REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITÉ SAAD DAHLEB BLIDA 01

Faculté des sciences

Département : Informatique

Mémoire de Fin d'Etude pour l'obtention du Diplôme de Master en Informatique

Option : Sécurité des Systèmes d'Information
&
Systèmes Informatiques et Réseaux

Thème :

Développement D'une Solution Optimisée Contre Les Attaques De Zombies Dans le Réseau ICN

Réalisé Par :

MelleSoualahChaima **Melle**Bakdache Rania

Devant le jury composé de:

Président : Mr Ouldaissa Ahmed.
Examinatrice : Mme Guessoum Dalila.
Promotrice : Mme Aroussi Sana.
Co- promotrice : Mme Arkam Meriem.

Année Universitaire : 2020/2021.

Remerciement

Tout d'abord, nous remercions ALLAH de nous avoir donné la patience et l'énergie pour réaliser ce travail.

Nos plus sincères remerciements pour nospromotrices Madame Aroussi Sanaa et Madame Arkam Meriem. Qui ont accepté de nous encadrer, ainsi que pour leur disponibilité tout au long de la réalisation de ce mémoire. La qualité de leurs conseils nous ont permis de mener à bien ce travail.

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 aussi les 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.

Enfin, nos remerciements les plus sincères à toutes les personnes qui ont contribué de près ou de loin à l'élaboration de ce mémoire ainsi qu'à la réussite de cette année universitaire.

Merci.

Dédicaces

C’est avec profonde gratitude et sincères mots, que je dédie ce modeste travail de fin d’étude aux personnes suivantes :

À ma mèrepour tous ses sacrifices, son amour, sa tendresse, son soutien et sesprières tout au long de nos études. Que Dieu la protège et l’accorde une longue vie pleine de santé et de bonheur.

À ma sœur Malak et mon frère Abderrahmane pour avoir contribué à la réussite de ce travail d’une manière indirecte, et pour leurs conseils et encouragements.

À mes amies, en particulier mon binôme Rania, mes amies Zola, Khaoula, Coumba, et Michou pour leurs soutiens et encouragements.

Finalement, à tous ceux qui me sont chers.

WARDA.

Dédicaces

C’est avec profonde gratitude et sincères mots, que je dédie ce modeste travail de fin d’étude aux personnes suivantes :

À mes chers parents pourleurs sacrifices, leur amour, leur tendresse, leur soutien et leurs prières tout au long de nos études. Que Dieu leur protège et l’accorde une longue vie pleine de santé et de bonheur.

À mon frère et surtout ma sœur Wissem pour avoir contribué à la réussite de ce travail d’une manière indirecte, et pour leurs conseils et encouragements.

À toute ma famille pour leur soutien tout au long de notre parcours universitaire en particulier mes chères cousines Fadia, Nour, Nada, Wiam. En particulier Hanine, Ouadie, et Abd el Madjid.

À mes amies en particulier mon binôme Warda, et mes amies Zola, Sekoura, Wiam, Coumba pour leur soutien et encouragement.

À tous ceux qui me sont chers.

RANIA.

Résumé

Information Centric Networking (ICN) a été proposé comme une nouvelle architecture pour l'Internet du futur, abordant de nombreux problèmes dans les réseaux IP d'aujourd'hui, tels que le processus de routage, le problème de l'évolutivité et des performances du partage de contenu. Dans notre travail, nous nous sommes concentrés sur certains problèmes de sécurité liés au routage, en nous concentrant sur l'opportunité de créer des attaques par déni de service distribué (DDoS) où les attaquants compromettent de nombreux ordinateurs tiers appelés zombies, et les utilisent pour lancer une attaque d'inondation d'intérêt (IFA).

C'est pourquoi nous avons étudié ce type d'attaque et les solutions proposées pour le combattre, comme les algorithmes Traceback, Poséidon et PAP (Producer-AssistedPushback). Nous avons ensuite proposé une version simplifiée de l'algorithme PAP et nous avons pu l'implémenter dans le simulateur CCN-lite pour simuler plusieurs scénarios d'attaque. Les résultats obtenus semblent prometteurs.

Mots clés : **ICN, DDoS, zombies, IFA, Traceback, Poséidon,PAP.**

Abstract

Information Centric Networking (ICN) has been proposed as a new architecture for the Internet of the future, addressing many issues in today's IP networks, such as the routing process, the problem of scalability and performance of content sharing. In our work, we have focused on certain routing-related security issues, specifically Distributed Denial of Service (DDoS) attacks where attackers compromise many third-party computers called zombies, and use them to launch an Interest Flooding Attack (IFA).

Therefore, we have studied this attack and the solutions proposed to fight against it, such as the Traceback, Poseidon and PAP (Producer-assisted Pushback) algorithm. We then proposed a simplified version of the PAP algorithm and we were able to implement it in the CCN-lite simulator to simulate several attack scenarios. The results obtained seem promising.

Keywords: ICN, DDoS, zombies, IFA, Traceback, Poseidon, PAP.

ملخص

تم اقتراح الشبكات المتمحورة حول المعلومات (ICN) كهيكل جديد للإنترنت في المستقبل ، ومعالجة العديد من المشكلات في شبكات IP الحالية ، مثل عملية التوجيه ، ومشكلة قابلية التوسع و فعالية مشاركة المحتوى. نركز في عملنا على بعض القضايا الأمنية ذات صلة بالتوجيه المتمحور حول المحتوى ، مثل فرصة إنشاء هجمات رفض الخدمة الموزعة أو هجمات DDoS (الحرمان الموزع للخدمة) حيث يقوم المهاجمون بخرق العديد من أجهزة الكمبيوتر من أطراف ثالثة ، تسمى الزومبي ويستخدمونها لشن هجوم إغراق الفائدة (IFA).

لذلك، قمنا بدراسة هذا النوع من الهجوم بالإضافة إلى الحلول المقترحة لمحاربة هذا الهجوم ، مثل خوارزمية Traceback و Poséidon و PAP (الرد بمساعدة المنتج). اقترحنا بعد ذلك إصدارًا مبسطًا من خوارزمية PAP التي تمكنا من تنفيذها في محاكي CCN-lite لاختبار عدة سيناريوهات هجوم. النتائج التي تم الحصول عليها تبدو واعدة.

الكلمات الرئيسية: ICN، DDoS، zombies، IFA، Traceback، PAP، Poseidon.

Table des Matières

Liste des Figures

Liste des Tableaux

Liste d'Acronymes

CCN :	Content Centric Network
CS :	Content Store
DDoS :	Distributed Denial Of Service
DNS :	Domaine Name System
DONA :	Data Oriented Network Architecture
Dos :	Denial Of Service
FIB :	PendingInterest Table
FIFA :	False Interest Flooding Attacks
FL :	Floride Liste
ICN:	Information Centric Network
IP:	Internet Protocol
INET :	Intelligent Network
IDE :	Integrated DevlopmentEnvironment
MAJ :	Mise à Jour
NACK :	NegativeAcknowledgment
NDN :	Named Data Networking
NDO :	Named Data Object
NetInf :	Network Information
OS :	Operating System
PARC :	Palo Alto Research Center
PFE :	Projet Fin Etude
PIT :	Forwarding Information Base
PAP :	Producer-AssistedPushback
PSIRP :	Publish Subscribe Internet Routing Paradigm
PURSUIT :	PublishSubscribe Internet Technology
SAIL :	Scalable and Adaptive Internet Solutions
SIA :	Satisfaction-basedInterest Acceptance
SP :	Satisfaction basedPushback
TCP :	Transmission Control Protocol
TTL :	Time To Live
URI:	Uniform Resource Identifier
WWW :	World Wide Web

Introduction Générale

1. Contexte

Le réseau centré sur l'information (Information Centric Network, ICN) est un nouveau paradigme de mise en réseau qui traite le contenu comme une entité de première classe. Il déplace le paradigme de mise en réseau du paradigme actuel centré sur l'hôte, où toutes les demandes de contenu sont adressées à un hôte identifié par son adresse IP, à un paradigme centré sur le contenu, qui dissocie les objets de contenu nommés des hôtes où ils se trouvent situé. En conséquence, le contenu nommé peut être stocké n'importe où sur le réseau, et chaque objet de contenu peut être adressé et demandé de manière unique. ICN a montré des possibilités pour résoudre plusieurs problèmes d'Internet, en particulier la distribution de contenu et la mobilité. Cependant, d'autres problèmes doivent être résolus afin de faire progresser cette architecture prometteuse notamment les problèmes de sécurité liés au contenu, à l'infrastructure ou aux terminaux. Il est donc crucial d'avoir une compréhension globale des attaques ICN, de leur classification et des solutions proposées. C'est dans ce contexte que s'inscrit notre projet de fin d'étude[1].

2. Problématique

L'architecture d'ICN permet de basculer vers la sécurisation de contenus au lieu de la sécurisation du chemin (l'infrastructure et des nœuds) des contenus comme dans Internet. En conséquence, de nouvelles attaques sont apparues avec ce nouveau modèle de sécurité, en plus des attaques héritées qui peuvent avoir un impact sur l'ICN. Par exemple, nous trouvons plusieurs attaques liées au routage centré contenu telles que les attaques des dénis de service distribuées ou DDoS (Distributed Denial of Service) où les attaquants pénètrent ou compromettent de nombreux ordinateurs de tiers, appelés zombies et les utilisent pour lancer une attaque Dos contre le réseau cible. Nous parlons ici des attaques de zombies qui consistent à utiliser une grande quantité de postes infectés pour envoyer une très grande quantité de paquets, dont la taille est relativement importante, en même temps, voire sur une longue période, dans l'intention de paralyser la réponse du serveur attaqué et le rendre indisponible. Pour lutter contre ces types d'attaque, il existe plusieurs algorithmes de contre-mesures dans la littérature comme l'algorithme de Traceback**[Hyj 13]** de Poséidon **[Amp 13]** de PAP (Producer-AssistedPushback)**[Zvj 18]**etc. Il est donc nécessaire d'étudier et d'évaluer ces solutions dans le cadre d'ICN afin d'arriver à la meilleure solution ou en proposer une autre solution (améliorée, voire nouvelle).

3. Objectifs

Étant une continuité de quatre travaux de PFE antérieurs **[Hak 17] [Raw 18] [Bof 20] [Drb 20]**, notre travail a pour objectif de développerune nouvelle solution contre les attaques zombies et la comparer avec les solutions déjà implémentées (Algorithme de Poséidon dans **[Raw 18]** et **[Bof 20]** et Algorithme de Traceback**[Hak 17]**). Cette nouvelle solution doit permettre de détecter les attaques de zombies et de réagir de manière efficace vers les attaquants.

