l'Internet actuel, certains problèmes n'ont pas été résolus. Plusieurs d'entre eux ont à voir avec la surveillance. Le contournement de la surveillance - ou plus généralement la confidentialité - n'est pas définie comme principe de conception primaire de CCNs, de sorte que des techniques supplémentaires devront être construites sur l'architecture pour la soutenir.

Premièrement, le fournisseur de contenu est encore facilement identifiable. Bien qu'il soit possible de le faire via son adresse IP dans l'Internet actuel, CCN l'oblige à signer tout élément de contenu avec cette signature. En conséquence, les censeurs peuvent vérifier si le contenu arrivant provient d'une source indésirable et peut-être le filtrer. Deuxièmement, les censeurs sont actuellement en mesure de bloquer les relais connus de Tor[1] et de tester les hôtes de manière proactive afin de savoir s'ils sont des ponts Tor. S'ils suspectent un hôte, ils peuvent simplement se comporter comme s'ils étaient un utilisateur bénin et essayer d'accéder à Tor par le pont. Si cela fonctionne, ils ont identifié un pont Tor. Dans CCNs, les réseaux d'anonymat sont également envisageables. Troisièmement, les censeurs peuvent filtrer les intérêts pour les mots-clés ou les préfixes interdits dans leurs noms. D'autres techniques comme Deep Packet Inspection (DPI) sont également applicables. Enfin, des tiers de confiance comme les Autorités de certification (CA) dans l'Internet actuel seront toujours nécessaires. Même si chaque élément doit être signé par l'éditeur, le destinataire a besoin d'un tiers de confiance fournissant la clé publique afin de pouvoir vérifier l'intégrité du contenu **[ROM 16]**.

[1] **TOR :** The Onion Router est un réseau informatique superposé mondial et décentralisé. Ce réseau permet d'anonymiser l'origine de connexions TCP.

## I.4. Conclusion :

Dans ce chapitre, nous avons présenté en premier temps les principaux projets d'ICNs, ses différentes fonctions de base de chaque projet et une petite comparaison entre ces projets avant de détailler le routage dans le CCN qui nous intéresse dans notre travail pour étudier ses problèmes de sécurité. Par la suite, nous avons discuté la sécurité dans les ICNs en général et en particulier la sécurité native offerte par l'architecture de CCN. Bien que l'architecture CCN offre plusieurs solutions pour la protection de la sécurité et de la vie privée que l'Internet actuel, notre travail aborde en générale les aspects de sécurité liés aux routages qui pourrait être implémentée sur CCN. Nous avons choisi le Flooding et le spoofing comme des types d'attaque à étudier. Ces attaques sont également connues dans la littérature comme Interest Flooding Attack, interception et le hijacking, ce qui signifie que le routage des données et leurs confidentialités sont toucher.

# Chapitre II : Interest Flooding Attack

## II.1 Introduction :

Après l'apparence de l'architecture réseau CCN dans le monde informatique, plusieurs attaques sont apparues par exemples des attaques liées à la mise en cache, des attaques liées au routage, c'est pour ses raisons que la partie sécurité a une grande importance sur ce domaine. Le type d'attaque que nous allons traiter est l'attaque d'inondation des intérêts IFA qui est une attaque liée au routage.

Jusqu'à maintenant les recherche se déroule de manière continue sur cette problématique. Parmi les contre-mesures qui sont proposer contre ce type d'attaque, des algorithmes qui contrôle le flux de paquet sur Le réseau CCN, et aussi des algorithmes qui sont inspirer des solutions existantes pour ddos (attaque par dénis de service) dans les réseaux IP (internet protocol).

## II.2 Interest Traceback :

Parmi les contre-mesures destinées à atténuer les attaques DDoS sur les CCN, il existe une appelée Interest Traceback, inspiré de IP Traceback décrit dans **[SAV 00]** et **[SNO 01]**. Cette dernière a pour but de trouver l'origine réelle d'un paquet IP (le trafic de l'attaquant peut être rempli avec des adresses source usurpées), il est significatif dans le traitement des attaques DDoS. Mais la traçabilité IP n'est pas triviale, elle est difficile et coûteuse à mettre en œuvre. Cependant, avec l'aide de PIT, CCN prend en charge la traçabilité des intérêts. Nous pouvons facilement retracer l'origine réelle d'un paquet d'intérêt qui demande le contenu inexistant en générant un paquet de données spoofé pour le satisfaire, et cet émetteur est un attaquant potentiel. Une fois l'attaquant identifié, nous limitons le débit de l'interface reliant cet attaquant à son routeur d'accès, afin de réduire les paquets d'intérêt attaquants entrant dans le réseau.

Interest Traceback se résume à la fonction FindAndSend() voir figure 1. Cette dernière est déclencher si la taille du PIT est supérieure à une taille prédéfinie.

Voici le pseudocode de Interest Traceback montrer ci-dessous :

```
Void   Traceback::FindAndSend()
{
        FOR EACH Entry in Pit
                IF IsOld(Entry)
```

```
                FOR EACH Face in Entry.FacesList
                    IF Face.IsConnectedToEndUser()
                        Block Face
                    ELSE
                        GENRATE SpoofedData
                        SEND SpoofedData through Face
                    END IF
                END LOOP
                RELEASE memory
            END IF
        END LOOP
}
```

La fonction FindAndSend() se compose de 5 sous-fonction :

- La première est IsOld(entry) qui vérifier l'ancienneté d'un intérêt en utilisant un paramètre prédéfini.
- La fonction IsConnectedToEndUser() qui vérifier le type de la machine (router ou terminal), connecter sous cette interface en utilisant un paramètre prédéfinis.
- La fonction block(), bloque ou attenue le débit de la bande passante du terminal en question.
- La fonction Generate(), génère un paquet falsifier.
- La fonction Send(), envoie le paquet falsifier au destinataire.
- La fonction Release(), libère la mémoire en supprimant du PIT l'intérêt traité.

Afin de mieux comprendre le comportement de la contre-mesure Interest Traceback, nous avons essayé de représenter les différentes étapes de son déroulement dans l'organigramme suivant, voir la figure 1.

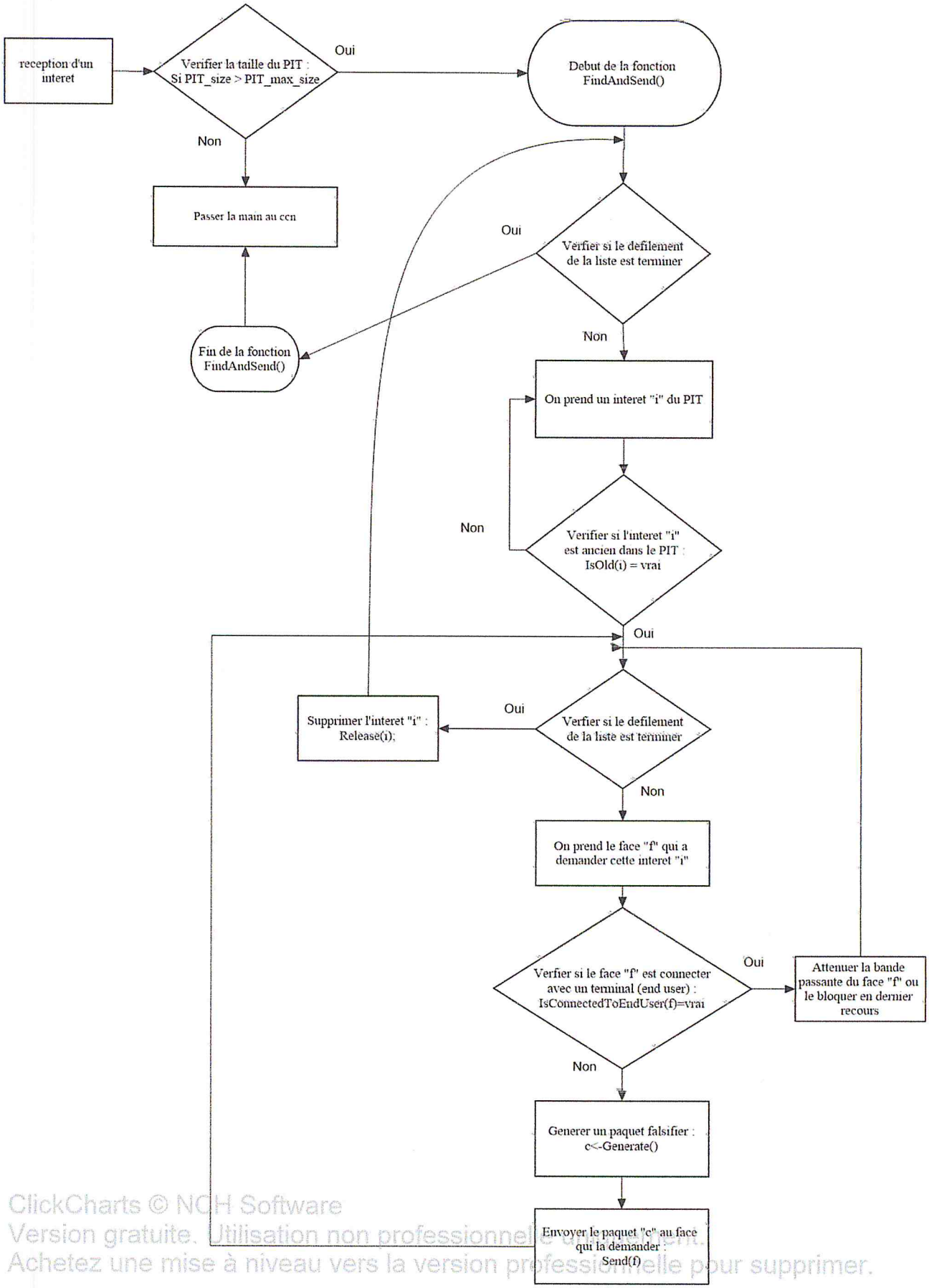




***Fig.II.1.*** *Organigramme de Interest Traceback.*

### II.2.1 Déroulement sur le réseau :

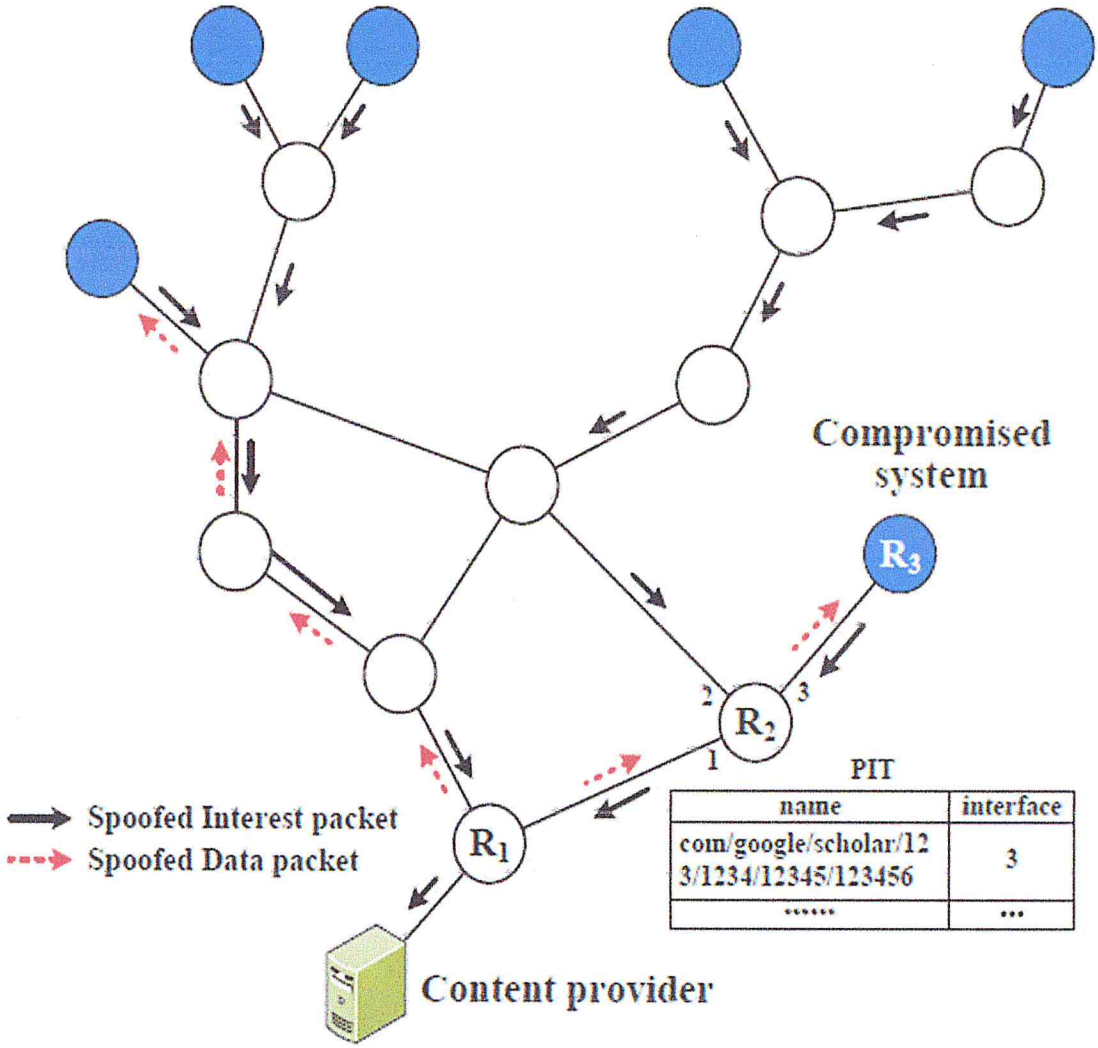


***Fig.II.2.*** *Les systèmes compromis envoient des paquets d'intérêt qui cible le même fournisseur de contenu* ***[DAI 13].***

Selon **[DAI 13],** Lorsque la taille PIT augmente à un rythme alarmant ou dépasse un seuil, le processus de retraçage d'intérêt est déclenché et fonctionne comme suit :

Un routeur répond à une attaque en remontant vers les initiateurs des intérêts, à savoir les attaquants, en générant des paquets de données usurpés pour satisfaire les paquets d'intérêt longtemps insatisfaits dans le PIT. Ces paquets de données usurpés sont remplis avec les mêmes noms falsifiés que dans les paquets d'intérêt, comme l'illustre la figure.II.3.

