MINISTERE DE L'ENSEIGNEMENT SUPERIEUR
ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE SAAD DAHLAB DE BLIDA 1
FACULTE DES SCIENCES

MEMOIRE DE MASTER II

Option : Systèmes Informatiques et Réseaux

THEME

PROPOSITION DE MECANISMES DE SECURITE CONTRE LES ATTAQUES ZOMBIES ET INTERCEPTION DE CONTENU DANS LES CCN

Réalisé par
Boukhtache Mohamed Abdelfetah
&
Fekirini Mohamed Billel

Devant le Jury compose de :

Mme ZAHRA F/Z.	**Présidente**
Mme ARKAM M.	**Promotrice**
M. MILOUD DAHMANE W.	**Co-Promoteur**
Mme BERAMDANE D.	**Examinatrice**

Année Universitaire : 2019-2020

REMERCIEMENT

Avant tout, nous remercions Allah de nous avoir accordé la force et le courage nécessaire afin d'accomplir ce travail, et Continuer à progresser durant ces longues années d'études.

Nous tenons à remercier vivement Madame ARKAM Meriem, pour son apport scientifique, qui a été mis à notre disponibilité, par ses conseils fructifiés et directives lors de l'élaboration de ce mémoire, sans oublier Monsieur MILOUD DAHMANE Walid, co-promoteur, qui nous a été d'une très grande aide dans la concrétisation de notre PFE.

Nous remercions également toute l'équipe pédagogique de l'université de Blida 1 ainsi que les intervenants professionnels responsables de notre formation, pour avoir assuré un enseignement de qualité.

Nos remerciements vont aussi aux membres du jury qui ont pris de leur temps pour juger ce modeste travail, qu'ils trouvent ici l'expression de notre gratitude et tout notre respect.

Il est naturel que nos pensées les plus fortes aillent vers nos parents. Qu'ils sachent que l'amour qu'ils nous donnent continue à nous animer et nous permet d'envisager l'avenir comme un défi.

Nous tenons à exprimer nos sincères remerciements à frères et sœurs, ainsi qu'aux familles Boukhtache & Fekirini pour leurs conseils, et leur soutien, à la fois moral qu'économique.

Enfin, nous souhaitons adresser nos remerciements à toutes les personnes qui ont contribué de près ou de loin à la réalisation de ce travail.

Merci à toutes et à tous.

RESUME

Le réseau centré sur l'information (Information Centric Networking ICN) est un nouveau paradigme de communication. Il a introduit de nouveaux concepts et idées sur les protocoles de routage. En mettant l'accent sur la récupération du contenu à partir du réseau, l'ICN propose des mécanismes et approches alternatives aux protocoles TCP/IP. Il envisage un réseau basé sur des périphériques de mise en cache intelligents transmettant non seulement des bits d'un endroit à l'autre, mais fournissant aux utilisateurs finaux ce qui les intéressent le plus : les données nommées indépendamment de l'emplacement, du stockage ou de la représentation physique de ce contenu. Cependant, les ICN sont très vulnérables coté sécurité, alors la sécurité du contenu est très importante, nous recensons plusieurs attaques détectées dans la littérature à savoir : Déni de service, Blocage mobil, interception, etc., classées selon la composante du réseau attaquées (cache, routage, infrastucture).

Dans notre travail nous nous intéressons à contrer quelques attaques liées au routage, à savoir le flooding (dénis de service par inondation) et le Spoofing (les attaques d'interception). Pour réaliser nos objectifs, nous avons tout d'abord commencé par étudier ce nouveau paradigme à savoir les ICN, par la suite nous avons attaqué l'aspect de flooding où une fois expliqué les différents scénarios d'attaques et les contre-mesures existantes, nous avons proposé une amélioration de la contre mesure « Poséidon » en proposant une nouvelle métrique au sein des ICN inspiré du TTL, cette solution a amélioré nettement les performances de la contre attaque existante. Par la suite nous avons étudié les différents sénarios des attaques par Spoofing, et nous avons proposé un mécanisme de sécurité basé sur la signature numérique des paquets de contenu. Nous avons simulé tous les scénarios d'attaques possibles et les solutions proposées en utilisant le package CCN-lite sous le simulateur OMNeT++. Les résultats obtenus sont très prometteurs.

Mots clés : Les Réseaux Centrés Information (ICNS), Signature numérique, TTL, OMNET++, ccn-lite, Poseidon, Interest Flooding

ABSTRACT

Information Centric Networking (ICN) is a new communication paradigm. It has introduced new concepts and ideas about routing protocols. With a focus on retrieving content from the network, ICN offers alternative mechanisms and approaches to TCP/IP protocols. It envisions a network based on intelligent caching devices that not only transmit bits from one location to another, but also provide end users with what they are most interested in: named data regardless of the location, storage or physical representation of that content. However, ICNs are very vulnerable on the security side, so content security is very important, we identify several attacks detected in the literature namely: Denial of Service, Mobile Blocking, Interception, etc., classified according to the component of the network attacked (cache, routing, infrastructure).

In our work we are interested in countering some attacks related to routing, namely flooding (denial of service by flooding) and Spoofing (interception attacks). To achieve our objectives, we first started by studying this new paradigm, namely the ICN, then we attacked the flooding aspect where once explained the different attack scenarios and the existing countermeasures, we proposed an improvement of the countermeasure "Poseidon" by proposing a new metric within the ICN inspired by TTL, this solution has significantly improved the performance of the existing counterattack. We then studied the different scenarios of Spoofing attacks, and proposed a security mechanism based on the digital signature of content packets. We simulated all possible attack scenarios and proposed solutions using the CCN-lite package under the OMNeT++ simulator. The results obtained are very promising.

Keywords: Information Centered Networks (ICNS), Digital Signature, TTL, OMNET++, ccn-lite, Poseidon, Interest Flooding

ملخص

الشبكات المتمحورة حول المعلومات (ICN) هي نموذج اتصال جديد. لقد أدخلت مفاهيم وأفكارًا جديدة حول بروتوكولات التوجيه. مع التركيز على استرداد المحتوى من الشبكة، يقدم ICN آليات وأساليب بديلة لبروتوكولات TCP / IP. إنه يتصور شبكة قائمة على أجهزة التخزين المؤقت الذكية التي لا تنقل البتات من موقع إلى آخر فحسب، بل تزود المستخدمين النهائيين أيضًا بما يهتمون به أكثر: بيانات مسماة بغض النظر عن الموقع أو التخزين أو التمثيل المادي لذلك المحتوى. ومع ذلك، فإن ICNs ضعيفة للغاية من ناحية الأمان، لذا فإن أمان المحتوى مهم جدًا، فنحن نحدد العديد من الهجمات المكتشفة في الأدبيات وهي: رفض الخدمة، وحظر الهاتف المحمول، والاعتراض، وما إلى ذلك، المصنفة وفقًا لمكون الشبكة المهاجم (ذاكرة التخزين المؤقت والتوجيه والبنية التحتية).

نحن مهتمون في عملنا بالتصدي لبعض الهجمات المتعلقة بالتوجيه، مثل الإغراق (رفض الخدمة عن طريق الإغراق) والخداع (هجمات الاعتراض). لتحقيق أهدافنا، بدأنا أولاً بدراسة هذا النموذج الجديد، وهوICN، ثم جانب هجمات الفيضانات (Flooding)حيث وضحنا سيناريوهات الهجوم المختلفة والإجراءات المضادة الحالية، واقترحنا تحسين الإجراء المضاد "Poseidon" من خلال اقتراح مقياس جديد له المستوحى من TTL، أدى هذا الحل إلى تحسين أداء الهجوم المضاد الحالي بشكل ملحوظ. ثم درسنا السيناريوهات المختلفة لهجمات الانتحال واقترحنا آلية أمان تعتمد على التوقيع الرقمي لحزم المحتوى. قمنا بمحاكاة جميع سيناريوهات الهجوم الممكنة والحلول المقترحة باستخدام حزمة CCN-lite ضمن محاكي OMNeT ++. النتائج التي تم الحصول عليها واعدة للغاية.

الكلمات الرئيسية: شبكات المعلومات المركزية (ICNS) ، التوقيع الرقمي ، TTL، OMNET ++، ccn-lite، بوسيدون ، إغراق الاهتمام

TABLE DES MATIERES

LISTE DES FIGURES

LISTE DES TABLEAUX

INTRODUCTION GENERALE

Ces dernières années, Internet s'est développé à un rythme incroyable, imprégnant presque tous les aspects liés à notre vie quotidienne. Travail, divertissement, réseautage social, mobilité, interactions longue distance : ce sont autant d'exemples de contextes qui ont été largement touchés et souvent révolutionnés par l'utilisation des technologies TIC. Néanmoins, cette tendance devrait s'accentuer encore plus dans le prochain avenir grâce à l'introduction de différents nouveaux paradigmes : Internet des objets, réalité virtuelle / augmentée, Cloud Computing...

Par ailleurs, ces technologies sous-jacentes apportent avec elle constamment l'innovation pour soutenir les applications informatiques modernes et aussi pour les rendre durables de différents points de vue : environnemental, culturel, social, politique. Parmi ces innovations, une nouvelle architecture réseau a été proposée par la communauté des chercheurs ces deux dernières décennies, qui pourra être adoptée pour l'infrastructure internet, c'est les réseaux centré sur l'information (Information Centric Networks ICN), cette technologie se focalise sur le contenu peu importe où se trouve ou comment faire pour l'avoir, les utilisateurs peuvent récupérer les données (contenus) n'importe où il se trouvent sans établir des connexions ou suivre les protocoles du réseau. Il suffit seulement retrouver un contenu par son nom, grâce aux mécanismes implémentés dans ce nouveau paradigme de réseaux qui lui rend plus en plus efficace. Plusieurs architectures ont été proposées dans la littérature, notre projet est dédié aux architectures des réseaux centrés contenu (CCN pour Content Centric Network).

Problématique

L'architecture CCN a introduit des nouveaux concepts et mécanismes qui règlent les problèmes liés aux anciennes architectures. Cependant, comme tout nouveau paradigme proposé dans le panorama des réseaux, plusieurs défis soulèvent continuellement de nouvelles questions, nous nous concentrerons sur les aspects de sécurité, un des aspects les plus important que les chercheurs essayent de traiter et d'améliorer dans les CCN, d'autant plus de

nouveaux types d'attaques sont apparus et d'autres ont hérité de l'ancienne architecture, nous citons comme exemple : les attaques de routage comme le Flooding, Jamming et Interception, ainsi que les attaques lié au nommage et au cache .

Malheureusement peu de contre-mesures ont été proposées pour toute cette panoplie d'attaques. Ne pouvant pas proposer une seule solution pour contrer toutes ces attaques, nous essayerons de se concentrer dans notre travail sur les attaques liées au routage en particulier le Flooding et d'interception.

Objectifs

Ce travail fait partie d'un projet initié en 2016/2017 par un groupe d'enseignants chercheurs au niveau de l'université Blida 1[*], c'est une continuité de trois travaux de PFE antérieurs **[HAC 16] [RAM 17] [LAT 18]**.

Notre objectif principal étant de proposer un mécanisme de sécurité contre les attaques liées au routage, malheureusement, les attaques qui ont été recensées dans la littérature sont très diversifiées et comportent des scénarios très variées, de ce fait, nous nous sommes fixés à deux objectifs principaux à savoir :

1) Proposer une solution améliorée pour conter les attaques de Flooding en se basant sur contre-mesures existantes du DDos.
2) Proposer un mécanisme de chiffrement du contenu pour éviter les interceptions de contenu.

La proposition de plusieurs mécanismes de sécurité à la fois doit passer par les sous-objectifs suivants :

- Etudier l'architecture des réseaux centrés contenu tout en explorant l'aspect sécurité dans les CCN.
- Comprendre les différents scénarios des attaques de Flooding et les limites des approches de contre mesure proposées dans la littérature, proposer une solution de contre-attaque Flooding et tester ses performances par rapport à la littérature.

[*] Le projet est intitulé « Proposition d'un mécanisme de sécurité pour les Réseaux ICN » dirigé par Pr. Boustia Narhimène avec la collaboration de Mme Arkam Merièm et Mme Arrousi Sana dans le cadre d'une thèse de doctorat Science au sein de l'Université Blida 1.

- Etudier les différentes attaques d'interception (Spoofing) et proposer une solution de sécurité basée sur la signature numérique contre ces types d'attaques.

Structure de la mémoire

Afin d'atteindre nos objectifs fixés, nous structurons notre mémoire en quatre chapitres distincts :

Chapitre I: Nous étudions dans ce chapitre les différentes architectures ICN, leurs caractéristiques et leurs composants.

Chapitre II: Nous aborderons dans ce chapitre l'une des attaques de routage qui est l'inondation des intérêts (Flooding), ses différentes approches de contre mesure et leurs limites.

Chapitre III: Dans ce chapitre, nous introduisant le concept de nombre de sauts (TTL) et nous allons l'intégrer dans l'algorithme de Poseidon dont le but de l'améliorer, puis, nous testons ce nouveau algorithme (poseidon_TTL) sur le simulateur omnet++ et le package ccn-lite.

Chapitre IV: Dans ce dernier chapitre, nous faisons une étude le l'art sur les attaques d'interception dans les CCN, puis nous proposons une solution basée sur la signature numérique des paquets de contenus contre ces attaques et nous la testerons sur omnet++ et ccn-lite.

CHAPITRE I.

CONCEPTS FONDAMENTAUX SUR LES ICN

CHAPITRE I. CONCEPTS FONDAMENTAUX SUR LES ICN

I-1 Introduction

La tâche de sécuriser un système ou un réseau est très complexe, en effet, avant de penser à sécuriser un système ou une architecture, il faudra tout d'abord prendre en considération tous les scénarios possibles dans lesquels un paradigme ou une technologie donnée peut fonctionner et la capacité de prévoir d'éventuels comportements malveillants qui ne sont pas ceux attendus. En conséquence, il est possible, en général, d'étudier les problèmes de sécurité sous différents angles : analyser les nouveaux protocoles ou architectures système, détecter différents types d'erreurs ou de défauts de conception.

Dans cette optique, nous essayerons dans ce premier chapitre d'étudier en détail l'architecture et le fonctionnement de ce nouveau paradigme à savoir les Réseaux Centrés Informations (ICN), leurs différentes architectures, leurs fonctionnements, leurs avantages et leurs limites avant de se concentrer sur l'architecture qui nous intéresse le plus à savoir celle des CCN.

I-2 Passage au Réseau Centré sur l'information

Au Passé le réseau est centré sur l'hôte pour déterminer la localisation d'une information reçue, mais, avec le développement technologique, la caractéristique le plus importante est le contenu et sa sécurité où un nouveau concept a été découvert sous le nom d'un réseau centré sur l'information (ICN).

Le paradigme ICN forme la future architecture Internet qui s'est contrée sur les données elles-mêmes plutôt que sur leurs emplacements dans le réseau. **[MAR 17]** Il s'agit d'un passage d'un modèle de communication centrée sur l'hôte vers un système centré sur le contenu en se basant sur des noms de contenu uniques et indépendants de la localisation, la mise en cache dans le réseau et le routage basé sur les noms. Grâce à ses avantages pertinents, l'ICN peut

être un Framework fiable pour soutenir l'internet des objets, interconnectant des milliards d'objets contraints hétérogènes **[MAR 17]**.

En effet, ICN permet l'accès facile aux données et réduit à la fois le délai de récupération et la charge des requêtes sur les producteurs de données.

I-3 Différentes fonctions de l'ICN

ICN tente de révolutionner l'infrastructure Internet actuelle en introduisant des nouvelles fonctionnalités qui n'étaient pas présent dans le passé dans le but est de passer de l'internet basé sur la localisation de contenu à l'internet basé sur le contenu seulement. Les principales fonctionnalités intégrées dans ICN sont :

I-3-1 Nommage des contenus

Le contenu dans les ICN doit avoir un nom unique pour le routage centré contenu, il doit avoir une certaine structure. Chaque architecture ICN possède son propre nommage des contenus.

Il existe trois types de nommage dans les ICN **[LIA 18]** :

1. Nommage Hiérarchique:

- ✓ Similaire au DNS actuel
- ✓ Est en corrélation avec les topologies de réseau sous-jacentes.

2. Nommage à Base Plate :

- ✓ Généralement fait par hachage

3. Nommage Hybride :

- ✓ Plus expressif, plus riche en structures sémantiques,
- ✓ Peut se combiner avec les deux schémas de nommage précédents

I-3-2 Routage Basé sur le contenu

L'utilisateur exprime seulement son intérêt (le contenu demandé) ensuite le réseau s'occupe de trouver des bons chemins pour arriver à la destination souhaitée (à la source du contenu demandé). Une fois l'intérêt retrouvé, la livraison du contenu se fait en traversant le chemin inverse du message de l'intérêt jusqu'au retour vers le demandeur. Le routage ICN est catégorisé principalement en deux, le routage par nom et routage par résolution de nom **[MAR 17]**.

- **Routage par résolution de nom :** Dans cette catégorie, le client envoie une demande de contenu, Le système de résolution (NRS) qui est chargé de stocker les informations de l'emplacement du fournisseur, relie le nom du contenu avec son localisateur.
- **Routage par nom :** Cette approche est basée sur la hiérarchie de nom de contenu pour acheminer la demande au fournisseur et lors de l'envoie de la demande, chaque routeur doit être au courant d'une partie des informations de routage.

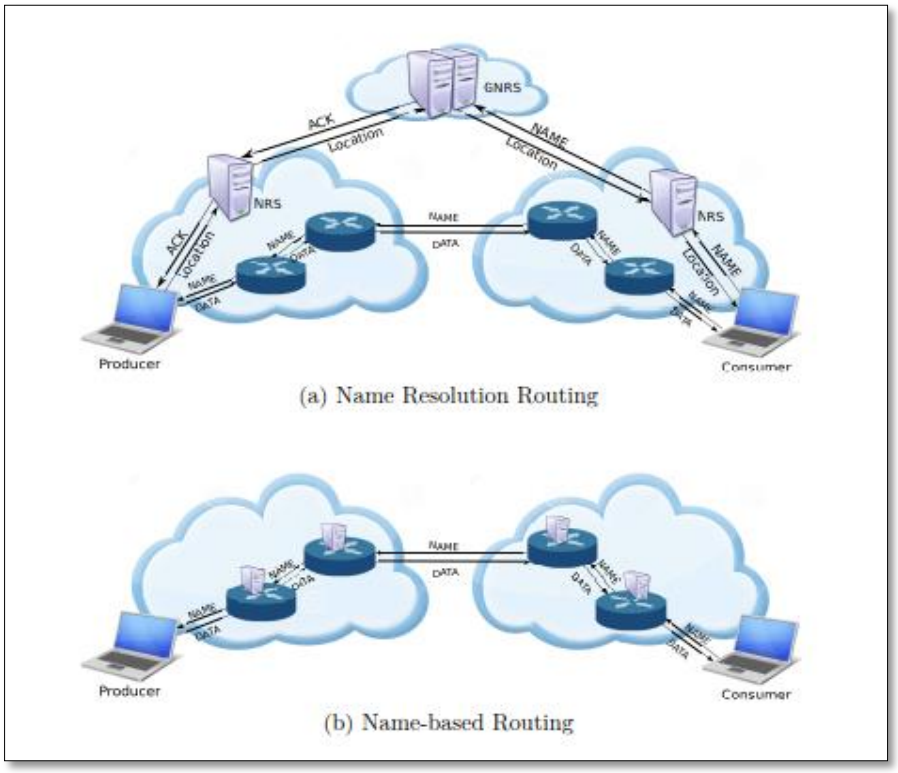

Figure I-1 Les Catégories de Routage Centré Contenu [MAR 17]

Parmi ces deux approches, Nous travaillons avec le routage par nom car nous disposons d'un environnement de développement (IDE) et un simulateur qui nous oblige à exploiter cette approche.

A travers le temps, elle nous a offert plusieurs avantages parmi ces avantages :

- ✓ La rapidité de réponse d'une demande d'intérêt.
- ✓ Chaque nœud est chargé du nommage et résolution des noms lui-même sans aller à un serveur du nommage.
- ✓ La flexibilité, car un nœud n'est pas affecté par un changement de la topologie (Mobilité).

I-3-3 Cache de contenus

La mise en cache des nœuds ICN fait partie intégrante de l'ICN. Tous les nœuds ont potentiellement des caches, y compris des nœuds dans des réseaux d'infrastructure gérés par l'opérateur et des réseaux domestiques gérés par l'utilisateur, ainsi que des terminaux mobiles. Les demandes des clients peuvent être satisfaites par n'importe quel nœud détenant une copie dans son cache. ICN combine ainsi la mise en cache à la périphérie du réseau, comme dans le réseau de bout en bout, avec la mise en cache établie dans le réseau. La mise en cache est générique, c'est-à-dire qu'elle est indépendante de l'application et s'applique à tous les fournisseurs de contenu, y compris le contenu généré par l'utilisateur **[BEN 12]**.

I-3-4 Sécurité du contenu

La sécurité d'ICN est largement assurée par l'authentification de nœud à partir laquelle le contenu est fourni et les communications sont sécurisées pour interdire les attaques de l'homme du milieu **[FAB 13]**.

En conséquence, l'authenticité et l'intégrité du contenu doivent pouvoir être vérifiées uniquement par les informations fournies à côté des données (client légitime).

En d'autres termes, l'essentiel est d'assure qu'un certain contenu a été fait publier par la bonne source et que le contenu n'a pas été modifiée en fonction du son passage, Il faut donc introduire un mécanisme de signature numérique. Donc on a exigé les demandeurs de contenu qu'ils maintiennent des paires de clés publiques et privée pour signer les paquets de données. Les données sont transférées non seulement avec leur nom, mais aussi avec la signature et certaines informations signées, Les informations signées ne doivent pas nécessairement être

Lisible durant la transmission Ainsi, le consommateur peut vérifier si le contenu en question a été publié sous ce nom par un clé donnée ou non

I-3-5 Mobilité du contenu

La mobilité dans ICN vise à permettre aux clients de recevoir à tout moment le contenu sans perturbation perceptible dans les applications ICN. En raison d'une extraction de contenu par récepteur dans ICN un changement de localisation physique d'un client n'interrompt pas la réception continue de contenu. D'autre part, côté serveur La mobilité est difficile à soutenir. En particulier, La mobilité de l'ICN doit travailler en collaboration avec le routage ICN pour découvrir un objet de donnée ou il est identique à celui qui est transmis **[KEP 19]**.

I-3-6 Interface de programmation d'application (API)

Une API dans ICN est utilisée pour demander et livrer le contenu. La source publie son contenu pour le rendre disponible pour les autres utilisateurs du réseau. Un utilisateur envoie un message d'abonnement pour le contenu qui l'intéresse. Les deux opérations (publication et abonnement) utilisent le nom du contenu comme paramètre principal. Le tableau I résume les différences importantes entre les architectures centrées sur l'hôte et ICN

Le tableau I.1 résume les différences importantes entre les architectures centrées sur l'hôte (réseaux TCP/IP) et les architectures centrées Informations (réseaux ICN)

Tableau I-1 Comparaison entre réseau centré sur l'hôte et ICN [ESL 15]

	Centré sur l'hôte	ICN
Nommage	Définir la position topologique de l'hôte	Définir le contenu indépendamment de son emplacement ou sa représentation
Routage	Entre hôtes utilisant des adresses IP.	À l'aide d'une entité de résolution de noms ou d'un routage basé sur un nom.
Cache	Points de mise en cache spécifiques (serveurs de cache).	Chaque nœud peut mettre en cache tout contenu qui le traverse
Sécurité	Canaux de communication sécurisés entre hôtes	Sécuriser le contenu lui-même
API	Envoyer des données à une adresse spécifique	Publier et souscrire des contenus

I-4 Principaux Projets ICN

Plusieurs projets ICN ont été mis en place par des laboratoires de recherches tels que DONA, NETINF**,** PSIR**,** CCN La faisabilité de ces nouvelles architectures proposée a été démontrée par des petits projets de 5 à 10 nœuds. L'objectif principal était de fournir un concept opérationnel des ICN à une échelle raisonnable avec des scénarios réalistes.

Par la suite, de nouvelles architectures ont vu le jour, dans cette section nous essayerons de présenter les principaux projets classiques d'ICN et puis les Nouvelles architectures.

I-4-1 Les Projets classiques

1) NetInf (Network of Information)

NetInf utilise également un espace de noms plat d'où un public l'infrastructure clé (PKI) n'est pas requise, Le modèle de contenu de NetInf est basé sur Standard MIME (Internet Mail Extensions) largement utilisé **[Has 15]**.

En outre, les primitives de recherche, qui fournissent des liens entre l'élément de recherche et le nom de l'objet, font également partie de l'architecture. Deux approches de récupération d'objets sont proposées dans l'architecture, à savoir la résolution de noms et le routage basé sur les noms. Selon le modèle utilisé, le nœud source peut soit s'enregistrer avec la résolution de nom service (NRS pour Named Resolution Service) pour publier le contenu appelé objet de données nommées (NDO pour Named Data object) ou utiliser un protocole de routage pour annoncer les informations de routage.

Comme illustré sur la Figure I-2, dans le premier cas :

- Le client transmet d'abord la demande au NRS, qui donne les localisateurs disponibles du nom NDO particulier.
- Le client récupère une copie des données à partir des meilleures sources disponibles.

Dans le deuxième cas de routage par nom :

- Le client peut envoyer directement une requête de demande (GET) d'objet de donnée nommées qui est transmis à la source.
- Les données sont transmises au client dès qu'une copie de le NDO est atteint.

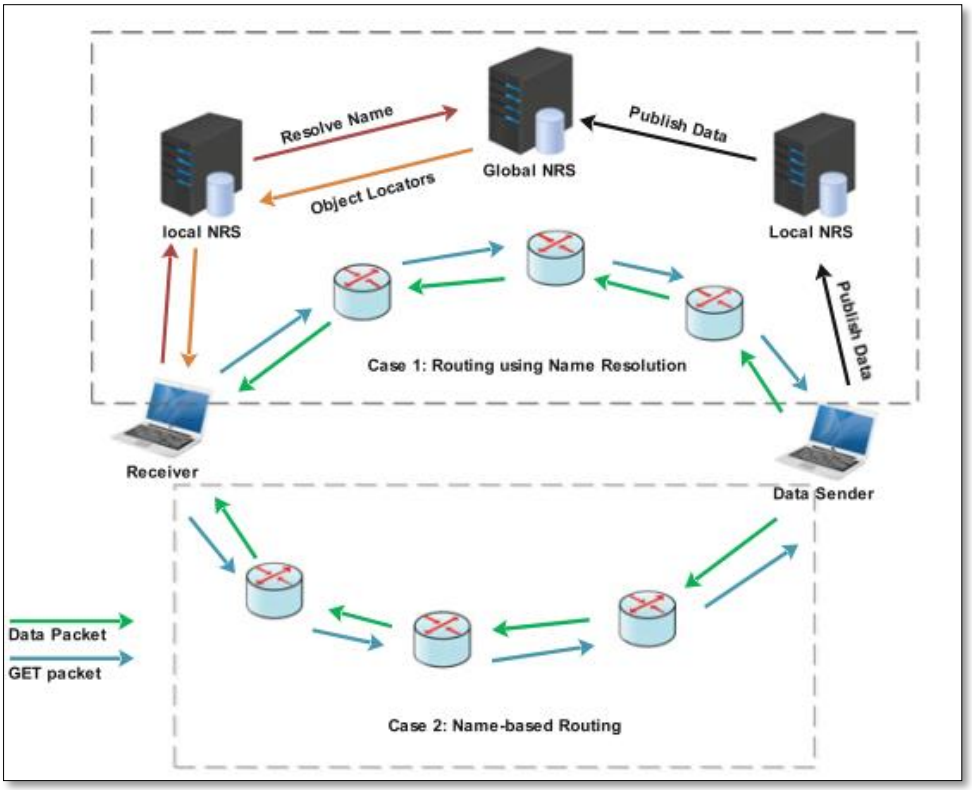

Figure I-2 Schéma de routage de NetInf [Has 15]

Ceux-ci peuvent être utilisés séparément dans le réseau ou fusionnés dans un schéma hybride, auquel cas la commutation entre les deux schémas est formée sur une base saut par saut. Ce schéma hybride permet à NetInf de s'adapter et d'évoluer lui-même aux différentes exigences du réseau telles que la mobilité du réseau à tolérance de retard (DTN) et connectivité globale. De plus, NetInf l'architecture peut être déployée en tant que couche supplémentaire au-dessus du réseau existant infrastructure, simplifiant ainsi la migration des applications vers le nouveau Infrastructure **[Has 15]**.