4. Structure de mémoire

Le reste de ce mémoire est structuré en trois chapitres :

- Chapitre 1 : Où nous présentons l'architecture, les fonctionnalités et les projets des réseaux ICNs.

[1]Ce travail fait partie du projet intitulé "proposition d'un mécanisme de sécurité pour les réseaux ICN" initié en 2016/2017 par un groupe d'enseignants chercheurs au niveau de l'équipe SIIR (Sécurité de Systèmes Informatiques et Raisonnement) de laboratoire LRDSI (Laboratoire de Recherche pour le Développement des Systèmes Informatiques) de notre faculté.

- Chapitre 2 : Où nous exploitons l'aspect sécurité dans les CCN tout en détaillant les attaques de zombies, ses différentes approches de contre mesurent et leurs limites. Nous présentons aussi un état de l'art sur les principaux algorithmes permettant de se protéger contre les attaques de zombies.
- Chapitre 3 : Où nous expliquons notre démarche dans ce projet, notre proposition et notre implémentation.

Et enfin, nous terminons avec une conclusion générale et perspective où nous résumons notre travail réalisé tout au long de ce projet.

Chapitre I : Réseaux Centrés sur l'Information (Information Centric Networks, ICN)

1. Introduction

Internet est un réseau de commutation de datagrammes qui offre un service sans aucune garantie quant à l'acheminement des messages. Il est construit sur un modèle où les différents hôtes du réseau communiquent de bout en bout et est basé sur la pile protocolaire TCP/IP **[Srt 15].** Les utilisateurs étant principalement intéressés par le contenu et non par sa localisation, de nouvelles architectures orientées contenu ont été proposées pour un Internet du futur. Ces architectures appelées génériquement Information-Centric-Networks(ICN) se concentrent sur les données elles-mêmes. L'acheminement des messages n'est donc plus réalisé en fonction de l'adresse de l'hôte (adresse IP), mais en fonction du nom de la donnée. Ces nouvelles architectures sont conçues pour permettre la diffusion de contenu à large échelle, tout en ajoutant de nouvelles fonctionnalités dans le réseau.

Dans ce chapitre, nous allons présenter l'architecture ICN, ses fonctionnalités tout en comparant avec l'architecture actuelle de l'Internet. Ensuite, nous décrirons les principaux projets ICN sur lesquels nous allons travailler.

2. Fonctionnalités

ICN (Information Centric Networking) est une norme de mise en réseau moderne qui ne fonctionne principalement que sur l'extraction de contenu à partir d'un réseau sans tenir compte de l'emplacement de stockage ou de la façon dont le contenu est représenté. L'approche de ICN considère que le contenu nommé comme l'élément central du réseau et supporte nativement la mise en cache dans le réseau **[Bpj 11]**.

Comme montré la figure I.1, pour la première demande de contenu, on doit parcourir tout le chemin jusqu'à arriver au serveur original pour trouver la donnée de la même manière comme dans le réseau IP. Ensuite, pour une autre demande, on peut trouver le contenu dans le cache des nœuds sen utilisant seulement son nom (étant un identifiant unique).

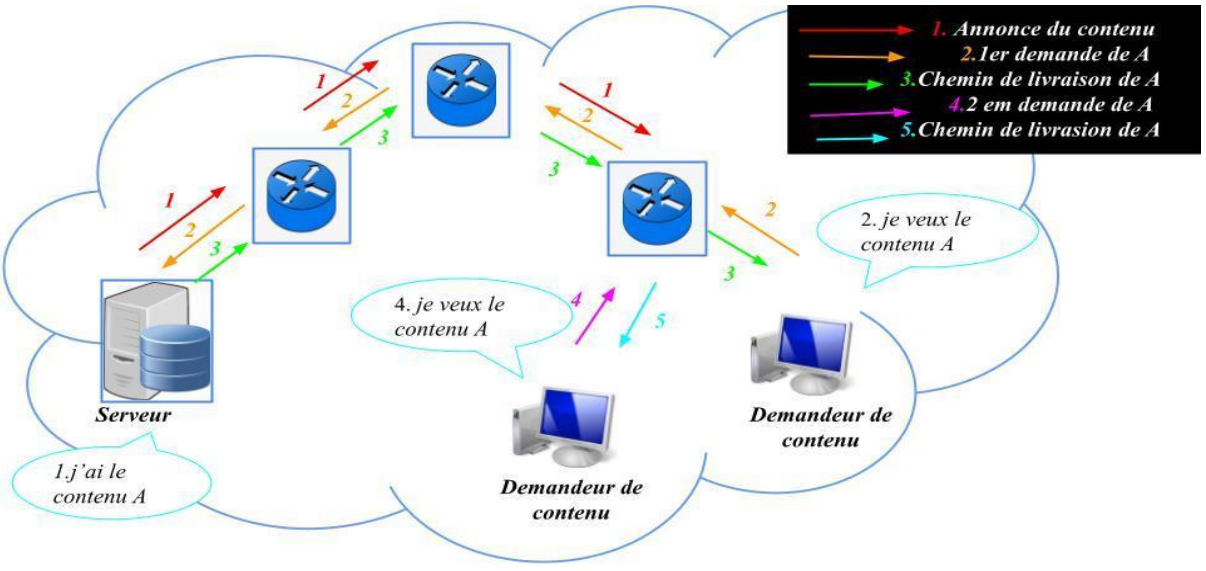

Figure I.1 – Le modèle de communication dans l'approche ICN [Bal 16].

Ce modèle de communication se base sur les fonctionnalités d'ICN suivantes : le nommage de contenu, la mise en cache distribuée du contenu, le routage basé sur le nom, la sécurité des données et la mobilité des utilisateurs. Chacune de ces fonctionnalités seront décrite par la suite.

2.1. Nommage de contenu

L'ICN est indépendant de l'emplacement, de la méthode de stockage, du programme d'application et de la méthode de transport. Cela signifie qu'un contenu conserve son nom, et donc son identité, quel que soit son emplacement et quelle que soit la manière dont il est copié, stocké et communiqué. Par exemple, tout nœud détenant une copie de contenu peut le fournir à un demandeur. Du point de vue du réseau, ces contenus peuvent être considérés comme des blocs de données nommés sans sémantique, mais certaines conceptions ICN ont un modèle d'abstraction de l'information y compris plusieurs représentations pour le même contenu **[Msr 12]**.

Le nom du contenu est l'élément clé de l'ICN, qui identifie de manière unique le contenu lui-même. Il doit être compact, persistant et pouvoir valider les contenus. Le schéma de nommage utilisé doit être évolutif et doit permettre l'agrégation de noms. Quatre principaux types de schémas de nommage ont été proposés dans l'ICN :

- **Nommages hiérarchiques :** Ils se composent de plusieurs composants pour identifier le contenu et décrire l'application/les services. Il a une structure similaire aux identificateurs de ressources uniformes (URI) actuels **[Sbm 18]**.
- **Nommages plats :** Ils sont généralement obtenus par le biais d'algorithmes de hachage appliqués au contenu. Il n'y a aucune structure dans le nom **[Bda 12]**.
- **Nommages basés sur la valeur d'attribut :** Le schéma de nommage basé sur la valeur d'attribut possède une collection d'attributs, où chaque attribut a un nom, un type et un ensemble de valeurs possibles (date/heure de création, type de contenu, emplacement, version, etc.). Collectivement, ils représentent un contenu unique et ses propriétés **[Osg 17]**.
- **Nommages hybrides :** Un schéma de dénomination hybride combine au moins deux des schémas décrits précédemment ou tous. Étant donné que les schémas de dénomination de base présentent des inconvénients différents **[Bof 20]**.

Le tableau suivant donne un aperçu des avantages et inconvénients de chaque type:

Type	**Exemple**	***Avantages***	***Inconvénients***
Nommage Hiérarchique	/bit.edu.cn/cs/recherche /2019/ Auteurs/titre.pdf	Évolutivité du réseau améliorée. Informations sémantiques incluses. Agrégation de noms possibles.	Noms plus longs. Normalisation requise.
Nommage Plat	Ni : //bi.edu.cn /sha -256 (titre)	Noms de longueur fixe. Facile à générer.	Aucune structure et sémantique du nom. Difficile pour le contenu dynamique. Pas de support d'agrégation.

Nommage attribut basé sur la valeur	Titre<str>:'Titre' Auteurs<list>[Aut_1,Aut_2] Journal<str>: 'Journal' Year<int> : 2019	Processus de recherche facile. Contenu additionnel informations disponibles.	Mêmes attributs pour différents contenus.
Nommage hybride	/bit.edu.cn/cs/ Year<int>: 2019, Auteurs<list>[Aut_1,Aut_2] /sha -256(titre)	Combiner différentes fonctionnalités.	Gestioncomplexe.

Tableau I.1–Types de schémas de nommage ICN [Bof 20].

2.2. Mise en cache

Tous les nœuds contiennent potentiellement des caches **[Med 17],** y compris des nœuds dans des réseaux d'infrastructure gérés par les opérateurs et des réseaux domestiques gérés par les utilisateurs, ainsi que des terminaux mobiles. Les requêtes de demande de contenu peuvent être satisfaites par n'importe quel nœud qui a une copie dans son cache. La mise en cache est publique.

Comme le montre la figure I.2, le consommateur 2 envoie une demande pour récupérer le contenu nommé X. Lors de l'envoi de la réponse, le nœud 2 et le nœud 6 stockent une copie du contenu. Ensuite, lorsque le consommateur 1 ou 3 demandes le même contenu X, la demande est satisfaite par les nœuds 2 ou 6 respectivement.

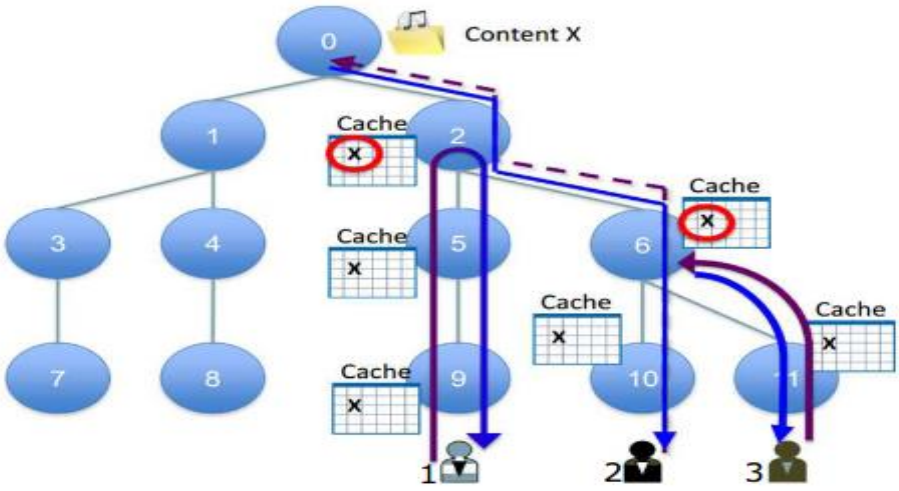

Figure I.2– mise en cache en réseau dans ICN[Med 17].