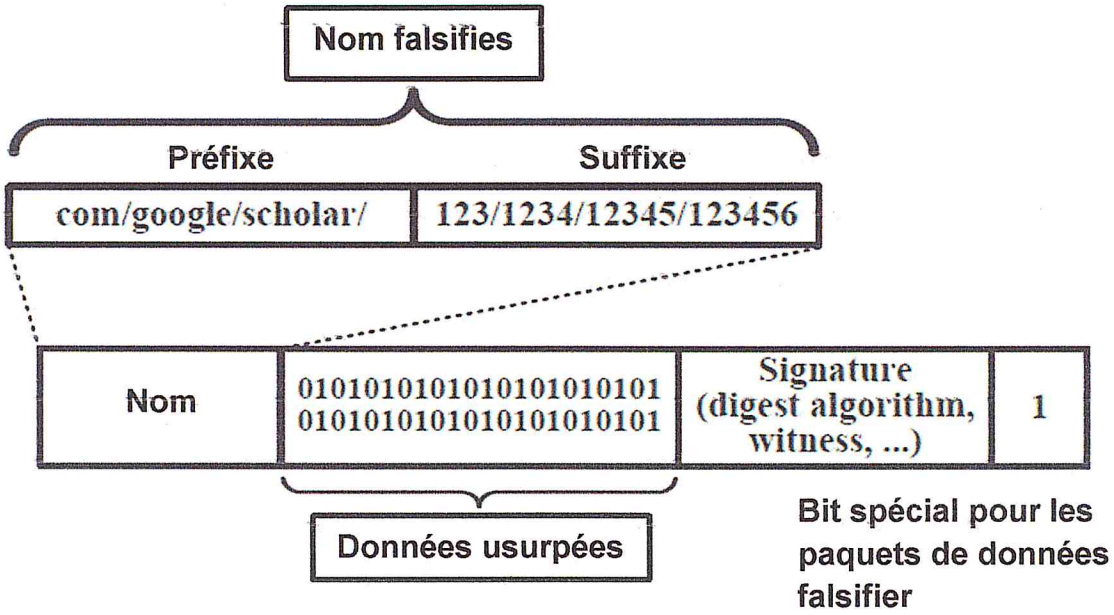


***Fig.II.3.*** *Un paquet de données avec un nom et un contenu falsifié* ***[DAI 13].***

Le routeur envoie les paquets de données falsifiés à l'expéditeur, en recherchant dans les PIT des routeurs intermédiaires. Dans notre cas, le routeur Edge *R*1 de la figure.II.2 Envoie un paquet données usurpées pour satisfaire l'intérêt de l'attaquant, et il est transmis au routeur *R*2. Le routeur *R*2 recherche le nom du contenu du paquet de données usurpées dans son PIT, et transmet le paquet de données usurpées à l'interface 3. Enfin, ce paquet de données usurpées est transmis à l'expéditeur du paquet d'intérêt attaquant *R*3. De même, le routeur *R*1 de la figure.II.4 envoie des paquets de données usurpés pour remonter à l'attaquant. Lorsque le paquet de données usurpé arrive à l'interface par laquelle le paquet d'intérêt attaquant entre dans le réseau, le routeur périphérique est averti que l'hôte directement connecté à cette interface est un attaquant. Par la suite, le routeur de périphérie bloque ou limite le débit de paquets entrant de cette interface en supprimant les paquets d'intérêt, atténuant ainsi l'attaque DDoS. Par souci de brièveté, la figure.II.4 ne montre que deux chemins de traçage flèches aux attaquants, les autres attaquants peuvent également être retracés de la même manière. La figure.II.4 décrit qu'un routeur retrace les trois attaquants.

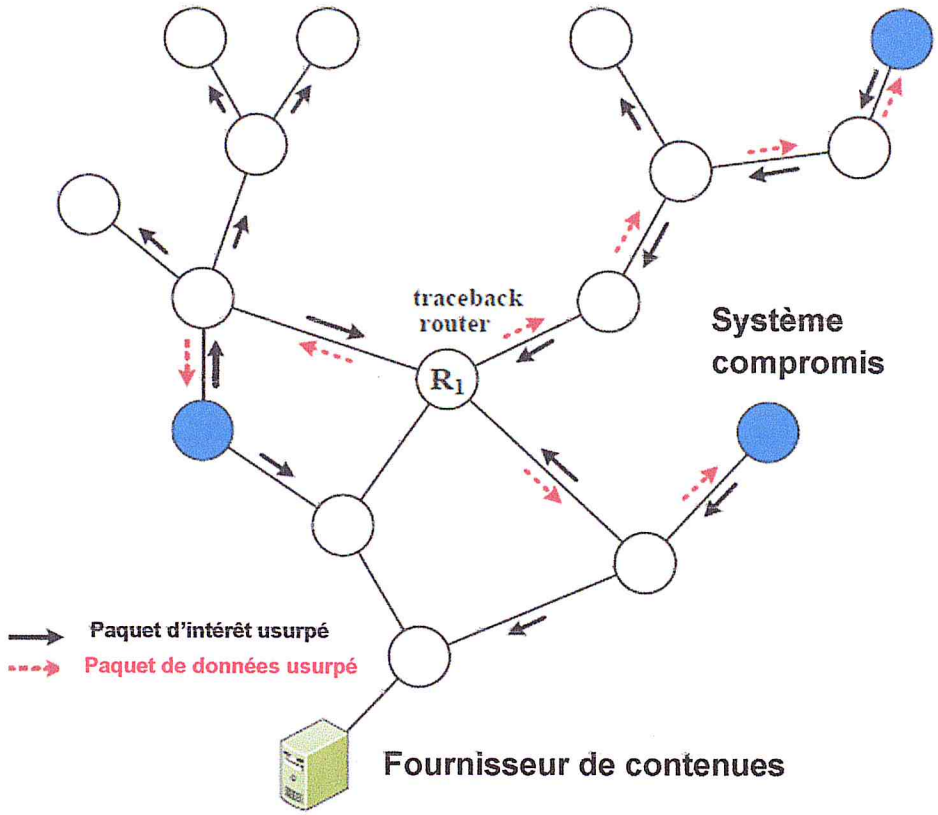


***Fig.II.4.*** *Attaques d'inondation d'intérêt faite pas des systèmes compromis* ***[DAI 13].***

### II.2.2 Paramétrage de l'algorithme :

Dans **[DAI 13]** et **[MAT 15]** les paramètres de l'algorithme sont différents car leurs scenarios d'attaque et leurs topologies sont différentes, c'est pour cette raison que dans notre cas on va paramétrer l'algorithme en faisons plusieurs tests, pour ainsi déduire le meilleur paramètre, les tests qui suivent ont été faite sur la topologie (voir figure.III.3).

Comme déjà expliqué dans la description de Interest Traceback, nous avons 3 paramètres à fixer à savoir.

- Seuil de PIT pour déclencher Interest Traceback.
- Temp de vie d'un intérêt dans le PIT.
- Seuil de nombre de contenue falsifier d'un interface pour le considérer comme étant un attaquant.

➢ **Seuil de faux contenu :**

On commence par faire un jeu de teste pour chercher le meilleur seuil du nombre de faux contenue, le seuil du PIT pour déclencher Interest Traceback et le temp de vie d'un intérêt

dans le PIT sont paramètrer de manière statique, leur valeur est respectivement 50 et 1 milliseconde.

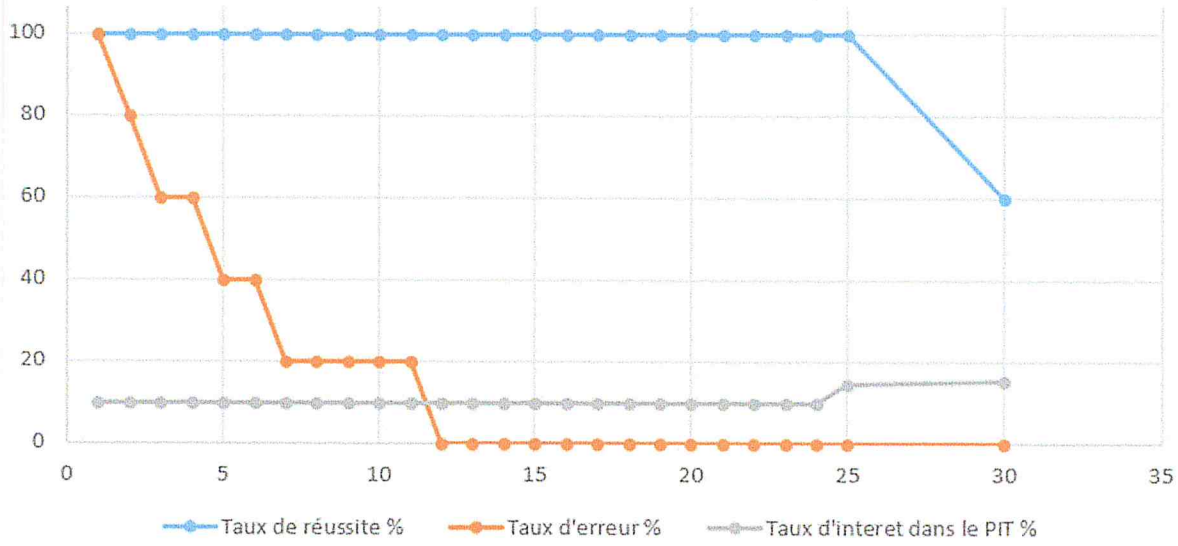


***Fig.II.5.*** *Variation du seuil de nombre de faux contenue sur le taux de réussite, taux d'erreur et le taux d'intérêt dans le PIT.*

De la figure.II.5 en déduit que le meilleur paramètre correspondant au seuil de nombre de contenue falsifier est entre 12 et 24, car celle-ci donne un taux de réussite avoisinant les 100 % et un très bas taux d'erreur, approchant les 0 %.

- **Temp de vie d'un intérêt :**

Après avoir trouvé le meilleur seuil de nombre de faux contenue, on applique ce dernier avec un seuil de PIT statique qui paramétrer a 50, et on varie le temp de vie d'un intérêt.

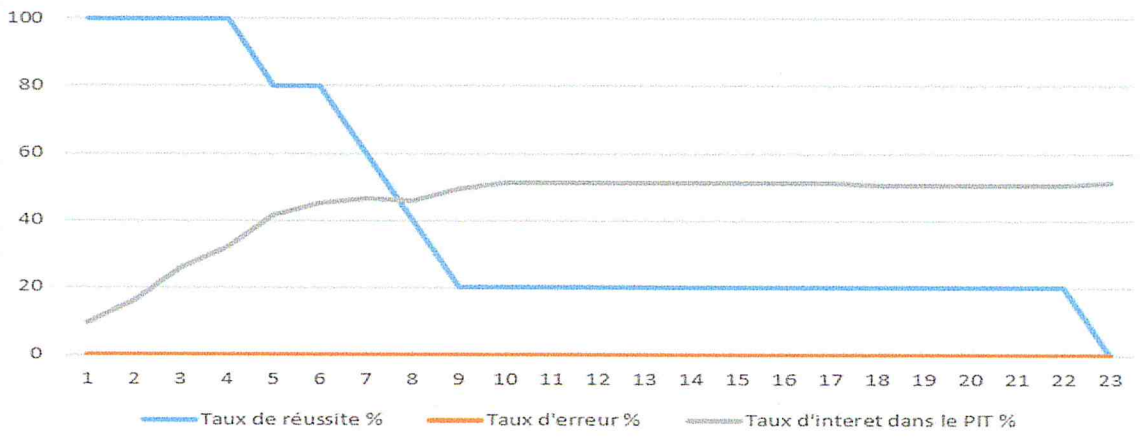


***Fig.II.6.*** *Variation du seuil du temp de vie d'un intérêt sur le taux de réussite, taux d'erreur et le taux d'intérêt dans le PIT.*

La figure.II.6 montre que le meilleur paramètre correspondant au temp de vie d'un intérêt est de 4 milliseconde, car celle-ci donne un taux de réussite avoisinant les 100 % et un très bas taux d'erreur, approchant les 0 %.

Enfin pour trouver le meilleur seuil de PIT, on applique les meilleur paramètres trouver précédemment, et on varie le seuil de PIT.