2) DONA (Data-Oriented Network Architecture)

L'architecture de DONA implique un remaniement de la dénomination actuelle de l'internet, c'est-à-dire que les noms DNS sont remplacés par des noms auto-certifiés, et la résolution de noms DNS est remplacée par un processus de résolution de noms distribués. En outre, ces changements sont incorporés au-dessus de la couche IP, ce qui permet d'exploiter les couches inférieures des mécanismes de découverte de chemin. Cette architecture permet une meilleure récupération des données ainsi qu'un meilleur service en assurant la persistance, l'authentification et la disponibilité **[Has 15]**.

Au sein de DONA, le fournisseur de source/contenu est responsable de la publication du

contenu sur le réseau. Pour servir les données, les nœuds doivent obtenir l'autorisation de l'infrastructure de résolution. Un paradigme de route par nom est utilisé pour la résolution des noms. Maintenant, au lieu d'utiliser des serveurs DNS, DONA s'appuie sur les entités du réseau appelées "gestionnaires de résolution" (RH). Les paquets de requête (FIND) sont transmis par plusieurs RH vers le nœud avec une copie du contenu, comme illustré dans la Figure I.3 Le contenu/les données peuvent être acquis par deux méthodes :

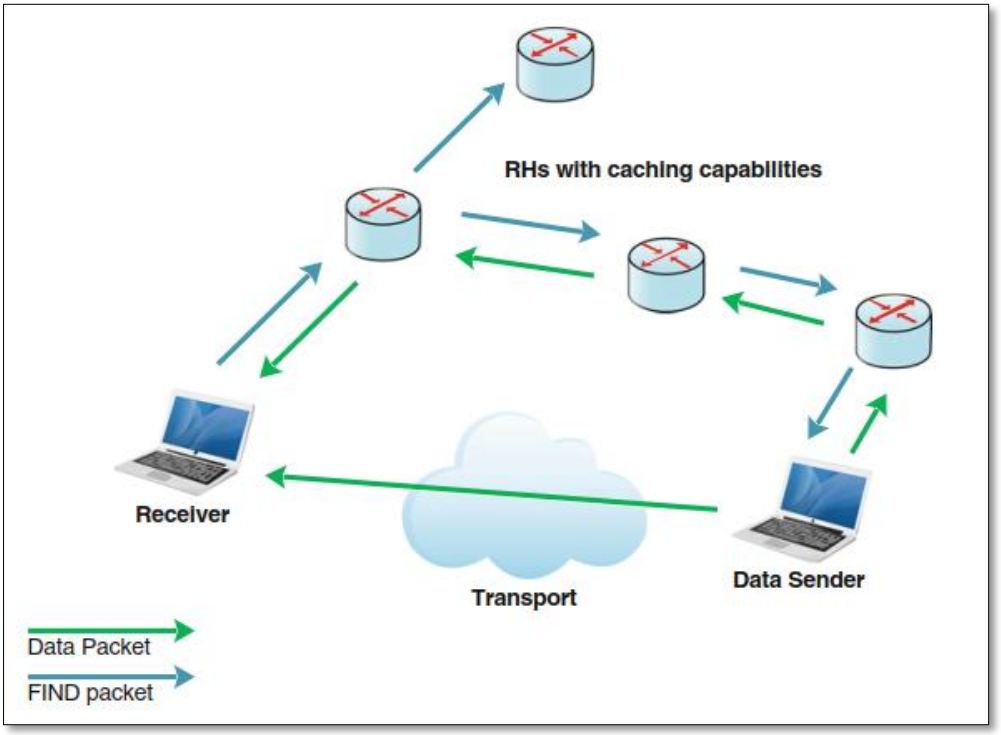

Figure I-3 Schéma de routage de DONA [Has 15]

1. Il est renvoyé par le même chemin que le paquet d'intérêt, la mise en cache étant activée pour chaque RH rencontré,
2. Il peut être renvoyé directement vers le consommateur. La source a également la possibilité d'enregistrer ses mandants auprès du RH afin que les paquets de demande puissent lui être envoyés directement. Toutefois, les enregistrements doivent être renouvelés périodiquement :
    - Le RH achemine les demandes en utilisant une approche hiérarchique vers FIND, le fournisseur de contenu le plus proche.
    - La résolution des noms de distribution utilisée dans DONA permet de prendre en charge les boîtes intermédiaires du réseau (par exemple firewalls, les proxy). En prévoyant un mécanisme distinct pour la découverte des chemins.

3) PSIRP (Publish-Subscribe Internet Technologie)

Le projet Publish-Subscribe Internet Technologie (PURSUIT) était auparavant connu sous le nom de paradigme de routage Internet Publish-Subscribe (PSIRP), Dans PURSUIT, les sources publient le contenu sur le réseau comme indiqué dans la Fig.I.3 Les destinataires peuvent s'abonner aux contenus publiés via les systèmes rendez-vous **[Has 15]**.

Un système de rendez-vous aide à localiser la portée et les publications dans le réseau. Chaque élément du contenu publié appartient à une portée nommée spécifique Les demandes d'abonnement contiennent l'identificateur de portée (SI) et l'identificateur de rendez-vous (RI), qui, ensemble, identifient / nomment le contenu particulier souhaité. En utilisant ces identifiants dans une procédure de correspondance conduit à un identifiant de transmission (FI) qui est utilisé par la source pour transmettre les données.

Un filtre Bloom est spécifié dans le FI, qui est utilisé par les routeurs intermédiaires pour sélectionner les interfaces à transférer contenu comme indiqué sur la Figure I.4. Cela évite au routeur de maintenir États protecteurs. Cependant, un filtre à efflorescence donne des résultats faussement positifs, donc conduisant à la transmission sur des interfaces où il n'y a pas de récepteurs **[Has 15]**.

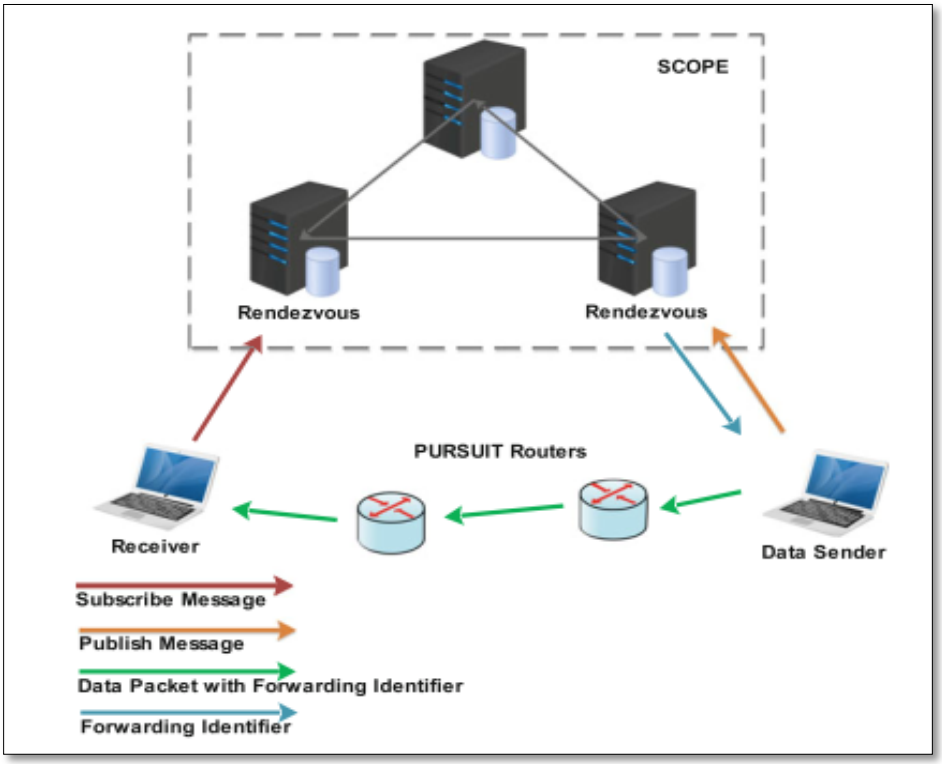

Figure I-4 Schéma de routage de PSIRP [Has 15]

4) Content-Centric Networking

Content Centric Networks **[FAB 13]** proposée par les chercheurs de centre de recherche Palo Alto en 2009, devenu l'un des projets à financer dans le cadre du futur internet en US, il utilise le nommage hiérarchique lisible par l'humain, la mise en cache du contenu dans le réseau, le routage basé sur le contenu. Par exemple un nom hiérarchique dans le CCN peut être comme cette figure (I-4) le montre :

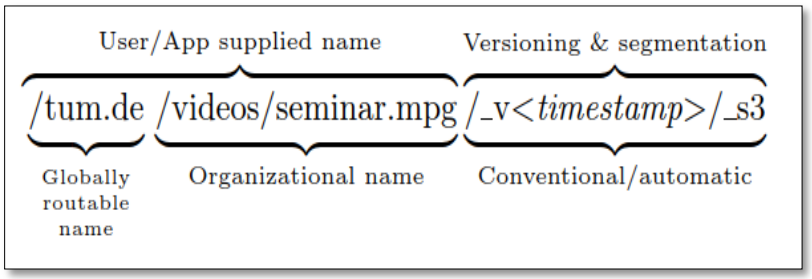

Figure I-5 Exemple de nom hiérarchique de CCN [FAB 13]

Cette figure nous montre un exemple d'un nommage dans les CCN dans d'un fichier vidéo est identifié par le nom : **/tum.de/videos/seminair.mpg/_v<timestamp>/** et ses morceaux avec des noms de type **/tum.de/videos/seminair.mpg/_v<timestamp>/_s_id** où **_s_id** représente l'identifiant du segment (paquet de données).

Cette dénomination hiérarchique permet de faciliter les décisions du routage dans les CCN. Comme notre travail est principalement sur les CCN, donc on doit bien les comprendre en couvrant ses différents aspects et fonctionnalités.

CCN fonctionne avec des données nommées. Chaque nom a une structure hiérarchique : il commence par un segment de nom global, est suivi d'un nom dépendant du système et de l'application, puis d'un numéro de version et d'un segment de protocole, cette structure réduit considérablement la taille de la table de transfert sur le routeur d'un CCN, afin d'accélérer le processus d'acheminement des paquets. Donc, elle remplace l'adresse IP et le numéro de port dans l'architecture actuelle de l'internet. Les informations sont traitées par des messages d'intérêt, envoyés par le client, et des messages de données, envoyés par le serveur.

I-4-2 Projets récemment apparus

1) GreenICN

Le projet GreenICN a été l'un des premiers projets de collaboration entre le groupe européen FP7 et le Japon : les projets de recherche en collaboration, qui ont débuté en Avril 2013 et

achevé en mai 2016. Ce projet a été dédié à travailler sur des systèmes hautement scalables et à haute efficacité énergétique pour les réseaux et les diapositifs ICN, un des objectifs de cet architecture est de mettre en œuvre une infrastructure pour une distribution efficace des notifications de catastrophe et les informations de sauvetage critique à travers une limite de ressources énergétique et de communication.

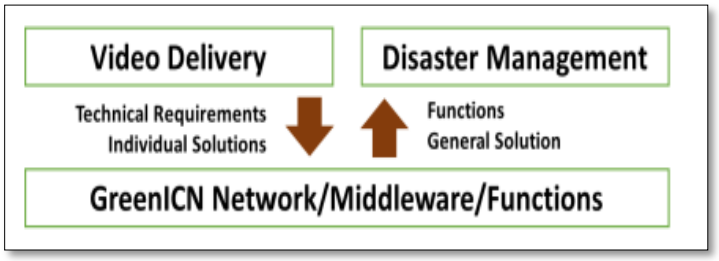

Figure I-6 Concept Green ICN présenté dans [KEP 19]

2) ICN2020

Commence en juillet 2016, ce projet **[KEP 19]** a été très financé et qui a basé sur des grandes études d'ICN pour réaliser plusieurs objectives, Parmi ces objectives :

- Adapter ICN pour compléter la 5G.
- Concevoir et développer les fonctionnalités des application IoT et des service ICN.
- Améliorer les Solutions au fonction vital de l'infrastructure basée sue le ICN.
- Le déploiement d'ICN dans le monde entier.

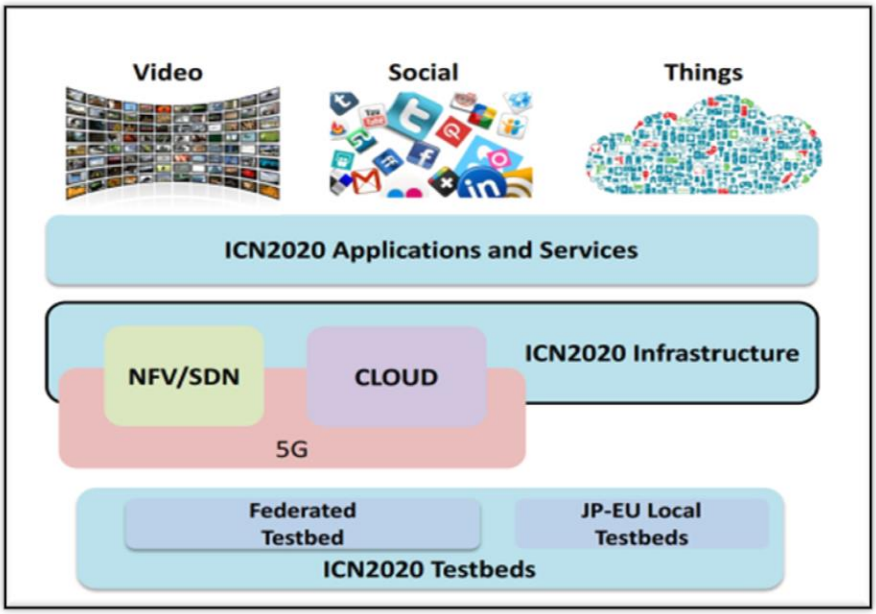

Figure I-7 Concept ICN2020 présenté dans [KEP 19]

I-5 Comparaison des architectures classiques des ICN

La comparaison entre les projets ICN classiques a été faite selon les fonctions de base reportées dans le tableau I-2 :

Tableau I-2 Comparaison entre les principes de base d'ICN [BEN 12]

	DONA	**CCN**	**PSIRP**	**Netlnf**
Espace du nom	Plat avec structure	Hiérarchique	Plat avec structure	Plat avec structure
Nom d'intégrité du domaine	Signature, PKI indépendant	Signature, source de confiance externe	Signature, PKI indépendant	Signature ou hachage de contenu, PKI indépendant
Nom lisible par homme	Non	Possible	Non	Non
Modèle d'abstraction d'information	Non	Non	Non	Oui
Type de données	Objet	Paquet	Objet	Objet
Agrégation du routage	Editeur explicite	Editeur	Porté explicite	Editeur
Routage de la demande	Basé sur le nom (via RHs)	Basé sur le nom	NRS rendez-vous	Hybride NRS et Basé sur le nom
Routage	Réserver le chemin de la demande ou une connexion IP directe	Réserver le chemin de demande à l'aide de l'état du routeur	Routage source à l'aide d'un filtre de bloom	Réserver le chemin de la demande ou une connexion IP directe
API	Obtenir la synchrone	Obtenir la synchrone	Editeur abonnement	Obtenir la synchrone
Transport	IP	Beaucoup, y compris IP	IP/PSRIP	Beaucoup, y compris IP

A travers nos études, la comparaison est faite entre les architectures ICN classique seulement, Pour cela nous n'avons pas pu comparer les architectures récemment apparus car ces projets sont en cours de développement et il n'y a pas d'assez d'information pour l'avoir tous comparer.

Nous nous intéressons beaucoup plus aux aspects liés aux CCN, d'après notre tableau récapitulatif des différentes caractéristiques de ces architectures, nous pouvons constater que

les CCN sont très vulnérables, un exemple très concret, le nommage des données est lisible par l'homme, d'autant plus le routage est basé sur le nom du contenu, ce qui rend le système vulnérable aux attaques d'inondations en falsifiant facilement l'intérêt envoyé, les intérêts ainsi que le contenus sont envoyé en paquets, ce qui rend leur interception très facile…

A la base de ces deux exemples de faille sécuritaire que nous venons de citer, notre travail s'articulera. Nous essayerons dans les chapitres suivants à les traiter en proposant des solutions plus fiables et plus performantes.

Puisque notre travail s'articule sur la sécurité liée au routage dans les CCN, nous détaillons dans la section suivante le fonctionnement du routage des CCN avant de passer aux attaques et aux contre-mesures existantes.

I-6 Routage dans CCN

Le routage dans les CCN fonctionne principalement comme suit : lorsqu'un client veut des données, il envoie un paquet d'intérêt à une ou toutes les faces qu'il voit. A chaque nœud du réseau, si le nœud a le paquet désiré est en cache, il consommera l'intérêt et répondra avec les données. Il est important de noter que les paquets sont les mêmes lorsqu'ils sont mis en cache et en mouvement dans ce système, de sorte que tous les paquets sont considérés comme égaux, ceux en mémoire et ceux sur le réseau. Si le nœud n'a pas les données désirées, il suivra un protocole de routage et transmettra l'intérêt au nœuds suivants pour arriver à la donnée souhaitée, une fois trouvée elle est envoyée à la source demandeuse en suivant le chemin inverse de l'intérêt et une copie de cette donnée est stockée dans chaque mémoire cache des routeurs intermédiaires pour une récupération rapide de la donnée dans les demandes ultérieures.

I-6-1 Type des paquets dans CCN

Donc on n'a pas besoin du mécanisme d'acheminement des adresses source et destination dans le réseau pour la demande d'une ressource ou une donnée. Alors, on distingue deux types de paquets dans Les CCN :

1) Paquet Intérêt

L'émetteur annonce sa demande pour un contenu nommé (données) par ce paquet d'intérêt. Il est simplement diffusé sur les interfaces disponibles dans l'espoir d'obtenir les données correspondantes, Naturellement, le paquet contient le nom du contenu souhaité (Name Content). De plus, il est accompagné d'informations de sélection, telles que la portée dans le

réseau d'où les données doivent provenir ou certaines informations de filtrage (Selector).

Enfin, il contient un Nonce, utilisé pour détecter les intérêts en double (Figure I-8).

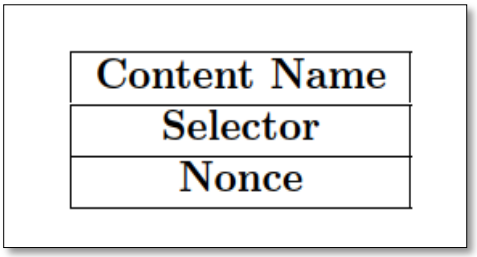

Figure I-8 Format de Paquet Intérêt [FAB 13]

2) Paquet de donnée

L'objectif d'un paquet de donnée est de " satisfaire " l'intérêt en ce qu'ils maintiennent une relation de un à un, où les données consomment l'intérêt. Cette règle de satisfaction permet de limiter la Congestion dans les réseaux et augmenter ses performances. Le paquet contient également une signature numérique d'un algorithme de digestion (cryptographique), ainsi que des informations signées. Le dernier champ mentionné donne des informations supplémentaires sur le paquet, comme le ID, où trouver la clé pour vérifier la signature, dont le but est d'assurer l'authenticité et l'intégrité du contenu, enfin il contient la donnée désirée (Figure I-9).

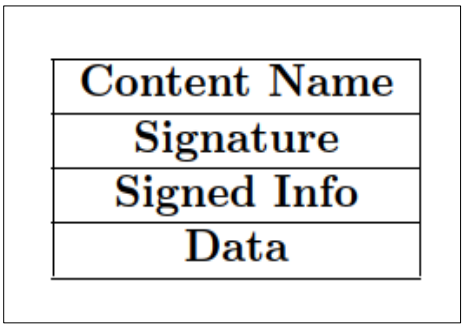

Figure I-9 Format de paquet de donnée [FAB 13]

I-6-2 Modèle d'un nœud CCN (routeur CCN)

Par rapport à l'internet actuel, les routeurs sont très différents (voir Fig.I.9). Ils contiennent essentiellement trois tables: le stockage de contenu (Content store), la table (PIT) et la table (FIB) **[WEI 14].**

1) Content Store (CS)

C'une mémoire cache (mémoire tampon) qui sert à sauvegarder le contenu reçu dans un nœud CCN (données et non pas intérêt). La possibilité de servir le contenu directement au lieu de

générer d'autres consultations minimise l'utilisation globale de la bande passante et latence et répondre rapidement aux demandes ultérieures.

2) Pending Interest Table (PIT)

Elle a deux rôles principaux. Le premier rôle est de garder une trace des intérêts émis sur les interfaces des nœuds. Ici, il n'est pas important que l'intérêt provienne du nœud lui-même ou est un transmis d'un autre nœud. Grace à ça, dès qu'un intérêt atteint une source de contenu, le PIT sert de marque dans la piste vers son (ses) demandeur(s) et suivre le chemin inverse pour arriver à son (ses) demandeur(s). Le Deuxième rôle de PIT est d'éviter l'envoi multiple des mêmes messages intérêt et un seul est envoyer et le résultat est distribué pour tous les demandeurs.

3) Forwarding Information Base (FIB)

Elle a le même comportement que la table de routage dans un routeur IP. La FIB stocke les informations sur les faces sur lesquelles les intérêts doivent être transmis vers la (les) source(s) du contenu en question

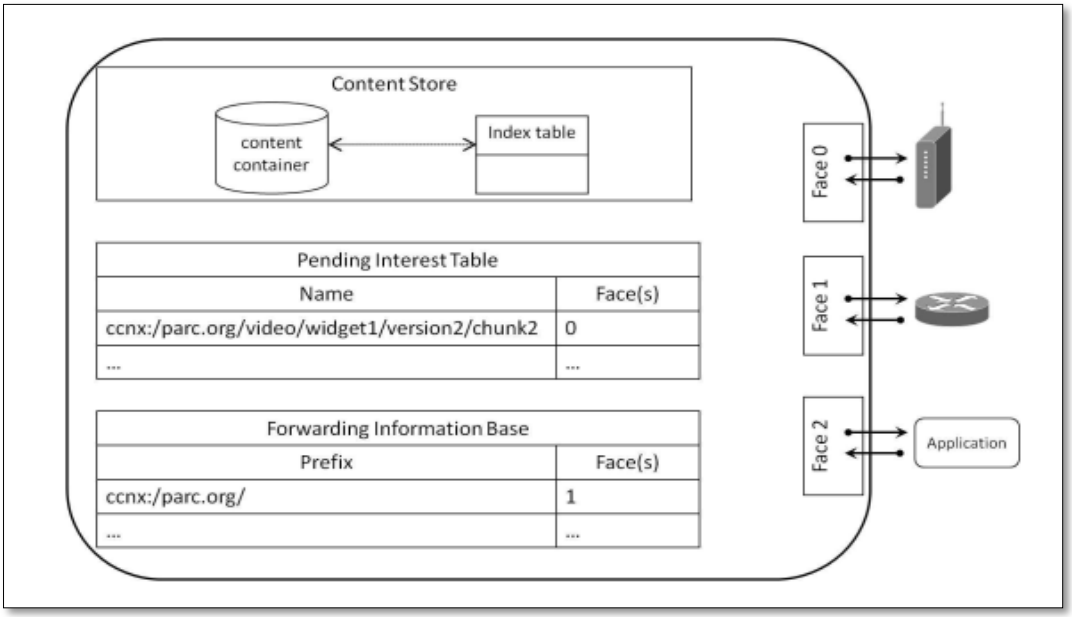

Figure I-10 Structure d'un nœud CCN [WEI 14]

La Figure I.11 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.

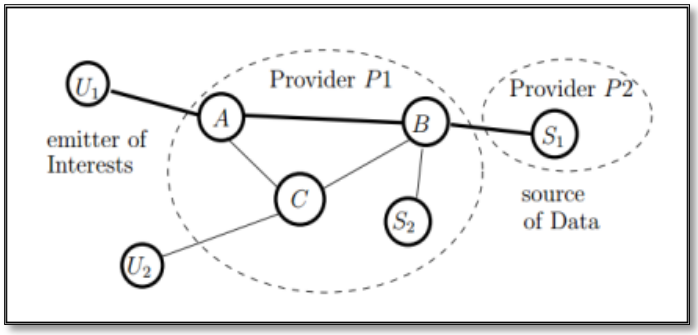

Figure I-11 Un segment du réseau CCN reliant un utilisateur U1 à une source S [NAD 14]

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 contenu 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 contenu sera typiquement stocké en même temps dans le Content Store

Dans l'exemple de la Figure I.12, le nœud A cherche les contenus "video" 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.

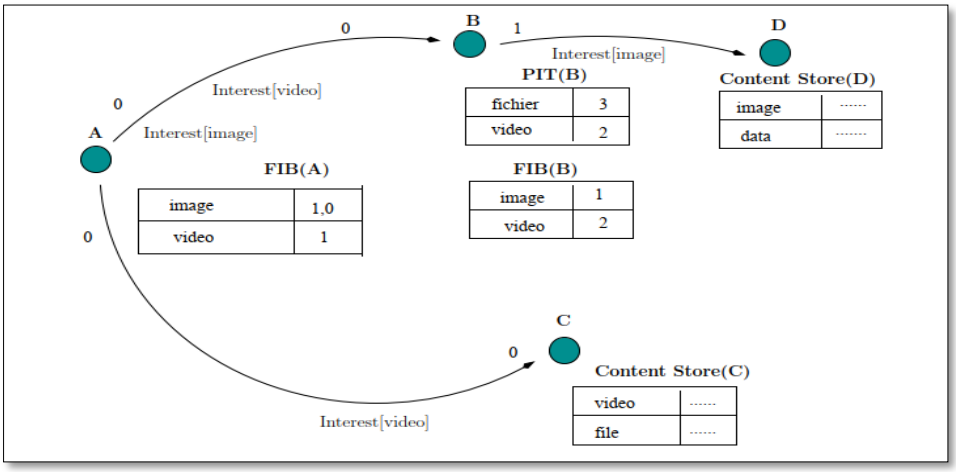

Figure I-12 La recherche des données dans CCN [NAD 14]

I-6-3 La gestion du trafic

La gestion du trafic consiste à contrôler le partage de la bande passante des liens afin d'assurer la qualité de service des diverses applications. Le partage équitable de la bande passante entre flots répond aux exigences des flux sensibles à débit faible et protégé les flux adaptatifs des flux gourmands. La gestion du trafic dans les réseaux IP a fait l'objet de plusieurs travaux de recherches, mais jusqu''a présenté aucune solution entièrement satisfaisante n'a été trouvée.

Jacobson et al **[NAD 14]** proposent un modelé CCN base sur un échange intérêt/donnée : pour recevoir un paquet Data, l'utilisateur devrait envoyer un paquet Interest. Ils assurent que ce mécanisme réalise une bonne gestion de trafic. Ceci est insuffisant en réalité. Tous les utilisateurs n'utilisent pas un protocole de transport unique ; ce dernier peut être modèle par l'utilisateur final. Il est alors nécessaire de définir des mécanismes pour gérer le partage de la bande passante. La détection de rejets par numéros de séquences des paquets n'est plus applicable sur CCN ; l'utilisation des multipaths ou multi chemins change l'ordre des paquets, et le timeout ne suffit pas pour assurer de bonnes performances du protocole de transport. D'autre part, dans CCN, l'utilisateur ne peut plus utiliser un protocole de transport multipath car il ignore les destinations possibles de ces paquets.

En plus des différences entre CCN et IP en termes de mode d'échange de données, les CCN implémentent des caches au niveau de chaque routeur. Ainsi, l'utilisation des caches réduit la consommation de bande passante, et les délais de transmission.