2.3. Routage

Dans ICN, le routage est basé sur le contenu, ainsi l'utilisateur ne montre que l'intérêt, puis le réseau s'intéresse à trouver de bons chemins pour atteindre la destination souhaitée. Une fois l'intérêt récupéré, le contenu est livré en parcourant le chemin inverse. Le routage ICN est principalement classé en deux types **[Med 17]** :

- **Routage par résolution du nom :**Dans cette catégorie, le client envoie une demande de contenu et le système chargé de distribution de contenu associe le nom du contenu à son localisateur.

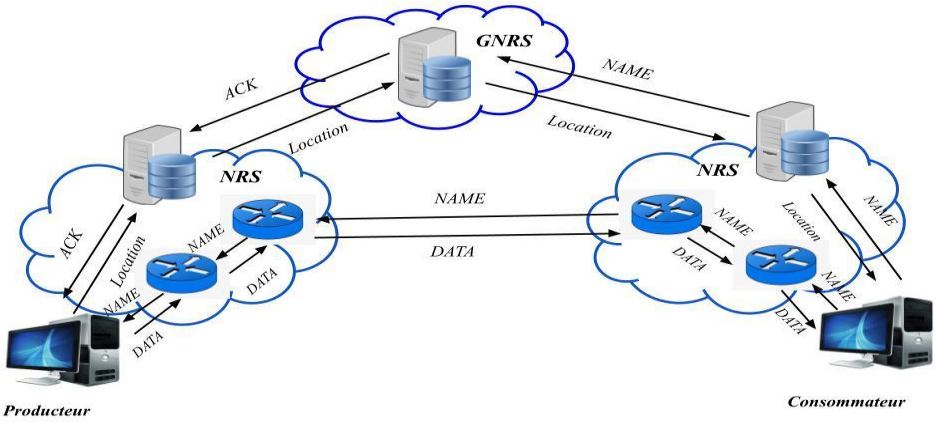

Figure I.3– Approche de Routage par résolution nom[Med 17].

- **Routage par nom :** Cette approche repose sur la hiérarchie des noms de contenu pour acheminer la demande vers le fournisseur. Lorsque la demande est envoyée, chaque routeur doit être conscient de certaines informations de routage.

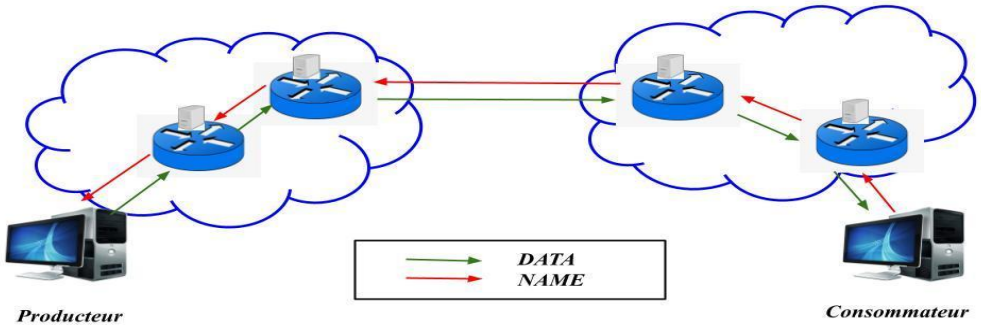

Figure I.4– Approche de Routage par nom[Med 17].

Dans le tableau suivant, on compare entre ces deux approches selon plusieurs points :

	Routage par nom	**Routage parrésolution de nom**
Nombres des étapes	1 seul	2
Acheminement des demandes	Directement au fournisseur.	Pas directement, par l'intermédiaire du NRS (Système de résolution de noms).
Récupération de Localisation du contenu	Non	Oui
Schéma de Nommage	Hiérarchique	N'importe
Envoie des données	A travers le chemin inverse.	A travers le NRS directement.
Utilisation du préfixe	Oui	Non

Tableau I.2–Comparaison entre approche de Routage par nom et approche de Routage par résolution nom.

2.4. Sécurité des données

L'approche de sécurit**é** dans ICN n'est pas liée à un nœud spécifié ou à un emplacement bien définie. Il faut sécuriser le contenue lui-même et non pas via des connexions de communication sécurisées comme IP.ICN fournit une garantie de sécurité auto-protégée via des contenus cryptés et auto certifiés, où seuls les utilisateurs autorisés peuvent déchiffrer les contenus. Il est toujours basé sur la sécurité de contenu directement plutôt que d'assumer la confiance des utilisateurs ou sécuriser les canaux de communication qui la délivrent. Cette liaison consiste généralement en un hachage de la clé privée du fournisseur de contenu au nom de l'objet de contenue. Cela permet l'authentification du fournisseur de contenu en créant une signature de l'objet de données à l'aide de la clé privée du fournisseur, qui peut être vérifiée avec ses clés publiques auto-certifiées sont envoyées sous forme de métadonnées aux récepteurs avec l'objet de données. La confidentialité et l'intégrité du contenu sont également garanties par un cryptage à clé publique. Un défi important pour l'auto-certification du contenu est de gérer la révocation des objets lorsque la clé publique est compromise ou que les données sont mises à jour **[Bal 16]**. L'échec de ces mécanismes de sécurité ouvre plusieurs attaques comme les attaques Dos en injectant du contenu contaminé dans le réseau **[Fab 13]**.

2.5. Mobilité des utilisateurs

Étant donné qu'ICN fonctionne sur un modèle non-connecté (il n'y a pas de connexion établie dans un réseau ICN), la mobilité des utilisateurs ne modifie pas le comportement des réseaux ICN. Leurs demandes, issues de différents endroits à différents moments, sont traitées indépendamment par les réseaux ICN, chacune comme une requête unique **[Wei 14]**. Au fait, la mobilité des producteurs et des consommateurs est indépendante dans ICN, donc si le producteur change son emplacement, le contenu est toujours accessible sans perturbation perceptible dans le réseau ICN. De plus, si le consommateur (client) se déplace, il faut redéfinir son intérêt pour un objet de données à partir du nouvel emplacement. Ainsi, la mobilité de l'ICN doit travailler en collaboration avec le routage ICN pour découvrir un objet de donnée où il est identique à celui qui est transmis**.**

3. Comparaison avec Internet

De même qu'ICN, les fonctionnalités citées précédemment sont assurées par Internet mais de manière différente :

- **Nommage des contenus :** Un internaute préfère, pour des raisons mnémotechniques, désigner le nom du contenu que l'adresse IP de la machine qui le contient **[Eli 17].** Par exemple,"www.wikipedia.fr" est plus facile à mémoriser que l'adresse IP "91.198.174.152", associée à ce serveur web. Lorsqu'un utilisateur effectue une requête web vers un nom de domaine, celui-ci sera converti en une adresse IP via le système DNS (Domaine Name System) **[Csc 02]**. Ce système de nommage garantit l'unicité des noms. Il est hiérarchique, sous forme d'arbre de nommage, tout en étant relatif à son emplacement.
- **Mise en cache :** La gestion de cache dans l'internet est réalisée par des applications (au niveau de l'utilisateur) permettant de choisir un algorithme donné pour diviser les données entre plusieurs serveurs spécifiques appelés souvent des caches. De plus, le même cache peut être utilisé par plusieurs applications.

- **Routage :** Les protocoles de routage dans Internet sont basés sur les adresses IPs des paquets permettant d'acheminer les datagrammes vers leurs destinations. Le routage est basé sur l'emplacement des hôtes contrairement au routage ICN qui est basé sur le nom de contenu.
- **Sécurité :** Les mécanismes de sécurité dans Internet sont intégrés dans les protocoles des couches IP (exemple, sécurité de la couche transport) ou dans les dispositifs de réseau supplémentaires (exemple, pare-feu ou systèmes de détection d'intrusion). Il n'y a pas de soutien inhérent à la sécurité, car les adresses IP ne fournissent pas de renseignements personnels sur les identités (elles peuvent être usurpées facilement) **[Paw 17]**.
- **Mobilité :** Dans Internet, les appareils mobiles doivent changer leur point d'accès lorsqu'ils sont transportés, ce qui les oblige à changer leur adresse IP. La modification de l'adresse IP interrompt la session de communication et nécessite le rétablissement d'une nouvelle session. Pire encore, lorsque le point final qui retient les données est mobile, le changement de son adresse le rend inaccessible par d'autres hôtes car il n'y a pas de mécanisme pour indiquer son adresse actuelle. En effet, ces approches sont superposées à l'architecture réseau actuelle. Avec une croissance aussi rapide du contenu et des utilisateurs simultanément, les modifications ou solutions incrémentielles de l'architecture Internet actuelle résistent difficilement à l'évolution d'Internet **[Bal 16].**

En résumé, le tableau suivant récapitule la différence entre ICN et Internet en termes de chaque fonctionnalité :

	Internet	**ICN**
Nommage	Relatif à l'emplacement de l'hôte.	Relatif au contenu, indépendant de son emplacement.
Mise en cache	Dans des serveurs spécifiques.	Dans n'importe quel nœud du réseau.
Routage	Entre les hôtes en utilisant des adresses IP.	Entre un demandeur et n'importe quel nœud du réseau détenant une copie du contenue, en utilisant le nom du contenu.
Sécurité	Intégrée dans les protocoles IP ou dans les dispositifs réseaux supplémentaires.	Intégrée au contenu, à l'infrastructure etaux terminaux finaux.
Mobilité	Des hôtes interrompent la session de communication et les rend inaccessible par d'autres hôtes ~~ge~~stion explicite	Gestion implicite ~~D~~es consommateurs permettent de reconfigurer l'emplacement de leur réseau sans interrompre la connectivité, tandis que la mobilité des producteurs permet aux sources de se déplacer sans perturber la disponibilité du contenu.

Tableau I. 3–Internet versus ICN [Ehm 15].

4. Projets de l'ICN

Les ICNs ont été explorée par de nombreux projets comme le montre le tableau II.2. Ce dernier compare les différents projets ICN disponibles dans les référentiels open source publics. La comparaison est basée sur la disponibilité de la littérature académique telle que le dernier article publié, le nombre de citations dans Google Scholar et la disponibilité de site web officiel.