***a. Seuil de PIT :***

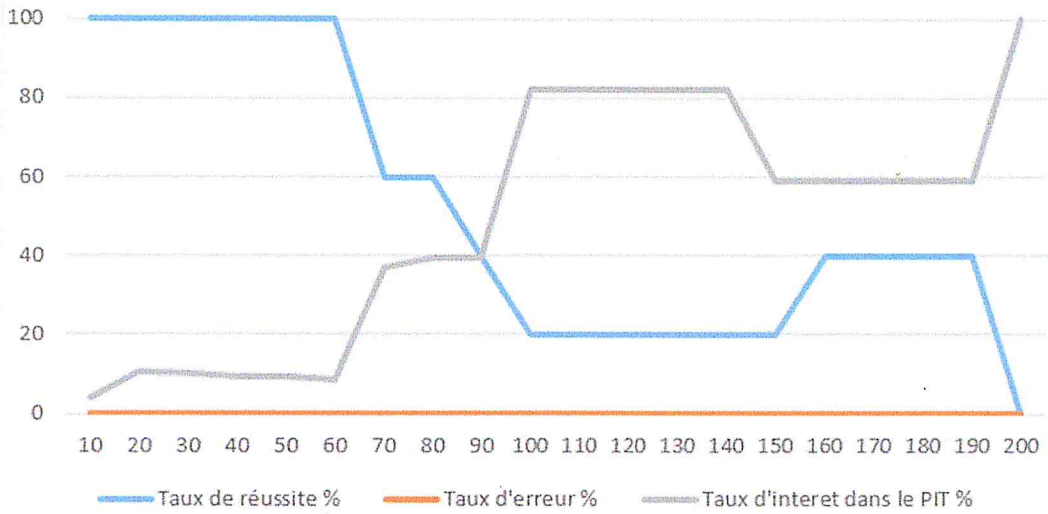


***Fig.II.7.*** *Variation du seuil du nombre d'intérêt contenue dans le PIT sur le taux de réussite, taux d'erreur et le taux d'intérêt dans le PIT.*

La figure.II.7 montre que le meilleur paramètre correspondant au seuil de PIT, est entre 10 et 60, car celle-ci donne un taux de réussite avoisinant les 100 % et un très bas taux d'erreur, approchant les 0 %.

Après avoir terminé les tests, on déduit que le paramétrage qui donne le meilleur résultat est :

- Seuil de PIT doit être entre 10 et 60
- Temp de vie d'un intérêt doit être entre 1 à 4 ms.
- Seuil de faux contenue doit être entre 12 et 24.

## II.3 Poséidon :

Poséidon consiste en une phase de détection et une phase de réaction. La phase de détection peut être locale ou distribuée (collaborative). Dans le premier cas, les routeurs s'appuient uniquement sur des métriques locales (par exemple, PIT utilisation, taux d'intérêts

insatisfaits, quantité de bande passante utilisé pour transférer du contenu) pour identifier une attaque. Dans ce dernier, les routeurs proches collaborent pour déterminer si une attaque est en cours et comment l'atténuer.

En cas d'attaque d'inondation d'intérêt réussie, la victime qui est le routeur peut facilement identifier une attaque en observant si son PIT est plein ou si la bande passante allouée à l'expédition de contenu est très petit. Cependant, il peut ne pas être possible pour un routeur sur le chemin de la victime pour détecter une attaque en cours. Les mécanismes de détection collaboratifs permettent aux routeurs d'échanger des informations sur leur état, dans le but de détecter une attaque en cours dès - et aussi proche de l'adversaire que possible. Avec la détection collaborative, les routeurs non seulement échange des informations sur l'existence d'une attaque, mais aussi les propriétés (détectées localement) de telles attaques, les stratégies peuvent prendre en compte les commentaires de plusieurs routeurs.

Nous considérons une approche connue comme push-back **[GAS 12]** pour contrer l'inondation d'intérêt. Poséidon est un ensemble d'algorithmes qui s'exécute sur les routeurs, dans le but d'identifier les anomalies de trafic (en particulier, les inondations d'intérêt) et atténuer leurs effets. Poséidon surveille en permanence les taux d'insatisfaction d'intérêts par interface. Si ces taux changent significativement entre deux intervalles de temps consécutifs, il filtrera l'interface (ou les interfaces) fautives (ce qui réduit le nombre des intérêts entrants). En outre, Poséidon peut émettre un message d'alerte "push" vers les mêmes interfaces, pour signaler qu'une attaque d'inondation d'intérêt est en cours. Poséidon conserve plusieurs statistiques sur les intérêts expirés. En particulier, pour chacun d'entre eux il enregistre l'espace de nom et d'informations sur les interfaces d'entrée / sortie. Dans les sections suivantes, nous introduiront la phase de détection et la phase de réaction de Poséidon. La notation utilisée est indiquée dans le tableau Tableau.II.1.

| | |
|---|---|
| R | Ensemble de tous les routeurs du réseau exécutant Poséidon |
| $r_i$ | i-ème routeur, $1 \leq i \leq \lvert R \rvert$ |
| $r_i^j$ | j-ème interface sur le routeur $r_i$ |
| $t_k$ | k-ème intervalle de temps |

| | |
|---|---|
| $\omega(r_i^j, t_k)$ | Taux entre l'intérêt entrant et le contenu sortant pour une interface donnée $r_i^j$ |
| $\rho(r_i^j, t_k)$ | L'espace PIT utilisé par les intérêts est arrivé sur l'interface $r_i^j$ mesurée à la fin de l'intervalle $t_k$ |
| $\Omega(r_i^j)$ | Seuil de détection d'inondation d'intérêt pour $\omega\left(r_i^j, t_k\right)$ |
| $P(r_i^j)$ | Seuil de détection d'inondation d'intérêt pour$\rho\left(r_i^j, t_k\right)$ |

***Tab.II.1*** *définition des notations de Poséidon*

- ***Phase de détection :***

**[COM 13]** Les attaques sont détectées en utilisant deux paramètres: $\omega(r_i^j, t_k)$, et $\rho(r_i^j, t_k)$. Le premier représente le nombre d'intérêts entrants divisé par le nombre de paquets de contenu sortant, observé par un routeur $r_i$ sur son interface $r_i^j$ dans l'intervalle de temps $t_k$:

$$\omega(r_i^j, t_k) = \frac{\text{nombre d'intérêts de } r_i^j \text{ à l'intervalle } t_k}{\text{nombre des paquets de contenu à } r_i^j \text{ à l'intervalle } t_k}$$

$\rho(r_i^j, t_k)$ Indique le nombre d'octets utilisés pour stocker les intérêts dans PIT, provenant de l'interface $r_i^j$ dans l'intervalle de temps $t_k$.

Poséidon détecte une attaque lorsque $\omega(r_i^j, t_k)$ et $\rho(r_i^j, t_k)$ dépassent leurs seuils respectifs $\Omega(r_i^j)$ et $P(r_i^j)$. L'algorithme de détection est exécuté à des intervalles de temps fixes et en présence d'événements particuliers (c'est-à-dire, des messages de refoulement, détaillé ci-dessous).

Le paramètre $\omega(r_i^j, t_k)$est une bonne représentation de la capacité des routeurs à satisfaire les intérêts entrants dans un intervalle de temps particulier. En particulier, $\omega(r_i^j, t_k)>$ 1 indique que le nombre de paquets de contenu transmis à $r_i^j$ est inférieur au nombre d'intérêts provenant de la même interface. Cependant, une petite explosion d'intérêts (réguliers ou non) ne peut être causée par une attaque. Par conséquent, en ne prenant en compte que $\omega(r_i^j, t_k)$(c'est-à-dire en ne considérant pas $\rho(r_i^j, t_k)$), l'algorithme de détection peut signaler

un grand nombre de faux positifs. L'application de contre-mesures peut, dans ce cas, produire des effets négatifs sur la performance globale du réseau.

Nous argumentons que ni augmenter $\Omega(r_i^j)$, ni calculer $\omega(r_i^j, t_k)$sur des intervalles plus longs, produit les effets indentés. En fait, dans le premier cas, la limite doit être suffisamment élevée pour éviter que les courtes attaques d'intérêts soient qualifiées d'attaques ; Cependant, cela pourrait inévitablement conduire à une détection tardive ou erronée des attaques réelles.

L'augmentation de la taille de l'intervalle sur lequel $\Omega(r_i^j)$ est calculé peut réduire la sensibilité de Poséidon à une brève explosion d'intérêts. Une longueur d'intervalle similaire ou supérieure au temps moyen d'intérêt / paquet de contenu aller-retour, en fait, peut permettre de renvoyer (partie de) le contenu demandé par la rafale, réduisant $\omega(r_i^j, t_k)$ à une valeur plus proche 1. Cependant, cela pourrait augmenter considérablement le temps de détection.

Au lieu de cela, pour améliorer la précision de la détection (en distinguant l'intérêt naturel des attaques), Poséidon prend également en compte $\rho(r_i^j, t_k)$. Cette valeur mesure l'espace PIT utilisé par les intérêts provenant d'une interface particulière. Cela permet à Poséidon de maintenir le nombre de faux positifs à un niveau bas - comparé à ne considérer que $\omega(r_i^j, t_k)$- tout en lui permettant de détecter les inondations à faible taux d'intérêt. Dans une attaque d'inondation à faible taux d'intérêt, l'adversaire limite le taux de faux intérêts pour maintenir $\omega(r_i^j, t_k)$en dessous de ses seuils. La surveillance du contenu du PIT permet à Poséidon d'observer les effets de l'attaque, plutôt que seulement ses causes, permettant une détection précoce.

Pour résumer, différents paramètres surveillés par Poséidon agissent comme des poids et des contrepoids pour la détection d'inondation d'intérêt. Lorsqu'un routeur est incapable de satisfaire les intérêts entrants sur une période relativement courte, $\rho(r_i^j, t_k)$ peut dépasser le seuil de détection mais $\omega(r_i^j, t_k)$ne le sera pas, lorsque le routeur reçoit de courtes rafales d'intérêts, $\omega(r_i^j, t_k)$peut devenir plus grand que $\Omega(r_i^j)$ mais l'utilisation de PIT sera probablement dans les limites des valeurs normales. Pour rester indétectable, un adversaire prêt à effectuer des inondations d'intérêt doit donc :

1- réduire le taux auquel il envoie des intérêts, ce qui limite les effets de l'attaque.

2- limiter l'attaque à une courte rafale, ce qui rend l'attaque inefficace.

Les seuils $\Omega(r_i^j)$ et $P(r_i^j)$ne sont pas constants et peuvent changer avec le temps pour s'adapter aux différentes conditions du réseau. A titre d'exemple, les messages de refoulement décrits ci-dessous fournissent une entrée pour déterminer des valeurs plus appropriées pour ces seuils.

- ***Phase de réaction :***

**[COM 13]** Une fois qu'une attaque d'inondation d'intérêt provenant de l'interface $r_i^j$ du routeur $r_i$ a été identifiée, Poséidon limite le taux d'intérêts entrants provenant de cette interface. Le débit d'origine est restauré lorsque tous les paramètres de détection retombent en dessous de leurs seuils correspondants.

Avec les contre-mesures collaboratives, une fois qu'un routeur détecte un trafic contradictoire à partir d'un ensemble d'interfaces, il limite son débit et émet un message d'alerte sur chacun d'entre eux. Un message d'alerte est un paquet de contenu non sollicité qui appartient à un espace de noms réservé, utilisé pour transmettre des informations sur l'attaque d'inondation d'intérêt en cours. Lors d'une attaque, le PIT du saut suivant connecté à l'interface incriminée peut être plein, et donc le message d'alerte peut être rejeté et les paquets de contenu sont signés, tandis que les intérêts ne le sont pas. Cela permet aux routeurs de déterminer si un message d'alerte est légitime.

Les routeurs exécutant Poséidon ne traitent pas les messages d'alerte en tant que contenu normal. Les alertes ne sont pas vérifiées par rapport au contenu PIT et ne sont plus transmises. La charge utile d'un paquet d'alerte contient l'horodateur correspondant à l'heure de génération d'alerte, le nouveau taux (réduit) auquel les intérêts offensants seront acceptés sur l'interface entrante et des informations détaillées sur l'attaque - telles que les espaces de noms utilisés dans des intérêts malveillants.

Le routeur $r_i$ recevant un paquet msg le traite comme détaillé dans l'algorithme 1 voir figure.II.8. Une attaque d'inondation d'intérêt persistant sur le routeur $r_i$ l'amène à envoyer plusieurs messages d'alerte vers la ou les sources de l'attaque. De telles sources vont diminuer leurs seuils $\Omega(r_i^j)$ et $P(r_i^j)$ jusqu'à ce qu'elles détectent l'attaque et mettent en œuvre une

Si le message réceptionne par l'interface $r_i^j$ est un contenu, il y a aucun traitement fait par le Poséidon sauf qu'il le renvoi à ce routeur pour le traiter comme un contenu normal.

Si le message est une alerte message qui est réceptionner par le routeur lui même $r_i$ à l'interface $r_i^j$ ,Poséidon va vérifier sa signature électronique, si le message est frais, et cet interface $r_i^j$ reçoit une seule alerte dans l'intervalle de temps $t_k$ ,si oui elle va diminuer les seuils de détection d'inondation d'intérêt pour $\omega(r_i^j, t_k)$ et $\rho(r_i^j, t_k)$ par un facteur spécial, sinon elle supprime ce message.

S'il le message reçut par l'interface $r_i^j$est un intérêt, elle vérifie si le taux entre l'intérêt entrant et le contenu sortant pour une interface $r_i^j(\omega(r_i^j, t_k))$ supérieur à le seuil $\Omega(r_i^j)$ et si l'espace PIT utilisé par les intérêts est arrivé sur l'interface $r_i^j$ $(\rho(r_i^j, t_k))$ supérieur au seuil $P(r_i^j)$, si la résultat est vrais elle supprime cet intérêt puis elle envoie une alerte message à l'interface $r_i^j$ si le dernière temps qui elle envoie l'alerte plus long que $t_k$. Sinon les paramètres $\omega(r_i^j, t_k)$et $\rho(r_i^j, t_k)$ ne dépasse pas les seuils, donc Poséidon laisse l'intérêt et met le routeur le traite de maniéré normale.