I-7 L'orientation future de ICN

Pour promouvoir une acceptation plus large et un déploiement à grande échelle d'ICN, les problèmes suivants doivent également être résolus

- **Compatibilité avec différentes architecture ICN [KEP 19]**

Au cœur des dernières années, plusieurs architecture ICN ont été proposé qui offrent différentes fonctionnalités et caractéristiques. Il est très difficile de sélectionne une architecture particulière selon les conditions du réseau donnée. Il peut être résolu en établissant une compréhension commune des différents architecture ICN et la compatibilité devait être une partie très importante.

En outres, plusieurs technologies à venir talque IoT, deep-learning, blockchain doit être prise

en compte dans le développement futur de ICN.

- **Support qualité de service et application 5G**

Avec l'apparition des réseaux 5G et l'augmentation du exigences logiciel et matériels des futures applications, la qualité de service est très importante d'où les architectures ICN sont très favorables pour atteindre les Objectifs de la 5G et l'intégrer dans ses infrastructures.

- **Plan d'affaire (puissance modelés)**

Un aspect clé du succès d'ICN est la définition de mécanisme d'incitation pour motiver les utilisateurs à traiter et transmettre des données qui pourraient ne pas les intéresser grâce à la mise en cache pour améliorer la disponibilité de l'information au sein du réseau. En général, cet aspect intéresse les grandes parties commerciales pour la publicité et la disponibilité de ses produits.

- **ICN in Edge-Computing (informatique de périphérique)**

Edge-Computing est une technologie qui appartient aux internet des objets ou elle répond aux préoccupations concernant le temps de réponse nécessaire, la durée de vie des objets IoT, les économies de bande passante, ainsi que la sécurité et la confidentialité des données. Ces dernières années l'ICN a été proposé de faire évoluer le modèle réseau, par conséquent cette architecture est un meilleur choix pour contenir le Edge Computing.

I-8 Conclusion

L'Informatique du futur s'accompagne d'exigences élevées en matière de diffusion de l'information, ce qui motive la communauté des chercheurs à trouver des solutions plus performantes et moins couteuses. ICN est l'une de ces solutions, elle se concentre sur les contenus afin de fournir une diffusion de contenu efficient évolutive. Il existe de nombreuses propositions d'architectures que nous avons illustrées dans ce chapitre, nous avons détaillé le fonctionne de l'une d'elle à savoir les CCN, cette architecture possède des attributs qui la rendent unique par rapport aux architectures centrées sur l'hôte, cependant, elle comporte plusieurs vulnérabilités coté sécurité que nous traiterons dans le chapitre suivant.

CHAPITRE II.

LES ATTAQUES FLOODING DANS LES CCN

CHAPITRE II. LES ATTAQUES FLOODING DANS LES CCN

II-1 Introduction

Avec L'apparition des Réseaux Centrées Informations et plus précisément les CCN dans le monde informatique, plusieurs attaques sont apparues telles les attaques liées à la mise en cache, les attaques liées au routage…Pour contrer ces attaques, et limiter leurs dégâts, les experts du domaine devront intégrer des mécanismes de sécurité assez puissants, mais sans perturber les performances de ces réseaux.

Jusqu'à maintenant, on ne peut trouver une solution unique et globale à toutes les attaques et vulnérabilités dans les réseaux informatiques en général, et les CCN en particulier, de ce fait, on s'intéresse, en premier lieu, aux attaques d'inondation d'intérêts (Flooding), qui sont liées au routage. Les recherches se déroulent de manière continue sur cette problématique. Parmi les contre-mesures proposées dans la littérature, des algorithmes contrôlant le flux de paquet sur Le réseau CCN, et aussi des algorithmes s'inspirant des solutions existantes pour DDOS (attaque de dénis de service) dans les réseaux IP.

Dans ce chapitre, nous commencerons tout d'abord par présenter les différentes attaques dans les ICN, les classer selon la littérature, par la suite nous aborderons les attaques Dos et Flooding en présentant plusieurs scénarios d'attaques, ensuite nous étudierons les contre-mesures existantes dans la littérature, puis nous les comparons selon différents critères d'évaluation.

II-2 Classification des attaques dans les ICN

Dans la littérature, plusieurs travaux **([ESL 15], [REZ 16])** ont essayé de recenser les différentes attaques ICN et de proposer une classification, qui classe les attaques ICN selon **[ESL 15],** en trois catégories, comme le montre dans la figure II-1 : nommage, routage, mise

en cache. Cette classification dépend de l'objectif principal de l'attaquant. Bien que chaque attaque soit incluse dans une seule catégorie, elle peut également influer sur d'autre catégorie.

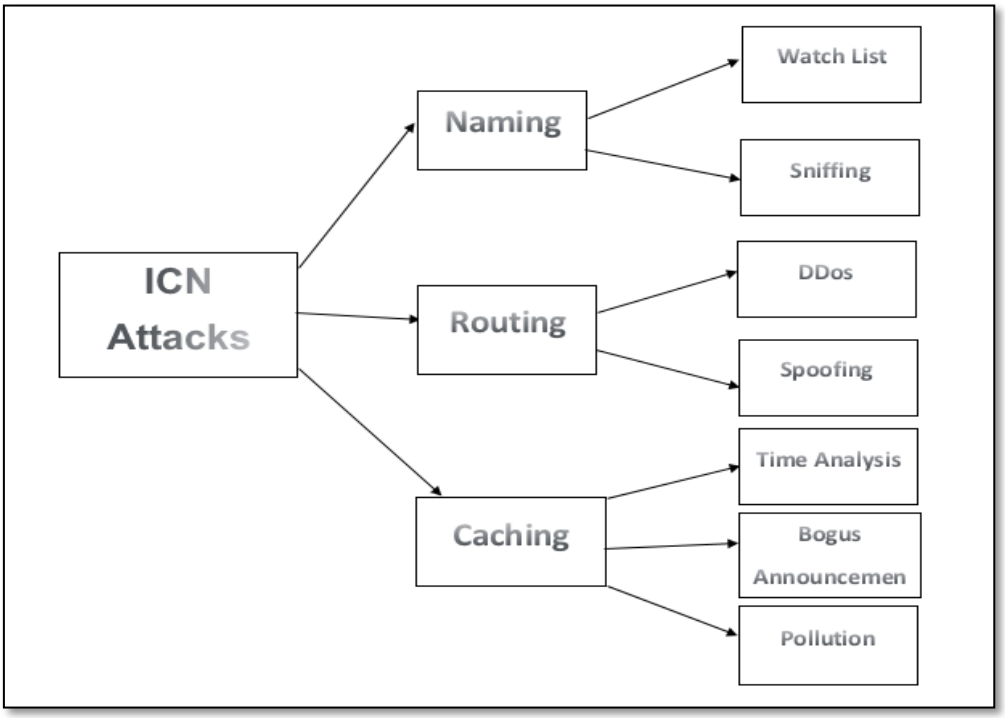

Figure II-1 Classification des attaques dans réseau ICN [ESL 15]

Les catégories proposées sont brièvement présentées dans les paragraphes suivants :

II-2-1 Les attaques de nommage

Les architectures CCN 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 CCN 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

II-2-2 Les attaques de 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.

II-2-3 Les attaques de 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.

II-3 Les attaques de Déni-Service dans CCN

Les attaques de déni de service (DOS) sont un ancien phénomène des réseaux informatiques qui visent principalement à dégrader ou à refuser complètement les services des diapositifs réseau et des serveurs aux utilisateurs légitimes.

II-3-1 Attaques de Zombies

Les attaques de Zombies font partie des attaques de déni de service ou un attaquant maitre contrôle plusieurs diapositifs finaux distribué dans le réseau dont le but de rendre un service indisponible en le saturant à travers des charges massives des requêtes malveillantes envoyées.

Dans CCN, ce genre d'attaque ressemble aux attaques par inondation (flooding) qui visent à dégrader les performances des routeurs dans le réseau et les rendre incapable de faire le routage.

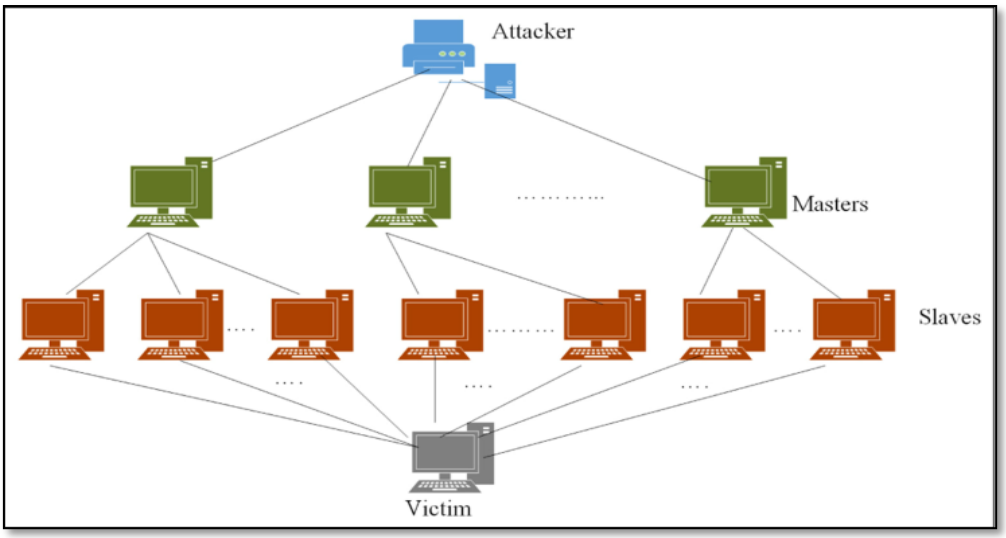

Figure II-2 Structure d'une attaque Zombie [Tas 17]

II-3-2 Attaques d'inondation d'intérêts

Certaines études récentes ont montré que même si CCN fournit une sécurité intégrée contre les attaques par inondation, il y aura une possibilité d'inonder les tables PIT (la surcharge de mémoire de PIT), ceci implique que le routeur ne pourra répondre aux intérêts valides. Le concept est d'envoyer des paquets d'intérêts falsifiés pour épuiser les ressources du routeur ou des sources de données spécifiques.

Comme le CCN est une architecture centrée sur le contenu, il est difficile de limiter le taux de demandes par utilisateur final car il n'y aura pas d'identification d'hôte et donc on risque de négliger les demandes des utilisateurs légitimes.

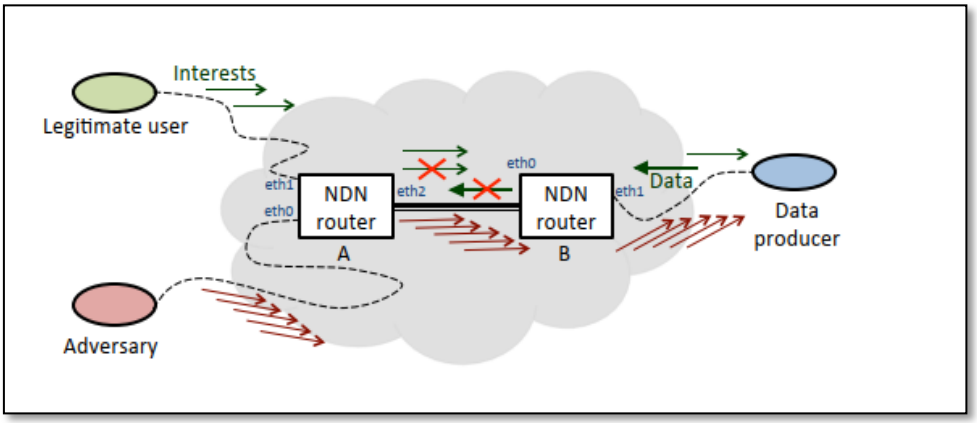

Figure II-3 Exemple d'une attaque d'Intérêts Flooding [Ale 13]

Nous constatons, suite aux recherches effectuées, que les attaques zombie (par

inondation) sont très difficiles à contrôler, les algorithmes de contre-mesure existants jusqu'à présent, à notre connaissance, sont toujours vulnérables.

Nous nous intéressons à contrer ces attaques de Zombies, pour cela nous dériverons en détail la composition de telles attaques.

II-4 Les types d'attaque d'inondation d'intérêts dans CCN

Les paquets d'intérêt dans un réseau CCN sont basés sur le nom de contenu préfixe et l'utilisation des ressources de mémoire dans les routeurs intermédiaire. Ceci peut constituer un outil très pratique pour lancer des attaques flooding et tenter de surcharger le réseau qui provoque des interruptions pour l'utilisateur légitime. **[Ale 13].**

Les recherches faites **[Nar 16]** ont déduit qu'il existe trois types d'intérêts flooding : Contenu existant ou fixe, contenu généré dynamiquement, contenu inexistant, nous les présentons brièvement comme suit :

II-4-1 Contenu existant ou fixe

L'impact de cette attaque est assez étroit puisque la mise en cache du contenu du CCN fournit une contre-mesure intégrée.

Si plusieurs attaquants des différentes voies génèrent un grand nombre d'intérêts pour attaquer un producteur. Après l'attaque initiale, Le contenu s'installe dans les caches de tous les routeurs intervenants. Les paquets d'intérêt ultérieurs pour le même contenu ne pourront plus se propager au producteur puisqu'ils sont satisfaits à partir des copies en cache des routeurs intermédiaires.

II-4-2 Contenu généré dynamiquement

Comme le contenu demandé est dynamique, tous les paquets d'intérêt sont acheminés vers le producteur de contenu, ce qui consomme de la bande passante et le PIT du routeur. Le résultat de cette attaque est que le producteur peut être surchargé d'intérêts malveillants et incapables de traiter les demandes d'autres consommateurs légitimes. Le routeur qui est le plus proche du producteur a le plus grand effet sur son PIT.

II-4-3 Contenu inexistant

Dans cette attaque, les attaquants génèrent des intérêts distincts et insatisfaisables pour un contenu non existant. Par conséquent, ces paquets d'intérêt sont dupliqués et propagés dans le réseau.

La duplication d'une telle quantité d'intérêts coûte beaucoup de ressources du nœud, ces derniers resteront dans le PIT pendant une période aussi longue que possible, ce qui épuisera certainement la mémoire et ressources informatiques sur les routeurs, dégradation des performances, ou même de les faire cracher.

Nous nous intéressons au troisième type des attaques d'inondation d'intérêts à savoir inonder le réseau avec de fausses demandes (de faux intérêts) en cherchant un contenu d'origine inexistant.

II-5 Description des attaques d'inondation d'intérêts

Une des structures principales des nœuds CCN : le PIT, en gardant la trace l'état de chaque flux de données, l'exploitation de cette structure se fera en envoyant des paquets d'intérêts par les utilisateurs finaux et la réponse à leurs demandes est représentée par des paquets de contenus (données) qui arrivent aux routeurs et finalement les utilisateurs demandeurs. Cependant, en tenant compte la vitesse d'accès au PIT et les types des mémoires physiques existants pour adapter ce module, il peut être considérer comme un point faible dans l'ensemble de l'infrastructure CCN avec sa taille mémoire limitée et son naïf fonctionnement dans l'environnement CCN.

Le problème réside dans l'utilisation intensive de la mémoire de PIT, c'est-à-dire augmenter le nombre de ses entrées et épuiser les ressources mémoire.

✓ Scénario de L'attaque

La figure 3 ci-dessous représente un réseau CCN ayant un fournisseur de contenu "CP", un attaquant "A", un utilisateur légitime "U", un router "R".

Tout intérêt avec un préfixe de nom de contenu "/data/Bank/" (inclus dans la table de recherche du FIB) est acheminé vers le CP par le CCN, l'attaquant "A" est capable de générer rapidement un grand nombre de faux intérêts en créant des noms de contenu basés sur le préfixe commun "/data/Bank/" et les envoyer. Par exemple : "/data/Bank/fake1.jpg", la partie "fake1.jpg" n'existe pas dans le CP et donc reste dans le PIT du router "R", puis chaque faux

intérêt est ajouté dans le PIT. En conséquence le PIT devient plein avec un maximum de faux intérêts, et "R" est incapable d'accepter et de traiter les intérêts de l'utilisateur légitime "U".

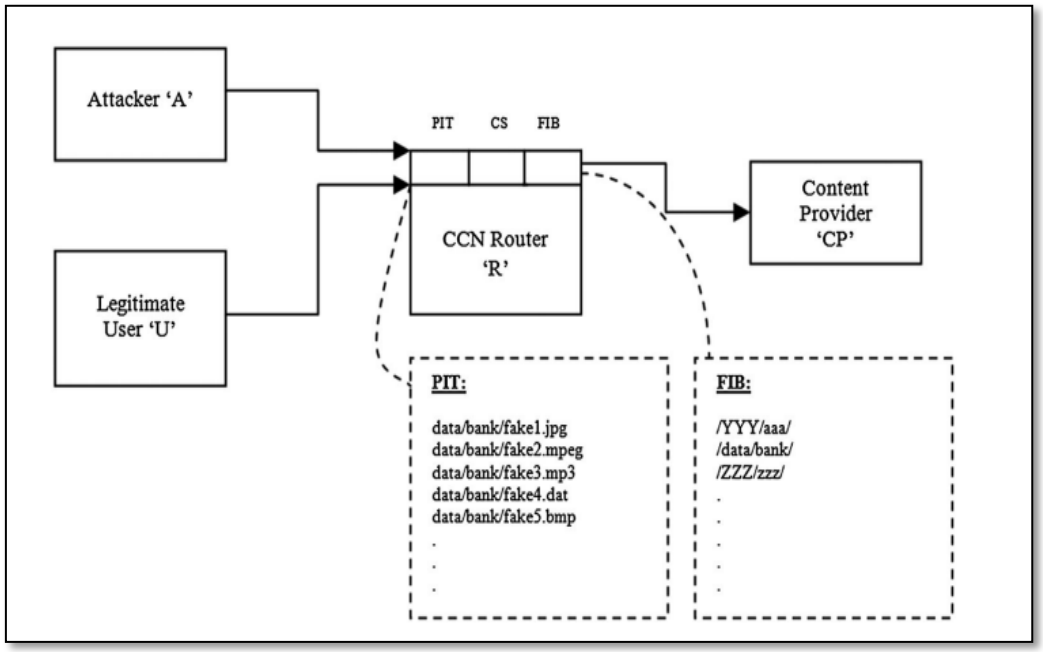

Figure II-4 Scenario d'attaque d'inondation d'intérêts [Muh 14]

II-6 Les Contre-mesures pour IFA

Pour se défendre contre ce type d'attaque d'inondations d'intérêts (Interest Flooding Attack IFA), il existe plusieurs algorithmes de contre mesure, selon la littérature, ils suivent tous l'une des deux approches :

II-6-1 Collection des statistiques par le routeur

Elle se base sur le processus de maintenir l'équilibre des flux entre les intérêts et les contenus. Les routeurs peuvent limiter le nombre total d'intérêts en attente pour un préfixe qui est attaqué et contrôle une ou plusieurs interfaces entrantes qui ont envoyé trop d'intérêts insatisfaits pour ce préfixe. **[Nar 16]**.

II-6-2 Les Mécanismes Push/Back

C'est le fait de remonter à la source de l'attaque et d'isoler l'attaque à la source **[Nar16]**, En outre, ce mécanisme permet aux nœuds d'interagir avec l'attaque dont le nœud décide lui-même en collaboration avec les autres nœuds comment réagir à l'attaque et quels sont les paramètres à prendre en considération et la nécessité de les rétablir ou modifier pour lutter contre cette attaque.

II-6-3 Problèmes liés à ces approches

Cependant, ces deux approches présentent quelques faiblesses contre les attaques d'inondation des intérêts, et requièrent des conditions d'implémentation spécifique.

1) Approche de collection des statistiques

Dans cette approche, les routeurs limitent le nombre d'intérêt entrants pour un interface ou plusieurs pour atténuer l'attaque. Cette méthode, peut enrichir les attaques d'inondation d'intérêt car un attaquant peut simplement atteindre la limite définit dans les interfaces en envoyant plusieurs requêtes malveillantes ce qui évoque l'interdiction de transmettre les requêtes légitimes.

2) Approche de Push/Back

Contrairement à l'approche précédente, push/back fournit un mécanisme plus ou moins intelligent basé sur la réaction avec l'attaquant tout en exécutant plusieurs algorithmes dans les routeurs et les faire collaborer entre eux en échangeant les informations nécessaires pour contrer l'attaque.

Le souci majeur de cette approche c'est le taux élevé de consommation des ressources physiques des nœuds et du réseau (la bande passante, l'utilisation CPU des routeurs, les caches…), ce qui produit un effet négatif sur la performance du réseau, et surtout pour les applications en temps réel qui nécessitent une transmission rapide des contenus (par exemple : vidéo Conferencing, streaming etc..).

Dans notre étude, on se basera sur l'approche de push/back car elle est plus efficace que l'approche des statistiques malgré ces limites, elle fournit des mécanismes de sécurité mieux développées et plus adaptés pour limiter les attaques d'inondation d'intérêts en tenant compte de leurs exigences matérielles.

II-7 Algorithmes de contre-attaque

A base de ces deux approches, les chercheurs ont proposé plusieurs algorithmes de contres attaques **[Ale 13]**. Nous présentons dans notre travail ceux qui sont les plus connus dans Les CCN.

II-7-1 Token Bucket Algorithme

Proposé par **[Ale 13]**, le Token Bucket est une technique qui a été utilisée dans les réseaux à commutation de paquets comme les réseaux étendus (réseaux IP, Frame Relay ...), elle se base

sur l'approche de collection des statistiques par le routeur. Son principe est simple, définir un taux maximal de consommation de la bande passante du réseau et vérifier que ce taux ne sera pas dépassé, pour éviter la congestion du réseau. De même, dans les CCN pour éviter les attaques d'inondation des intérêts, il faut fixer une limite des intérêts envoyés par les utilisateurs dans les routeurs, dès que cette limite est atteinte aucune autre demande n'est acceptée.

Le calcul de cette limite se fait en prenant compte de la capacité de l'interface du routeur, la bande passante et le temps de réponse moyenne pour chaque requête. La figure ci-dessous (II-5) montre une formule mathématique pour calculer la limite, ou Le Delay est le temps nécessaire pour satisfaire un intérêt en secondes, Bandwidth c'est la capacité de l'interface en Octets par secondes et la taille du paquet de donnée en Octets.

$$\text{Interest Limit} = Delay\ [s] \cdot \frac{\text{Bandwidth [Bytes/s]}}{\text{Data packet size [Bytes]}}$$

Figure II-5 Formule Mathématique pour calculer la limite des intérêts [Ale 13]

Cependant, cet algorithme peut nourrir les attaques d'inondation en lançant plusieurs demandes par un attaquant dans le but est d'atteindre la limite du routeur et par conséquent les demandes d'intérêt légitime ne peuvent plus être transmit et selon **[Ale 13]** cette approche a été proposée sans faires des tests.

II-7-2 Token Bucket with per interface fairness

Cet Algorithme représente une amélioration du naïf Token Bucket, ou Les intérêts acheminés par un routeur sur chaque interface représentent une combinaison équilibrée des intérêts reçus des nœuds voisins. Afin d'assurer un partage équitable des intérêts de tous les nœuds voisins, les chercheurs ont étendu le tableau PIT pour soutenir le signalement des intérêts qui ne peuvent pas être transmis immédiatement et mettons en œuvre des files d'attente hiérarchiques pour chaque interface.

Ce mécanisme est fondé essentiellement sur une file d'attente basée sur les classes pour chaque interface sortante et entrante. Nous constatons que, contrairement aux files d'attente normales, les files d'attente d'intérêt ne stockent pas réellement un paquet, mais simplement un pointeur bidirectionnel vers l'entrée PIT existante. Ainsi, une entrée PIT peut

être rapidement mise à jour lorsque l'intérêt est effectivement transmis, et l'élément peut être facilement retiré de la file d'attente lorsque l'intérêt expire.

Il est également important de définir des tailles de file d'attente appropriées et de fixer une valeur raisonnable pour la durée pendant laquelle un intérêt peut être mis en file d'attente.

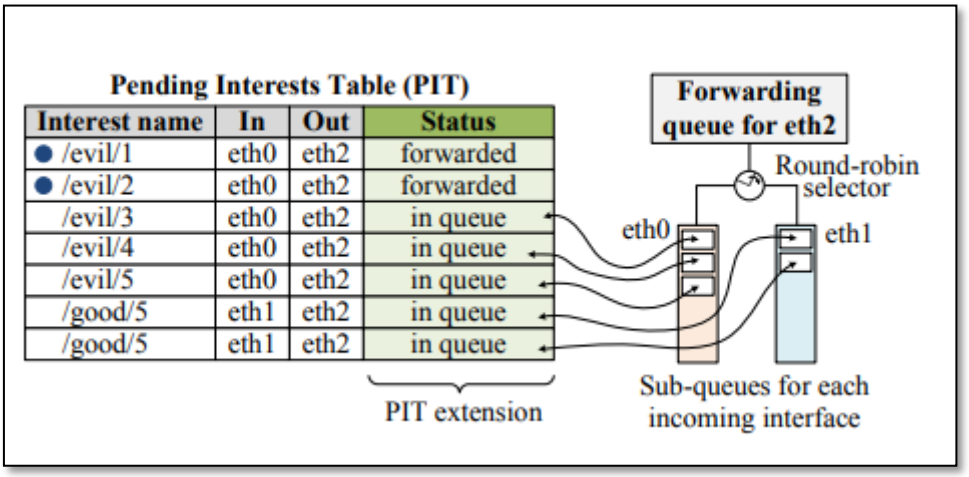

Figure II-6 Mise en file d'attente des intérêts [Ale 13]

II-7-3 Satisfaction-based Interest acceptance

Après avoir mise en œuvre la technique de collection statistique sur le taux de satisfaction d'intérêt, le prochain défi consiste à utiliser cette mesure pour bloque les intérêts malveillants. Cette simple méthode permet d'utiliser le taux de satisfaction comme une probabilité de transmission ou de rejet.

Ce modèle probabiliste permet de garantir que la surcharge d'intérêts dans une interface particulier ne pas vraiment faible et fournir une occasion pour l'utilisateur légitime de recevoir le contenue demandée ,l'inconvénient de cette méthode est que chaque routeur prendre la décision de transmettre ou de rejeté le contenue de façon indépendante (décentralisé) c'est à dire que chaque routeur de réseau CCN a une probabilité d'acceptation d'intérêt diffèrent a d'autre routeur ce qui influence de façon directe sur l'utilisateur légitime si le nombre de saute est plus grand (croissance de réseau) .une façon d'éviter cette réaction excessive est l'échanges des notification explicité tel qu'un protocole entre les voisins pourrait atténuer le problème .

II-7-4 Satisfaction-based push back

Cet algorithme consiste à permettre faire d'applique une limite d'intérêt explicite (distribution du jeton) pour chaque taux de satisfaction d'intérêt d'une interface sans réaction excessive, car avant de déclenche cette opération les routeurs doivent annoncer ces limites à leurs voisins au début, ce qui donne de véritable statistique sur la satisfaction des intérêts.