Projet	Date	Cité en Google Schollar	Dernier article	Site Web
DONA	2007	1246	2007	ND[2]
AsiaFI	2007	3	2007	http://www.asiafi.net/
COMET	Jan 2010-Dec2012	10	2013	http://www.comet-project.org/
ANR Connect	Jan2011-Dec 2012	ND	ND	http://www.anr-connect.org/
Convergence	Jun 2010-Fev 2013	105	2013	http://www.ict-convergence.eu/
CCN	2009	2665	2016	http://www.ccnx.org/
NDN	Sep2010-Aoû2013	2519	2017	http://www.named-data.net/
NetInf	2010	ND	ND	https://sail-project.eu/about-sail/netinf/
PSIRP	Jan2008-Jun2010	ND	ND	http://www.psirp.org/
PURSUIT	Sep2010-Fev 2013	ND	2011	http://www.fp7-pursuit.eu/
4WARD	Jan 2008-Jun 2010	35	2010	http://www.4ward-project.eu/
SAIL	Aoû2010-Jan 2013	ND	2013	http://www.sail-project.eu/
Mobility First	Sep2010-Sep 2013	60	2015	http://mobilityfirst.winlab.rutgers.edu/

Tableau I.4– Les projets ICNs [Drb 20].

Bien qu'ils soient encore en cours de développement, ces architectures ICN abordent l'ensemble de fonctionnalités clés citées précédemment, mais avec différentes approches. Dans notre travail, nous nous sommes intéressés par ces deux projets : CCN et NDN, qui reposent sur un grand nombre de chercheurs académiques ainsi que industriels.

4.1. Projet CCN

Le réseau contré sur contenu ou CCN (Content-Centric Network) a été proposé par Palo Alto Research Center (PARC) en 2009 **[Ham 19]**. C'est l'un des projets de recherche ICN les plus attractifs. Il propose une architecture de mise en réseau purement basée sur le contenu, depuis le nommage du contenu, le routage basé sur le contenu, la découverte et la livraison de contenu, jusqu'à la sécurité basée sur le contenu et la mise en cache en réseau.

Au fait, la récupération de contenu dans CCN peut être principalement divisée en deux parties :

- **La découverte de contenu :** qui est liée à la façon dont un contenu est nommé, comment il est publié et comment un nœud CCN l'adresse.

[2] Non Défini

- **La livraison de contenu :** qui définit le protocole de routage CCN qui concerne la façon dont un fournisseur de contenu propage son contenu dans le réseau, comment un nœud CCN achemine les demandes des utilisateurs finaux vers les meilleures sources de contenu et comment un nœud CCN fournit le contenu aux utilisateurs finaux.

4.1.1. Nommage CCN

Le nom du contenu dans CCN est conçu comme une structure hiérarchique selon un ordre préfixe-suffixe. La figure suivante montreun exemple de nom de contenu CCN : ccnx : / parc.com / video / widget1 / version2 / chunk2

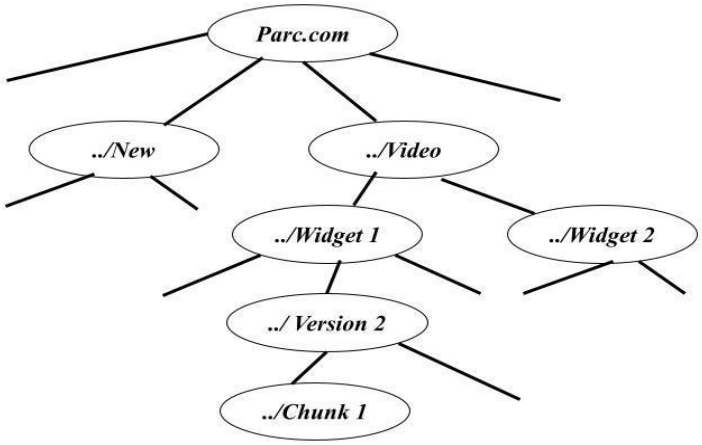

Figure I. 5– Le nom de CCN.

4.1.2. Paquets CCN

Toutes les communications en CCN sont effectuées à l'aide de deux types de paquets distincts (figure I.6)**[Ham 19]**:

- Paquet d'intérêt pour l'envoi des requêtes.Il porte les éléments suivants :
  - Nom du contenu.
  - Sélecteurs : ce sont des éléments optionnels utilisés pour découvrir et sélectionner les données qui correspondent aux besoins de l'application.
  - Nonce : il s'agit d'un nombre aléatoire utilisé par un routeur pour détecter si le paquet d'intérêt est dupliqué ou non.

- Paquet des données pour la réponse au contenu correspondant.il contient les champs suivants :

  - Nom du contenu,
  - Signature et infos signées : selon les informations signées, les utilisateurs peuvent obtenir la clé publique de l'éditeur de contenu pour vérifier les données reçues avec la règle spécifiée dans le champ signature et décider d'accepter ou non les données.
  - Données : ce sont les données qui correspondent au nom du contenu dans le paquet d'intérêt reçu.

Les deux types de paquets portent un Nom, qui identifie de manière unique une donnée pouvant être transportée dans un paquet de données. Un consommateur met le nom des données souhaitées dans un paquet d'intérêt et l'envoie au réseau. Les routeurs utilisent ce nom pour transmettre l'intérêt au producteur de données, et le paquet de données dont le nom correspond le mieux à l'intérêt est renvoyé au consommateur. Tous les paquets de données portent une signature qui lie le nom aux données. En effet, les producteurs sécurisent le contenu en signant de manière cryptographique avec sa clé secrète chaque élément de données, ce qui garantit l'intégrité des données. En conséquence, les consommateurs peuvent déterminer facilement les données**[Zvj 18]**.

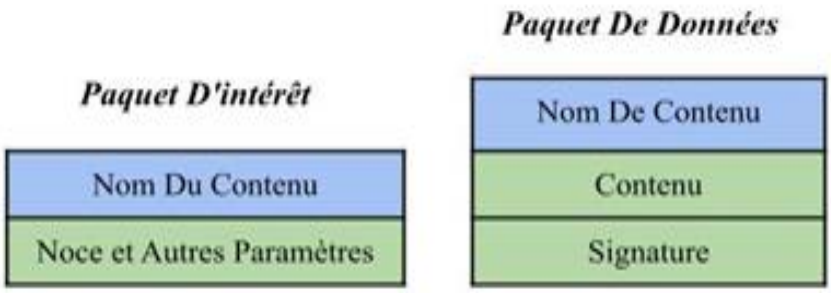

Figure I. 6– Paquets CCN [Zvj 18].

4.1.3. RoutageCCN

Le réseau CCN est composé des nœuds CCN. Un nœud CCN peut être : un routeur, un commutateur ou même un terminal d'utilisateur final. CCN utilise le concept de « face » au lieu de la notion d'interface qui est familière à IP. La face CCN est un concept général relatif, définissant non seulement l'interface physique qui relie l'équipement mais également la connexion interne entre les applications logicielles supérieures et les couches matérielles inférieures. La FigureI.7illustrel'architecture d'un nœud CCN.

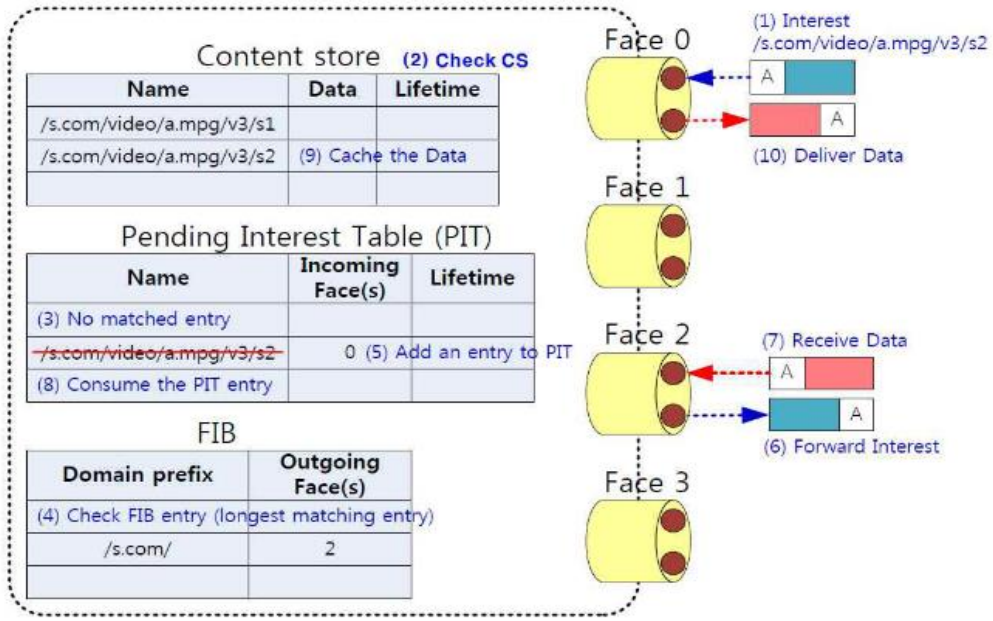

Figure I.7– Architecture d'un nœud CCN [Mam 14].

Chaque nœud CCN est composé de trois éléments principaux :

- **Tableau des intérêts en attente (PendingInterest Table, PIT)** qui stocke les intérêts actuellement insatisfaits et leurs interfaces entrantes/sortantes. Chaque entrée PIT enregistre un paquet d'intérêt qui a été transmis, en attendant le retour du paquet de données. L'entrée enregistre le nom, l'(les) interface(s) entrante(s) de l'Intérêt(s) et l'(les) interface(s) sortante(s) vers laquelle l'Intérêt aété transmis. Le tableau I.5donne un exemple d'un PIT d'un CCN, qui consiste à conserver le chemin inverse du paquet d'intérêt propagé afin que le paquet de données renvoyé puisse suivre ces « fils d'Ariane » en aval jusqu'aux consommateurs.

CCN PendingInterest Table	
Content names	**Faces**
Ccnx:/youtube.com/news/baby_born/	face308,face321
Ccnx:/google.fr/	Face103
Ccnx:/orange.fr/news/meteo/	Face201
…	…

Tableau I.5– CCN Forwarding Information Base.

- **Forwarding Information Base (FIB)** est à peu près similaire au FIB d'un routeur IP, sauf qu'il contient des préfixes de nom au lieu de préfixes d'adresse IP, et il peut afficher

plusieurs interfaces pour un préfixe de nom donné. De plus, chaque routeur CCN dispose d'un module de stratégie qui prend les décisions de transmission pour chaque paquet d'intérêt.

- **Content Store (CS)** est un cache ou un tampon défini dans chaque nœud CCN. Il peut mettre en cache les objets de contenu qui passent et fournir la fonction de mise en cache en réseau dans le but de minimiser les demandes de bande passante et de latence du réseau, ainsi que la charge du serveur.