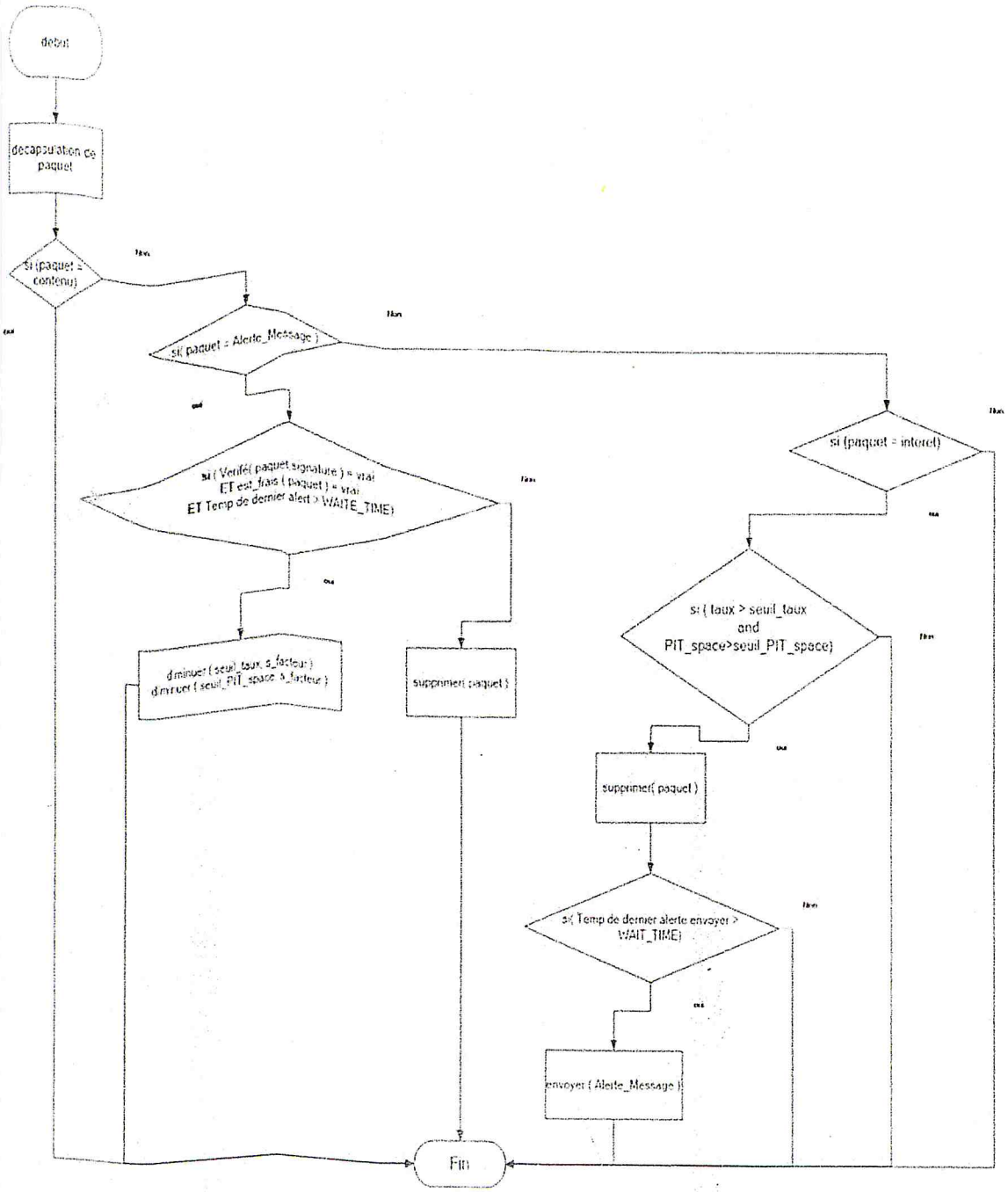


***Fig.II.9*** *l'organigramme de l'algorithme Poséidon*

## II.3.2 Paramétrage de l'algorithme :

Après implémentation de Poséidon, on configure ses paramètres qui sont seuil de détection d'inondation d'intérêt pour $\omega(r_i^j, t_k)(\Omega(r_i^j))$, seuil de détection d'inondation d'intérêt pour$\rho(r_i^j, t_k)(P(r_i^j))$, k-ème intervalle de temps $t_k$,et le facteur utiliser pour diminuer les seuils en cas elle réceptionne un message d'alerte qui est nommé le facteur 's'.

Notre réseau (voir figure.II.10), se compose d'un attaquant, un routeur, et un serveur on adapte l'attaquant pour faire des IFA, puis on essaie de paramétrer les paramètres pour lutter contre ce type d'attaque.

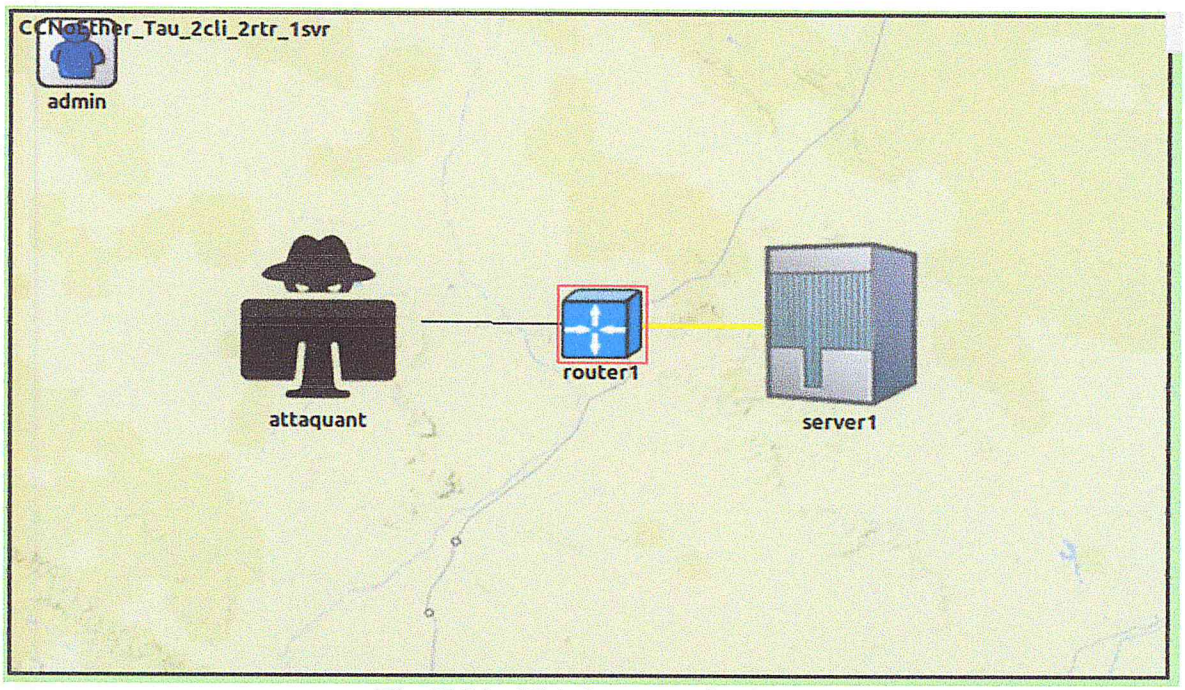


***Fig.II.10.*** *IFA dans simple réseau*

Notre point de départ se consiste sur un test expérimental sur une topologie plus important que celle cité dans **[COM 13]** le test est fait avec ces valeurs, $\Omega(r_i^j)$ =3, $P(r_i^j)$=1/8 de l'espace de PIT, $t_k$= 60 ms, le facteur 's'=2. Le résultat obtenu de cette configuration n'est pas satisfiable pour nous, le taux d'erreur est 0.35 (car la topologie et les types des nœuds sont différents), dons nous avons améliorés ces derniers par fixer les 3 paramètres et mettre un paramètre comme une variable lorsque on obtient des résultats acceptables.

On fixe les valeurs comme la suite $\Omega(r_i^j)$ =3, $t_k$= 60 ms, le facteur 's'=2, et on met $P(r_i^j)$ variable. Dans notre ressource, ce dernier est 1/8 de l'espace de PIT et dans notre projet CCN la taille PIT n'est pas fixé car elle dépend à la taille de la RAM, alors on commence par une autre taille qui est 1500 bit.

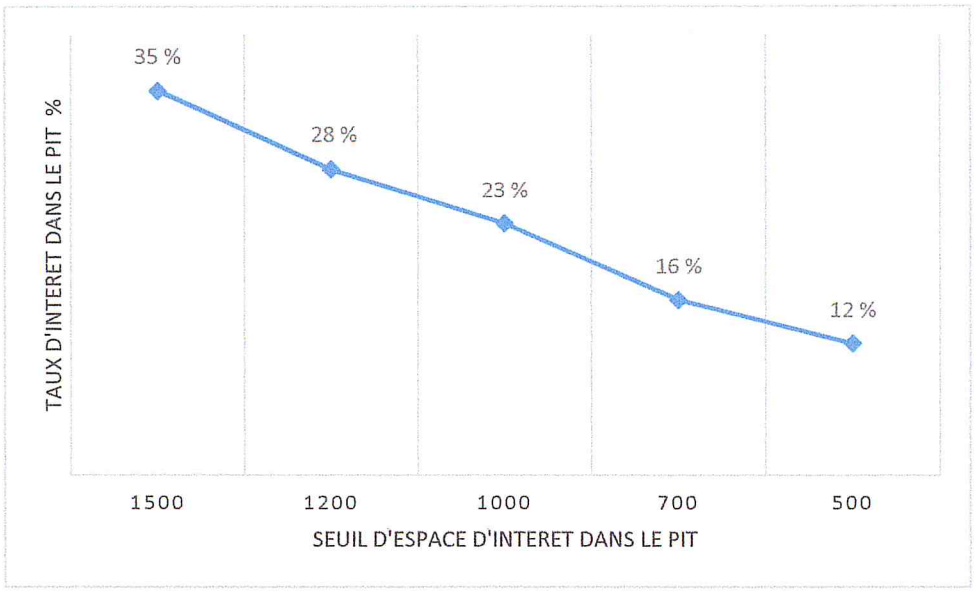


***Fig.II.11*** *évaluation le taux d'erreur par rapport le seul PIT space*

Du graphe figure.II.11 nous déduisons que les valeurs 500 et 700 qui représente le $P(r_i^j)$ impliquent à un bon résultat (le taux d'erreur est diminué). On prend 700 comme un bon résultat car si le seuil est configuré avec une taille faible, le Poséidon va être très sensible (déclenche rapidement), il peut tomber dans le cas où il pense qu'un faux positif est un attaquant.

On fixe les valeurs comme la suite : $P(r_i^j)$=700, $t_k$= 60 ms, le facteur 's'=2 ; et met $\Omega(r_i^j)$comme une variable.

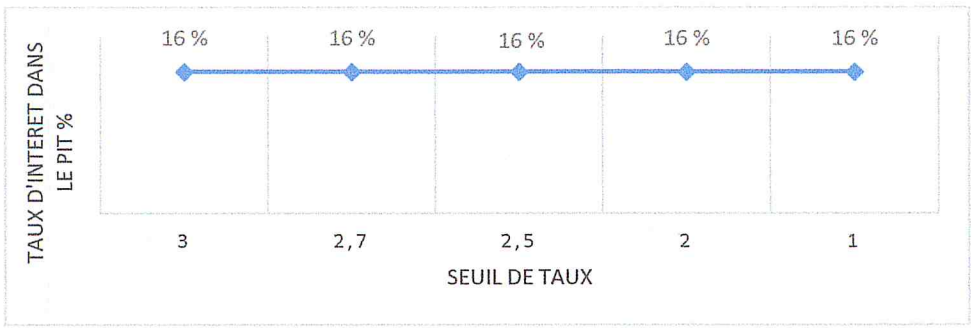


***Fig.II.12*** *évaluation le taux d'erreur par rapport le seuil de taux*

Le graphe figure.II.12 présente l'évaluation de taux d'erreur par rapport le seuil de taux $\Omega(r_i^j)$, nous voyons que tous les seuils nous donnent de bons résultats, mais pour notre étude nous intéressons par le seuil avec la valeur 2.5 car il ne force pas Poséidon faire une défense de manière rapide où lent.

On fixe les valeurs comme la suite :$\Omega(r_i^j)$ =2.5, $P(r_i^j)$=700, le facteur 's'=2 ; et met $\mathbf{t_k}$ comme une variable.

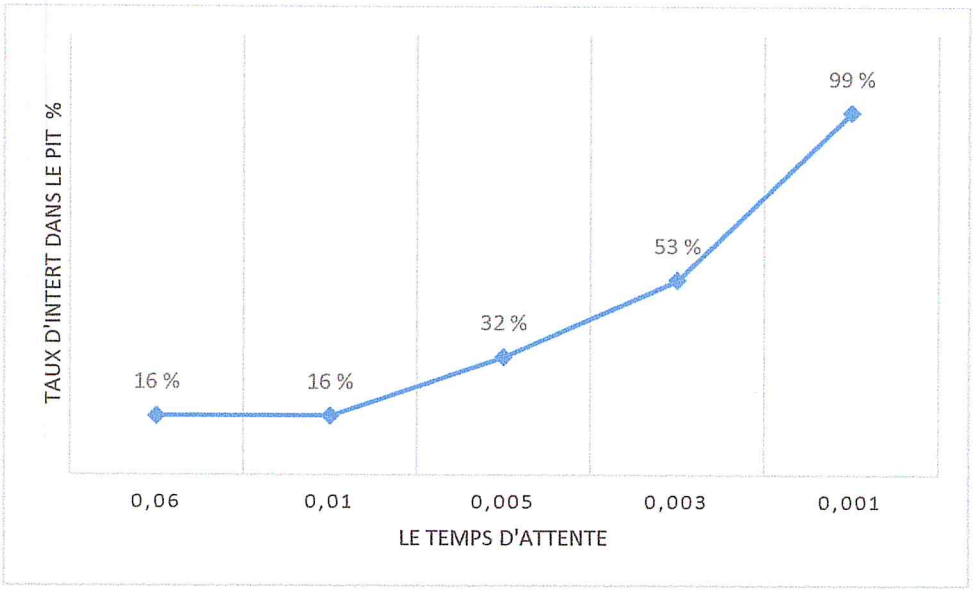


***Fig.II.13*** *évaluation le taux d'erreur par rapport temps d'attente*

Le graphe figure.II.13 présente l'évaluation de taux d'erreur par rapport le temps d'attente $t_k$, nous voyons que les valeurs 0.06 et 0.01 $t_k$ nous donnent un bon résultat par rapport autres, car le rôle de ce paramètre est de réinitialiser les calcule des valeurs $\rho(r_i^j, t_k)$ et $\omega(r_i^j, t_k)$ au début, alors si le $t_k$ est plus élevé, le temps de blocage de l'interface est plus long, mais il ne faut pas oublier que l'interface peut être relier avec des clients légitimes, donc $t_k$ doit être une valeur presque élevé comme 0.01.