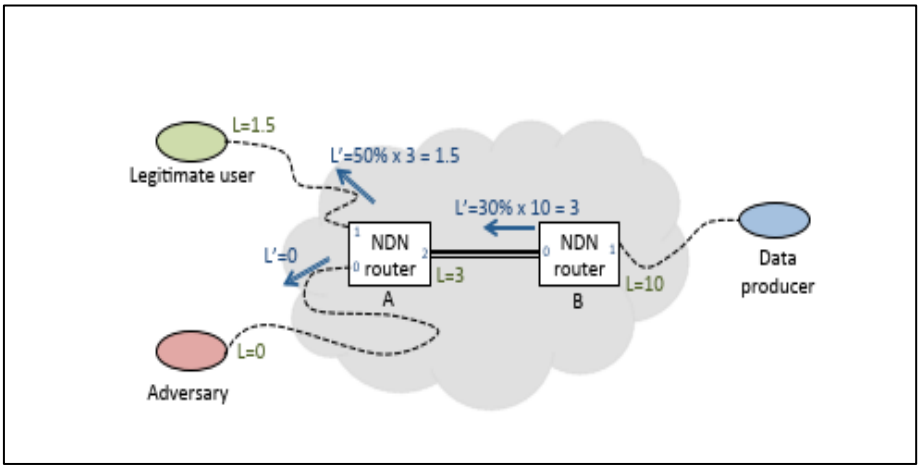

Figure II-7 Satisfaction-based push back Example [Ale 13]

La Figure II-7 illustre comment l'algorithme fonctionnera, en supposant que la limite initiale du jeton L = 10 et que le taux de satisfaction pour le routeur A est de 50% pour eth1 et de 0% pour eth0, et pour le routeur B, le rapport est de 30 % pour eth0, chaque nœud fixera et annoncera la limite d'interface entrante suivante L′ :

1) le routeur B établira et annoncera l'interface entrante limite L′ = 3 ;

2) le routeur A, après avoir reçu l'annonce de B réajuster ses limites d'interface entrante à L′eth1 = 1,5 et L′eth0 = 0 ;

3) et les utilisateurs légitimes et les adversaires peuvent soit respecter, soit ignorer la limite annoncée, qui sera de toute façon appliqué par le routeur A.

La limite zéro pour le lien de l'adversaire implique que le routeur A n'est temporairement pas disposé à accepter des intérêts de cette interface. Lors de la prochaine itération de l'algorithme de push back sur la satisfaction, un utilisateur légitime sera en mesure d'améliorer progressivement les statistiques sur les deux routeurs A et B à mesure que tous les intérêts de l'utilisateur passeront et retourneront des données, ce qui conduit finalement à une autorisation complète (L′ = L = 10) dans les liaisons entre les routeurs A et B, et l'utilisateur et

le routeur A. Satisfaction-based Interest push-back est un algorithme de mécanisme push-back qui permet d'atténue les attaques qui peuvent être déployés à tout moment sans dégrader la performances du réseau, même en l'absence d'attaquants actifs.

II-7-5 Poséidon

C'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'intérêts) et en atténuer les effets. Poséidon surveille en permanence les taux par interface des intérêts par rapport au trafic global. Si ces taux changent de manière significative entre deux intervalles de temps consécutifs, il réduit le débit sur la ou les interfaces incriminées (ce qui réduit le nombre des intérêts entrants) ou les bloquent carrément. En outre, Poséidon utilise mécanisme push/back qui peut émettre un message d'alerte aux mêmes interfaces, pour signaler qu'une inondation d'intérêts est en cours.

Il est composé de deux phases, une phase de détection et une phase de réaction. Pour introduire ces deux phases, on doit faire recours à des notations importantes qui sont utilisées dans le cœur de cet algorithme. Le tableau ci-dessous montre ces notations et ses significations :

Tableau II-1 Définition des notations de Poséidon

R	Ensemble de tous les routeurs du réseau exécutant Poséidon
r_i	i-ème routeur, $1 \leq i \leq \|R\|$
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 mesurée à la fin de l'intervalle t_k
$\Omega(r_i^j)$	Seuil de détection d'inondation d'intérêt pour $\omega\ (r_i^j, t_k)$
$P(r_i^j)$	Seuil de détection d'inondation d'intérêt pour $\rho(r_i^j, t_k)$

1) Phase de détection

Les attaques sont détectées à l'aide de deux paramètres : $\omega(r_i^j, t_k)$ et $\rho(r_i^j, t)$, Le premier représente le nombre des intérêts divisés par le nombre de paquets de contenus sortants, observé par un routeur r_i sur son interface r_i^j dans l'intervalle de temps t_k **[COM 13]**:

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

P (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 les deux valeurs $\omega\ (r_i^j, t_k)$ et $\rho\ (r_i^j, t_k)$ excède leurs seuils respectivement $\Omega(r_i^j)$ et $P(r_i^j)$. Cet algorithme de détection est exécuté dans des intervalles de temps bien fixés et en réagissant à des évènements spécifiques. Le paramètre $\omega\ (r_i^j, t_k)$ est une bonne image de la capacité des routeurs à répondre aux intérêts entrants dans un intervalle de temps donné. En général, quand $\omega\ (r_i^j, t_k) > 1$ que le nombre de paquets de contenu envoyés est inférieur que le nombre d'intérêts provenant de la même interface. C'est un facteur important dans la détection de l'attaque mais une petite explosion (régulière ou non) des intérêts ne peuvent pas être causés par une attaque. Par conséquent, la prise en compte de $\omega\ (r_i^j, t_k)$ uniquement peut amener l'algorithme de détection à signaler un grand nombre de faux positifs. L'application de contre-mesures peut, dans ce cas, produire des effets négatifs sur les performances globales du réseau.

Pour améliorer la précision de la détection (en distinguant les explosions d'intérêts naturelles des attaques), Poséidon prend en compte également $\rho(r_i^j,t_k)$. Cette valeur mesure l'espace PIT utilisé par les intérêts provenant d'une interface particulière, ce qui lui permet de maintenir le nombre de faux positifs à un faible niveau par rapport à la seule considération de $\omega\ (r_i^j, t_k)$ tout en lui permettant de détecter les inondations d'intérêts à faible taux. La surveillance du contenu du PIT permet à Poséidon d'observer les effets de l'attaque, plutôt que ses seules causes, ce qui permet une détection anticipée et rapide. Les deux mesures $\omega(r_i^j, t_k)$ et $\rho(r_i^j,t_k)$ ils se complètent l'un l'autre, 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 fera pas, lorsque le routeur reçoit une courte série d'intérêts, $\omega(r_i^j, t_k)$ peut devenir plus grand que $\Omega(r_i^j)$ mais l'utilisation du PIT sera probablement dans les valeurs normales ($P(r_i^j)$ dans ses limites).

2) Phase de réaction

Une fois qu'une attaque d'inondation d'intérêts depuis l'interface r_i^j du routeur r_i a été identifiée, Poséidon limite le taux d'intérêts entrants depuis cette interface. Le taux initial est rétabli lorsque tous les paramètres de détection tombent à nouveau en dessous de leurs seuils correspondants **[COM 13**].

Grâce aux contre-mesures collaboratives, une fois qu'un routeur détecte un trafic anormal à partir d'un ensemble d'interfaces, il limite leur taux 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, utilisé pour transmettre des informations sur l'attaque d'inondation d'intérêt en cours. Il y a deux raisons d'utiliser des paquets de contenu plutôt que des intérêts pour transmettre des informations de type "push-back" :

- lors d'une attaque, le PIT du prochain saut connecté à l'interface en infraction peut être plein, et donc le message d'alerte peut être rejeté.
- les paquets de contenu sont signés, alors que les intérêts ne le sont pas. Cela permet aux routeurs de déterminer si un message d'alerte est légitime.

Un paquet d'alerte contient : l'horodatage correspondant à l'heure de génération de l'alerte, le nouveau taux (réduit) auquel les intérêts malveillants seront acceptés sur l'interface entrante, et des informations détaillées sur l'attaque, telles que le ou les espaces de noms utilisés pour les intérêts malveillants. Une attaque par inondation d'intérêt persistante sur un routeur provoque l'envoi de multiples messages d'alerte vers la ou les sources de l'attaque. Ces sources diminueront leurs seuils $\Omega(r_i^j)$ et $P(r_i^j)$ jusqu'à ce qu'elles détectent l'attaque et mettent en œuvre une limitation de l'intérêt malveillant. Si aucune attaque n'est signalée pendant une période de temps prédéfinie, les seuils sont rétablis à leurs valeurs initiales.

Ce mécanisme de push/back permet aux routeurs qui ne sont pas la cible de l'attaque, mais qui transmettent involontairement des intérêts malveillants, de détecter rapidement l'inondation d'intérêts. En particulier, les messages d'alerte permettent aux routeurs de détecter une inondation d'intérêts même lorsqu'ils sont loin de la victime visée - c'est-à-dire près des nœuds contrôlés par l'adversaire, où les contre-mesures sont plus efficaces

II-7-6 Interest Traceback

L'algorithme Traceback **[DAI 13]** il est conçu pour libérer les entrées PIT indésirables lorsque la quantité d'espace mémoire disponible atteint un seuil prédéfini, La phase de détection est assez simple et ne nécessite que de surveiller l'utilisation de la mémoire PIT dans

le temps.

Après avoir détecté une occupation de mémoire anormale, le processus de trace est déclenché et un ensemble de paquets de données usurpés est généré pour les entrées qui sont restées longtemps insatisfaites. Les paquets de données usurpés portent le nom nécessaire pour satisfaire les intérêts fautifs et sont transmis en aval pour libérer des ressources tout au long du chemin. Afin d'exploiter au maximum l'espace mémoire disponible pour le PIT, il faut défini le seuil pour le Traceback soit activé, à 90% de la mémoire occupée. Une telle limite agressive évite une réaction excessive de l'algorithme et permet au réseau de prendre en charge un pic de trafic temporaire sans déclencher de mécanisme de blocage des intérêts.