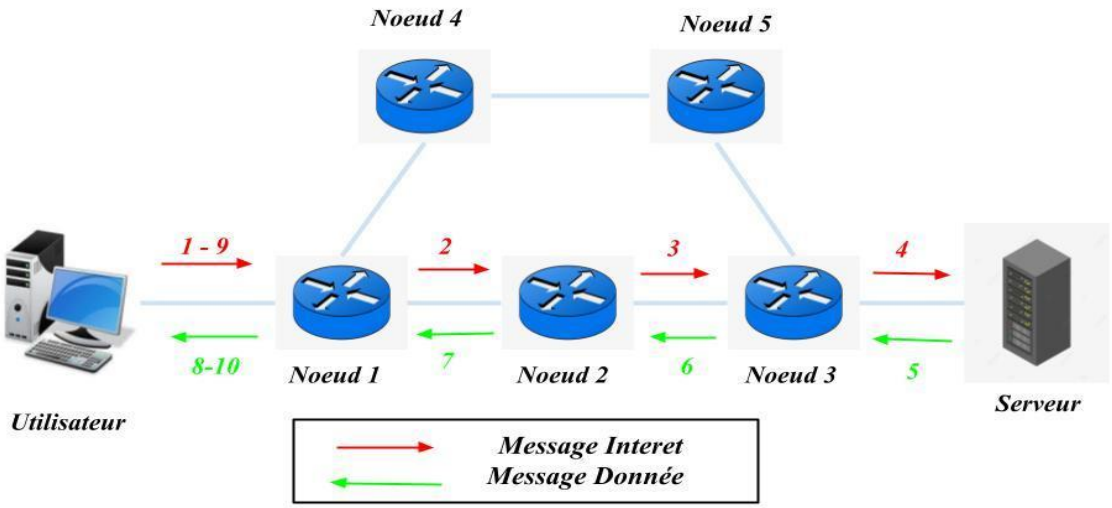

Figure I.8–routage CCN [Aub 17].

Les grands principes du fonctionnement du routage CCN sont représentés sur la figure I.8. Lorsqu'un utilisateur veut accéder à un contenu, il publie/envoie un message d'intérêt « Interest » (message 1). Quand le Noeud1 reçoit ce message d'intérêt, plusieurs cas de figure sont possibles :

(i) Soit il connaît le prochain nœud pour transmettre le message car une entrée est déjà dans sa FIB, auquel cas il lui transmet donc la requête (message 2) ;

(ii) Soit-il ne possède pas la règle pour atteindre ce contenu et il va transmettre cette requête sur toutes ses autres interfaces (flooding).

Supposant le premier cas (i), le Noeud2 transmet à son tour ce message au Noeud3 (message 3) qui la transfère au serveur (message 4). Ce dernier répond grâce à un message de données « Data » transportant la donnée. Ce message de données va utiliser le chemin inverse du message d'intérêt pour arriver à l'utilisateur (messages 5 à 8), car chaque nœud a conservé cette entrée dans sa PIT. Tous les nœuds ayant retransmis le message « Data » vont pouvoir conserver une copie locale dans leurs mémoires cache (CS) et ce, en fonction des mécanismes de cache qui ont été instanciés sur chacun des nœuds. Ainsi, lorsqu'un utilisateur émet à nouveau une requête pour ce même contenu (message 9), le Noeud1 va pouvoir utiliser sa copie locale pour répondre (message 10). C'est ce principe qui permet au contenu d'être propagé au plus près des utilisateurs, de réduire le délai d'accès au contenu et de limiter la portée d'une requête afin de ne pas surcharger le réseau **[Aub 17]**.

4.2. Projet NDN

Le réseau des données nommés ou NDN (Named Data Network) est l'un des quatre projets financés dans le cadre Le programme Future Internet Architecture de la NSF **[Aub 17]**. Il partage les mêmes principes de conception que CCN, comme la hiérarchie du schéma de nommage, la mise en cache de contenu et le routage par nom. La principale différence réside dans la structure hiérarchique du nom de contenu qui est formée par trois parties (Figure I.9). Chaque partie comporte un ou plusieurs composants lisibles, délimités par le caractère '/'. La première partie des noms fournit des informations de routage globales. La deuxième partie contient des informations de routage organisationnelles et la dernière partie donne des informations sur la version et le numéro de segment.

Figure I.9– Structure d'un nom de contenu dans NDN.

La notion de segmentation est utilisée lorsque les données sont volumineuses ; elle permet de découper ces données en plusieurs parties. Chacune de ces parties est identifiée par un numéro de segment et elle peut être récupérée individuellement. La notion de version permet d'effectuer des changements à un contenu préalablement publié et de le publier sous un nouveau nom. En effet, une fois publié, un contenu ne peut pas être modifié et republié avec le même nom puisqu'il peut être déjà dispersé dans plusieurs caches. La Figure représente un exemple de la structure de nom dans NDN, /univ-Blida représente la partie routable. /master/Informatique/SIR & SSI est la partie organisationnelle et /1/1 indique qu'il s'agit de la première version du premier segment de la thèse. Les noms NDN sont lisibles et sont souvent significatifs, portant des informations sur les contenus associés.

5. Conclusion

ICN est une nouvelle architecture d'internet qui remplace IP. Elle repose sur le nommage indépendant de l'emplacement, de la mise en cache dans le réseau, et le routage basé sur le nom. Plusieurs projets ICN en cours de développement abordent l'ensemble de ces fonctionnalités de différentes manières. Dans ce chapitre, nous avons décrit les deux projets qui nous allons utiliser dans notre travail : CCN et NDN. Dans le chapitre suivant, nous allons étudier les problèmes de sécurité dans les ICN et leurs solutions, notamment ceux liés au routage basé sur le nom.

Chapitre II: Attaques des Zombies dans les CCN

1. Introduction

Dans ICN, la sécurisation du contenu lui-même est bien plus importante que la sécurisation de l'infrastructure ou des terminaux. Pour atteindre les objectifs de sécurité dans ce nouveau paradigme, il est crucial d'avoir une compréhension globale des attaques ICN, de leur classification et des solutions proposées. Cela fait l'objet de ce chapitre, dans lequel nous commençons par présenter une taxonomie des attaques ICN. Ensuite, nous nous concentrons sur les attaques liées au routage notamment les attaques de DDoS en décrivant les attaques par inondation d'intérêts (IFA). Enfin, nous présentons les approches de contres mesures des attaques IFA et les principaux algorithmes.

2. Classification des attaques dans ICN

ICN a de nombreux problèmes de sécurité à résoudre. Il existe de nouveaux types d'attaques dans ICN qui n'ont pas eu lieu auparavant ou qui n'ont pas eu d'impact significatif dans d'autres environnements. Dans la littérature, plusieurs travaux ont essayé de recenser les différentes attaques ICN et de proposer une classification qui classe les attaques ICN selon quatre catégories **[Ehm 15]**, comme la montre dans la Figure II.1. 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.

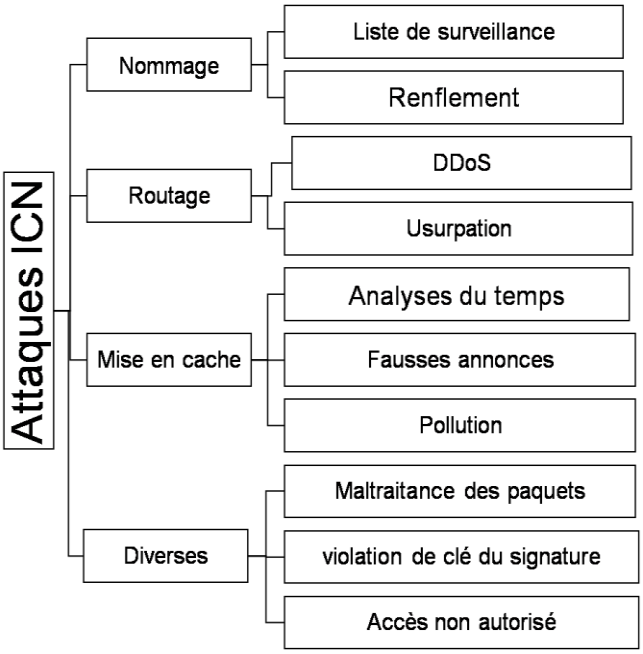

Figure II. 1– Taxonomie des attaques ICN [Ehm 15].

a. **Attaques liées au nommage** : Les architectures ICN sont confrontées à une plus grande menace en ce qui concerne la confidentialité car les demandes de contenu sont visibles sur le réseau. 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, on trouve :
   - **Liste de surveillance (watchlist) :** L'attaque surveille le réseau pour filtrer et enregistrer les requêtes et/ou le contenu en fonction de sa liste prédéfinie.
   - **Renflement (Sniffing) : L'attaquant** surveille le réseau pour vérifier si les données doivent être signalées afin de les filtrer ou de les supprimer.

b. Attaques liées au routage

Les attaques liées au routage ont un impact sur les éléments suivants **[Hak 17]**:

- **Déni de service :** Les Dos peuvent survenir en raison de nombreuses attaques dans cette catégorie, telles que l'envoi de nombreuses demandes de contenu indisponible ou d'une seule source, de blocage mobile, d'inondation (flooding), de Hijacking et de Timing. En conséquence, les temporisations intermédiaires suppriment les requêtes avec les délais expirés, ce qui peut entraîner des Dos ou au moins de longs délais.
- **L'exténuation des ressources :** Il existe de nombreuses sources d'exténuation des ressources dans l'infrastructure ICN qui proviennent d'un mauvais usage ou d'un trafic incontrôlé, comme l'envoi d'un grand nombre de requêtes et d'inondations.
- **L'infiltration du chemin :** Dans ICN, des copies de contenu sont généralement distribuées à de nombreux sites non approuvés, et il est donc difficile d'authentifier des origines valides pour le contenu. Le Hijacking et l'interception sont les principales sources d'infiltration de chemin dans l'ICN, car les attaquants peuvent annoncer des routes non valides et les réclamer comme faisant confiance.
- **Intimité :** La violation de la vie privée (privacy) dans l'attaque d'interception donne à l'attaquant un accès non autorisé aux demandes de l'utilisateur, surtout lorsque l'attaquant est topologique ment proche ou sur la route vers l'utilisateur.
- **Usurpation :** (spoofing) Dans ICN, l'intérêt est utilisé pour exprimer les demandes des utilisateurs tandis que le paquet de données contient un NDO (Named Data Object) .Cependant, étant donné que le routeur CCN n'accepte pas les messages non sollicités Données, l'envoi de Données sans précédent L'intérêt n'entraîne pratiquement aucun Endommager le routeur.