On fixe les valeurs comme la suite :$\Omega(r_i^j)$ =2.5, $P(r_i^j)$=700, $t_k$ = 0.01 ; et met le facteur 's'comme une variable.

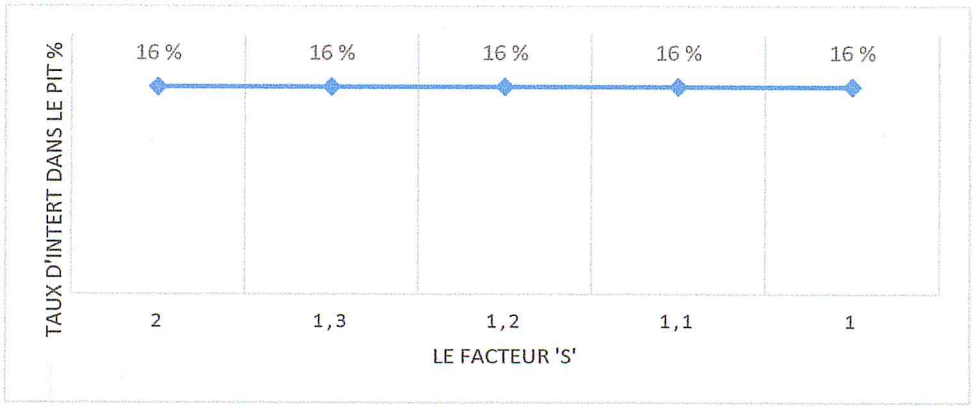


***Fig.II.14*** *évaluation du taux d'erreur par rapport au facteur 's'*

Le graphe figure.II.14 présente l'évaluation du taux d'erreur par rapport le facteur 's', le rôle de facteur « s » est diminuer les seuils en cas d'un IFA. Le résultat se change si l'attaquant fait plusieurs attaques progressives sur un intervalle de temps très court. Nous intéressons à la valeur 1.3 car il celle-ci laisse l'interface ouvert dans un durée de temps raisonnable.

## II.4 Conclusion :

Dans ce chapitre nous avons essayé de déployer et de parametrer les contre-mesure Interest Traceback et poseidon, ce qui nous nous permet de démontrer que les paramètres influent beaucoup sur les performances de ces algorithmes, la négligence de la phase de paramétrage peut causer des dégâts néfastes parmi elle la surcharge de PIT, et aussi blocage de clients légitimes, c'est pour cela que l'administrateur du réseau devra choisir avec prudence les bonnes valeurs.

# Chapitre III : Comparaison des solutions pour IFA

## III.1 Introduction :

La simulation des réseaux en générale consiste principalement en la reproduction du comportement et du fonctionnement des nœuds dans un environnement informatique pour des raisons tel que : la répétition d'expérience, l'adressage des systèmes complexes, le gain de temps et la variation des paramètres de simulation. Alors que la simulation réelle s'avère coûteuse, voire impossible dans quelque cas comme la simulation réelle des CCNs.

Dans ce qui suit, nous présenterons le simulateur "OMNET++" que nous avons choisi dans notre travail, ainsi que le package CCN-lite qui nous aide à faire des simulations légères des CCNs. En fin, nous feront des comparaisons des attaques en utilisant différentes solutions.

## III.2 Déploiement de l'environnement de travail :

Dans notre simulation on a utilisé des outils bien spécifique, qui sont détailler ci-dessous.

### III.2.1 Omnet++ :

OMNeT ++ est une bibliothèque et une structure de simulation C ++ extensible, modulaire et basée sur des composants, principalement pour la construction de simulateurs de réseau. Le terme "réseau" est entendu dans un sens plus large qui inclut les réseaux de communication câblés et sans fil, les réseaux sur puce, les réseaux de mise en file d'attente, et ainsi de suite. Les fonctionnalités spécifiques au domaine telles que la prise en charge de réseaux de capteurs, de réseaux ad-hoc sans fil, de protocoles Internet, de modélisation de performances, de réseaux photoniques, etc., sont fournies par des frameworks de modèles, développés en tant que projets indépendants. OMNeT ++ offre un IDE basé sur Eclipse, un environnement d'exécution graphique et une foule d'autres outils. Il existe des extensions pour la simulation en temps réel, l'émulation de réseau, l'intégration de base de données, l'intégration SystemC et plusieurs autres fonctions **[OMN 18]**.

OMNeT ++ fournit une architecture de composants pour les modèles. Les composants (modules) sont programmés en C ++, puis assemblés en composants plus grands et en modèles utilisant un langage de haut niveau (NED). La réutilisabilité des modèles est gratuite. OMNeT ++ a une prise en charge étendue de l'interface graphique et, grâce à son architecture modulaire, le noyau de simulation (et les modèles) peuvent être facilement intégrés dans vos applications **[OMN 18]**.

Même si OMNeT ++ n'est pas un simulateur de réseau en soi, il a acquis une grande popularité en tant que plate-forme de simulation de réseau dans la communauté scientifique et académique ainsi que dans les milieux industriels, et a constitué une importante communauté d'utilisateurs, sa modularité fait de lui le meilleur parmi les environnements de simulation open source et freeware, si pour cette raison on l'a choisi.

➢ ***Composants :***

- Bibliothèque de noyau de simulation
- Langage de description de topologie NED
- IDE OMNeT ++ basé sur la plateforme Eclipse
- Interface graphique pour l'exécution de la simulation, liens vers un exécutable de simulation (Tkenv)
- Interface utilisateur en ligne de commande pour l'exécution de la simulation (Cmdenv)
- Utilitaires (outil de création de makefile, etc.)
- Documentation, simulations d'échantillons, etc.

➢ **Plateformes :**

Selon **[OMN 18]** OMNeT ++ fonctionne sous Windows, Linux, Mac OS X et d'autres systèmes de type Unix. L'IDE OMNeT ++ nécessite Windows, Linux ou Mac OS X.

### III.2.2 CCN-lite :

CCN-LITE est parmi les extensions, plateformes et simulateurs, qui nous permis de manipuler les CCN dans OMNET++ avec aisance et documentation fournis.

CCN-lite est une implémentation légère et cross-compatible de Content Centric Networking. Il est écrit en C pur et s'exécute (optionnellement) dans l'espace du noyau Linux. De plus, il contient une extension de CCN appelée Named-Function-Networking[6].

- Ccn-lite se caractérise par **[LIT 15]**: base de code minuscule : le noyau a moins de 2.000 LoC, C pur, fonctionne sur UDP et Ethernet brut.
- Plates-formes multiples : le même code s'exécute dans l'espace utilisateur (UNIX), le noyau Linux, OMNeT ++, Android, Arduino (Uno et AtMega328, 2 KiB RAM), RFduino (32 KiB RAM) et Docker.
- Le support de plate-forme actuel inclut IntelX86 ainsi que ARM (Raspberry Pi).
- Formats de paquets multiples : ccnb, NDN, CCNx1.0, IoT-TLV, Cisco-TLV.
- Support de la fragmentation des paquets (déploiement natif possible, sans couche IP).

- Prise en charge de l'ordonnancement à plusieurs niveaux (au niveau du bloc et du niveau de paquet).
- Options de compilation : mise au point de la mémoire, serveur HTTP, configuration à distance, fragmentation, planificateur, signature HMAC256, etc.
- Bibliothèque cliente Python (rightnow, elle ne supporte que le format fil ndn2013).
- Est livré avec la licence ISC entièrement permissive.

L'intégration de ccn-lite dans omnet++ se fait par le biais INET framework, ce dernier. Est une bibliothèque des modèles open source pour l'environnement de simulation OMNeT ++. Il fournit des protocoles, des agents et d'autres modèles pour les chercheurs et les étudiants qui travaillent avec les réseaux de communication. INET est particulièrement utile lors de la conception et la validation de nouveaux protocoles, ou en explorant des nouveaux ou exotiques scénarios **[INE 18]**.

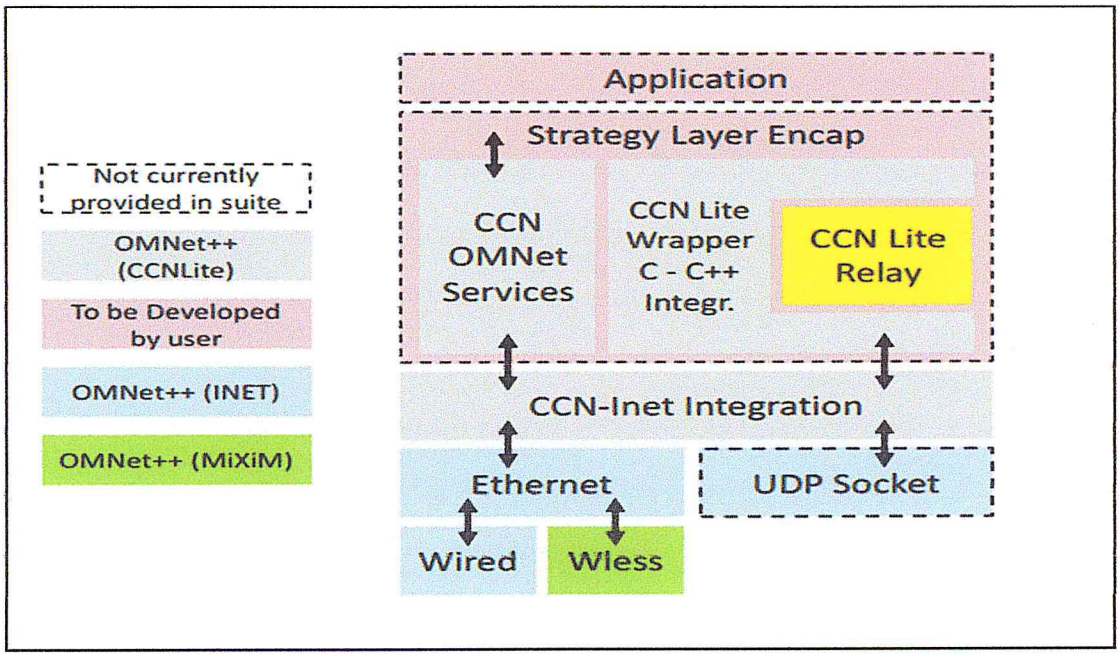


***Fig.III.1*** *représentation de l'intégration de CCN-lite avec OMNeT++ et INET framework***[GIT15]**

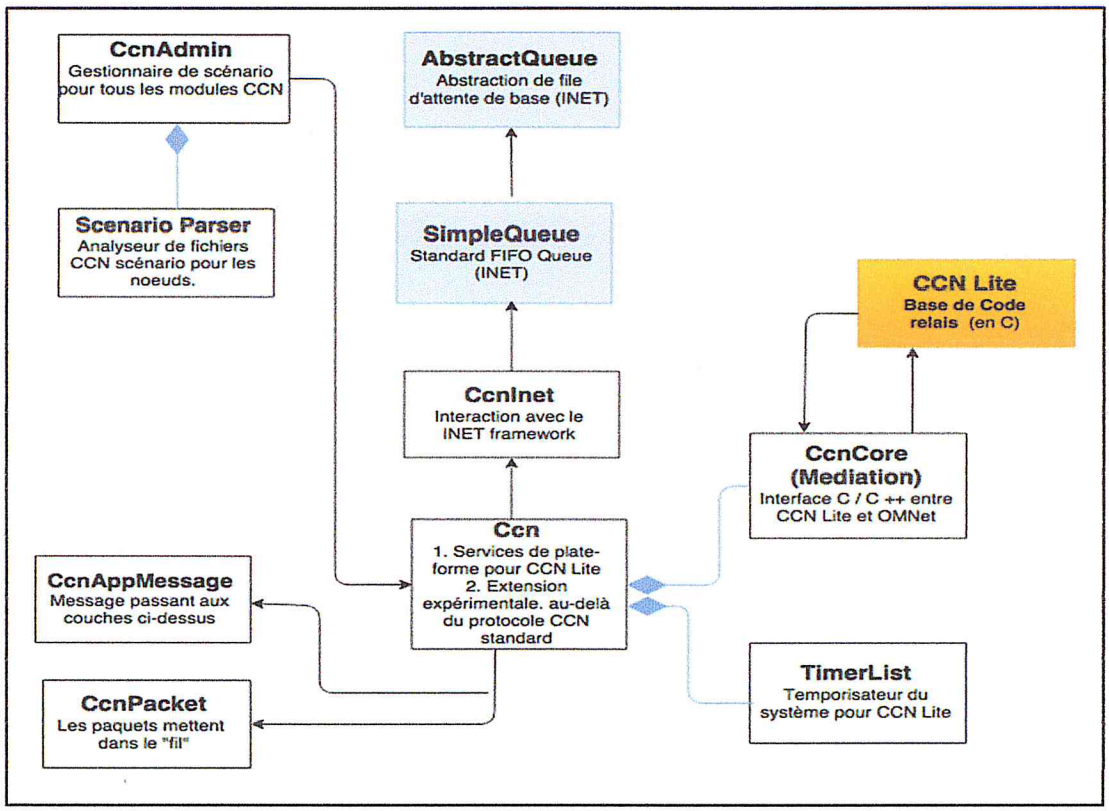


***Fig.III.2*** *Diagramme de classe UML des Composants CCN-lite/OMNeT++* **[GIT15]**

### III.2.3 Les matérielles et les logiciels utilisés :

La configuration suivante a été utiliser lors des tests des contre-mesures. (Voir figure 14)

| CPU | RAM | Disque dure | OS | Omnet++ | Inet Framework | CCN-lite |
|---|---|---|---|---|---|---|
| I7-6700 HQ 2.6 GH | 8 GO DDR4-L 2400MH | 1TB 7200RPM | Ubuntu 16.04.4 desktop amd64 | OMNET++ 4.5 | INET 2.6 | CCN-LITE 0.3.0 |

***Tab.III.1*** *la configuration utiliser lors de la simulation*

## III.3 Mise en œuvre du projet :

Pour pouvoir mettre en œuvre le projet, nous avons besoin de 3 choses essentiels et qui sont, la topologie du réseau, le dataset (les données) et la liaison des scenarios.