Les développeurs ont conçu un code **[MAT 15]** pour qu'il se rapproche le plus possible de la description de la contre-mesure. En particulier, ils simulent pour chaque routeur un processus de surveillance qui est planifié toutes les secondes pour vérifier si l'occupation de la mémoire dépasse sa valeur alarmante. Si (et seulement si) il dépasse notre seuil (plus de 90% d'occupation de mémoire), nous invoquons la fonction **FindAndSend ()** pour générer des paquets de données usurpés et les faire voyager vers l'initiateur de l'attaque. Une vision simplifiée de haut niveau d'implémentation **[MAT 15]** qui est présente par le code suivant :

```
Void Traceback:: FindAndSend ()
{
   FOR EACH Entry in Pit
      IF IsOld (Entry)
         FOR EACH Face in Entry.FacesList
            IF Face.IsConnectedToEndUser()
               BLOCK Face
            ELSE
               GENERATE 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 prédéfini.
- 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.

II-7-7 Comparaison des contre-mesures :

Nous récapitulons les différents critères de comparaison pour les algorithmes de contre mesure IFA dans le tableau suivant (II-2) :

Tableau II-2 Comparaison des algorithmes avec différents critère

Algorithme / Critère	Traceback	Satisfaction Based Acceptance	Satisfaction Based Push/Back	Token Bucket with Interface Fairness	Poseidon
Vitesse d'exécution	Lent	Rapide	Rapide	Rapide	Moyen
Précision	76%	Max 87,5%	Max 90%	Max 23%	84%
Taux d'erreur	24%	Max 12,5%	Max 10 %	Max 77%	16 %
Paquetage de programmation	CCN lite	NdnSim	NdnSim	NdnSim	CCN lite
Possibilité de bloquer un utilisateur légitime (faux positif)	Oui	Oui	Oui	Oui	Oui
Possibilité de ne pas bloquer un utilisateur malveillant (faux négatif)	Oui	Oui	Oui	Oui	Oui
Type d'approche de contre mesure	Basé sur les statistiques du routeur	Basé sur les statistiques du routeur	Push/Back	Basé sur les statistiques du routeur	Push/Back
Collaboration entre les routeurs	Non	Non	Oui	Non	Oui

Possibilité de d'installer dans tous les routeurs	Oui	Oui	Oui	Oui	Oui

La comparaison a été faite entre les contre-mesures existantes dans la littérature par rapport au diffèrent critères, cela nous donne un aperçu des avantages de chaque algorithme. Nous constatons que les contre-mesures selon l'approche Push/Back sont meilleurs que les autres solutions, avec une précision dépassant 76% (pour traceback) jusqu'à 90% (pour Satisfaction Based Push/Back) contre une précision variant entre 23% et 87% pour les algorithmes basés sur les statistiques des routeurs, ce qui nous laisse choisir les solutions de la première approche, sur avec l'algorithme Satisfaction Based Push/Back qui a eu le meilleur taux de précision avec la plus petite marge d'erreur.

Cependant, suite aux travaux déjà réalisés au sein du projet dont lequel s'inscrit le notre, nous allons choisir entre les deux algorithmes basés sur Push/Back à savoir Poséidon et Trace Back dans l'optique d'une meilleure amélioration de leurs performance et diminution du taux d'erreurs (surtout en termes de faux positifs).

Afin de choisir entre Traceback et Poséidon, nous étudierons les inconvénients de l'un et de l'autre dans le tableau suivant (II-3) :

Tableau II-3 Les inconvénients des deux contre mesure TraceBack et Poséidon

Inconvénients Traceback	Inconvénients Poséidon
• Risque de libérer les intérêts satisfaits dans la table PIT • Possibilité de bloquer l'interface de façon définitive • Possibilité de bloquer un client légitime (faux positive) • Possibilité de ne pas bloquer un attaquant (faux négatif) • Pas de collaboration entre les routeurs • La consommation exhaustive de l'bande passante du réseau en envoyant les paquets falsifiés • Assure une défense plus agressive	• Possibilité de bloquer un client légitime (faux positive) • Pas de mise à jour dans la table PIT • Possibilité de ne pas bloquer un attaquant (faux négatif)

D'après l'analyse des différentes points négatifs de ces deux algorithmes, nous déduisons que l'algorithme **Traceback** prend un comportement très agressif qui permet de dégrader les performances du réseau en bloquant le passage des paquets envoyée par les client légitime de manière définitif par rapport au Poséidon dont nous allons l'améliorer par la suite.

II-8 Conclusion

Les attaques Zombies ont été toujours un grand problème dans la sécurité informatique car ils sont simples à lancer et difficiles à les contrer. La nouvelle architecture centrée contenu a malheureusement héritée ce genre d'attaque des anciennes architectures. Cette attaque est nommée attaque d'inondation d'intérêts ou l'adversaire vise à saturer la mémoire PIT des nœuds CCN dont le but de déstabiliser le processus de routage centré contenu, plusieurs algorithmes de contre-mesure existent toujours est-t-il les limites de ces algorithmes (tel (précision, taux d'erreur, faux positifs, faux négatifs …) peuvent renforcer les attaques d'inondations.

Donc il faut limiter ces vulnérabilités en renforçant ces contre-mesures ce qui va être l'objet du prochain chapitre ou nous allons proposer un mécanisme TTL inspiré du réseau IP dans l'optique d'améliorer l'algorithme de Poséidon en limitant le nombre de faux positifs et faux négatifs produites par l'algorithme initial.

CHAPITRE III.

CONTRE-MESURE

«POSÉIDON_TTL»

CHAPITRE III. CONTRE-MESURE « POSEIDON_TTL »

III-1 Introduction

Les attaques de Déni de Services (DoS) ont été toujours une menace majeure dans les réseaux informatiques, que ce soit avec ses architectures et protocoles actuels (modèle TCP/IP, IPV4...), ou dans les futures architectures comme les réseaux CCN (l'inondation des intérêts). Ces attaques visent à perturber le processus du routage dans les CCN en envoyant un grand nombre d'intérêts au routeur dont le but de le saturer pour ne plus transmettre les requêtes des utilisateurs.

Pour lutter contre ces attaques, les chercheurs ont proposé un ensemble d'algorithmes de contre-mesures fonctionnant selon deux approches, celles basées sur les statistiques, et celles basées sur la contre-attaque (push/back), ils offrent tous une bonne défense contre ces attaques.

Cependant, ces algorithmes comportent toujours quelques limites pouvant dégrader leurs performances et les performances du réseaux CCN par conséquent, citons à titre d'exemple, le taux d'erreurs assez significatif, en effet il y a une grande chance que ces algorithmes bloquent des utilisateurs légitimes non seulement en bloquant un faux positif (un utilisateur pris pour un attaquant), mais le plus souvent c'est le fait de bloquer toute une interface du routeur, cette dernière peut être reliée à tout un sous-réseau avec un nombre important de nœuds. Comme ils peuvent omettre quelques utilisateurs malveillants et les considérer comme légitimes (faux négatifs), ces cas paraissent dans la plupart des cas quand il s'agit d'attaques de zombies où le routeur est submergé sans pouvoir détecter la source de l'attaque (plusieurs requêtes/intérêts non valides arrivant de partout). On peut dire que ces faiblesses sont dues à un manque de collaboration efficace entre les différents routeurs dans les CCN, à un taux élevé de consommation des ressources, tout nourrit de plus en plus les attaques d'inondations. Il est inévitable alors de trouver, et/ou proposer des solutions pour

diminuer faiblesses de ces algorithmes, en les rendant plus efficaces et plus performants afin d'améliorer le routage centré contenu.

C'est dans cette optique que nous réalisons ce chapitre, en effet, afin de créer une liaison plus collaborative entre les différents routeurs du réseau CCN sans pour autant affecter négativement ses performances, nous proposons un protocole de calcul du nombre de saut des paquets CCN semblable au TTL des réseaux IP. Une fois le protocole implémenté dans le CCN, nous l'incorporons à l'intérieur de l'algorithme de contre mesure « Poséidon » que nous appelons « Poséidon_TTL ». Cette proposition est effectuée dans le but d'éliminer le problème de faux positif, et contrer aux attaques de Zombies dans les CCN.

III-2 Notre contribution

Dans cette première partie de réalisation de notre travail, nous proposons un protocole de sécurité dans les réseaux CCN basé sur le principe du mécanisme TTL (Time To Live) du réseau conventionnel IP (Internet Protocol).

Le TTL indique le nombre de sauts (également le nombre de routeurs) qu'un paquet IP ou intérêt (dans CCN) peut avoir en traversant les routeurs intermédiaires entre la source et la destination. De ce fait, nous supposons dans notre protocole qu'à chaque fois que le nombre de saut est grand, on aura plus d'utilisateurs à l'extrémité ; C'est le principe d'une arborescence à plusieurs branches, la branche la plus longue aura probablement plus de nœuds. Cette métrique du nombre de sauts permet d'estimer le nombre d'utilisateurs pour chaque interface du routeur CCN.

Par ailleurs, le principe des algorithmes classiques de contre mesure de Flooding est de bloquer chaque interface du routeur ayant un nombre d'intérêts supérieur à un seuil « s » indépendamment du nombre d'utilisateurs finaux venant de cette interface. Notons ici que le seuil est calculé en moyennant le nombre d'intérêts des autres interfaces sans prendre en compte le nombre de nœuds venant par la même interface.

Pour implémenter notre protocole, nous allons tout d'abord implémenter le principe du TTL dans le réseau CCN afin de calculer le nombre de saut. Par la suite, nous intégrons cette métrique à l'algorithme de contre mesure Poséidon dans le but de réduire le taux d'erreur de cet algorithme en minimisant le nombre de faux positifs.

III-3 TTL et CCN

Le temps de vie (TTL) fait référence au nombre de saut pendant lequel un paquet est censé exister dans un réseau avant d'être rejeté par un routeur, Lorsqu'un paquet d'informations est créé et envoyé sur le réseau, il existe un risque qu'il continue de passer indéfiniment de routeur en routeur. Pour atténuer cette possibilité, les paquets sont conçus avec une expiration appelée durée de vie ou limite de saut (TTL) qui indique à un routeur réseau si le paquet a été dans le réseau trop longtemps et doit être rejeté.

Dans ce qui suit nous expliquons en détail le mécanisme du TTL dans les réseaux IP, avant de proposer notre propre TTL_CCN. Notre protocole sera mis en œuvre et tester sur une petite topologie du réseau CCN vu les moyens matériels disponibles.

Notons que, dans ce travail, le TTL_CCN a été proposé dans l'objectif d'estimer, approximativement, la longueur d'une branche (un sous-réseau) à partir d'une interface d'un routeur dans le réseau CCN pour des fins d'amélioration des mécanismes de sécurité. Cependant le protocole TTL_CCN peut faire office d'autres tâches comme celle du TTL dans les réseaux IP.

III-3-1 Le mécanisme TTL dans les réseaux IP

Dans un paquet IP, Le TTL est initialement défini par le système d'exploitation **[GOTO 12],** Il peut être réglé sur n'importe quelle valeur comprise entre 1 et 255 qui définissent des valeurs par défaut différentes, lorsque des données doivent être véhiculées sur un réseau, les machines les encapsulent dans un paquet muni d'un en-tête comme illustrer dans la figure III-1, il comporte toutes les informations nécessaires au routage des données.

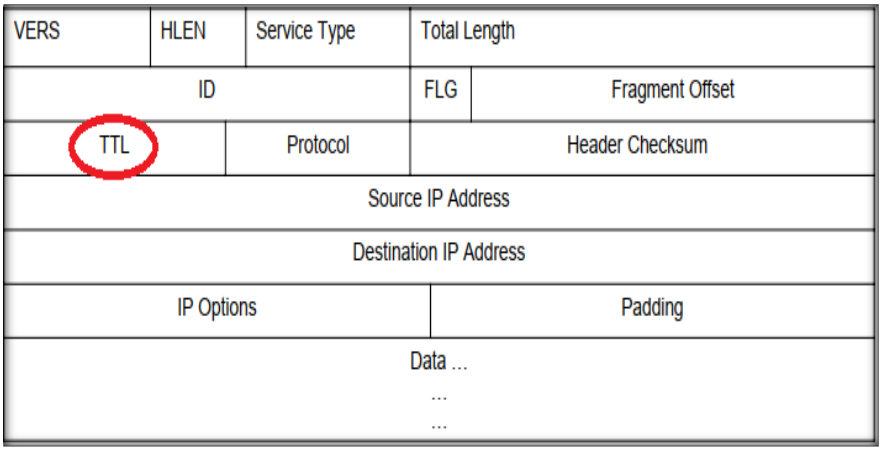

Figure III-1 en-tête d'un paquet IPV4 [IBM 06]

Le champ TTL peut également être utile pour déterminer la durée de circulation d'un paquet,

il permet à l'expéditeur de recevoir des informations sur le chemin d'un paquet via le réseau, son but est de garder les flux de paquets non livrables coincés dans les boucles de routage de circuler indéfiniment et de boucher les réseaux en question.

Chaque paquet a un endroit où il stocke une valeur numérique déterminant combien de temps il doit continuer à se déplacer sur le réseau. Chaque routeur qui reçoit le paquet soustrait un du comptage TTL ; Si le nombre reste supérieur à zéro, Le routeur transfère le paquet, sinon il le rejette et renvoie un message ICMP (Internet Control Message Protocol) à l'hôte d'origine, Ce qui peut déclencher un renvoi (Figure III-2).

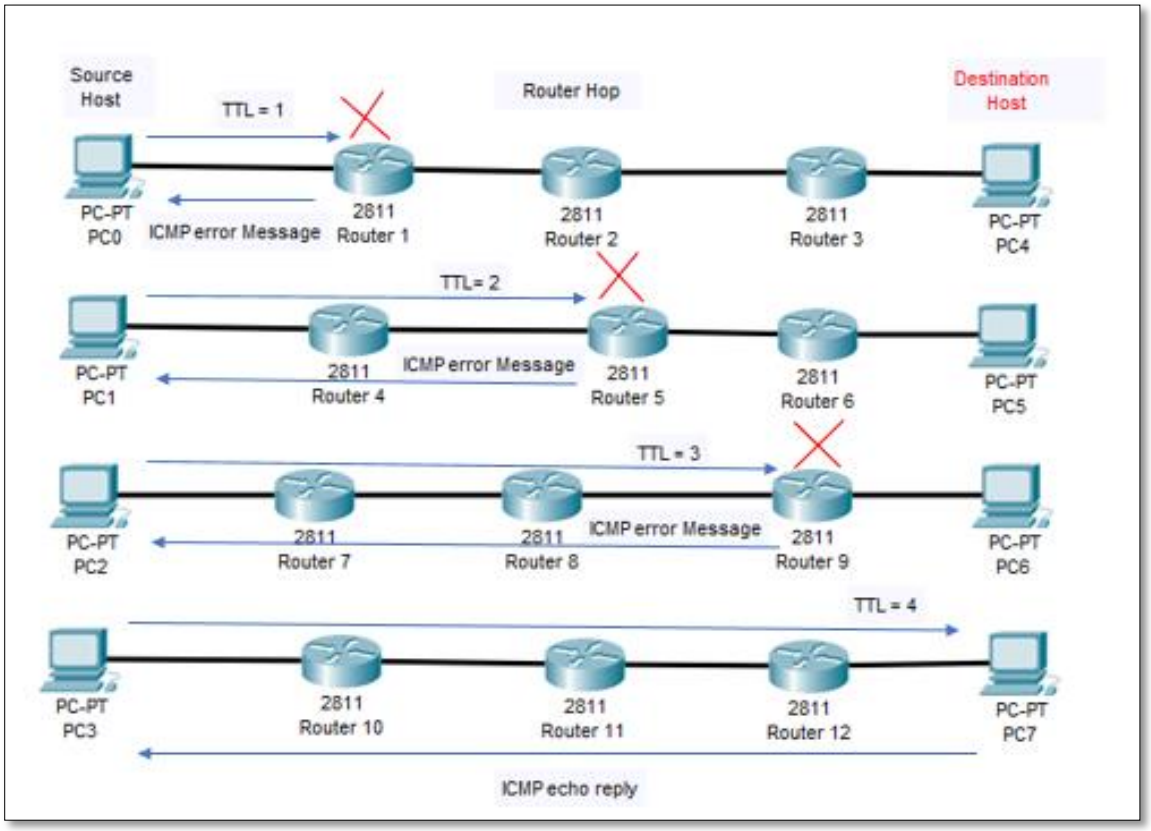

Figure III-2 Déroulement du mécanisme TTL dans Réseau IP

III-3-2 Mise en place du TTL dans les CCN

Le mécanisme du TTL est donc très important dans les réseaux IP, pour éliminer les boucles de transmissions dans le réseau et même peut être utilisé pour la détection des attaques de déni de services **[GOTO 12]**.

Dans notre travail nous allons nous inspirer du TTL des réseaux IP, et proposer un protocole similaire dans les CCN afin de l'exploiter dans l'algorithme de Poséidon. Le TTL dans les CCN sera diffèrent au TTL du réseau IP, il s'agit d'une valeur affectée à l'intérêt, elle sera

initialisée par le demandeur du contenu à zéro et incrémentée à chaque passage par un routeur, contrairement au réseau IP où cette valeur générée par le SE de l'hôte émetteur, est décrémentée à chaque saut.

Pour pouvoir mette en œuvre le TTL dans les réseaux CCN, nous allons modifier l'entête du paquet d'intérêt CCN en ajoutant un nouveau champ nommée le Time To Live (TTL). Ce nouveau champ va être initialisé à zéro par le demandeur de contenu avant de le transmettre au nœud du prochain saut, et sera incrémenté chaque fois que l'intérêt traverse un nœud CCN jusqu'à arriver au fournisseur de contenu demandé.

En outre, Pour le nombre de bits que le TTL peut occuper, on s'est inspiré de **[GOTO 12]** qui considère qu'un paquet IP généralement prend un maximum de 30 sauts pour arriver à la bonne destination. De même le paquet intérêt ne pourra pas prendre plus que 30 routeurs sans arriver à le satisfaire surtout avec le mécanisme de cache implémenté dans les routeurs CCN où un routeur CCN lui-même peut satisfaire un intérêt. En conséquence, nous estimerons que 5 bits de plus dans l'entête de l'intérêt CCN est assez bonne pour ce TTL. La Figure III-2 montre le nouvel entête d'un intérêt CCN après avoir ajouté le TTL :

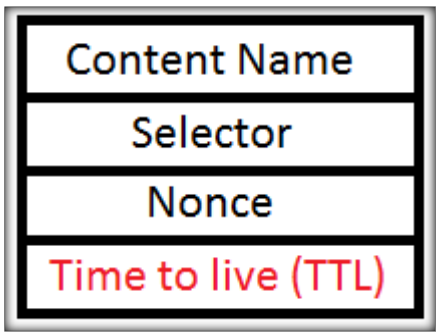

Figure III-3 Le nouveau paquet intérêt avec le TTL [FAB 13]

III-3-3 Comparaison entre le TTL TTL-CCN

Nous avons proposé un mécanisme TTL pour les réseaux CCN (TTL-CCN) en s'inspirant du réseau informatique TCP/IP. Cependant, les spécificités du réseau centré contenu nous a imposer des choix différents que le TTL classique, par exemple la taille d'intérêt qui doit être minimale par rapport à la taille d'un paquet IP.

Les points communs et les différences entre les deux mécanismes TTL sont présentés dans le tableau suivant (Tableau III-1) :

Tableau III-1 Les points communs et les différences entre les deux mécanisme TTL_IP et TTL_CCN

Les points communs	Les différences
• La valeur du TTL est représentée dans un champ de l'entête des paquets CCN et IP • Les deux valeurs TTL sont générées par le nœud émetteur • Les deux valeurs TTL prennent des valeurs entières	• La valeur de TTL d'un réseau CCN est initialisée à zéro, alors que celle du réseau IP est générée par le système d'exploitation du nœud émetteur. • La valeur de TTL est incrémentée à chaque saut d'un routeur CCN, et elle est décrémentée à chaque passage de routeur IP. • L'objectif principal du TTL_CCN est d'améliorer le niveau de sécurité avec les algorithmes de contre mesure des attaques d'inondation des intérêts, par contre, l'objectif principal du IP est d'éliminer les boucles de retransmissions des paquets entre les routeurs.

Malgré les similarités des deux TTL, nous avons besoin de modifier un peu le principe de TTL classique (TTL du réseau IP) afin de faciliter son intégration dans les réseaux CCN.

III-3-5 Implémentation technique du TTL dans le CCN

Comme un routeur IP, le routeur CCN est plus puissant et possède plus de fonctions (cache, résolution des noms …), et puisque le CCN est toujours laboratoire, il donc est capable d'aborder des nouvelles améliorations pour un routage efficace et sécurité. Parmi ces changements nous avons introduit le mécanisme de TTL inspiré du réseau IP.

Après avoir proposé une nouvelle représentation de l'intérêt dans la section précédente, nous passons maintenant à sa mise en place dans le réseau CCN. Afin d'expliquer le mécanisme sur le réseau CCN, nous illustrons dans la figure III-4 un exemple d'un intérêt émis par le client 1 nommé « /b3c/wowmom/movie15/c0 » avec TTL=0, l'intérêt parcoure tout le réseau,

nous constatons qu'au troisième routeur, TTL a la valeur 3 ce qui confirme que

l'incrémentation se fait convenablement.

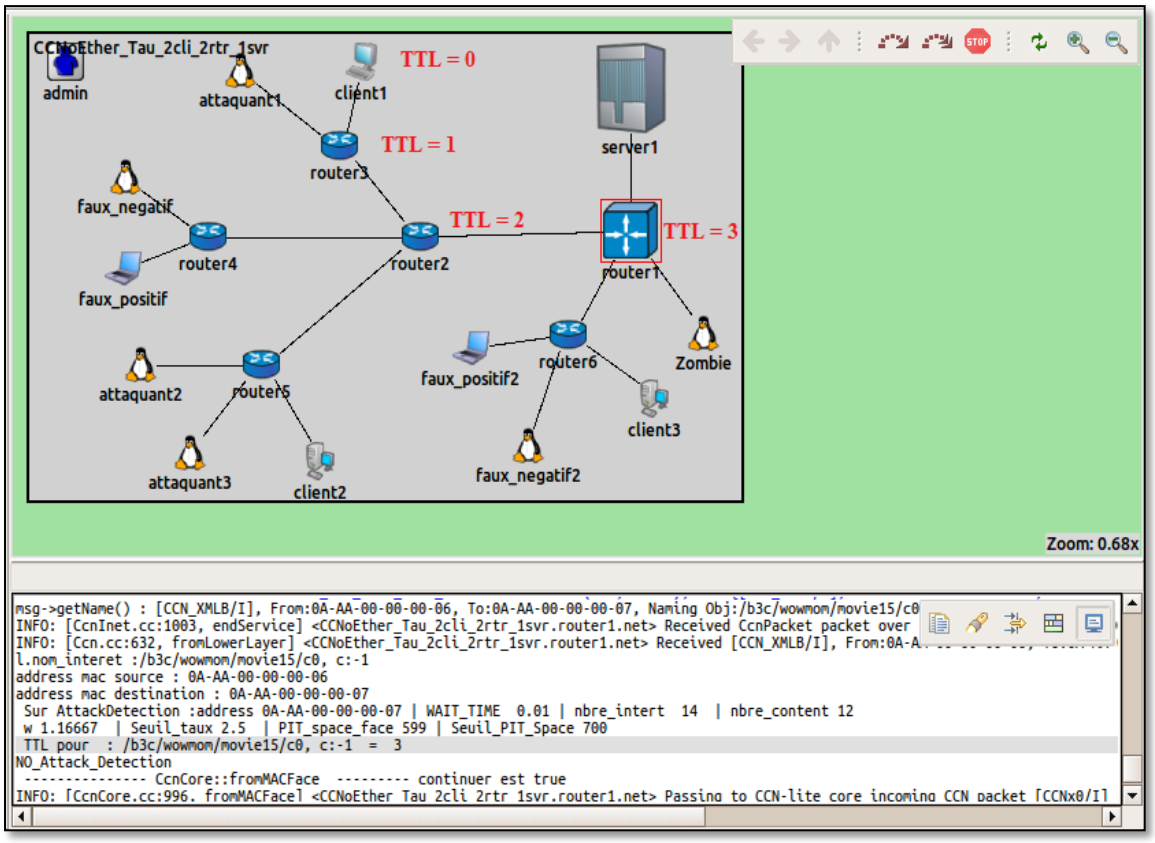

Figure III-4 Aperçu de TTL d'un intérêt dans notre implémentation

III-4 Intégration de TTL dans l'algorithme Poséidon

Nous avons choisi l'algorithme de Poséidon comme étant un algorithme de contre mesure des attaques d'inondation d'intérêts basé sur une approche push/back. Nous intégrons la métrique de sauts TTL et testons notre nouvel algorithme nommé « Poséidon _TTL » dans un environnement CCN.

III-4-1 Le choix de Poséidon

Ce choix a été effectué car Poséidon offre une protection progressive contre les attaques d'inondation d'intérêts, et possède moins de points négatifs que Traceback comme vu dans le chapitre précédent.

Contrairement à Traceback, l'algorithme de Poséidon peut bloquer une interface qui n'est pas

lié directement à un utilisateur final, c'est-à-dire une interface d'un ou plusieurs routeurs intermédiaires qui se situent entre le producteur et le consommateur de l'intérêt, ce qui, en conséquence, peut amener à un grand nombre de faux positifs générés tout en dépendant de la taille et la structure du réseau CCN : « Plus le nombre de saut est grand plus la possibilité de bloquer des clients légitimes augmente ».

III-4-2 Poséidon_TTL

Dans notre étude, nous proposons l'algorithme « Poséidon _TTL » qui est une amélioration du fameux algorithme de Poséidon afin d'atténuer le problème de faux positifs, ceci nous a été possible en ajoutant le concept du nombre de saut (le TTL) dans le CCN, et intégrant comme métrique avec les paramètres d'exécution de Poséidon, l'algorithme vérifie le TTL de l'intérêt reçu et compare cette valeur avec un seuil TTL prédéfini dans l'algorithme (ce seuil une valeur référentielle définie par l'administrateur réseau, ce dernier se base sur la hiérarchie de la topologie pour le fixer),

Si la valeur TTL de l'intérêt reçu est supérieure ou égale au seuil prédéfini, l'algorithme ne bloque l'interface même si le seuil de détection d'une attaque est atteint,

Sinon, si la valeur du TTL est inférieure au seuil TTL prédéfini, l'algorithme s'exécute normalement, donc peut bloquer l'interface de l'intérêt reçu si le seuil de détection d'attaque est atteint.

En ce qui suit, l'organigramme du déroulement de notre algorithme Poséidon_TTL tel qu'expliqué précédemment

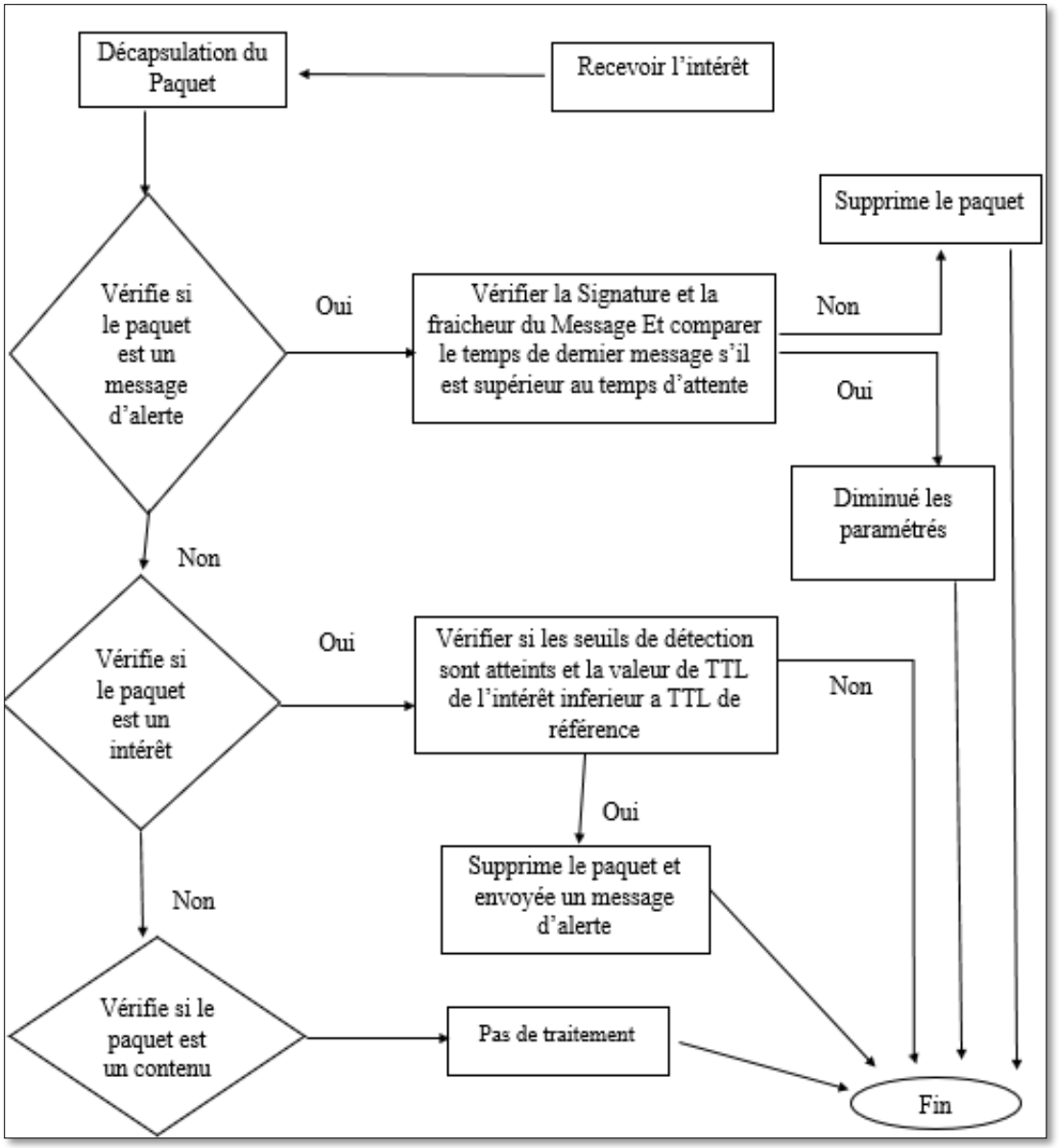

Figure III-5 Organigramme de Poséidon_TTL sur réseau CCN

III-5 Simulation de Poséidon et Poséidon _TTL

La simulation des réseaux en générale consiste principalement à modéliser le comportement 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. Dans le cas des CCN il est pratiquement impossible d'avoir une simulation réelle.

Nous allons utiliser dans notre travail le simulateur OMNET++ avec le package ccn-lite qui simule légèrement le fonctionnement des CCN. On présentera le matériel et les logiciels utilisés pour tester les deux algorithmes, ainsi que les fichiers nécessaires pour faire marcher

la simulation et la topologie utilisée. A la fin, nous testerons Poséidon et Poséidon_TTL et on compare les résultats de la simulation.

III-5-1 Les matériels et logiciels utilisée

La simulation a été réalisée suivant la configuration logiciel/matériel suivante :

Tableau III-2 La configuration utilisée pour la simulation

CPU	RAM	Disque Dur	OS	Omnet++	Inet Frame Work	CCN-lite
I3-6006U 2.0 GHz	4 GB DDR3 L	1 TB HDD	Ubuntu 16.04.4 desktop amd64	OMNET++ 4.5	INET 2.6	CCN-LITE 0.3.0

III-5-2 Mise en œuvre de la simulation

Pour pouvoir effectuer la simulation de ces deux contres mesures, on a besoin de :

- Définir la topologie du réseau utilisé
- Définir les fichiers nécessaires au routage et les types des données et structures utilisées dans la simulation
- Faire la liaison entre les fichiers de la simulation

1) Topologie utilisée

Notre choix de la topologie est basé sur les scénarios possible, en effet on utilise une topologie présentant un scénario où le nombre d'attaquants est supérieur au nombre des utilisateurs légitimes, ou nous avons 5 clients légitimes, 6 utilisateurs malveillants (attaquants) et 6 routeurs, un serveur qui contient des contenus nommés ayant le préfixe « /b3c/wowmom », un admin qui remplit le serveur avec les contenus et synchroniser les nœuds avec la simulation.

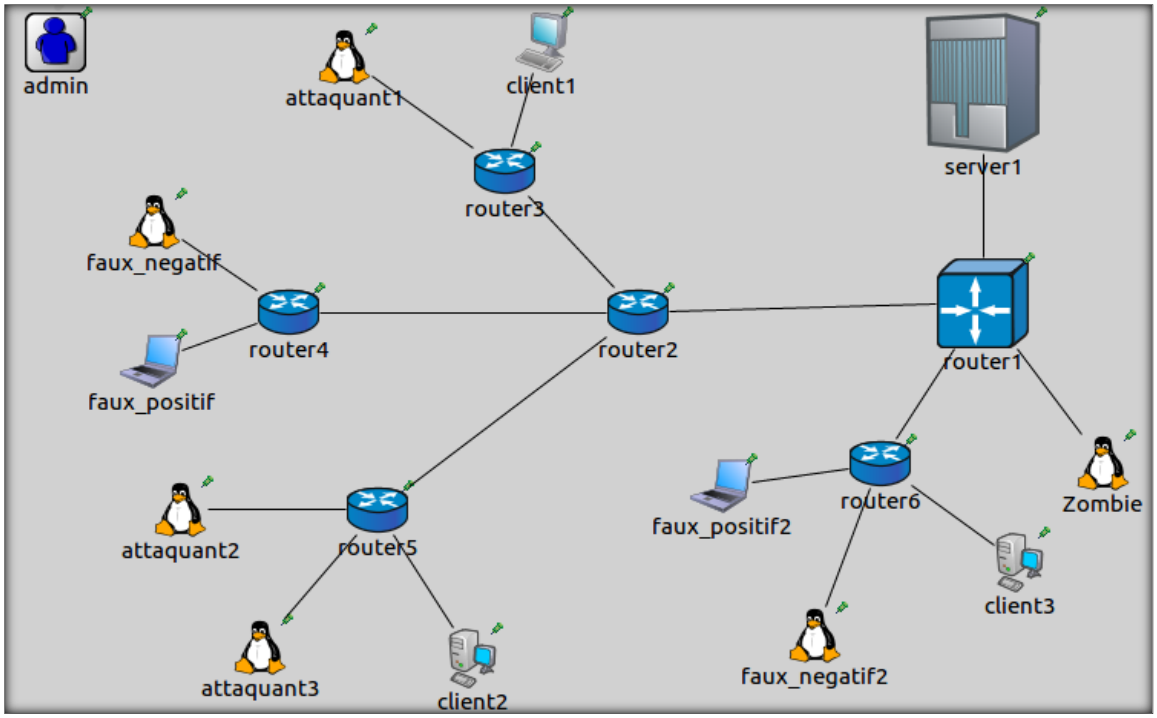

Figure III-6 Topologie utilisée pour la simulation

La topologie est représentée dans notre simulateur par un fichier de l'extension. NED, il contient le code source de la topologie, aussi les paramètres DeLay et data rate qui représentent le temps de retard dans un canal qui est fixé à 0.5 us et le débit de transmission des données (100 Mbps) respectivement. De plus, les nœuds sont liés entre eux avec des liens fast Ethernet pour la couche de liaison des données dans une partie de ce fichier.

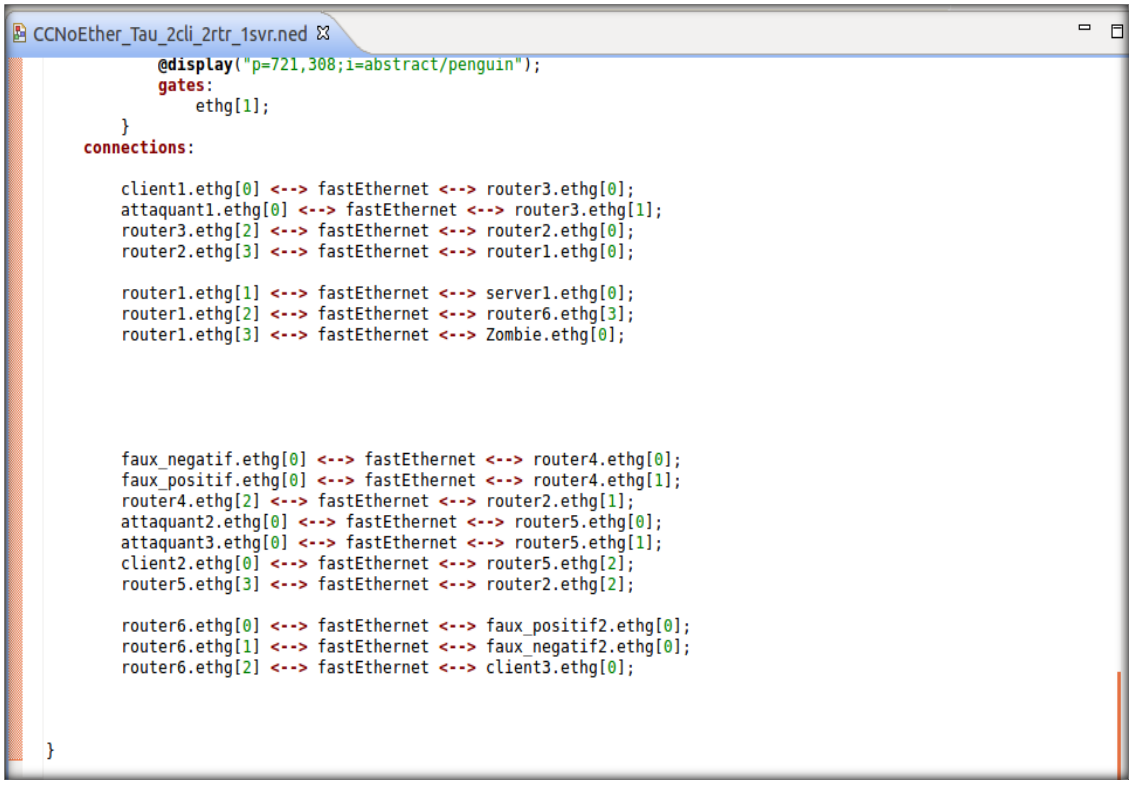

Figure III-7 Les connections fast Ethernet entre les nœuds dans le fichier de topologie. NED

5.2.2 Routage et les données nommées

Chaque nœud CCN dans la topologie requiert un fichier de configuration, de l'extension .cfg, le fichier est composé de trois parties principales et une partie de commentaires.

- **[eInterestMode] :** c'est le PIT du nœud CCN, contient les intérêts nommés demandés par le nœud source (les données nommées), avec les attributs **ContentName** qui est le nom du contenu, **StartChunk** qui est le début du segment, **ChunksCount** (le nombre des segments) et le **Request Time** (le temps de réponse)
- **[ePreCacheMode] :** qui représente le CS (cache) pour stocker les contenus, elle agit comme une mémoire physique, dans notre cas le serveur utilise cette mémoire pour stocker les contenus demandés par les autres nœuds, elle possède les mêmes attributs que le **[eInterestMode].**
- **[eFwdRulesMode] :** Elle équivalente à la table FIB, elle contient les informations de routage, ContentPrefix pour définir le préfix du nom de contenu, NextHop (interface de nœud suivant), AccessFrom (interface de sortie) et UpdateTime (temps de mise à jour).

- **[eCommentsMode] :** là on peut insérer des commentaires.

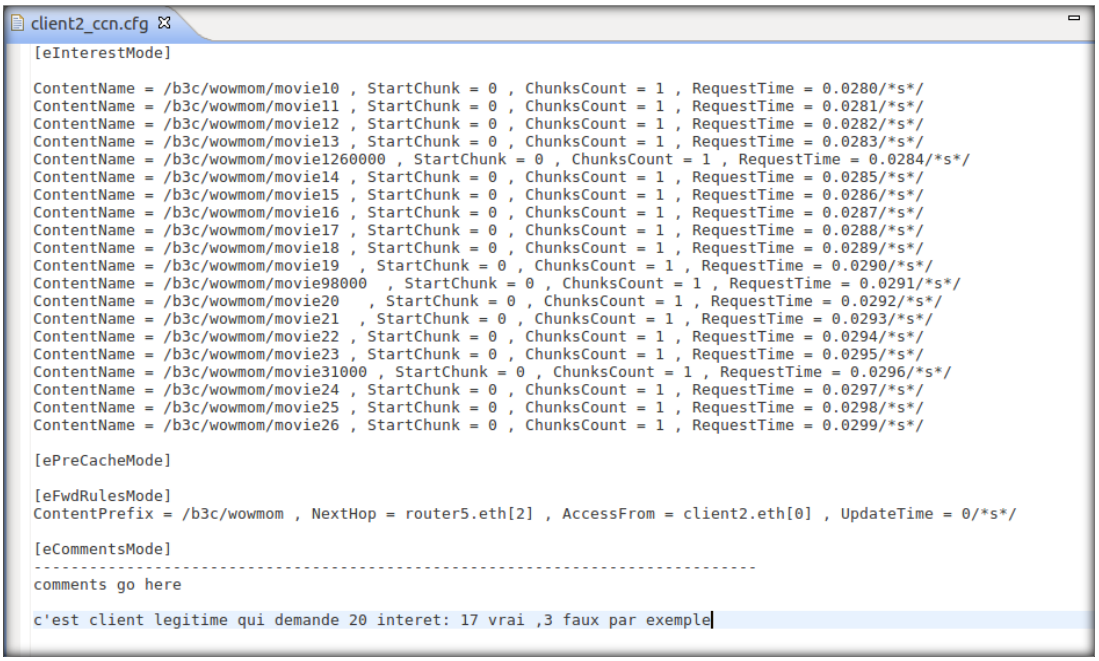

Figure III-8 Exemple d'un fichier de configuration d'un client légitime

5.2.3 La liaison entre les fichiers. NED et .cfg

Pour pouvoir lier le fichier de topologie (.NED) avec les fichiers de configuration de chaque nœud (.cfg), on doit créer un fichier de l'extension.INI qui contient des liens vers ces deux composants et initialise les paramètres d'entrée de la simulation comme la taille maximale du mémoire cache de chaque nœud et la version du ccn lite qu'on utilise.

omnetpp.ini

```
network = unibas.ccnlite.topology.CCNoEther_Tau_2cli_2rtr_1svr
description = "Example CCN over Ethernet Tau topology with 2 clients, 2 routers, 1 servers. Scenario setup taken fr