De ce fait, les attaques dans cette catégorie peuvent être classées dans les attaques de déni de service distribué (DDoS) et les attaques de Spoofing (attaques de falsification ou d'usurpation). Certains attaques de spoofing comme jamming et la synchronisation visent à échouer à la cohérence entre les états de données distribuées, ce qui peut entraîner des flux de trafic indésirables et / ou un déni de service. D'autres attaques de DDoS, 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. Dans notre travail, nous nous concentrons sur les attaques de DDoS qui font l'objet de la section suivante.

c. Attaques liées à la mise en cache

La mise en cache est l'un des composants importants d'ICN, car les performances de l'infrastructure ICN sont basées sur une mise en cache pilotée par le récepteur qui vise à fournir la copie disponible la plus proche à un utilisateur. Par conséquent, ICN est vulnérable à toutes les opérations qui polluent ou corrompent le système de mise en cache. De ce fait, on trouve les attaques suivantes :

- **Attaque par fausses annonces :** un utilisateur demande du contenu ICN nommé (x), tandis qu'un attaquant envoie un grand nombre de mises à jour pour le contenu, y compris(x), avec une fréquence qui dépasse le temps de convergence du routage des demandes de contenu. Les

routeurs ICN ne pourront pas mettre à jour sa table de routage à cause de ces fausses annonces, ce qui entraîne une absence ou une fausse récupération de contenu.

- **Analyse du temps :** Dans cette attaque, un adversaire mesure la différence de temps entre le temps de demande de réponse pour le contenu mis en cache et non mis en cache. Pour arriver si un utilisateur a déjà demandé le même contenu que l'adversaire ou pas.Cette attaque touche la vie privée de l'utilisateur car l'adversaire peut obtenir des informations sur cet utilisateur proche.
- **Pollution :** Le but de cette attaque est de modifier de manière malveillante la localité du contenu du cache. Cette attaque peut être mise en œuvre en émettant des demandes de (nouveau) contenu non populaire. Une attaque de fausse localité vise malicieusement augmenter la popularité d'une fraction (généralement petite) de contenu disponible. L'attaquant met en œuvre ce virement attaque en émettant un grand nombre de demandes pour les mêmes pièces du contenu.

d. Attaques diverses : Les menaces dans cette catégorie visent à dégrader certains services ICN et à permettre à un attaquant de faire un accès non autorisé. Ces attaques entraînent une distribution de données insuffisante ou erronée. Il ya plusieurs attaques dans cette catégorie comme : Maltraitance des paquets (PacketMistreatment), violation de clé de signature (BreachingSigner's Key) et L'accès non autorisé.

3. Attaques de DDoS

En sécurité informatique, un zombie est un ordinateur connecté à un réseau qui a été compromis par un pirate, un virus ou un cheval de Troie. Il peut être utilisé à distance pour des tâches malveillantes d'où l'appellation des attaques de Zombies **[Gup 17]**. Dans ces attaques, la plupart des propriétaires d'ordinateurs zombies ne réalisent pas que leur système est infecté et employé dans des tâches malveillantes, dans le but est de saturer le réseau en envoyant un grand nombre des requêtes malveillantes pour rendre le service indisponibles (Figure II.2).

Ces attaques de réseau distribué sont souvent appelées attaques par déni de service distribué (DDoS). Elles consistent à envoyer un très grand nombre des données (requêtes, intérêts ou autre) à la victime pour surcharger le réseau et/ou interrompre le réseau **[Xue 17]** Dans CCN, les attaques DDoS liés au routage ciblent les routeurs intermédiaires et/ou les fournisseurs de contenu dans le but de dégrader les performances des routeurs dans le réseau et les rendre incapable de faire le routage.

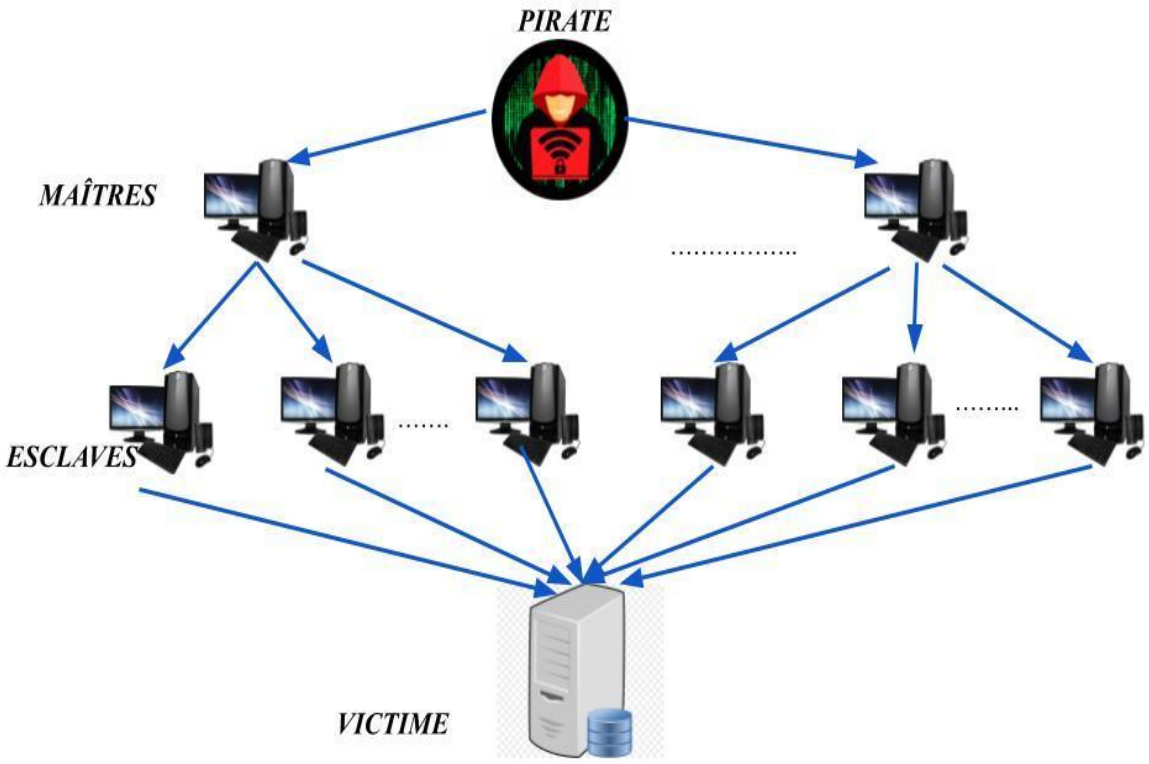

Figure II.2– Structure d'une attaque zombie[Tyg 17].

3.1. Types des attaques DDoS

Selon la classification des attaques ICNs citée auparavant, les attaques DDoS peuvent être classées en deux types (Figure II.3). Le premier type d'attaque vise l'épuisement des ressources comme les attaques en blocs d'infrastructure, de source, de blocage mobile et d'inondation(flooding), tant dis que le deuxième type s'agit des attaques temporelles :

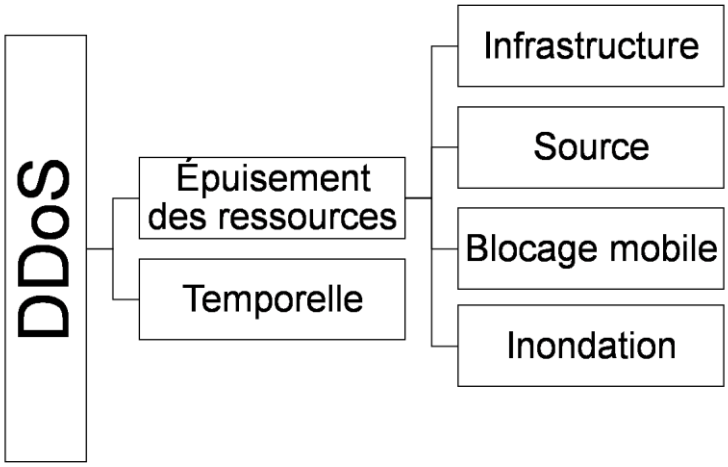

Figure II.3– Taxonomie des attaques DDoS [Ehm 15].

- **Infrastructure :** Un attaquant envoie un grand nombre de demandes de contenu disponible/ indisponible. Lorsque les architectures ICN tentent de trouver la copie la plus proche du meilleur emplacement disponible, ces requêtes empruntent différentes routes vers la source provoquant des conditions de surcharge. Si le nombre de ces demandes est significativement élevé, cela entraîne un déni de service.Cette attaque peut être amplifiée, car les utilisateurs

réguliers envoient des demandes de retransmission après un temps spécifié. Cette menace peut être atténuée car les mécanismes de routage dans ICN tentent de se diriger vers plusieurs emplacements.

- **Source :** Un attaquant envoie un grand nombre de requêtes à une source de contenu spécifique pour dégrader des performances. En conséquence, cette attaque augmente le temps de réponse de livraison de contenu pour cette source de contenu ou son routeur d'accès. De plus, l'attaque peut réduire le taux de retour des données et affecter les demandes de tous les nœuds dans les chemins vers les récepteurs.
- **Blocage mobile :**Un attaquant mobile peut surcharger une région en traversant des réseaux voisins sur des chemins circulaires tout en envoyant un nombre important de demandes de contenu. L'attaquant vise à surcharger les routeurs d'accès mobiles pour qu'il dépasse le délai d'exécution de l'état qui entraîne un blocage des réseaux régionaux. La retransmission des demandes fait partie de l'aspect de la mobilité dans un environnement ICN qui ajoute de la difficulté à détecter cette attaque.
- **Temporelle :**Il s'agit d'augmenter le délai d'attente de la demande pour certains nœuds ICN pour violer la cohérence entre la publication asynchrone ICN et le processus d'abonnement. Un attaquant envoie un grand nombre de demandes pour dégrader la performance de certains routeurs, de sorte que le routage des demandes et le transfert de données présentent des retards plus longs.
- **Inondation :** Un attaquant envoi un grand nombre de demandes qui dépassent la limite des nombres de requêtes capables de traiter. Le nœud attaqué accepte un certain nombre de demandes et ignore les demandes restantes. En conséquence, l'attaquant réussit à surcharger l'infrastructure globale et nuit à tous les utilisateurs à proximité. En outre, comme ICN est une architecture centrée sur le contenu, il est difficile d'appliquer des limites pour le taux de demande par utilisateur final car il n'y a pas d'identifiant hôte. Dans notre travail, nous nous intéressons à ce type d'attaque que nous détaillons dans les sections suivantes.