### III.3.1 Topologie utilisée :

Pour tester les deux contre-mesures, nous créons une topologie générique compose de 5 clients légitime, 5 attaquant, 6 routeurs, serveur distant remplit par des contenus, et un admin qui permet de donner la main à ses nœuds pour envoyer les intérêts, remplir le serveur par les contenus et de déconnecter les nœuds à la fin de la simulation.

Notre but de cette topologie est de simulé la réalité dans tous les cas possibles, voir figure.III.3.

Le fichier de l'extension. NED contient le code source de la création de topologie, le code compose aussi les paramètres « Delay » et « datarate », le premier représente le retard de message envoyé sur le canal, pratiquement est 0.5 us, et le deuxième paramètre est le débit des donnés sur un canal spécifié par la taille des bits sur le second, pratiquement il est à 100 Mbps. Puis nous lions les nœuds entre aux par un fast Ethernet dans une partie de connexion dans le fichier.

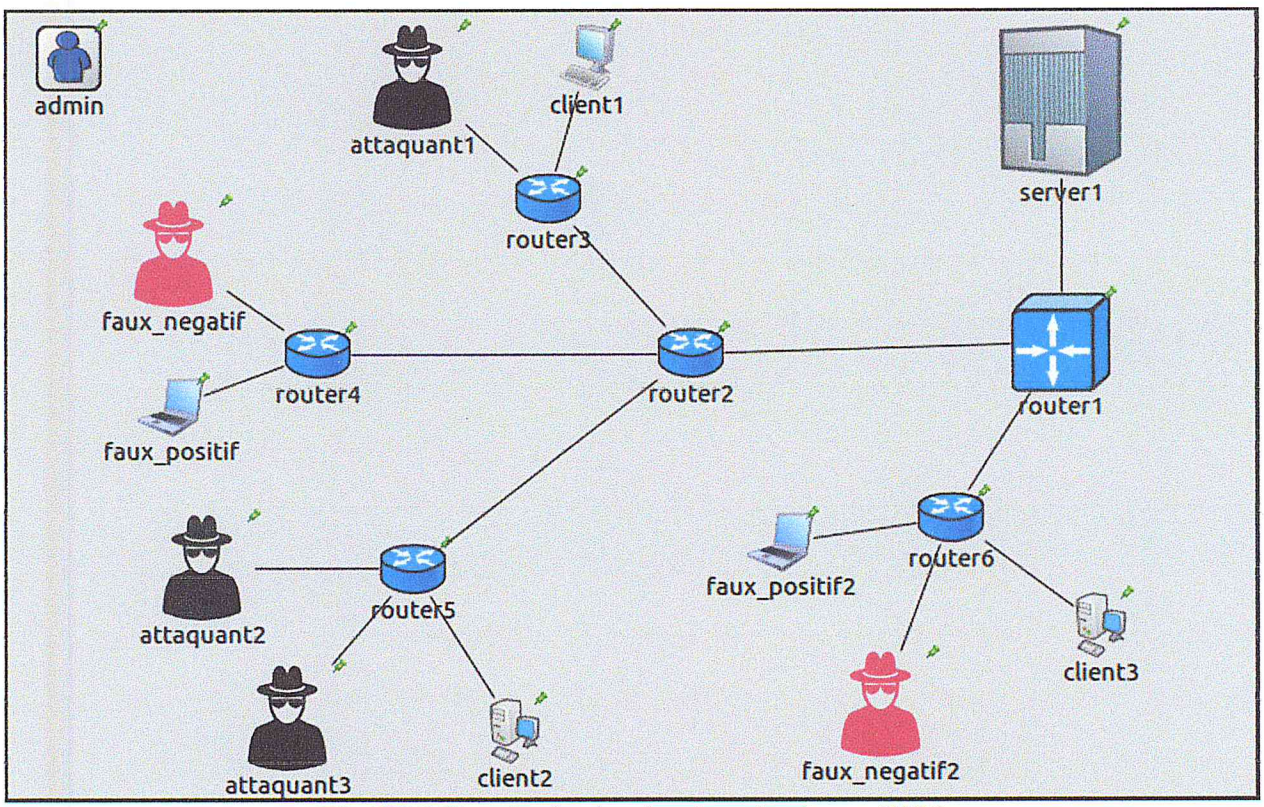


**Fig.III.3** Topologie simuler dans le réseau CCN

### III.3.2 Routage et chargement des donnés :

Chaque nœud CCN dans la topologie a un fichier de configuration, il a extension .cfg. Le fichier compose par 3 parties principales.

La première (eInterestMode) contient les intérêts demandés par le nœud source avec le temps de demande.

La deuxième(ePreCacheMode) est la partie de routage, elle équivalente en réalité le tableau FIB.

Une partie troisième(eFwdRulesMode) qui représente la mémoire physique (pré-cache) pour stocker les contenus, dans notre cas le serveur utilise cet espace de stockage pour les contenus demandés par les autres nœuds, voir figure.III.4.

```
[eInterestMode]

ContentName = /b3c/wowmom/movie104 , StartChunk = 0 , ChunksCount = 1 , RequestTime = 0.0280/*s*/
ContentName = /b3c/wowmom/movie110 , StartChunk = 0 , ChunksCount = 1 , RequestTime = 0.0281/*s*/
ContentName = /b3c/wowmom/movie115 , StartChunk = 0 , ChunksCount = 1 , RequestTime = 0.0282/*s*/
ContentName = /b3c/wowmom/movie120 , StartChunk = 0 , ChunksCount = 1 , RequestTime = 0.0283/*s*/
ContentName = /b3c/wowmom/movie1260000 , StartChunk = 0 , ChunksCount = 1 , RequestTime = 0.0284/*s*/
ContentName = /b3c/wowmom/movie130 , StartChunk = 0 , ChunksCount = 1 , RequestTime = 0.0285/*s*/
ContentName = /b3c/wowmom/movie135 , StartChunk = 0 , ChunksCount = 1 , RequestTime = 0.0286/*s*/
ContentName = /b3c/wowmom/movie140 , StartChunk = 0 , ChunksCount = 1 , RequestTime = 0.0287/*s*/
ContentName = /b3c/wowmom/movie145 , StartChunk = 0 , ChunksCount = 1 , RequestTime = 0.0288/*s*/
ContentName = /b3c/wowmom/movie149 , StartChunk = 0 , ChunksCount = 1 , RequestTime = 0.0289/*s*/
ContentName = /b3c/wowmom/movie93  , StartChunk = 0 , ChunksCount = 1 , RequestTime = 0.0290/*s*/
ContentName = /b3c/wowmom/movie98000  , StartChunk = 0 , ChunksCount = 1 , RequestTime = 0.0291/*s*/
ContentName = /b3c/wowmom/movie7   , StartChunk = 0 , ChunksCount = 1 , RequestTime = 0.0292/*s*/
ContentName = /b3c/wowmom/movie14  , StartChunk = 0 , ChunksCount = 1 , RequestTime = 0.0293/*s*/
ContentName = /b3c/wowmom/movie21 , StartChunk = 0 , ChunksCount = 1 , RequestTime = 0.0294/*s*/
ContentName = /b3c/wowmom/movie28 , StartChunk = 0 , ChunksCount = 1 , RequestTime = 0.0295/*s*/
ContentName = /b3c/wowmom/movie31000 , StartChunk = 0 , ChunksCount = 1 , RequestTime = 0.0296/*s*/
ContentName = /b3c/wowmom/movie126 , StartChunk = 0 , ChunksCount = 1 , RequestTime = 0.0297/*s*/
ContentName = /b3c/wowmom/movie98 , StartChunk = 0 , ChunksCount = 1 , RequestTime = 0.0298/*s*/
ContentName = /b3c/wowmom/movie31 , StartChunk = 0 , ChunksCount = 1 , RequestTime = 0.0299/*s*/

[ePreCacheMode]

[eFwdRulesMode]
ContentPrefix = /b3c/wowmom , NextHop = router5.eth[2] , AccessFrom = client2.eth[0] , UpdateTime = 0/*s*/

[eCommentsMode]
-------------------------------------------------------------------------
comments go here

est client legitime qui demmende 20 interet: 17 vrai ,3 faux par hasar
```

***Fig.III.4*** *fichier de configuration d'un nœud de type client légitime*

### III.3.3 Liaison des nœuds avec le fichier de configuration :

Précédemment on a parlé sur la création des nœuds de topologie sur le fichier. NED puis la configuration de chaque nœud. Un fichier d'extension .**INI** lie les deux derniers composants (nœud, fichier de configuration) afin de remplir le nœud par son routage, ses intérêts et ses contenus après le lancement de la simulation voir figure.III.3.

```
## topology/scenario settings
*.defaultDebugLevel = 4      ## for all simulation: 0=none, 1=Error, 2=Warning, 3=Info, 4=Detail
*.auxDebug = true            ## enable console debugging output

## per node settings
**.debugLevel = 4            ## per host: 0=none, 1=Error, 2=Warning, 3=Info, 4=Detail
**.minTxPace = 10ms
**.maxCacheSlots = 400
**.maxCacheBytes = 524288000Bytes
**.ccnCoreVersion = "CCN Lite v0.3.0"


*.attaquant1.net.ccnScenarioFile = "attaquant1_ccn.cfg"
*.attaquant2.net.ccnScenarioFile = "attaquant2_ccn.cfg"
*.attaquant3.net.ccnScenarioFile = "attaquant3_ccn.cfg"
*.client1.net.ccnScenarioFile = "client1_ccn.cfg"
*.client2.net.ccnScenarioFile = "client2_ccn.cfg"
*.client3.net.ccnScenarioFile = "client3_ccn.cfg"
*.faux_negatif.net.ccnScenarioFile = "faux_negatif_ccn.cfg"
*.faux_positif.net.ccnScenarioFile = "faux_positif_ccn.cfg"
*.faux_negatif2.net.ccnScenarioFile = "faux_negatif2_ccn.cfg"
*.faux_positif2.net.ccnScenarioFile = "faux_positif2_ccn.cfg"
*.router1.net.ccnScenarioFile = "router1_ccn.cfg"
*.router2.net.ccnScenarioFile = "router2_ccn.cfg"
*.router3.net.ccnScenarioFile = "router3_ccn.cfg"
*.router4.net.ccnScenarioFile = "router4_ccn.cfg"
*.router5.net.ccnScenarioFile = "router5_ccn.cfg"
*.router6.net.ccnScenarioFile = "router6_ccn.cfg"
*.server1.net.ccnScenarioFile = "server1_ccn.cfg"
```

***Fig.III.5*** *fichier INI de notre topologie*

## III.4 Les résultats de chaque contre mesure :

Après avoir fait plusieurs itérations de test pour chaque contre-mesure, les résultats sont montrés ci-dessous.

### III.4.1 Interest Traceback :

Après le lancement de la simulation voir la figure.III.3, les nœuds transmettent les paquets entre aux, ces paquets sont sur quatre types. Paquets qui est un contenu demandé, des vrais intérêts qui retourne un résultat(contenu), ou un faux intérêt qui demande un contenu qui n'existe pas, et un fau contenu en réponse un faux intérêt.

Nous installons Interest Traceback dans les routeurs router 3, router 4 et router 5 (les routeurs d'extrémités) pour mettre le résultat clair et interprétables (voir la note au-dessous). Ils contrôlent tous ses interfaces à travers les paquets entrants et sortants. Depuis ce traitement il peut détecter si l'interface reçoit une attaque ou pas.

Les paramètres de Poséidon sont configurés comme la suite, le seuil de PIT est 60, le seuil de faux contenue est 12, le temp de vie d'un intérêt est de 4ms, pour le coté matérielle nous avions déjà citées précédemment Tableau.III.1.

Le tableau.III.2 ci-dessous présente les états des nœuds (bloqués ou pas) dans la simulation, et par qu'elle routeur, en précisément la conséquence de chaque cas.

| Temps en (ms) | Entré | Bloqué | Par | Conséquence |
|---|---|---|---|---|
| 9.50658 | Attaquant 1 | Oui | Router 3 | Le chemin entre le router 3 et l'attaquant1 est bloqué |
| 23.0065 | Faux négatif 1 | Oui | Router 4 | Le chemin entre le router 4 et le faux négatif 1 est bloqué |
| 33.7066 | Attaquant 3 | Oui | Router 5 | Le chemin entre le router 5 et attaquant 3 est bloqué |
| 20.0066 | Attaquant 2 | Oui | Router 5 | Le chemin entre le router 5 et attaquant 2 est bloqué |
| 32.6066 | Faux négatif 2 | Oui | Router 3 | Le chemin entre le routeur 6 et le faux négatif est bloqué |

***Tab.III.2*** Déroulement de la simulation avec interest traceback

### III.4.2 Poséidon :

Après le lancement de la simulation voir le figure.III.3, les nœuds transmettent les paquets entre aux, ces paquets sont sur trois types. Paquets qui est un contenu demandé, des vrais intérêts qui retourne un résultat(contenu), ou un faux intérêt qui demande un contenu qui n'existe pas.