## 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"
*.Zombie.net.ccnScenarioFile = "Zombie_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"
```

Figure III-9 Fichier .INI de la simulation

III-6 Tests, Résultats et Discussion

Après avoir implémenté l'algorithme de Poséidon dans tous les routeurs de notre réseau avec les seuils avec le protocole TTL non activé au départ, ceci nous permet de tester Poséidon avant, et puis activer le protocole TTL et refaire les tests afin de comparer les résultats obtenus.

III-6-1 Poséidon

Nous lançons notre simulation, le client 1 envoie ses demandes normalement et le réseau s'en occupe pour satisfaire ses demandes, l'algorithme de Poséidon n'est pas déclenché car ses seuils ne sont pas atteints pour le moment, notre point de départ est un test expérimental sur une topologie qui a été définie précédemment, le test est fait avec les valeurs de **[RAM 17]** qui a fixé les paramètres suivants : $\Omega(r_i^j)$ =2.5, $P(r_i^j)$=700, tk = 0.01, le facteur "s" = 1.3 tel que ces paramètre et leur valeurs selon **[RAM 17]** représentent

- $\Omega(r_i^j)$: Seuil de détection d'inondation d'intérêt pour ω (r_i^j, t_k) qui a été fixe sur 2.5 car il ne force pas Poséidon faire une défense de manière rapide où lent.
- $P(r_i^j)$: Seuil de détection d'inondation d'intérêt pour ρ (r_i^j, t_k) qui a été fixe sur 700 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.

- Tk : est le k-ème intervalle de temps qui prendre en valeur 0.01 car le rôle de ce paramètre est de réinitialiser les calcule des valeurs ρ (r_i^j, t_k) et ω (r_i^j, t_k) au début, alors si le Tk 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 Tk doit être une valeur presque élevée comme 0.01

- Facteur ''s'' : diminuer les seuils en cas d'un IFA et prendre en valeur 1.3 car il celle-ci laisse l'interface ouvert dans un durée de temps raisonnable

Résultats

Après avoir terminé, les attaquants Zombie et attaquant_1 commencent à lancer leurs attaques mais après un certain temps ils seront bloqués par les routeurs qui leur sont proches. Une fois terminée, l'attaquant 2 commence à attaquer le réseau mais ses requêtes sont bloquées par le routeur 1.

L'attaquant 2 est bloqué par Poséidon à cause de la surcharge faite par les anciens attaquants sur routeur 1. En conséquence, le routeur 1 bloque tout accès légitime de client 1, 2, faux positif, ce qui augmente le nombre de faux positifs car une grande partie du réseau s'est bloqué pour rien.

Après cet ensemble d'attaques, l'attaquant faux_négative2 envoie agressivement ses demandes ce qui résulte un blocage du chemin entre routeur 1 et router 6 par le routeur 1 qui a subi une grande charge de la demande, donc faux_positif 2 et client 3 seront incapables d'envoyer leurs requêtes a cause de ce blocage.

Les nœuds faux_négatif et faux_positif ne sont pas impliqués directement dans le scénario, nous les avons ajoutés au cas où le router 1 ne sera pas surchargé, puisque nous avons notre résultat souhaité, ils n'envoient plus les intérêts.

Arrivé à l'attaquant 3, avec ses attaques agressives il arrive à bloquer le chemin entre router2

et router5 ce qui rend le client2 un faux positif donc il est bloqué pour rien, puis ce dernier va être bloqué par le router5 grâce à la charge d'attaque et le nombre des attaquants qui sont dans le même réseau.

A la fin, tous les attaquants seront bloqués au niveau de leurs routeurs les plus proches, mais ceci est venu après avoir bloqué des chemins aux utilisateurs légitimes, nous avons ainsi un nombre important de faux positifs.

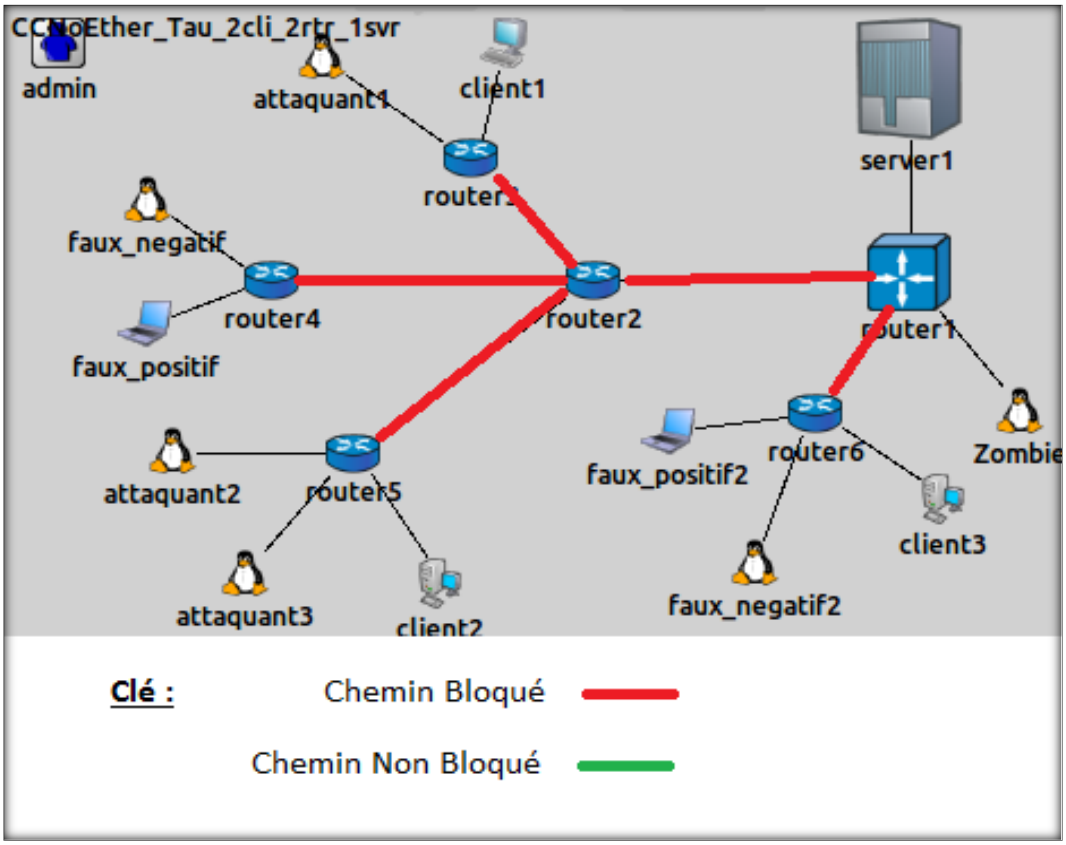

Figure III-10 Résultat de l'exécution du Poséidon sans le TTL

III-6-2 Poséidon_TTL

- **Cas de TTL=2**

On va rajouter le nouveau protocole de nombre de sauts comme étant une métrique qui va déterminer si l'algorithme va être lancé ou pas. Selon **[Goto 12]**, dans un réseau IP la valeur moyenne de nombre de sauts pour un paquet IP c'est 30, c'est-à-dire un paquet IP généralement passe par 30 routeurs aux moyennes pour arriver à la bonne destination. Cependant, nous avons choisi la valeur de TTL dans notre réseau CCN en fonction de la

largeur et la hiérarchie du réseau, plus le réseau est grand plus le seuil TTL doit augmenter. Il n'existe pas une formule mathématique exacte pour calculer ce seuil TTL dans un CCN (valeur définie par l'admin réseau), cette limite reste comme perspective pour les travaux futurs.

Résultats

Dans notre cas, on a pris une valeur de TTL égale à 2, On relance notre simulation en intégrant le TTL dans Poséidon, les mêmes étapes sont répétées pour le client 1, attaquant 1 et zombie, sauf que maintenant le TTL existe pour chaque intérêt d'un nœud. Toutefois, on remarque que l'interface entre router 1 et router 2 n'est pas bloquée car la valeur de TTL pour les intérêts du client 1, faux positif, client 2 égale à 3 ce qui est supérieur à la valeur référentielle fixée dans notre algorithme.

L'interface entre router 1 et router 6 ne se bloque pas car le nombre de saut des intérêts du faux_positif 2 et du client 3 se trouvant en dessous du router 6 sont égaux à 2, ce qui ne satisfait pas la condition pour bloquer l'interface (TTL>2).

Par conséquent, chaque attaquant présent dans la figure est bloqué dans l'une des routes proches, par exemple faux_négatif 2 est bloqué par routeur 6, attaquant 2 est bloqué par le routeur 5, sauf l'attaquant 3 ou il bloque le chemin entre router 5 et router 2. Par conséquent, le client2 sera un faux positif bloqué par Poséidon.

Dans le cas de l'attaquant 3, la métrique de TTL n'a pas pu fonctionner dans notre simulation à cause de la surcharge énorme d'attaques faites par l'attaquant 2 et 3 qui sont dans le même réseau que le client2

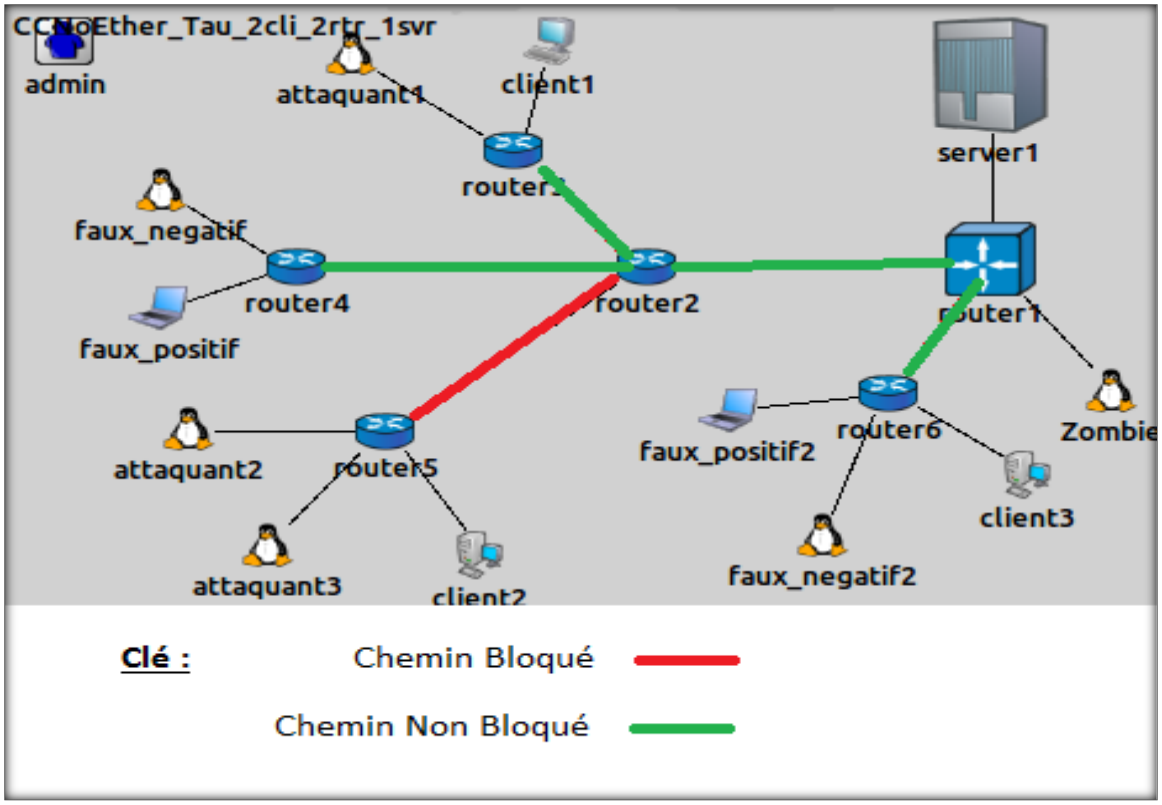

Figure III-11 Le résultat de l'exécution du Poséidon _TTL égale à deux

- **Cas de TTL=3**

Dans ce cas, on a pris une valeur de TTL égale à 3 qui est la valeur maximale de sauts qu'un intérêt peut prendre pour arriver au serveur.

On relance ce test à travers la simulation sans oublier que le paquet intérêt possède l'entête TTL qui sera vérifié par chaque routeur si l'intérêt a été satisfait par les demandeurs légitime.

Résultats

On remarque que l'interface entre routeur 1 et routeur 6 est bloquée par faux négatif 2 sa valeur de TTL est égale à 2, pour satisfaire les données qui existent dans le serveur, sachant que la valeur du seuil TTL est fixée dans notre algorithme à 3.

Toutefois, on remarque que les interfaces entre routeur 1 et routeur 2 ne sont pas bloquées car la valeur de TTL pour les intérêts du client 1, faux positif, et client 2 est égale à 3 ce qui est supérieur ou égale à la valeur référentielle fixée dans l'algorithme Poséidon _TTL.

Par conséquent, client 1, faux positif et client 2 ne seront pas bloqués et peuvent envoyer leurs requêtes normalement et l'attaquant 2 sera bloqué au niveau de son routeur le plus proche

(router 5) et non pas par le routeur 1 ce qui a augmenté le nombre de faux positifs.

L'attaquant 3 reste toujours un problème en bloquant le chemin entre router2 et router5.

Ceci nous permet de déduire que L'algorithme Poséidon_TTL fonctionne très bien.

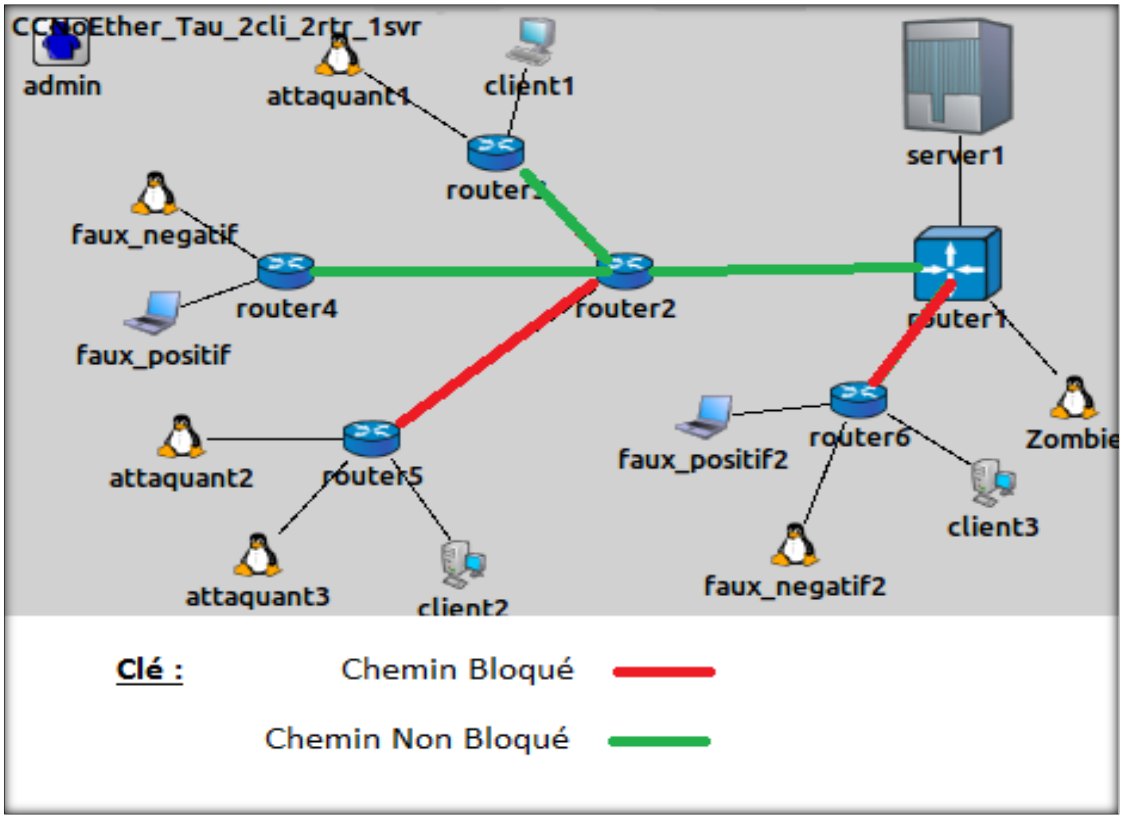

Figure III-12 Le résultat de l'exécution du Poséidon _TTL égale à trois

La simulation des différents cas a produit le temps du blocage de chaque chemin qui peut créer des faux positifs, ce temps est présenté dans le tableau ci-dessous

Tableau III-3 Tableau qui représente le temps du blocage de chaque chemin

Type d'algorithme	Temps du blocage pour le chemin R2 R5 (ms)	Temps du blocage pour le chemin R1 R2 (ms)	Temps du blocage pour le chemin R1 R6 (ms)	Total (ms)
Poséidon	1.3 ms	1.3 ms	0.7 ms	3.3 ms
Poséidon_TTL =2	1.3 ms	0 ms	0 ms	1.3 ms
Poséidon_TTL =3	1.3 ms	0 ms	0.7 ms	2 ms

III-6-3 Discussion

Après plusieurs jeux de test, les résultats de l'exécution de l'algorithme Poséidon_TTL sont récapitulés dans un Histogramme montrant le nombre de client légitimes qui ont le risque d'être bloqués pour rien selon différentes valeurs du champ TTL.

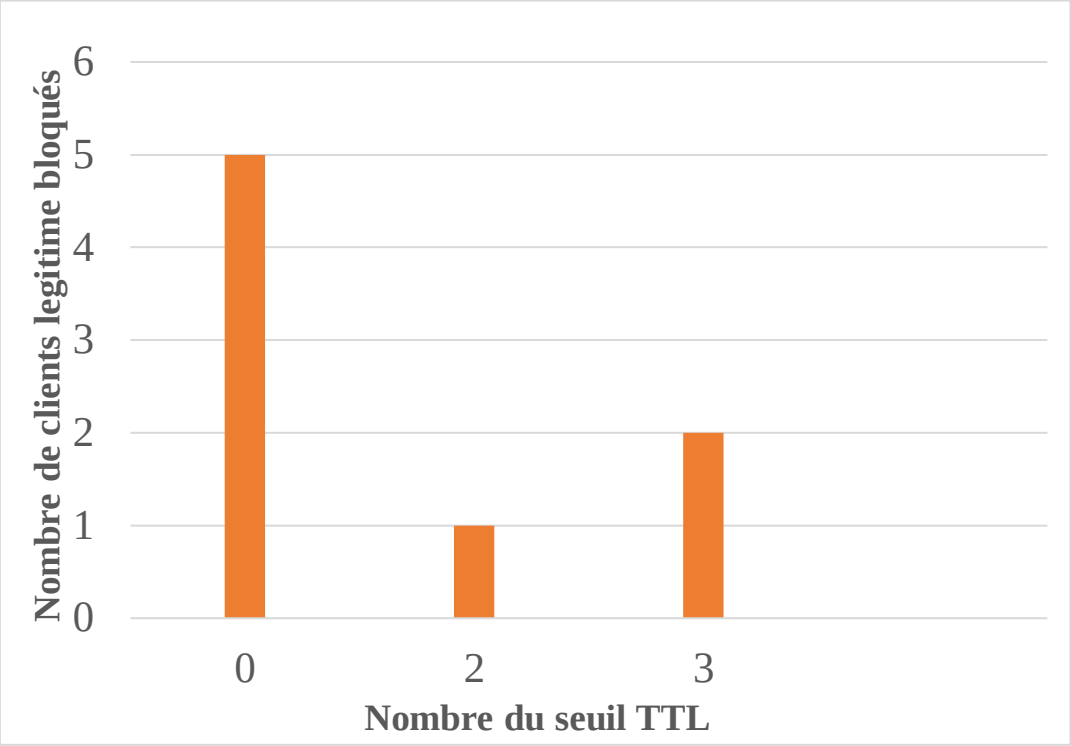

Figure III-13 Résultats des tests sur Poséidon_TTL

Après avoir représenté les résultats du test sous un histogramme la valeur du champ TTL=1 n'a pas été simulée par CCN Lite, car d'après la condition implémentée dans l'algorithme Poséidon_TTL il refuse tout intérêt malveillant qui est inférieur ou égale au nombre de saut 1, ce qui est impossible dans le réseau en générale car il n'existe jamais un paquet ou bien un intérêt dans un réseau ccn qui ne passe pas par au moins un routeur et donc ne traite même pas la condition proposée pour la détection de l'attaque (sera toujours fausse) ce qui rend Poséidon faible et exposé aux attaques sans y rien faire.

Toutefois, une métrique TTL égale à 1 provoque plusieurs problèmes et il est préférable de ne pas l'utiliser (Au minimum TTL=2).

III-7 Conclusion

Nous avons proposé au niveau de ce chapitre un protocole de calcul de saut dans les réseaux CCN similaire au protocole TTL dans les réseaux IP tout en respectant les spécificités des réseaux CCN et leurs caractéristiques.

Par la suite, nous avons modifié l'algorithme Poséidon en ajoutant la métrique Seuil TTL aux paramètres d'exécution, cette dernière nous a permis de diminuer le taux d'erreurs considérablement en éliminant au maximum le cas des faux positifs.

En effet, après avoir fait les tests de notre nouveau protocole nommé TTL_CCN, nous avons constaté qu'il est énormément performant dans la cadre de réduire le nombre de faux positifs bloqués par Poséidon dans un réseau CCN. Toutefois, il est nécessaire de bien choisir la valeur de TTL qui doit correspondre aux hiérarchies du réseau et sa structure ainsi que le nombre des nœuds pour avoir une meilleur efficacité et performance.

CHAPITRE IV.
LA SIGNATURE NUMÉRIQUE CONTRE LES INTERCEPTIONS

CHAPITRE IV. LA SIGNATURE NUMERIQUE CONTRE LES INTERCEPTIONS

IV-1 Introduction

L'internet a connu une énorme évolution lors de ces dernières années avec l'apparition de nombreuses applications qui traitent les différentes types de données (texte, audio ,vidéo ..), le e-commerce, le devise électronique (par exemple le bitcoin) , les systèmes bancaires et beaucoup des entreprises qui leur travail est basé sur l'internet , toutes ses transactions sont vulnérables à un ou plusieurs adversaires qui peuvent manipuler l'information facilement, ce type d'attaque est l'attaques d'interception , ou un ou plusieurs attaquants se place entre un fournisseur de données et un consommateur et essaient d'extraire les données importantes et les modifier et se faire espionner . Avec l'entrée des architecture ICN dans le monde informatique, le problème de ces attaques reste toujours, ce qui requiert un protocole de sécurité plus efficace et fiable pour atténuer les menaces de ces attaques, nous allons présenter dans ce chapitre une étude de l'art sur ces types d'attaques et une solution basée sur la signature numérique dans les Réseaux Centrés Contenu.

IV-2 Classification des types d'attaques interception

L'interception est un type d'attaque qui se fait sans l'autorisation ou la connaissance des utilisateurs. Il enfreint les règles de confidentialité du principe de sécurité. En termes simples, on peut dire que l'interception entraîne la perte de la confidentialité des messages. Il s'agit d'un type d'attaque passive. Elle a quatre types :

IV-2-1 Release of message

Lorsque vous envoyez un message à votre ami, vous voulez que seule cette personne puisse le lire. Grâce à certains mécanismes de sécurité, nous pouvons empêcher la divulgation du contenu des messages. Par exemple, nous pouvons chiffrer le message à l'aide d'un algorithme.

IV-2-2 Traffic analysais

Si de nombreux messages passent par un seul canal, l'utilisateur peut donner des informations à l'attaquant qui surveille le trafic généré dans ce canal, car il pense que le message provient de son émetteur légitime.

IV-2-3 Sniffing

Le sniffing est une méthode permettant de renifler les données transférées qui ont été envoyées par l'expéditeur. L'adversaire essaie simplement de savoir quel type de message ou de données est transféré par l'expéditeur sans son autorisation.

IV-2-4 Keyloggers

Il s'agit d'un programme qui fonctionne en arrière-plan, enregistrant tous les frappes de clavier. Une fois que les frappes sont enregistrées, elles sont cachées dans la machine pour une récupération ultérieure, ou expédiée à l'état brut à l'agresseur. L'attaquant les examine ensuite attentivement dans l'espoir de trouver des mots de passe, ou éventuellement d'autres informations utiles qui pourrait être utilisé pour compromettre le système ou être utilisé dans une attaque d'ingénierie social. Par exemple, un keylogger révélera le contenu de tous les courriers électroniques composés par l'utilisateur.

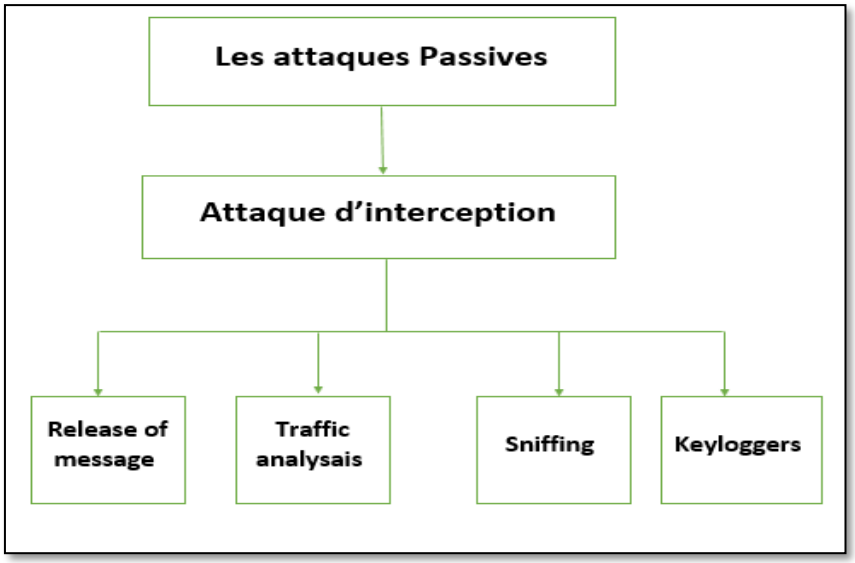

Figure IV-1 Classification des types d'attaques passive [KHA 11]

Selon **[LAT 18]** les impacts de l'attaque d'interception conduit vers l'infiltrations du chemin ce qui la source principale dans CCN, car les attaquent peuvent annoncer des routes non valides et le revendique comme des router faible, ce problème revient au comportement des nœuds CCN qui possède des copies de contenu a travers la mise en cache qui est

généralement distribue vers des emplacements différents non approuvés ce qui rend l'authentification des origines des chemins très difficile.

IV-3 L'attaque de l'homme du milieu

Dans la sécurité informatique, une attaque de type "homme au milieu" (MITM) est une attaque où l'agresseur transfère furtivement et peut-être modifie la correspondance entre deux parties qui se font confiance communiquer directement entre eux. L'attaque de l'homme au milieu (MITM) est un terme général pour lorsqu'un coupable se positionne dans une discussion entre un client et une application qui lui offre un service ; soit pour écouter furtivement, soit pour imiter l'une des parties, en faisant croire qu'un échange d'informations ordinaire est en cours, comme le montre la figure ci-dessous (Figure IV-2)

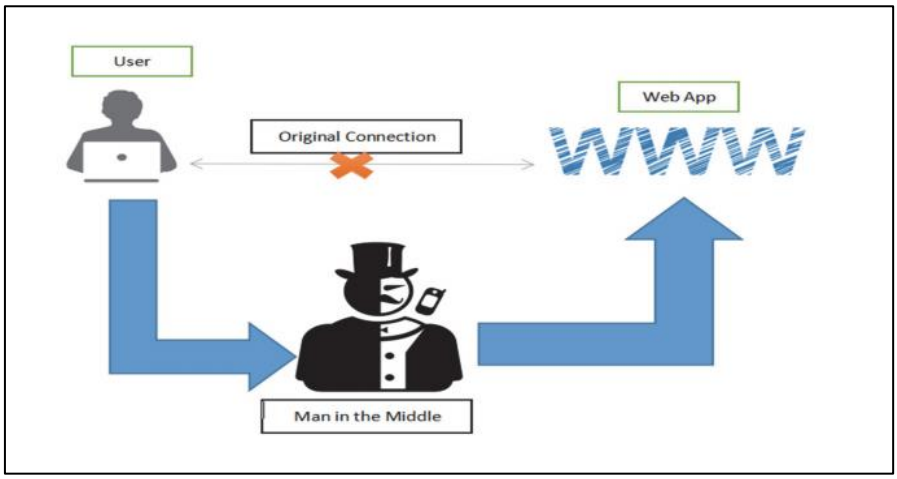

Figure IV-2 Idéologies de l'attaque de l'homme du milieu

L'objectif d'une attaque est de prendre des informations individuelles, par exemple, les certifications de connexion, les points de compte d'intérêts et de numéros de cartes de paiement. Les cibles sont normalement les clients des applications financières, les entreprises, les locaux commerciaux en ligne et d'autres sites où il est nécessaire de se connecter, les informations obtenues pendant une attaque pourrait être utilisée à de nombreuses fins, y compris la fraude, les échanges de soutien non approuvés ou un changement de mot d'ordre illégal.

IV-4 Les attaques d'interception dans les CCN

[ESL 15] Cette attaque est similaire à l'attaque de "homme du milieu". Contrairement à l'attaque du Hijacking, un attaquant qui se fait passer pour un éditeur de confiance qui annonce des routes non valables, tout en conservant un registre des routes valables vers le