3.2. Attaques par Inondation d'intérêts

Les attaques par inondations d'intérêts (Interest Flood Attacks, IFA) sont le fléau de la sécurité CCN. Un IFA est une attaque qui exploit le tableau des intérêts en attente (PIT), Le point de départ est un botnet composé de nombreux zombies potentiellement distribués, qui génère rapidement un grand nombre des intérêts en saturant le PIT du routeur victime. Une fois que le PIT est complètement plein, tous les intérêts entrants ultérieurs sont abandonnés et le routeur s'engage à supprimer des entrées du PIT, soit en raison du retour de contenu ou de l'expiration. C'est le but des attaques par inondation d'intérêts **[Api 13]**.

Dans la catégorie des attaques IFA, nous distinguons trois types d'attaque selon le contenu des paquets d'intérêt envoyés par les attaquants **[Nah 16]**:

1. **Contenu existant ou fixe**

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

2. Contenu généré dynamiquement

Lorsqu'un intérêt pour des données générées dynamiquement arrive, le serveur doit traiter la demande (par exemple, requête de base de données, calcul) avant de pouvoir générer les données et répondre. Ce qui peut imposer plus de charge au producteur, en raison du nombre accru de demandes de contenu qui ne peuvent pas être prés calculés. Cela pourrait entraîner une latence aller-retour plus élevée et un taux plus élevé de paquets abandonnés, ce qui force les intérêts de l'attaquant à rester plus longtemps dans le PIT de la victime.

3. Contenu inexistant

Lorsque les attaquants envoient des intérêts à des paquets de données inexistants ou que les paquets de données ne peuvent jamais être générés, les paquets d'intérêt ne peuvent pas être satisfaits. Ce qui permet à l'attaquant de créer des entrées dans le PIT de la victime pour lesquelles aucun contenu ne sera jamais renvoyé. Ces derniers resteront dans le PIT pendant une période aussi longue que possible, ce qui épuisera certainement la mémoire et les ressources des routeurs, dégrader a les performances des routeurs, et ralentira (voire abandonnera) le traitement des intérêts des clients légitimes. Dans notre travail, nous nous intéressons à ce troisième type des attaques d'inondation d'intérêts.

3.3. Scénario d'attaque par Inondation des Faux Intérêts en utilisant un réseau des zombies

La figure II.4 montre un exemple de scénario d'attaque IFA générée par un réseau des zombies, où des clients légitimes et des attaquants connectés à un routeur de périphérie, demandent des contenus spécifiques. Les attaquants inondent le réseau avec de fausses intérêts (« /attack/B* » et « /attack/C* ») en cherchant un contenu d'origineinexistant. Cela va remplir la table des intérêts (PIT) et les ressources de traitement et de mémoire d'échappement du routeur de l'extrémité. En conséquence, les routeurs attaqués transmettent ces requêtes aux nœuds voisins, qui, à leur tour, les transmettent aux prochains nœuds voisins et ainsi de suite jusqu'à arriver aux fournisseurs de contenu. Si le nombre de demandes invalides est élevé, toute demande légitime prend un temps de réponse plus long. Par conséquent, si le temps de réponse dépasse la période d'expiration de la demande, la demande peut ne pas être répondu. C'est pourquoi, il faut mettre en place des contre-mesures pour ce type d'attaques. Cela fait l'objet de la section suivante.

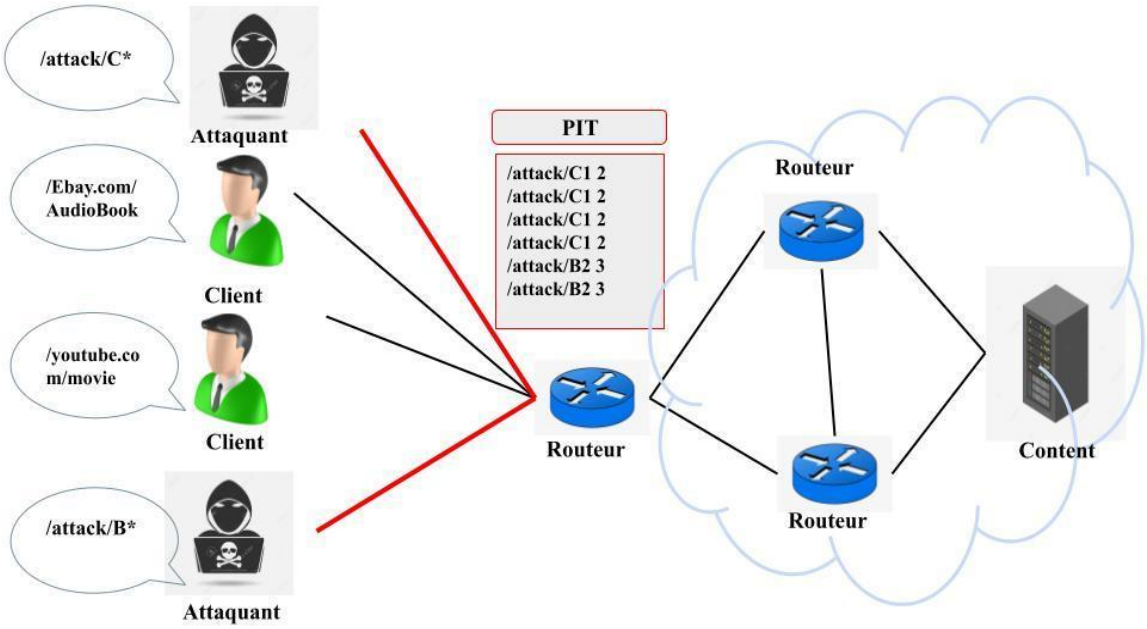

Figure II.4– Scénario de Attaque IFA-Zombie[Rst 17].

4. Approches des contres mesures pour les attaques par inondation d'intérêts

Pour se défendre contre ce type d'attaque d'inondations d'intérêts (IFA), il existe plusieurs algorithmes de contre mesure (Figure II.5). Selon la littérature, ces algorithmes sont basés soit sur les statistiques du routeur, soit sur le mécanisme push/back [**Nah 16]**.

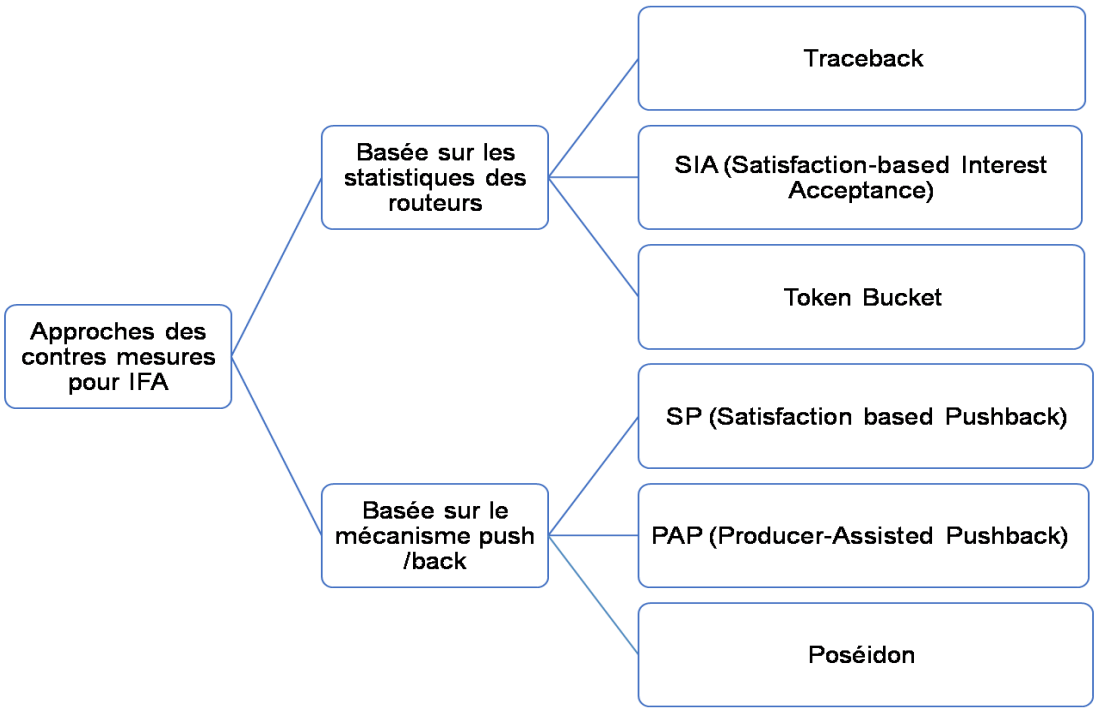

Figure II. 5– Approches contres mesures IFA.

- **Approche Basée sur les statistiques du routeur :**Les statistiques du routeur peuvent être utilisées pour 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 les interfaces entrantes qui en ont envoyé trop des intérêts insatisfaits pour ce préfixe. Cependant, cette approche, peut enrichir les attaques IFA car un attaquant peut simplement atteindre la limite définie dans les interfaces en envoyant plusieurs requêtes malveillantes ce qui évoque l'interdiction de transmettre les requêtes légitimes. Des exemples d'algorithmes de cette catégorie : Traceback**[Hyj 13]**, SIA (Satisfaction-basedInterest Acceptance) **[Api 13]**, AlgorithmeTokenBucket**[Api 13]**.
- **Approche Basée sur le mécanisme push/back :**Ce mécanisme permet de remonter à la source de l'attaque et d'isoler l'attaque à la source. Il permet aussi 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. Contrairement à l'approche précédente, le 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..). Des exemples d'algorithmes de cette catégorie : Poséidon **[Amp 13],** SP (Satisfaction basedPushback) **[Api 13],** PAP (Producer-AssistedPushback) **[Zvj 18].**

Dans notre étude, nous allons se concentrer sur l'approche basée sur le mécanisme de push/back car elle est plus efficace que l'approche basée sur 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.

5. Principaux algorithmes de contres mesures

Comme déjà mentionné dans la section précédente, ils existent plusieurs algorithmes de contres mesures pour IFA. Dans notre projet, qui est une continuité des PFE antérieurs, nous allons se contenter de présenter l'algorithme de Traceback (déjà étudié dans le PFE**[Hak 17]**), l'algorithme de Poséidon (déjà étudié dans le PFE **[Bof 20]** et **[Raw 18]**) et l'algorithme de PAP comme un nouvel algorithme à étudier et à comparer avec les algorithmes déjà étudiés.