Nous installons Poséidon dans les routeurs router 3, router 4 et router 5 (les routeurs d'extrémités) pour mettre le résultat clair et interprétables (voir la note au-dessous). Ils contrôlent tous ses interfaces à travers les paquets entrants et sortants. Depuis ce traitement il peut détecter si l'interface reçoit une attaque ou pas.

Les paramètres de Poséidon sont configurés comme la suite, le seuil de détection d'inondation d'intérêt pour $\omega(r_i^j, t_k)$ $(\Omega(r_i^j))$ = 2.5, seuil de détection d'inondation d'intérêt pour$\rho(r_i^j, t_k)$ $(P(r_i^j))$ = 700, le k-ème intervalle de temps $(t_k)$ = 0.01 et le facteur 's'= 1.3 ; pour le coté matérielle nous avions déjà citées précédemment Tableau.III.1.

Le tableau (voir figure 19) ci-dessous présente les états des nœuds (bloqués ou pas) dans la simulation, et par qu'elle routeur, en précisément la conséquence de chaque cas.

| Temps en (ms) | Entré | Bloqué | Par | Conséquence |
|---|---|---|---|---|
| 5.60658 | Attaquant 1 | Oui | Router 3 | Le chemin entre le router 3 et l'attaquant 1 est bloqué |
| 0.01 | Attaquant 1 | Non | / | Réouverture du chemin entre le routeur 3 et attaquant 1 |
| 11.30658 | Attaquant 1 | Oui | Router 3 | Le chemin entre le routeur 3 et attaquant 1 est bloqué |
| 15.60658 | Attaquant 2 | Oui | Router 5 | Le chemin entre le routeur 5 et attaquant 2 est bloqué |
| 15.706499999 | Faux négatif | Oui | Router 4 | Le chemin entre le routeur 4 et le faut négatif est bloqué |
| 20.00 | Faux négatif et attaquant 2 | Non | / | Réouverture du chemin entre le routeur 4 et faut négatif<br>Réouverture du chemin entre le routeur 5 et attaquant 2 |
| 21.3065 | Faux négatif | Oui | Router 4 | Le chemin entre le routeur 4 et faut négatif est bloqué |
| 21.30658 | Attaquant 2 | Oui | Router 5 | Le chemin entre le routeur 5 et attaquant 2 est bloqué |
| 25.8065 | Faux_negatif2 | Oui | Router 6 | Le chemin entre le routeur 6 et faux négatif 2 est bloqué |
| 29.60658 | Attaquant 3 | Oui | Router 5 | Le chemin entre le routeur 5 et attaquant 3 est bloqué |
| 30.00 | Attaquant3 et faux_negatif2 | Non | / | Réouverture du chemin entre le routeur 5 et attaquant 3<br>Réouverture Le chemin entre le routeur 6 et faux négatif 2 |
| 31.3065 | Faux_negatif2 | Oui | Router 6 | Le chemin entre le routeur 6 et faux négatif est bloqué |
| 31.30658 | Attaquant 3 | Oui | Router 5 | Le chemin entre le routeur 5 et attaquant 3 est bloqué |

***Tab.III.3*** *Déroulement de la simulation avec Poséidon*

## III.5 Synthèse :

Depuis les tests effectuer précédents, nous concluons que la comparaison entre les deux contre-mesures est délicate car chaque solution se différencier par son fonctionnement et sa stratégies différentes, mais on peut dire que les deux solutions ont réussi à réagir au IFA (interest flooding attack) efficacement.

## III.6 Conclusion :

Après avoir testé les 2 algorithmes, nous avons recapitulé la comparaison entre Interest traceback et Poséidon dans le tableau suivant :

| | Interest Traceback | Poséidon |
|---|---|---|
| Atténuation progressive d'attaquant | Oui | Oui |
| Possibilité de blocage d'un client légitime | Oui | Oui |
| Blocage définitif d'une interface | Oui | Non |
| Peut configure chaque interface (chemin) relier par un nœud de manière différent que les autres interfaces. | Oui | Oui |
| Chaque interface (chemin) d'un nœud a un niveau de sensibilité que les autres interfaces. | Oui | Oui |
| On peut installer dans un chaque nœud **CCN** | Oui | Oui |
| Il fait des mises à jour de **PIT** | Oui | Non |
| Si on fait une mise à jour sur le réseau, il faut faire une nouvelle configuration sur la contre-mesure | Oui | Oui |
| Collaboration entre routers | Non | Oui |

***Tab.III.4*** *comparaison des Deux contre-mesure*

Au final, nous constatons que Interst traceback opte pour une défense plus agressive que Poséidon, bien que ce dernier attenue aussi les attaque mais de façons plus progressives que son homologue Interest Traceback.

# Conclusion générale et perspective

Les Réseaux Centrés Information (ICNs) a bâti une nouvelle architecture réseau qui commence à se propager progressivement dans le monde, cette dernière a apporté de nouveaux concepts et idées dans le domaine de recherche des protocoles de routage de prochaine génération, proposant une approche alternative à la suite de protocoles TCP/IP bien connue et consolidée. Un ICN envisage un réseau de périphériques de mise en cache intelligents qui transmettent non seulement des bits d'un endroit à l'autre, mais aussi un support du réseau pour fournir aux utilisateurs finaux ce qui les intéresse : les données nommées. Cependant, bien qu'une grande partie de la littérature existante souligne les avantages de ce nouveau paradigme de réseau toutefois, comme toute architecture ouverte, elle reste très vulnérable aux problèmes de sécurité tel l'opportunité pour le spoofing ou de créer des dénis de service distribuées (DDoS), communément appelées les attaques d'inondations d'intérêt (Interest Flooding Attack, IFA). Dans notre cas nous nous intéressons a ces types de problèmes.

Notre travail consistait à concevoir et à implémenter plusieurs solution (Poséidon, interest traceback) de lutte contre IFA et proposer des solutions pour contrer le spoofing. Une fois implémenté, plusieurs tests ont été faite sur les algorithme pour ajuster leurs paramètres et calculer leurs performances.

Après avoir configure et paramétrer les algorithmes sur le réseau (voir chapitre 2 et 3), nous avons fait plusieurs tests comparatifs, qui nous a donner des résultats acceptables et satisfaisons, ces derniers se présente presque identiques malgré leurs stratégies et fonctionnements différents de l'un de l'autre, le choix entre ces deux solutions est délicat car chaque contre-mesure présente des avantages et des inconvénients, aussi nous avons trouvé que les solutions proposer pour le spoofing sont les plus adéquate.

Lors du déploiement des deux solutions sur le réseau, on a rencontré des problèmes lors de la configurations des paramètres, car jusqu'à maintenant leurs configurations sont faites de manière statique, c'est pour cette raison nous proposons une amélioration de ce problème de manière dynamique, en apprentissage automatique, avec à cette amélioration, les chances d'avoir de mauvais paramétrage seront considérablement réduit, et aussi le temp de maintenance sera réduit, qui en résulte un réseau plus sécuriser, bien que on a essayer de suggerer les solutions les plus efficace pour contrer les attaque de types spoofing par exemple : attaque d'interception et hijacking, nous proposons une implémentation de ces solutions en utilisant le même environnement de simulation utiliser lors ce travail.

# Annexe : Spoofing

## IV.1 Introduction :

Dans les CCN's la notion de spoofing, appelé plus communément usurpation d'identité, reste toujours aussi fidèle à son homologue IP, mais avec quelques différences mineures dans son fonctionnement et de sa mise en place, ce type d'attaque cause des dégâts non négligeable entre l'utilisateur légitime et le service qui fournis ou transmet les données.

Au cours ce chapitre, nous allons présenter deux types d'attaque qui appartient aux attaques liées au spoofing qui s'intitule attaque d'interception et hijacking, ainsi nous allons aborder son déroulement, les failles qui exploite et sa gravité sur le réseau de façon générale, pour aux finales proposer des solutions pour contrer ces types d'attaques.

## IV.2 Attaque d'interception :

Connue sous le nom d'attaque man in the middle dans les réseau IP, ce type d'attaque n'a pas grandement changer dans sa nouvelle architecture réseau CCN.

### IV.2.1 Les failles exploiter :

Le tableau.A.1 citer dans **[ESL 15]** montre les caractéristiques de CCN, que l'attaque d'interception peut en dépendre totalement ou partiellement, pour qu'elle soit exécuter.

| Caractéristiques | Nommage indépendant de l'emplacement | Dans la mise en cache réseau | Publication / abonnement omniprésente |
|---|---|---|---|
| Dégréé de dépendance | Totalement | Partiellement | Totalement |

***Tab.A.1*** *dépendance des défauts de CCN face à l'attaque interception*

Description des caractéristiques :

- **Nommage indépendant de l'emplacement :** Cet attribut permet la récupération de contenu à partir de plusieurs emplacements inconnus ou non approuvés. ICN a besoin d'un système de nommage sécurisé pour nommer le contenu indépendamment de son emplacement et de sa représentation.
- **Mise en cache dans le réseau :** La mise en cache est l'une des principales caractéristiques des architectures CCN. Tout nœud du réseau peut mettre en cache tout élément qui le traverse. Le contenu peut être délivré à partir du cache le plus proche qui contient le contenu au lieu d'aller sur le serveur d'hébergement.

- **Publication / abonnement omniprésente :** Tout utilisateur peut accéder au réseau CCN depuis n'importe quel endroit et agir en tant que fournisseur de contenu ou consommateur de contenu. Certains utilisateurs peuvent envoyer des contenus ou des demandes indésirables.

### IV.2.2 Impacte :

Selon **[ESL 15]** :

- **Infiltration de chemin :** Dans CCN's, les copies de contenu sont généralement distribuées dans de nombreux emplacements non approuvés. Il est donc difficile d'authentifier les origines valides pour le contenu.
- **Vie privée :** La violation de la confidentialité dans l'attaque d'interception donne à l'attaquant un accès non autorisé aux demandes de l'utilisateur, en particulier lorsque l'attaquant est topologiquement proche ou sur la route vers l'utilisateur.
- **Difficulté de fixation des dégâts :** ne peut pas réparer depuis l'attaque, Si l'attaquant a accédée à des informations privées.

## IV.3 Hijacking :

Toujours aussi dangereux le hijacking, a su attirer les foudres des utilisateurs et administrateurs réseau, ce dernier a était sujet a de nombreux articles scientifique.

### IV.3.1 Les failles exploiter :

Le tableau.A.2 citer dans **[ESL 15]** montre les caractéristiques de CCN, que le hijacking peut en dépendre totalement ou partiellement, pour qu'il soit exécuter.

| Caractéristiques | Nommage indépendant de l'emplacement | Dans la mise en cache réseau | Publication omniprésente / abonnement |
|---|---|---|---|
| Dégréé de dépendance | Totalement | Partiellement | Totalement |

***Tab.A.2*** *dépendance des défauts de CCN face au hijacking*

### IV.3.2 Impacte :

Selon **[ESL 15]** :

- **Déni de service :** DoS peut se produire en raison de nombreuses attaques dans cette catégorie, telles que l'envoi de nombreuses demandes de contenu indisponible ou à

une source unique. Par conséquent, les temporisateurs intermédiaires suppriment les demandes avec les délais expirés, ce qui peut conduire à des DoS ou au moins à des retards importants.

- **Infiltration de chemin :** Dans CCN's, les copies de contenu sont généralement distribuées dans de nombreux emplacements non approuvés. Il est donc difficile d'authentifier les origines valides pour le contenu.
- **Vie privée :** La violation de la confidentialité dans l'attaque d'interception donne à l'attaquant un accès non autorisé aux demandes de l'utilisateur, en particulier lorsque l'attaquant est topologiquement proche ou sur la route vers l'utilisateur.

## IV.4 Solutions existantes :

Selon **[REZ 16]** la technique proposée par Tourani et al. A la plus faible complexité de calcul et le coût d'infrastructure. Certaines des solutions anti-censure existantes **[DIB 11]** et **[TOU 15]** ont permis une communication anonyme par tunneling sécurisé, où le contenu est crypté entre les fournisseurs / mandataires et les clients. D'autres approches incluent un schéma d'obfuscation de noms **[ARI 11]** et un réseau de courtage hiérarchique **[ARI 14]** pour la récupération de contenu anonyme. Les opérations cryptographiques coûteuses **[ARI 11]**, **[DIB 11]**, **[CHU 14]**, l'exigence d'un canal arrière sécurisé **[ARI 11]**, et l'affaiblissement de la mise en cache dans le réseau **[DIB 11]**, **[TAO 15]**, **[TOU 15]** sont les principaux pièges de ces mécanismes. Sauf le travail de Fotiou **[ARI 14]** qui cible les architectures avec un réseau de courtage (par exemple, PSIRP et PURSUIT), d'autres solutions proposées (par exemple tunneling, obfuscation de noms et mécanismes de codage réseau) sont applicables à toutes les architectures CCN. Il existe des pistes potentielles pour de futures recherches sur l'optimisation de l'utilisation du cache et la réduction du coût des opérations cryptographiques. L'application d'opérations cryptographiques sur un sous-ensemble de segments de contenu pour réduire les coûts n'a pas encore été explorée. Exploiter des techniques de codage de réseau de faible complexité **[TAO 15]**, **[TOU 15]** au lieu de la cryptographie traditionnelle serait une bonne idée pour étendre l'applicabilité des schémas de tunnelisation. Ceci est particulièrement important étant donné que la majorité des dispositifs à l'avenir seront des dispositifs à ressources limitées (par exemple, des dispositifs mobiles, l'Internet des objets, etc.).