contenu. Les demandes de contenu peuvent ensuite être saisies et envoyées aux services de l'emplacement. Bien que le récepteur reçoive le contenu normalement, l'attaquant prend connaissance du contenu demandé. Comme montre la figure ci-dessous, l'adversaire annonce des routes non valables pour attirer les demandes de l'utilisateur. Lorsque des utilisateurs légitimes envoient des demandes pour l'un des routes malveillantes, les nœuds ICN transmettre ces demandes au nœud malveillant de l'attaquant. L'attaquant enregistre les personnes qui ont demandé ce contenu et le transmet ensuite pour obtenir les données réelles. Lorsque les données réelles arrivent à l'attaquant, l'attaquant le renvoie au nœud ICN, qui à son tour le transmet à l'utilisateur légitime. Pour l'utilisateur, le scénario semble normal, mais en réalité, l'utilisateur malveillant viole la vie privée de l'utilisateur et peut même violer l'intégrité des données.

Contrairement à ces types d'attaques d'interception, ils ne souhaitent pas modifier le contenu du message original. Il est très difficile à détecter car il n'altère pas les données, l'attaque de l'homme du milieu (Man in the middle) est une attaque de type active qui se base sur la modification des données qui circule entre deux utilisateurs par un malveillant utilisateur qui se place au milieu des deux.

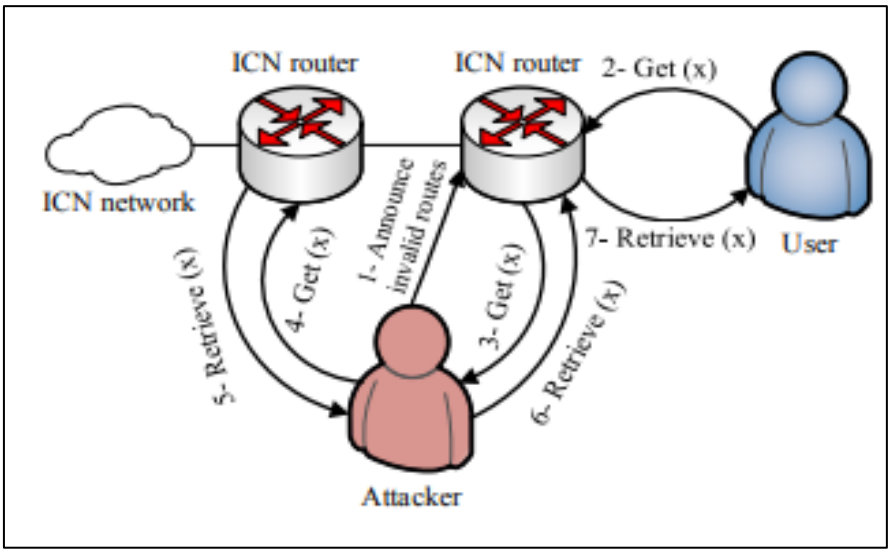

Figure IV-3 Exemple d'attaque d'interception [ESL 15]

IV-5 La sécurité Contre L'attaque d'interception

Pour protéger le contenu d'un client légitime plusieurs solutions ont été proposées dans l'architecture actuelle de l'internet (multipath, vpn, circuit éphémère, etc), mais vu que dans les Réseaux Centrés Contenu le processus du routage ne supporte pas les changements de

chemin (multipath) ni de créer un réseau virtuel (VPN) car parmi les concepts clés du CCN, le contenu prend le même chemin de retour que l'intérêt a entrepris pour retrouver le nœud demandeur du contenu (initiateur de l'intérêt), les nœuds CCN possèdent la notion de mobilité. (La topologie du réseau n'est pas fixe ni peut être définie). Donc la meilleure solution propose de chiffrer le paquet contenu demandé par le client et pas de tout l'intérêt, car l'attaquent veut voler les informations contenant dans le paquet du contenu et non l'intérêt, ce dernier ne l'aide pas à atteindre son objectif. Le chiffrement permet de résoudre trois problèmes différents

- **La confidentialité** Le texte chiffré ne doit être lisible que par les destinataires légitimes, Il ne devra pas pouvoir être lu par un intrus.
- **L'authentification** Le destinataire d'un message doit pouvoir s'assurer de son origine, un intrus ne doit pas être capable de se faire passer pour quelqu'un d'autre.
- **L'intégrité** Le destinataire d'un message doit pouvoir vérifier que celui-ci n'a pas été modifié en chemin. Un intrus ne doit pas être capable de faire passer un faux message pour légitime.

Ceci a été réalisé par L'algorithme de Signature Numérique qui sera présenté ci-dessous.

IV-5-1 Algorithme de la Signature numérique

La signature numérique est un procédé de sécurité qui permet d'identifier l'expéditeur d'un message. Elle est analogue à la signature manuscrite sur papier C'est un mécanisme de base qui permet la mise en œuvre de l'authentification des messages et de l'intégrité des données. Chaque nœud expéditeur signe numériquement son message avant de l'envoyer et chaque nœud récepteur vérifie la signature du message qu'il a reçu. La signature numérique repose sur une fonction mathématique appelée fonction de hachage. Cette fonction génère une empreinte « hash » du message qui sera chiffré par la suite avec la clé privée **[BEN 17].**

L'algorithme de signature numérique est composé par le cryptage asymétrique en appliquent le chiffrement RSA et le hachage MD5 qui est représenté par la figure ci-dessous.

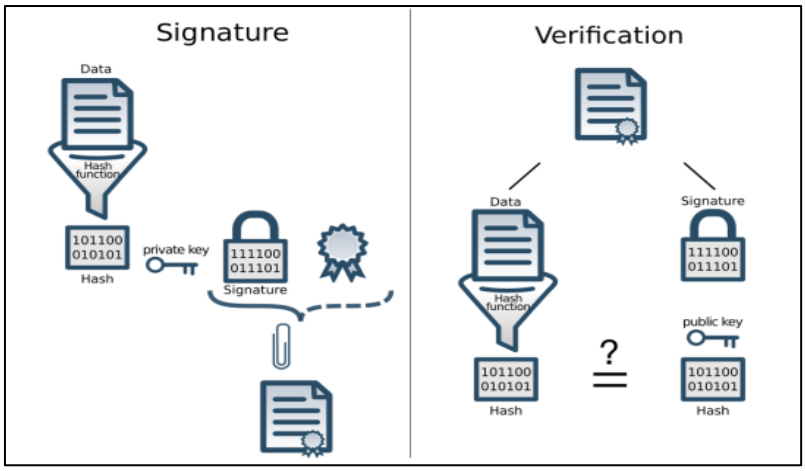

Figure IV-4 Fonctionnent du l'algorithme signature numérique [NIT 16]

IV-5-2 Algorithmes asymétriques

Les algorithmes asymétriques ont été inventés pour pallier précisément le problème de transmission sécurisée de la clé **[Ben 11].**

On parle d'algorithmes asymétriques car ce n'est pas la même clé qui sert au chiffrement et au déchiffrement. Dans le cas de ces algorithmes, on parlera alors de clé privée et de clé publique. Ces deux clés, clé privée et clé publique, sont intimement liées par une fonction mathématique complexe **[Ben 11].**

Les algorithmes asymétriques possèdent deux modes de fonctionnement

- Le mode chiffrement dans lequel l'expéditeur chiffre un fichier avec la clé publique du Destinataire pour chiffrer. Le destinataire utilise sa clé privée pour déchiffrer le fichier. Dans ce mode, l'expéditeur est sûr que seul le destinataire peut déchiffrer le fichier.

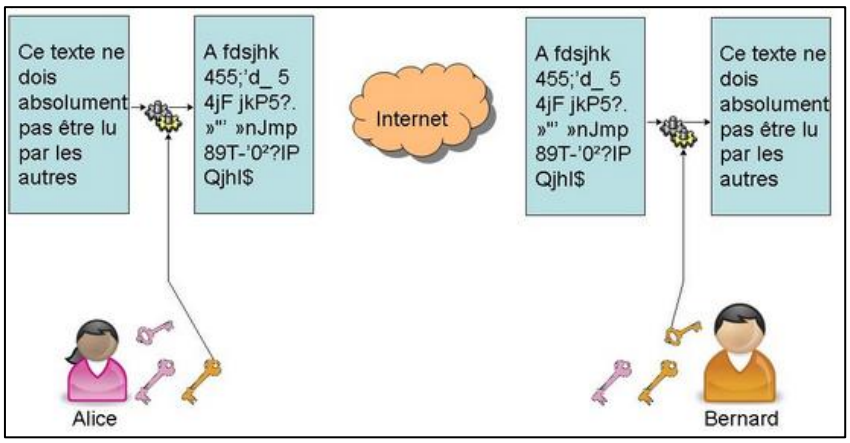

Figure IV-5 Chiffrement avec l'algorithme asymétrique [Ben 11]

- Le mode signature dans lequel l'expéditeur signe un fichier avec sa propre clé privée. Le destinataire utilise la clé publique de l'expéditeur pour vérifier la signature du fichier. Dans ce mode, le destinataire est sûr que c'est bien l'expéditeur qui a envoyé le fichier.

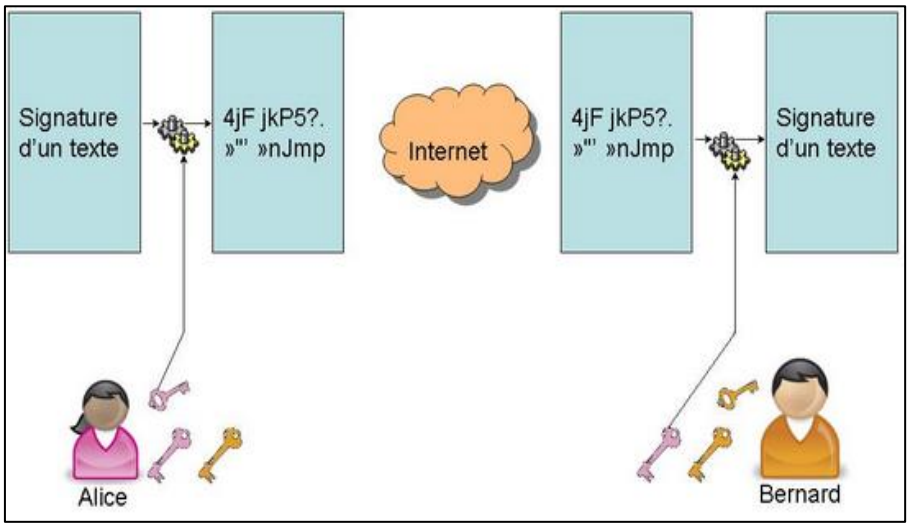

Figure IV-6 Signature avec l'algorithme asymétrique [Ben 11]

Plusieurs algorithmes de chiffrement asymétrique existent dans la littérature, dans notre solution on a utilisé RSA (Rivest, Shamir et Adleman en 1977). Ce chiffrement donne une mesure de sécurité plus fort et fiable pour partager les contenus entre les clients légitimes.

IV-5-3 Fonctionnement de RSA

Le plus solide algorithme qui est utilisé actuellement, est le système RSA qui est Publié en 1977 par Ron **R**ivest, Adi **S**hamir et Leonard **A**dleman de l'Institut de technologie du Massachusetts (MIT), RSA est fondé sur la difficulté de factoriser de grands nombres, et la fonction à sens unique utilisée est une fonction "puissance".

1) Chiffrement

On travaille principalement avec des nombres premiers (donc entiers naturels). A priori, il est assez facile de dire si un petit nombre est premier ou non. Cependant, dès que ce nombre devient plus conséquent, il est très rapidement difficile de dire s'il s'agit d'un premier ou non. Voici un atout principal du RSA **[MOR 05]**

- On choisit donc deux nombres premiers très grand p et q qui serviront à former les clés publiques et privées.
- On calcul N, qui est un constituant de la clé publique et de la clé privée en faisant : $\boldsymbol{N = p * q}$.
- Ensuite on calcul e, qui fait partie de la clé publique, avec

Φ(N)= (p-1) *(q-1) et

L'exposant public e de telle sorte qu'il est un nombre premier avec

(P-1) *(q-1). C.à.d. Le PGCD (e, φ(N)) =1.

- La couple (N, e) forme ainsi la clé publique de chiffrement.

On peut commencer le cryptage. On va se servir de la clé publique de chiffrement. Tout d'abord, il faut un message préalablement codé. On doit donc transformer en nombres le message d'origine (en utilisant la valeur ASCII de chaque lettre ou encore en remplaçant chaque lettre par son rang dans l'alphabet par exemple).

On a un message codé nommé $\boldsymbol{M}$, Il faut ensuite découper le message en blocs strictement inférieurs à N. On nomme alors chacun des blocs codés $\boldsymbol{Mi}$, le texte crypté $\boldsymbol{C}$ et chacun des blocs cryptés $\boldsymbol{Ci}$. Dans la réalité, les blocs sont très longs et les clés contiennent une centaine de chiffres, Afin de crypter, on fait $\boldsymbol{Ci = Mie \ mod \ N}$. Autrement dit, $\boldsymbol{Ci}$ (étant le bloc de texte crypté), équivaut au reste de la division de $\boldsymbol{Mie}$par$\boldsymbol{N}$.

2) Déchiffrement

- Il s'agit désormais de calculer, à partir de p et q, la clé privée d. Pour cela, il faut satisfaire l'équation $\boldsymbol{d.\ e \equiv 1\ mod\ \varphi(N)}$.
- Avec la clé $\boldsymbol{d}$ générée, on peut décrypter bloc par bloc le texte crypté. On retrouve alors le message$\boldsymbol{M}$.
- $\boldsymbol{Mi = Cid \ mod \ N}$. Autrement dit, Mi (étant le bloc de texte décrypté), et qui vaut au reste de la division de $\boldsymbol{Cid}$par$\boldsymbol{N}$. On retrouve alors nos blocs codés et il ne reste plus que les transférer vers leur équivalant dans l'alphabet défini .

3) Signature RSA

Comme pour l'opération de déchiffrement, la signature RSA S d'un message $\boldsymbol{M}$ consiste en une autre exponentiation modulaire avec l'exposant privé : $\boldsymbol{S = Md \ mod \ N}$. La validité de cette signature est vérifiée en utilisant la clé publique du signataire : $\boldsymbol{M}$ = ? $\boldsymbol{Se \ mod \ N}$.

Dans la pratique, on préfère, signer une empreinte du message $\boldsymbol{M}$ plutôt que de signer directement sa valeur. Pour ce faire, on utilise une fonction de hachage $\boldsymbol{H}$ pour calculer l'empreinte du document à signer$\boldsymbol{M}$ = ($\boldsymbol{M}$). Dans ce cas, la signature devient$\boldsymbol{S = Md \ mod \ N}$.

De même, le calcul de l'empreinte est aussi nécessaire pour la vérification de signature

[BER 10]

Comme pour les protocols fondés sur le logarithme discret, la sécurité du système RSA est

calculatoire, elle dépend essentiellement de la difficulté de factoriser un entier qui est le produit de deux grands nombres premiers. Si on sait factoriser n, il est facile de trouver d. mais la factorisation d'un nombre est un problème complexe (complexité NP) donc il est conseiller d'utiliser des clés de 1024 bits

Enfin le problème réel du RSA n'est pas la sécurité, mais la lenteur. Tous les algorithmes à clé publique sont 100 à 1000 fois plus lents que les algorithmes à clé privée, quelle que soit leur implémentation (logicielle ou matérielle)

IV-6 Conception et réalisation de l'algorithme de Signature Numérique

Le but principal de notre solution est de faire une simulation de l'attaque sur RSA, pour sa réalisation nous passerons par plusieurs étapes à savoir :

- Génération des clés publiques et privées pour client et serveur.
- Envoie les demandes de données (intérêt) qui seront vérifiées par la suite au niveau du serveur
- La condition effectuée permet de détermine si la donnée existe dans la table puis chiffrement du message lors de son transfert selon la méthode
    - Avec Signature : Chiffrement du message haché avec la clé privée de l'expéditeur pour générer la signature, puis chiffrer le message avec la clé publique du consommateur.
- Sinon rejet de l'intérêt demandé.
    - Déchiffrement du message reçu selon la méthode choisie
    - Avec Signature : Déchiffrement de la signature avec la clef publique du de l'expéditeur et comparaison avec le message haché reçu (après l'avoir déchiffré avec la clé privée du consommateur) afin de garantir l'authenticité et l'intégrité.
- Interception et déchiffrement des messages transmis entre client et serveur

Ces étapes peuvent se résumer par l'organigramme suivant :

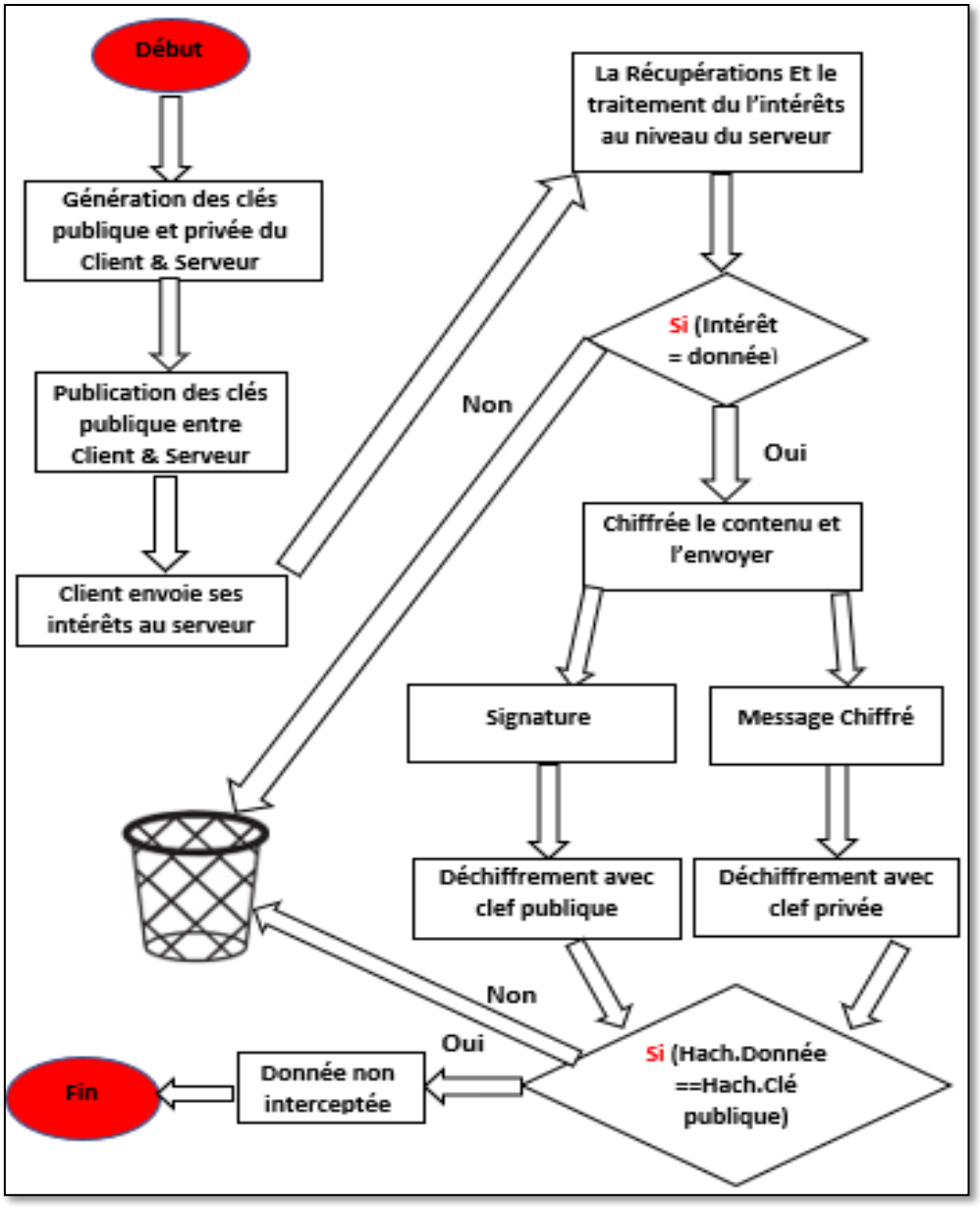

Figure IV-7 Organigrame de l'algorithme de la signature numérique

IV-7 Simulation de la solution proposes

Dans cette section, nous essaierons d'implémenter la solution de la signature numérique pour les paquets de contenus contre les attaques d'interception et d'intrusions, dans les CCN avec les outils que le simulateur OMNET++ avec le package ccn-lite nous offre et nous présentons les différents acteurs et paramètres dans cette simulation ainsi que ses résultats finaux.

Les configurations logicielles et matérielles utilisées sont identiques à celles du chapitre précédant.

IV-7-1 Topologie et nœuds utilisée

Nous testerons notre solution dans un segment de la topologie utilisée dans le chapitre 3, la partie concernée est encadrée en rouge comme le montre la figure ci-dessous.

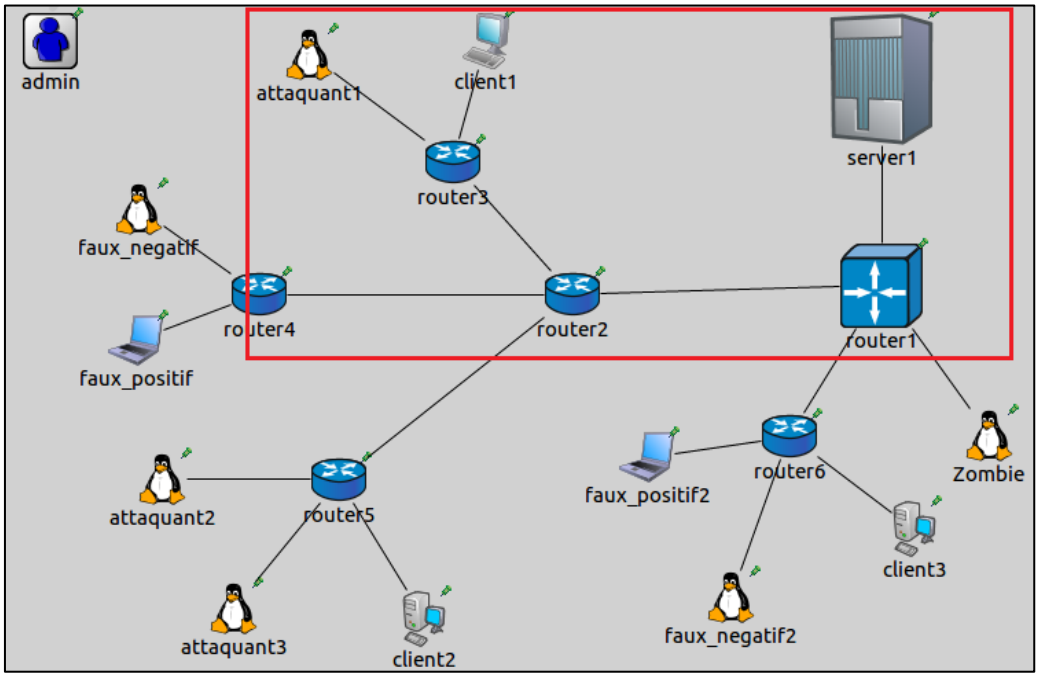

Figure IV-8 La partie de la topologie dont la solution est implémentée

Les nœuds concernés par la simulation sont: client1, attaquant1, router3, router2, router1 et server1.

Le client1 envoie ses demandes de contenu (intérêts) au serveur et le serveur lui répond par les contenus souhaités.

IV-7-2 Test de l'algorithme de la signature numérique

L'algorithme de la signature numérique se base sur quatre fonctions principales.

- **Signature (message en claire, clé privée)**

  Elle permet de générer une signature numérique pour le message en paramètres, en calculant le hash md5 du message et chiffrer (par RSA) cet hash avec la clé privée.

- **Verifier_signature (message reçu, signature de ce message, clé publique)**

  Cette fonction vérifie si le message reçu est valable ou non en vérifiant sa signature numérique : c'est en la déchiffrant par la clé publique du l'émetteur, ce qui lui donne un hash md5, après le récepteur déchiffre le message reçu avec sa clé privé et calcule le hash de ce message, puis le compare avec le hash de la signature, s'ils sont

identiques, le message est bien valable, sinon, il s'agit d'un contenu manipulé par un attaquant qui était en train d'écouter dans le réseau.

- **Chiffrement_RSA (message claire, clé de chiffrement)**
  Elle assure le chiffrement asymétrique RSA avec la clé de chiffrement
- **Dechiffrement_RSA (message chiffre, clé de déchiffrement)**
  C'est celle qui fait le déchiffrement avec la clé de déchiffrement et rendre le message en claire.

Le test de cet algorithme est réalisé en implémentant l'algorithme de la génération de la signature dans le server1 avec l'algorithme de chiffrement RSA, et le déchiffrement avec la vérification de la signature dans le client1.Les valeurs des clés utilisées dans la simulation sont calculées par les méthodes de génération des clés définit précédemment dans ce chapitre, on a pris par exemple dans notre simulation :

- ✓ Clé publique de chiffrement de contenu pour le client 1 : **(5,323).**
- ✓ Clé privée de déchiffrement de contenu pour le client 1 : **(173,323).**
- ✓ Clé privée pour la génération de signature numérique au niveau de serveur : **(317,437)**.
- ✓ Clé publique de serveur pour la vérification de la signature numérique au niveau du client 1 : **(5,437)**.

IV-7-3 Déroulement de l'algorithme de la signature numérique dans OMNET++

Quand l'intérêt est arrivé au serveur, il va produire un contenu relatif aux demandes du client, puis il le chiffre avec la clé publique de celui-ci, ainsi il lui généré une signature numérique avec sa clé privée et l'affecte a ce contenu chiffré pour assurer l'intégrité et la confidentialité de ce contenu.

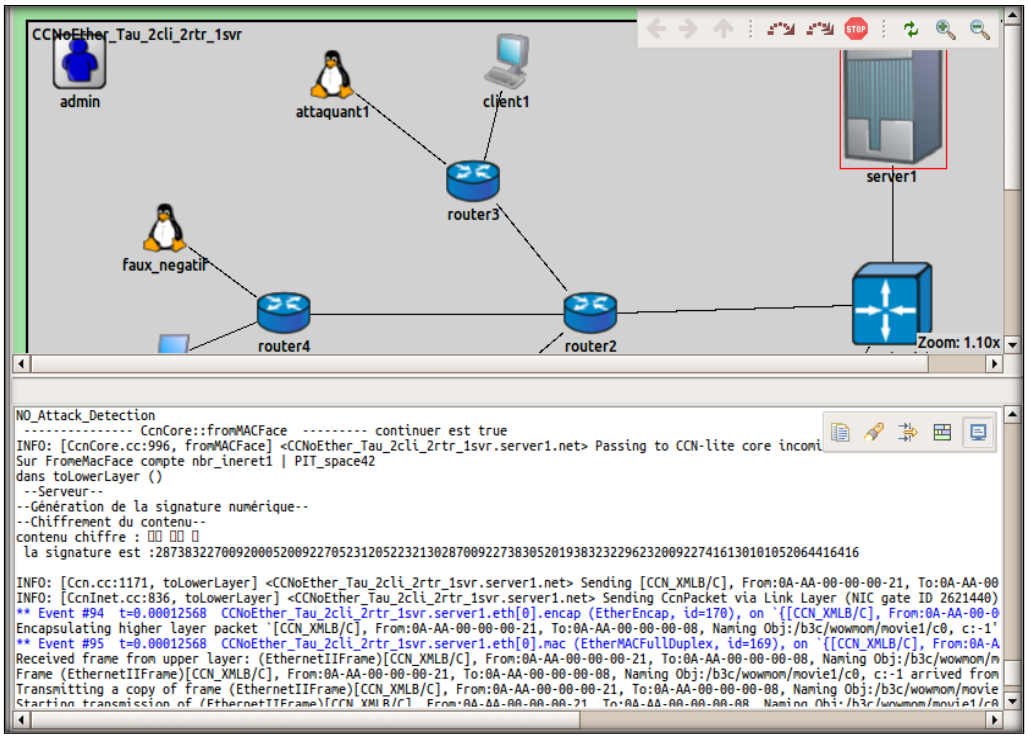

Figure IV-9 Le serveur génère la signature numérique et chiffre le contenu

Le contenu chiffré et la signature sont acheminés vers le client en suivant le chemin inverse de l'intérêt via le CCN.

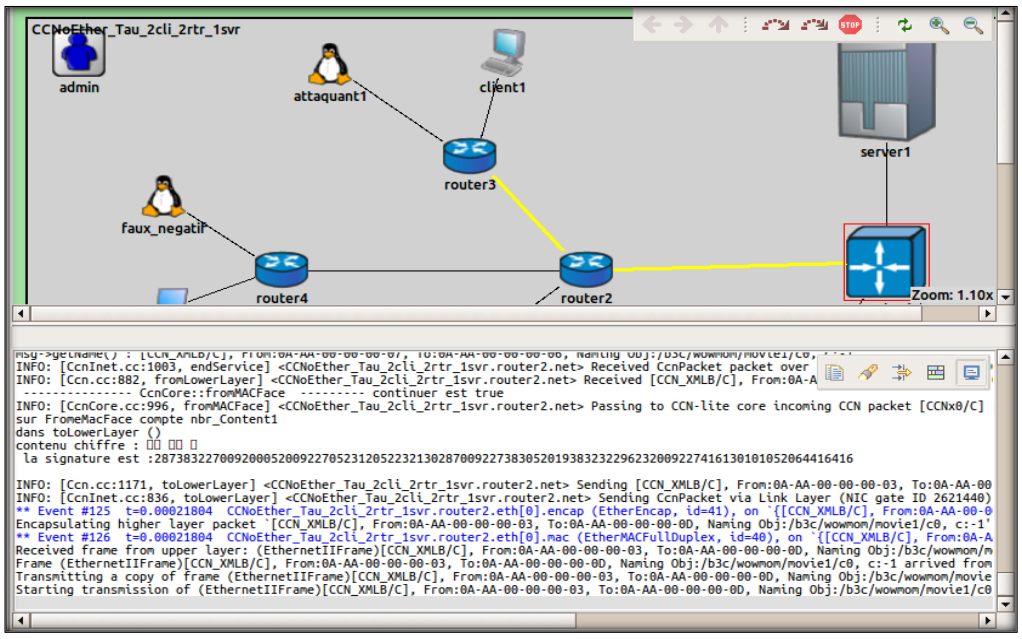

Figure IV-10 Le contenu chiffré et sa signature passant par les nœuds intermédiaires entre le client et le serveur

Finalement, le contenu chiffré et sa signature sont arrivés au client dont il déchiffre le contenu avec sa clé privée et vérifie sa signature avec la clé publique du serveur, et décide s'il est valable ou pas.

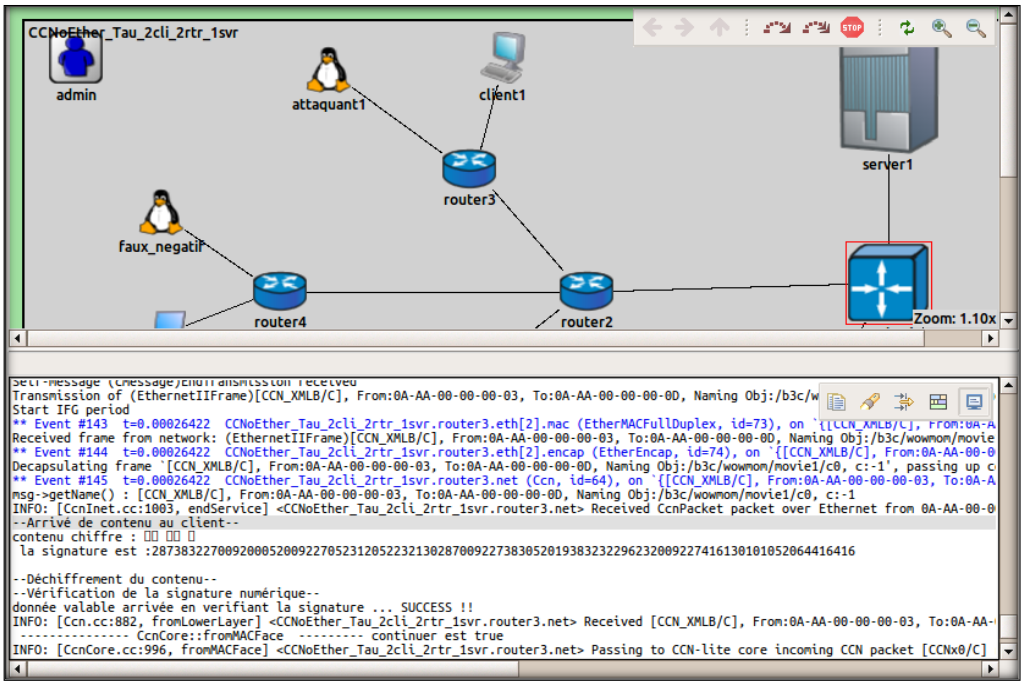

Figure IV-11 Le contenu arrivé au client qui vérifie lui déchiffre et vérifie sa validité avec sa signature.

IV-7-4 Scenario d'attaque

Nous simulons une attaque d'interception dans notre réseau CCN, en supposant que l'attaquant 1 a capturé le contenu avant d'arriver au client 1 et arrive à le modifier puis le repasse au client dans un premier lieu, puis nous allons simuler une attaque d'usurpation d'identité ou l'attaquant produit un contenu avec une signature numérique falsifiée et essaie d'usurper l'identité du serveur.

Quand le client reçoit le contenu falsifié il va le déchiffrer et vérifie son intégrité avec la signature numérique associé, puisque le contenu est modifié illégalement le client arrive à détecter qu'il s'agit d'une alternation du contenu via la signature numérique qui ne correspond pas au contenu déchiffré (les deux haschs ne sont pas identiques), il renvoi un message d'erreur que le contenu n'est pas valable.

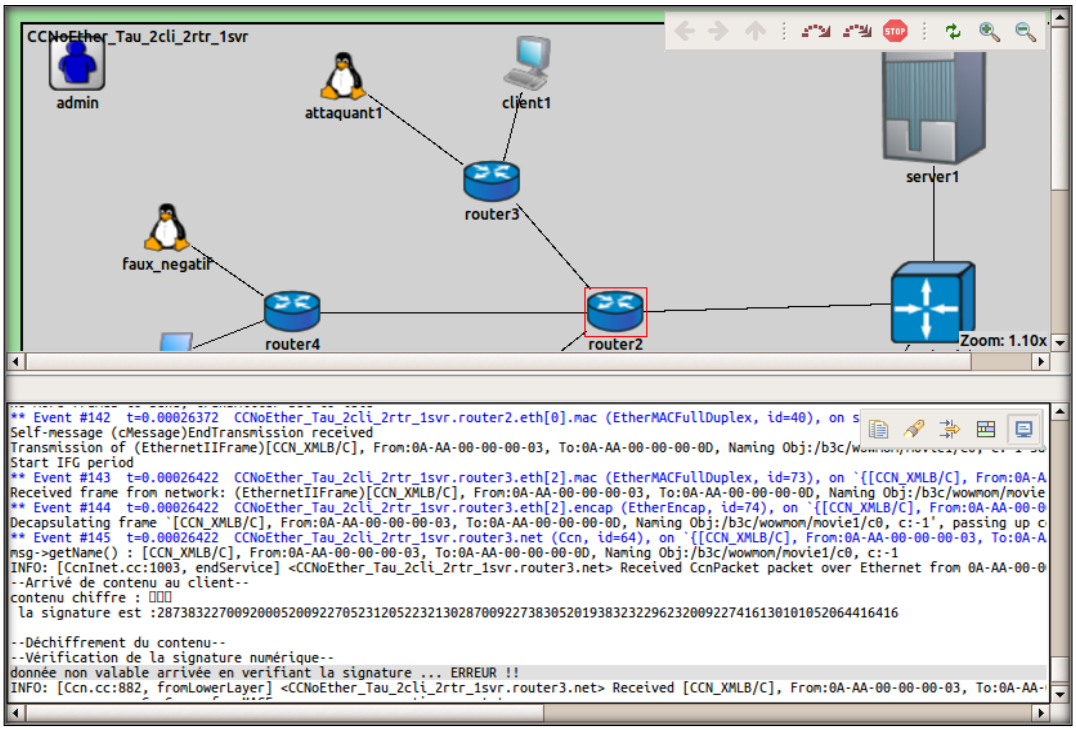

Figure IV-12 Le contenu falsifié arrive au client et le revoie d'erreur

Maintenant, avant que le contenu arrive au client, l'attaquant génère un contenu falsifié et lui affecte une signature avec une valeur de clé privé de son choix (nous testerons avec **(7,291)**), et l'envoie au client en se faisant passer comme un vrai serveur.

En conséquence, le client détecte l'anomalie dans le contenu arrivé car sa signature ne correspond pas à la clé publique du serveur original qui l'a, et renvoi le message d'erreur. Donc il s'agit d'une usurpation d'identité du serveur.

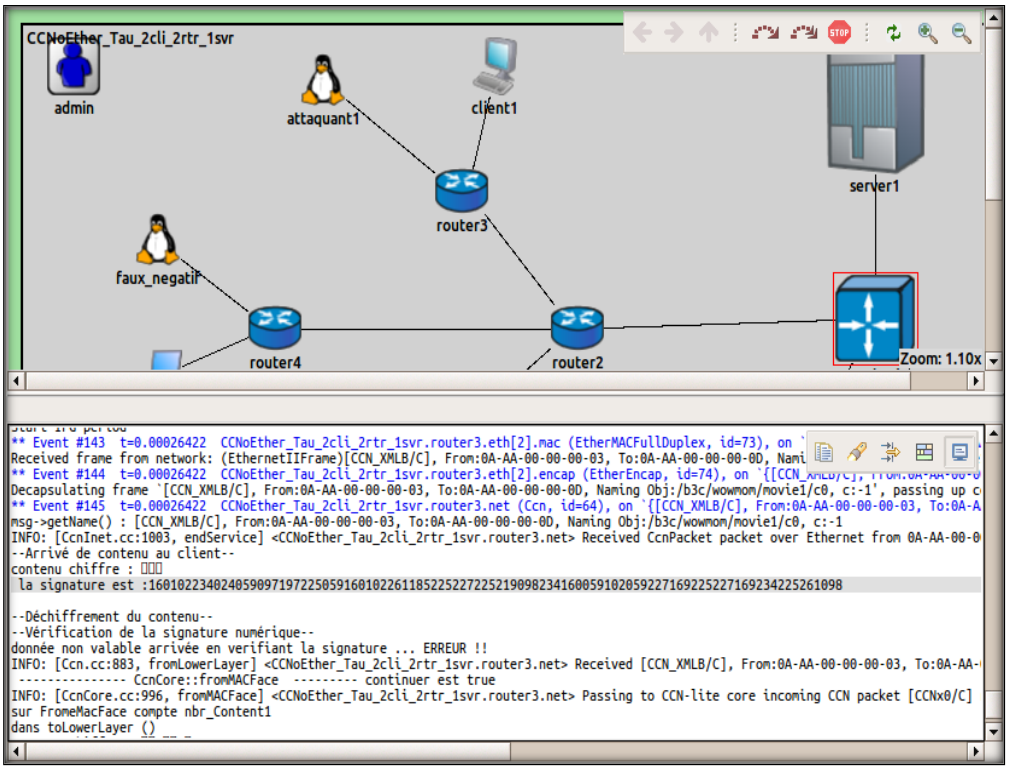

Figure IV-13 Résultats de l'attaque d'usurpation d'identité avec un contenu falsifié

IV-8 Conclusion

Comme la sécurité dans les CCN est basée sur la sécurité du contenu, avec l'implémentation du protocole de sécurité à base de la signature numérique dans un réseau CCN, on assure la confidentialité, l'intégrité et la non-répudiation du contenu par le processus du chiffrement asymétrique et la signature numérique dont le but est de détecter et de prévenir les attaques d'interception avec tous ses types dans les réseaux centrés contenu.

CONCLUSION GENERALE

Plusieurs initiatives ont été prises au cours de la dernière décennie pour améliorer l'architecture et les performances des réseaux centrés sur l'information (ICN). Dans ce contexte, Van Jacobson a présenté une nouvelle architecture appelée "Réseau Centré sur le Contenu" (CCN), Le principal objectif du CCN était de transformer la communication centrée sur l'hôte en communication centrée sur le contenu.

Dans le CCN, le demandeur est connu comme un consommateur qui envoie un intérêt et tout nœud disposant des données souhaitées peut renvoyer le contenu au consommateur en empruntant le chemin inverse, cette simple vue générale rend le routage très vulnérable aux attaquent, c'est ce volet qui a fait l'objet de notre travail, en effet, nous nous intéressons à renforcer la sécurité du routage à travers des contre-mesures luttant contre ces attaques.

Lors de notre projet de fin d'étude, nous avons étudié en premier lieu les CCN, leur architecture, leur fonctionnement ainsi que leurs vulnérabilités, par la suite notre travail a été divisé en deux grandes parties.

La première partie traite le problème des attaques par inondation d'intérêts, après avoir mis en revue les mesures de contre attaques existantes dans la littératures, plusieurs limites ont été constaté, nous avons proposé d'améliorer ces mesures en prenant l'algorithme de Poséidon comme exemple d'applications. Nous avons mis en place un procotole de calcul de sauts inspiré du TTL dans le réseau conventionnel IP (Time To Live) qui indique le nombre de sauts (également le nombre des routeurs) qu'un paquet IP peut avoir en traversant les routeurs intermédiaires entre la source et la destination, nous l'avons nommé TTL_CCN, ce protocole nous a permis d'estimer approximativement la profondeur des sous réseaux du point de vu d'un nœud (un routeur). Cette donnée une fois intégré dans Poséidon nous a permis d'éliminer au maximum le nombre de faux positifs bloqués auparavant par cet algorithme. Notre algorithme amélioré a été appelé Poséidon_TTL.

Plusieurs tests ont été effectués selon plusieurs essais et paramétrages, nous avons constaté que notre algorithme « Poséidon_TTL » a diminué de 80% le risque de bloquer des utilisateurs légitimes, ce qui rend notre solution plus performante et plus efficace que sa précédente, plus encore, il a été plus efficace contre les attaques de dénis de service distribués/

Ce type d'attaques permet d'inonder le réseau de façon consécutive dans le but de charger la table PIT en envoyant des intérêts falsifiés et bloquer les clients légitimes, nous nous sommes focalisés sur la modification de cet algorithme en ajoutant un nouveau champ appelée « TTL ».

La deuxième phase de notre travail est dédiée au chiffrement du contenu transmis, et ce dans le but de sécuriser les données contre les attaques d'interception, cette solution nous permet d'assurer l'intégrité et la confidentialité des données.

Nous avons choisi la signature numérique comme solution de chiffrement à notre problème, elle est composée d'un cryptage asymétrique en appliquant le chiffrement RSA, et d'un hachage MD5 pour renforcer la sécurité du paquet du contenu CCN et garantir confidentialité et l'intégrité des données.

Lors de notre travail de fin d'étude, nous avons pu effectuer plusieurs recherches bibliographique, approfondir nos connaissances théoriques dans le domaine de la sécurité des réseaux et plus spécialement les CCN, mais aussi appliquer ces connaissances en apprenant à travailler sur l'environnement OMNet++V4.5 ou nous avons implémenté CCN-Lite V0.3, avec l'aide de Dieu nous avons réussi à répondre aux objectifs établis au début de notre PFE.

Cependant, plusieurs limites ont été observées tel

- Le problème du paramétrage de la valeur de TTL qui est toujours manuelle, elle dépend de la taille du réseau, nous pensons qu'il sera plus intéressons d'automatiser cette valeur en la mettant à jour selon la moyenne des valeurs TTL entrantes à chaque routeur.
- Le problème des faux négatifs, il faudra en rajouter plusieurs tests pour vérifier si vraiment notre proposition reste toujours valable, et nous arrivons à bloquer les attaques zombies.

- Plusieurs autres attaques sont à traiter pour aboutir à un mécanisme de sécurité plus global.
- Introduire le concept de Block-Chain » dans les CCN, le Block Chain est une base de données informatique capable de stocker tous type d'informations (liste de courses, photos, données bancaires, données patients, dossiers de lot, …). Une des caractéristiques fondamentales est que cette base de données est répliquée chez les utilisateurs de cette blockchain. Ainsi, tous les utilisateurs possèdent une copie de la base de données, et sont reliés les uns aux autres directement ou indirectement par un réseau, ce concept sera utilisé pour renforcer la sécurité du réseau dans tous ses cotés.

ANNEXES

1- Simulateur OMNET++

OMNET++ est un simulateur à évènements discrets orienté objet et basé sur C++. Il a été conçu pour simuler les systèmes réseaux de communication, les systèmes multi processeurs, et d'autres systèmes distribués, OMNET++ est un projet open source développé à l'université de Budapest., Actuellement, Ce simulateur est utilisé par des dizaines d'université pour la validation de nouveaux matériels et logiciels, ainsi que pour l'analyse de performance et l'évaluation de protocoles de communication.

Les avantages de OMNET ++ ce sont principalement sa facilité d'apprentissage, d'intégration de nouveaux modules et la modification de ceux déjà implémentés.

2- Architecture d'OMNET++

L'architecture d'OMNET++ est hiérarchique composée de modules. Un module peut être soit un module simple soit un module composé. Les feuilles de cette architecture sont les modules simples qui représentent les classes C++. Pour chaque Module simple correspond un **fichier.cc** et un **fichier.h.** Un module composé comporte de simples modules ou d'autres modules composés connectés entre eux. Les paramètres, les sous modules et les ports de chaque module sont spécifiés dans un **fichier.ned.**

La communication entre les différents modules se fait à travers les échanges de messages. Les messages peuvent représenter des paquets, des trames d'un réseau informatique, des clients dans une file d'attente ou bien d'autres types d'entités en attente d'un service.

Les messages sont envoyés et reçus à travers des ports qui représentent les interfaces d'entrer et de sortie pour chaque module, La conception d'un réseau se fait dans un **fichier.ned** et les différents paramètres de chaque module sont spécifiés dans un fichier de configuration**(.ini).**

3-CCN Lite

CCN-lite est une implémentation ICN légère appartenant à la licence ISC permissive qui est développé à l'Université de Bâle en suisse Transitaire au format multi-paquets : NDN, CCNx, etc. CCN-lite fonctionne sur plusieurs plateformes comme x86 / 64 sur Linux, BSD et MacOs,

cela couvre :

- Le protocole de réseau centré sur le contenu de PARC
- Le Projet Named-Data Networking (NDN)
- Le projet Naming-Function Networking
- Un encodage expérimental et compact pour les environnements IOT

La Figure.A.1 représente l'intégration de CCN-Lite avec OMNet++ et INET Framework qui est nécessaire pour faire la simulation souhaitée, concernant notre travaille on a utilisé les Outils technique suivante :

- IDE OMNet ++ V 4.5
- CCN-Lite 0.3.0
- INET Framework V 2.6

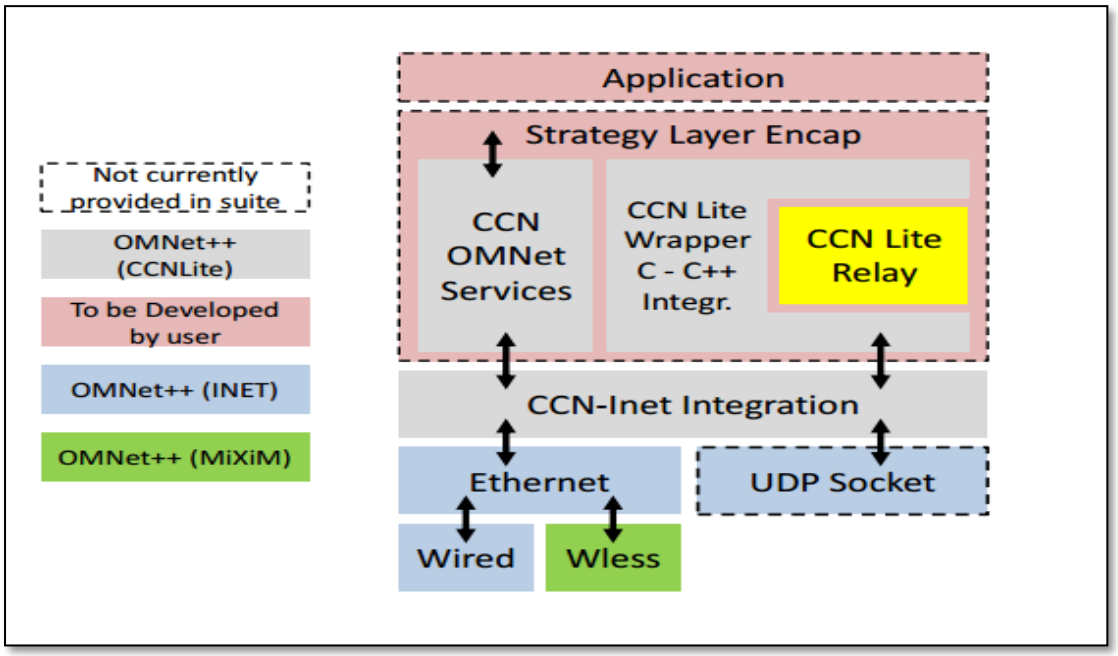

Figure A-1 schéma d'intégration de CCN-lite avec OMNeT++ et INET Framework **[GIT15]**

CCN-Lite cette extension légère qui se dispose au niveau de OMNet++ permet de fournit un énorme fonctionnement dans l'cote expérimentale ou développement, qui est a répondue au plusieurs besoins des chercheurs de domaine recherche académique et qui offre plusieurs caractéristiques parmi eux :

- Un noyau CCNx écrit en langage C

- Un support de multiple plate-forme
- Une mise en œuvre partielle du protocole de gestion interopérable
- Un service http simple pour afficher la configuration interne du relais
- Quelques extensions intéressantes

Les classes les plus importants dans le package CCN-Lite sont être présenté par le diagramme de classe UML ci-dessous :

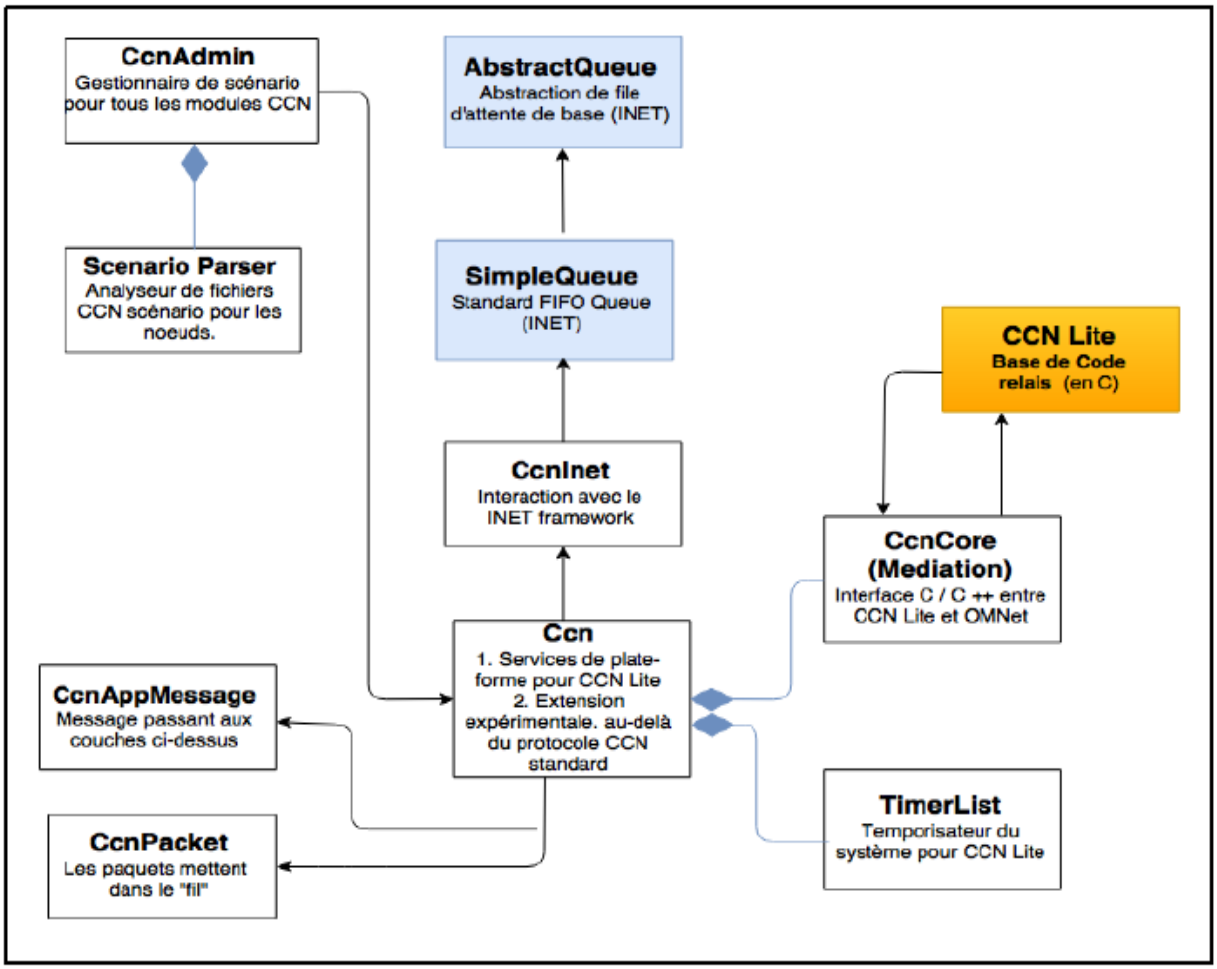

Figure A-2 Diagramme de classe UML des Composants CCN-lite/OMNeT++**[GIT15]**

4- INET Framework

INET Framework est une bibliothèque de modèles open source pour l' environnement de simulation OMNeT ++ , Il fournit des protocoles, des agents et d'autres modèles aux chercheurs et aux étudiants travaillant avec des réseaux de communication. INET est particulièrement utile lors de la conception et de la validation de nouveaux protocoles ou lors de l'exploration de scénarios nouveaux ou exotiques.

INET bénéficie de l'infrastructure fournie par OMNeT ++, Au-delà de l'utilisation des services fournis par le noyau et la bibliothèque de simulation OMNeT ++ (modèle de composant, paramétrage, enregistrement des résultats, etc.), cela signifie également que les modèles peuvent être développés, assemblés, paramétrés, exécutés et leurs résultats évalués dans le confort de l'IDE de simulation OMNeT ++ ou depuis la ligne de commande.

REFERENCES BIBLIOGRAPHIQUES

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

[HAC 16] - Hachad amine et khasnissi balaoua“Etude et Implémentation des Mécanismes de Sécurité pour le Routage Centré Contenu ", 2016-2017, UNIVERSITE SAAD DAHLAB, Blida

[RAM 17] - Ramla Mohamed et Walid Miloud Dahmane, “Etude et Implémentation des Mécanismes de Sécurité pour le Routage Centré Contenu ", 2017-2018, UNIVERSITE SAAD DAHLAB, Blida

[LAT 18] - LATRECHE FATMA ZOHRA“Etude et Implémentation des Mécanismes de Sécurité contre les intrusions pour le Routage Centré Contenu ", 2018-2019, UNIVERSITE SAAD DAHLAB, Blida

[MAR 17] - Maroua Meddeb, ‘Information Centric Networking, A naturel design for IoT applications ‘. Thèse de Doctorat, Université de Toulouse, 2017

[LIA 18] - Liang Wang, Information-Centric Networking from Point-to-Point Communication to Content Distribution, Université de Cambridge,2018

[BEN 12] - Bengt Ahlgren, Christian Dannewitz, Claudio Imbrenda, Dirk Kutscher, and Börje Ohlman, A Survey of Information-Centric Networking, IEEE Communications Magazine • July 2012

[FAB 13] - Fabian Ohlman, ‘Content-Centric Networking ‘, Faculty for Informatics, Technische Universität München, 2013

[KEP 19] - KEPING YU, SUYONG EUM, TOSHIHIKO KURITA, QIAOZHI HUA5TAKURO SATO, HIDENORI NAKAZATO, TOHRU ASAMI, AND VED P. KAFLE, Information-Centric Networking: Research and Standardization Status, date of current version September 17, 2019.

[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.

[Has 15] - Hassan Syed Ahmed, Safdar Hussain, Bouk Dongkyun Kim, Content-Centric Networks an Overview, Applications and Research Challenges, Kyungpook National University Research Fund, 2015**.**

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

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

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

[Ale 13] - Alexander Afanasyev, Priya Mahadevan, Ilya Moiseenko, Ersin Uzun , Lixia Zhang, Interest Flooding Attack and Countermeasures in Named Data Networking, University of California, Los Angeles, †, Palo Alto Research Center, 2013

[Nar 16] - Narayan Chhetry, Hemanta Kumar Kalita , Interest Flooding Attack in Named Data Networking: A Survey , Department of Information Technology, North Eastern Hill University , AJET, ISSN: 2348-7305, Volume 4(1), 2016

[Muh 16] - Muhammad Aamir and Syed Mustafa Ali Zaidi, Denial-of-service in content centric (named data) networking: a tutorial and state-of-the-art survey, Information Technology Division, National Bank of Pakistan, Karachi, Pakistan Faculty of Computing, SZABIST, Karachi, Pakistan, Published online 30 October 2014 in Wiley Online Library

[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

[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.

[Tas 17] - Tasnuva Mahjabin, Yang Xiao, Guang Sun and Wangdong Jiang, A survey of distributeddenial-of-service attack,prevention, and mitigation techniques , International Journal of Distributed Sensor Networks 2017, Vol. 13(12)

[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

[IBM 06] - ibm.com/Redbooks, Eighth Edition (December 2006), TCP/IP Tutorial and

[GOTO 12] - Goto Shigeki Ryo Yamada and, using abnormal TTL values to detect malicious IP packets, Department of Computer Science and Engineering, Waseda University, 3-4-1 Okubo, Shinjuku, Tokyo 169-8555, Japan

[KHA 11] - Khaleel Ahmad, Shikha Verma, Nitesh Kumar and Jayant Shekhar, Classification of Internet Security Attacks, March 2011

[BEN 17] - BENDOUM AMINA, DÉVELOPPEMENT D'UNE INFRASTRUCTURE DE GESTION DE CLÉS DE CRYPTAGE DANS LES RÉSEAUX AD HOC VÉHlCULAIRES (VANET), JUILLET 2017, UNIVERSITÉ DU QUÉBEC(Canada)

[NIT 16] - Nitulescu, Anca,Authentification et Integrité:Signature numériqueet Hachage,2016,Ecole Normale Supérieure(Paris)

[Ben 11] - Benzenine Hadjira, Amara Khadidja, La cryptographie appliquée sur les fichiers audio(son),28 septembre 2011, Université Abou Bakr Belkaid(Tlemcen)

[MOR 05] - Morges-Beausobre, ES,La Cryptographie & Le RSA,2005

[BER 10] - Berzati, Alexandre,Analyse cryptographique des altérations d'algorithmes,29 Septombre 2010,Thèse Doctorat de l'Université de Versailles Saint-Quentin en-Yvelines