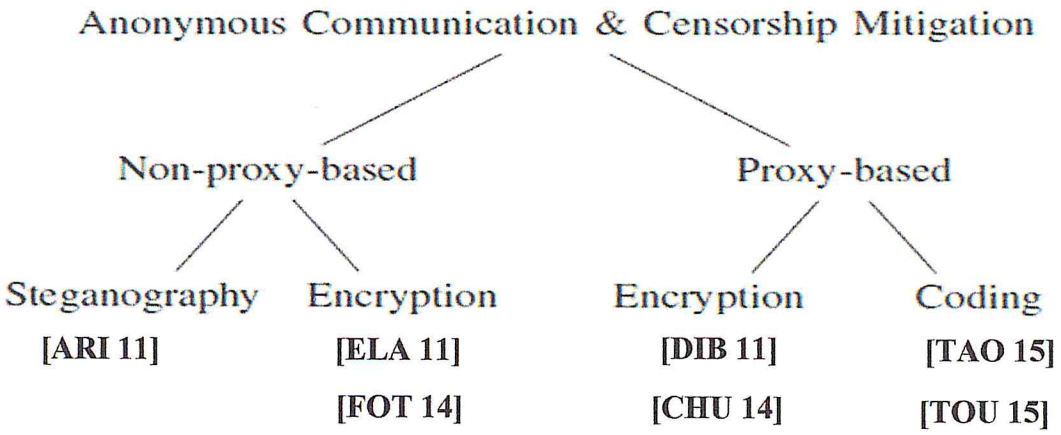


***Fig.A.1*** *l'approche de la communication anonyme et de l'atténuation de la censure sont catégorisées selon qu'elles utilisent un proxy ou non* **[REZ 16].**

## IV.5 Les solutions proposer :

Après avoir décortiquer les solutions offertes par la communauté scientifique. Nous avons trouvé parmi elles, des contre-mesures plus aptes à être implémenter et déployer sur le réseau.

### IV.5.1 Pour l'interception :

Selon **[REZ 16]** Dans les approches basées sur un proxy, un client doit interagir et partager un secret avec un proxy (un réseau de proxys). Le proxy est responsable du décryptage / décodage des requêtes des clients, de la récupération du contenu demandé et du renvoi du contenu codé / codé aux clients. Les approches sont similaires dans l'esprit au populaire Tor (protocole de routage, aussi outil anti-censure populaire pour les réseaux IP). Sur la base de la manière dont le chiffrement en couches est effectué.

ANDaNA, un protocole anti-censure basé sur le tunnel, utilise deux proxy, un proxy adjacent au demandeur et un autre proxy plus proche de la destination pour créer un tunnel avec deux couches de chiffrement.

Dans [101], ont pris une approche similaire à ANDaNA et Tor. Dans cette approche, le client crypte le paquet d'intérêt avec deux clés symétriques qui seront partagées avec deux routeurs anonymes (RA). L'ordre de chiffrement de l'intérêt suit le modèle de routage de Tor. Différent du routage traditionnel de Tor, un identifiant (un hachage du nom de contenu) est incorporé dans l'intérêt chiffré pour permettre l'utilisation du cache (c.-à-d. Recherche CS) et l'agrégation d'intérêts (recherche PIT) au premier AR. Le fournisseur transmet le contenu au

AR le plus proche en texte clair. La réponse de contenu sur le chemin du retour peut être mise en cache sur le second AR, qui crypte le contenu et le transmet au premier AR. Le premier AR déchiffre le contenu pour la mise en cache avant de le recrypter et de le transférer vers le client. Similaire à ANDaNA, ce système souffre du même coût élevé du multiples chiffrements / décryptions par paquet.

### IV.5.2 Pour le hijacking :

Dans **[REZ 16]**, La catégorie de transfert sécurisé inclut des mécanismes qui sécurisent le plan de transfert ou créent un mappage d'espace de noms sécurisé, ce qui permet le transfert d'intérêt pour les préfixes de noms qui ne figurent pas dans les tables FIB des routeurs. **[YI 13]** ont augmenté le plan d'acheminement NDN pour contrecarrer les problèmes de sécurité, tels que le détournement de préfixes et la surcharge PIT (cas de déni de service authentifié). Dans le détournement de préfixe, un attaquant annonce le préfixe de la victime et abandonne l'intérêt. Les auteurs ont suggéré l'utilisation de NACK's (Accusé de réception négatives) d'intérêt lorsque les demandes ne sont pas satisfaites pour des raisons telles que l'encombrement du réseau, le contenu inexistant et le contenu en double. L'intérêt NACK aide à réduire la taille du PIT en raison du NACK supprimant une entrée PIT. En outre, il atténue la vulnérabilité de détournement de préfixes, en fournissant plus de temps au routeur pour interroger les autres visages pour une correspondance de contenu. Cependant, cela nécessite que chaque routeur stocke des informations RTT (temps de voyage) pour chaque intérêt, ce qui représente un surcoût important pour les routeurs centraux. De plus, étant donné que la NACK a un intérêt dans le PIT, il n'y a aucune possibilité de regroupement d'intérêts fictif, cela pourrait exacerber les attaques DoS basées sur les intérêts.

## IV.6 Discussion :

Nous concluons que parmi les solutions existantes dans la littérature, les solutions proposer se montre tout à fait à la hauteur pour ce type d'attaque, ces derniers peuvent agir de manière efficace lors de leur déploiement, et aussi leurs performance leur donne avantages non négligeable, s'est pour cette raison nous feront de ces proposition dans un future proche, sujet à un prochain travail qui consiste à faire une étude approfondie sur eux en les implémentant, testant pour ainsi les comparer aux autres solutions existantes.

# Bibliographie

**[CCNA1]** CCNA Exploration : Notion de base

**[CCNA4]** CCNA Exploration : Accès au réseau étendu

**[MIC 06]** Michel Riguidel ; la sécurité des réseaux et des systèmes ; ENST Paris ;2006.

**[ORG 15]** Orange Consulting ; Etude Prospective et Stratégique - Etude sécurité Internet 2030 ; Version 1.1 ; Date : 06/01/2015 ; Paris, France.

**[CAB 14]** Url : http://www.cablemap.info/

**[KASPERSKY]** ABC de la sécurité : Types des menaces connues ; Url : https://support.kaspersky.com/fr/614

**[POL RI]** Mr RIAHLA Med Amine, Polycop 1 : Généralité sur les réseaux informatiques, Université de BOUMERDES UMBB.

**[AVT 17]** Url : https://av-test.org

**[RED 05]** Red Hat Enterprise Linux 4 : Guide de sécurité. Url : http://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-sg-fr-4/index.html

**[WEI 14]** Wei You. A Content-Centric Networking Node for a Realistic Efficient Implementation and Deployment. Networking and Internet Architecture. Télécom Bretagne, Université de Rennes 1, 2014.

**[Cis 14]** Cisco visual networking index : Forecast and methodology, 2014-2019.

**[SAI 13]** SAIL. FP7-ICT-2009-5-257448/D-2.4. Technical report, Sail, Febreury 2013.

**[KOP 07]** T. Koponen, M. Chawla, G.-B. Chun, A. Ermolinskiy, H. Kim, S. Shenker, and I. Stoica, "A data-oriented (and beyond) network architecture," ACM SIGCOMM Computer Communication Review, 2007.

**[LAG 12]** Lagutin D, Visala K, Tarkoma S. Publish/subscribe for internet: PSIRP perspective. IOS Press; 2012.

**[JAC 09]** V. Jacobson, D. Smetters, J. Thornton, M. Plass, N. Briggs, and R. Braynard, "Networking named content," in CoNext 2009, 2009.

**[ESL 15]** Eslam G. AbdAllah, Hossam S. Hassanein, and Mohammad Zulkernine.: A Survey of Security Attacks in Information-Centric Networking, IEEE COMMUNICATION SURVEYS & TUTORIALS, VOL. 17, NO. 3, THIRD QUARTER 2015.

**[ALI 14]** Thèse Ali MAKKE, "Détection d'attaques dans un système WBAN de surveillance médicale à distance", Université Paris Descartes, Télécommunications et Electronique (ED 130), 2014.

**[SAN 15]** Sandvine. "Global Internet Phenomena Report - 2H 2014. Technical report, Sandvine, 2015.

**[NAD 14]** Mme. Nada SBIHI. "Gestion du trafic dans les réseaux orientés contenus". Thèse de Doctorat, Université Pierre et Marie Curie - Paris 6, 2014.

**[ROM 16]** Roman Lutz, "Security and Privacy in Future Internet Architectures", College of Information and Computer Sciences, University of Massachusetts Amherst, USA, 2016.

**[MAT 13]** Matteo Virgilio, Guido Marchetto and Riccardo Sisto, PIT Overload Analysis in Content Centric Networks, Department of Control and Computer Engineering, Politecnico di Torino, Torino, Italy, 2013.

**[MAT 15]** Matteo Virgilio, Guido Marchetto and Riccardo Sisto, Interest Flooding Attack Countermeasures Assessment on Content Centric Networking, Department of Control and Computer Engineering, Politecnico di Torino, Torino, Italy, 2015.

**[DAI 12]** H. Dai, B. Liu, Y. Chen, and Y. Wang. On pending interest table in named data networking. In ANCS '12, Austin, TX, USA, Oct. 2012.

**[DAI 13]** H. Dai, Y. Wang, J. Fan, and B. Liu. Mitigate ddos attacks in ndn by interest traceback. In NOMEN'13, Turin, Italy, Apr. 2013

**[YOU 12]** W. You, B. Mathieu, P. Truong, J. Peltier, and G. Simon. Dipit: a distributed bloom-filter based pit table for ccn nodes. In ICCCN '12, Munich, Germany, July 2012.

**[AFA 13]** A. Afanasyev, P. Mahadevan, I. Moiseenko, E. Uzun, and Lixia Zhang. Interest flooding attack and countermeasures in named data networking. In IFIP Networking Conference, 2013.

**[LIT 15]** CCN-lite Project and Community. Url : http://ccn-lite.net/

**[ATH 15]** Athanasios V. Vasilakos, Zhe Li, Gwendal Simon, Wei You. Information centric network : Research challenges and opportunities Athanasios, 2013.

**[REZ 16]** Reza Tourani, Travis Mick, Satyajayant Misra, and Gaurav Panwar. Security, Privacy, and Access Control in Information-Centric Networking : A Survey, 2016.

**[COM 13]** A. Compagno, M. Conti, P. Gasti and G. Tsudik. Poseidon: Mitigating Interest Flooding DDoS Attacks in Named Data Networking. In 38th Annual IEEE Conference on Local Computer Networks (LCN 2013), Sydney, Australia

**[MSK05]** C. Mallanda, A. Suri, V. Kunchakarra, S.S. Iyengar, R. Kannan and A. Durresi "Simulating Wireless Sensor Networks with OMNeT++", S. Sastry The University of Akron, Akron, Ohio.

**[OMN 04]** "OMNeT++, Discrete Event Simulation System Version 4.0", User Manual.

**[GIT15]** CCN-lite en GitHub : https://github.com/cn-uofbasel/ccn lite/blob/master/doc/internal/omnetpp-getting-started.pdf

**[INE 18]** INET Framework: https://inet.omnetpp.org/Introduction.html

**[SAV00]** S. Savage, D. Wetherall, A. Karlin, and T. Anderson, Practical network support for ip traceback, in *Procedings of ACM SIGCOMM'00*, 2000, pp. 295–306.

**[SNO01]** A. C. Snoeren, C. Partridge, L. A. Sanchez, C. E. Jones, F. Tchakountio, S. T. Kent, And W. T. Strayer, "Hash-based ip traceback," in *Procedings of ACM SIGCOMM'01*, 2001, pp. 3–14.

**[GAS 12]** P. Gasti, G. Tsudik, E. Uzun, and L. Zhang. DoS & DDoS in named-data networking. Technical report, University of California, Irvine, 2012.

**[OMN 18]** https://omnetpp.org/intro/what-is-omnet

**[ARI 11]** S. Arianfar, T. Koponen, B. Raghavan, and S. Shenker. On preserving privacy in content oriented networks. In Proceedings of the ACM SIGCOMM workshop on Information-centric networking, pages 19–24, 2011.

**[ELA 11]** A. Elabidi, G. Ben Ayed, S. Mettali Gammar, and F. Kamoun. Towards hiding federated digital identity: Stop-dissemination mechanism in content-centric networking. In Proceedings of the international conference on Security of information and networks, pages 239–242. ACM, 2011.

**[FOT 14]** N. Fotiou, D. Trossen, G. F. Marias, A. Kostopoulos, and G. C. Polyzos. Enhancing information lookup privacy through homomorphic encryption. Security and Communication Networks, 7(12):2804–2814, 2014.

**[DIB 11]** S. DiBenedetto, P. Gasti, G. Tsudik, and E. Uzun. Andana: Anonymous named data networking application. arXiv preprint arXiv:1112.2205, 2011.

**[CHU 14]** S. Chung, T. Kim, and M. Jang. A privacy-preserving approach in content centric networking. In Proceedings of IEEE Consumer Communications and Networking Conference (CCNC), pages 866–871, 2014.

**[TAO 15]** F. Tao, X. Fei, L. Ye, and F. J. Li. Secure network coding-based named data network mutual anonymity communication protocol. In Proceedings of International Conference on Electrical, Computer Engineering and Electronics (ICECEE), pages 1107–1114, 2015.

**[TOU 15]** R. Tourani, S. Misra, J. Kliewer, S. Ortegel, and T. Mick. Catch Me If You Can: A Practical Framework to Evade Censorship in Information- Centric Networks. In Proceedings of the International Conference on Information-Centric Networking, pages 167–176. ACM, 2015.

**[ARI 14]** N. Fotiou, S. Arianfar, M. S¨arel¨a, and G. C. Polyzos. A framework for privacy analysis of icn architectures. In Privacy Technologies and Policy, pages 117–132. Springer, 2014.

**[YI 13]** C. Yi, A. Afanasyev, I. Moiseenko, L. Wang, B. Zhang, and L. Zhang. A case for stateful forwarding plane. Computer Communications, 36(7):779–791, 2013.