5.1. Algorithme de Traceback

L'algorithme Traceback**[Hak 17]** 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 faux intérêts 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 que 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. Une version simplifiée de l'algorithme Traceback est donnée dans la figure suivante :

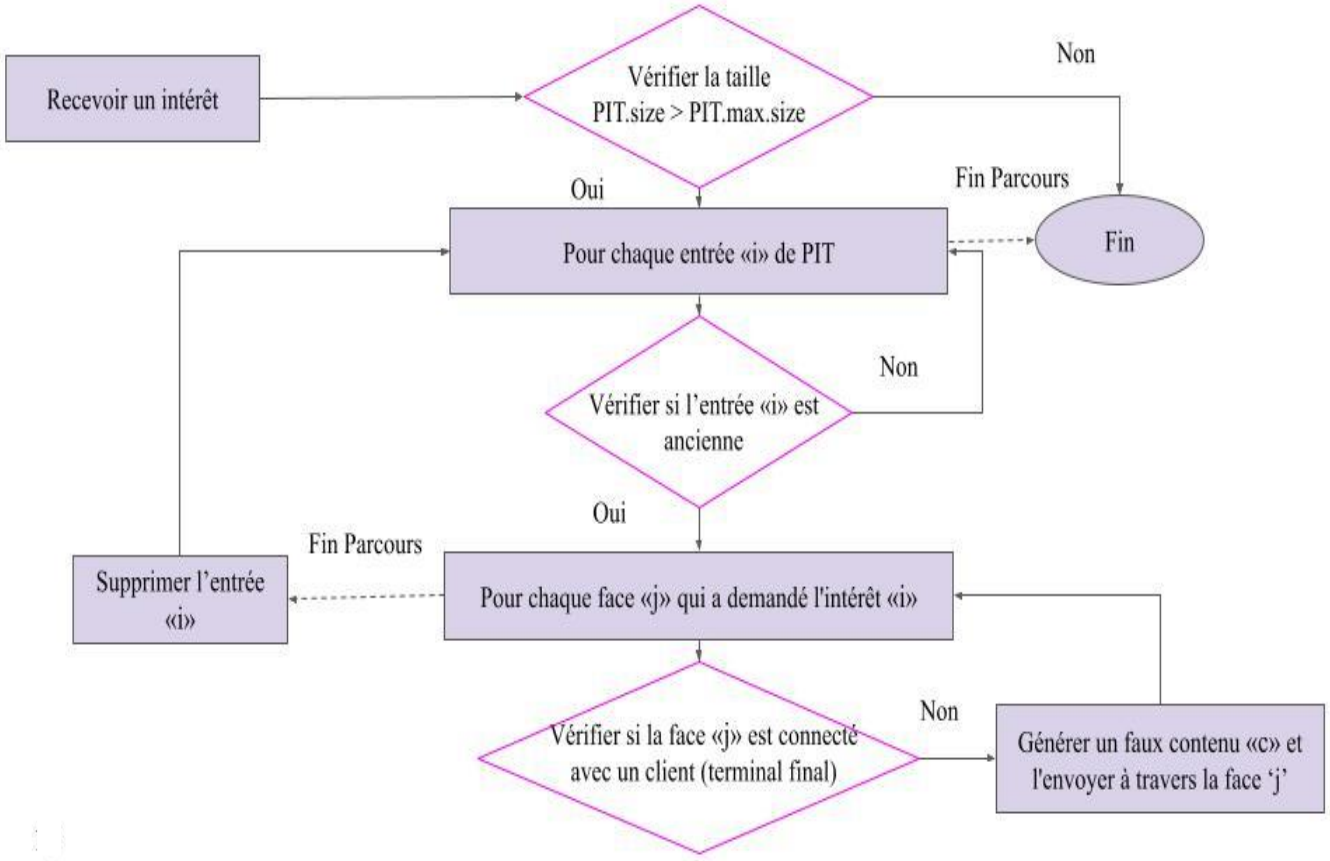

Figure II.6– Algorithme de Traceback [Raw 18].

5.2. Algorithme de Poséidon

Poséidon est un ensemble d'algorithmes qui fonctionne sur des routeurs dans le but d'identifier les anomalies de trafic (en particulier, les inondations d'intérêts) et d'atténuer leurs effets **[Amp 13]**. Il est composé de deux phases, une phase de détection et une phase de réaction. Dans la phase de détection, il surveille en permanence les taux d'insatisfaction par interface des intérêts par rapport au trafic global. Si ces taux changent significativement entre deux intervalles de temps consécutifs, il fixe un filtre sur la ou les interfaces incriminées (ce qui réduit le nombre des intérêts entrants).Ainsi, dans la phase de réaction, 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 le mécanisme push/back qui peut émettre un message d'alerte aux mêmes interfaces, pour signaler qu'une inondation d'intérêts est en cours.

Notons ici que la phase de détection peut être locale ou distribuée (collaborative). Dans le premier cas, les routeurs s'appuient uniquement sur des métriques locales (par exemple, taux d'occupation de PIT, taux d'intérêts insatisfaits, quantité de bande passante utilisé pour transférer du contenu) pour identifier une attaque. Dans le deuxième cas, les routeurs proches collaborent pour déterminer si une attaque est en cours et comment l'atténuer. En cas d'attaque d'inondation d'intérêt réussie, la victime qui est le routeur peut facilement identifier une attaque en observant si son PIT est plein ou si la bande passante allouée à l'expédition de contenu est très petit. Cependant, il peut ne pas être possible pour un routeur sur le chemin de la victime pour détecter une attaque en cours.

Dans **[Bof 20]**, les développeurs ont proposé une version améliorée du Poséidon, appelée « Poséidon _TTL » afin d'atténuer le problème de faux positifs, en ajoutant le concept du nombre de saut (le TTL) dans le CCN, et intégrant comme métrique avec les paramètres d'exécution de Poséidon. Cet algorithme (figure II.7) vérifie le TTL de l'intérêt reçu et compare cette valeur avec un seuil TTL prédéfini. Si la valeur TTL de l'intérêt reçu est supérieure ou égale au seuil prédéfini, l'algorithme ne bloque pas 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.

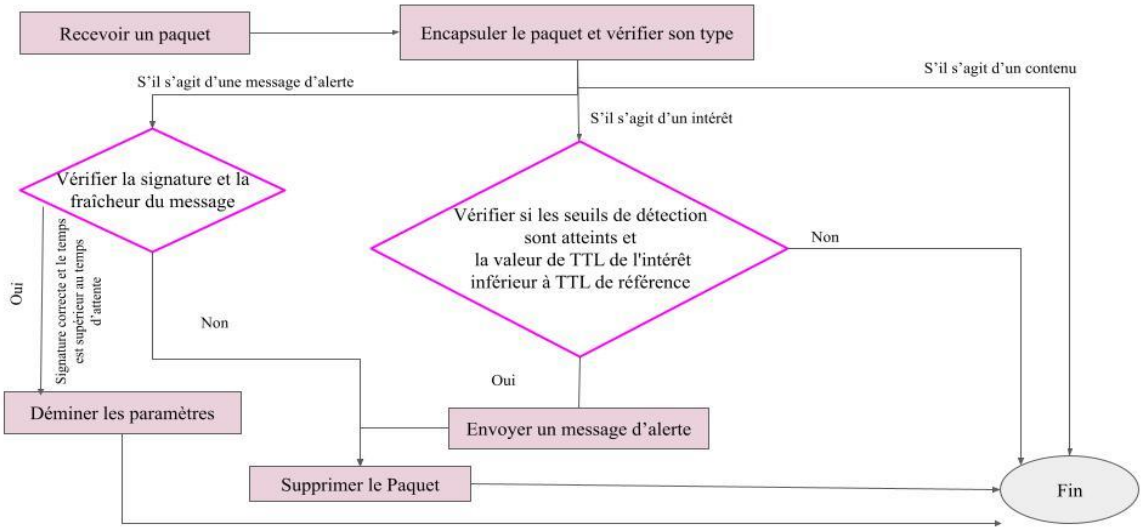

Figure II.7– Algorithme de Poséidon_TTL[Bof 20].

5.3. Algorithme de PAP

L'algorithme de PAP (Producer-AssistedPushback), proposé dans **[Zvj 18]**, permet de repousser le trafic DDoS dans NDN vers des entités qui se comportent mal, à une granularité beaucoup plus fine. Il est conçu pour être exécuté sur chaque routeur dans le cadre de stratégie de transfert. Le mécanisme de push/back est déclenché par le serveur victime en envoyant un paquet NACK (Figure II.8). Un paquet NACK est un mécanisme de rétroaction CCN saut par saut utilisé pour signaler un problème dans la transmission ultérieure d'un intérêt. Il contient les informations suivantes :

- Un code Raison (R) pour informer le routeur du type d'attaque : A (Intérêts pour les paquets de données statiques ou existants), B (Intérêts pour les paquets de données inexistants qui ne peuvent être générés) ou C (Intérêts pour les paquets de données générés dynamiquement). Autrement dit, l'attaque B inonde le réseau par des faux intérêts alors que l'attaque A et C inondent le réseau par des intérêts valides.
- Le Préfixe (P) des intérêts suspects.
- Le taux de réception des Intérêts Faux/Valides (T/C) que le serveur peut gérer actuellement sous le préfixe P. Ce taux est appelé « tolérance » en cas des faux intérêts ou « capacité » en cas des intérêts valides.
- Une liste des noms des faux intérêts(FL, Floride Liste) dans le cas d'attaque B uniquement.

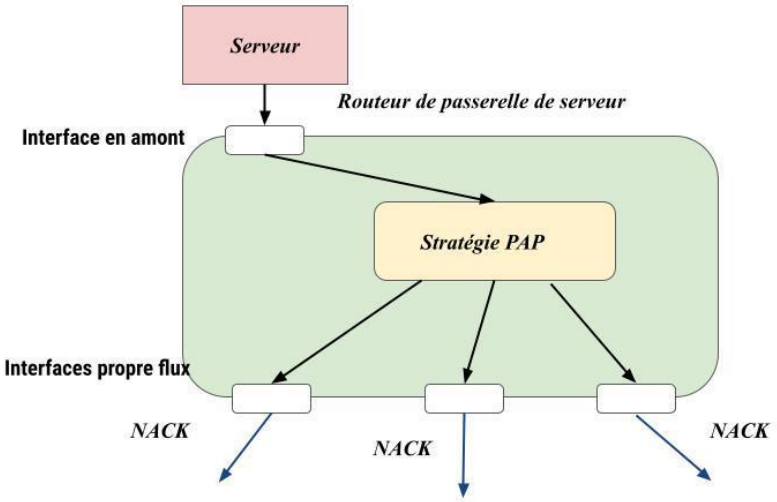

Figure II.8–Présentation du système PAP [Zvj 18].

Lorsque les routeurs reçoivent un tel paquet NACK, une action appropriée est déclenchée dans le routeur en fonction du code de raison utilisé dans le paquet NACK :

1. S'il s'agit d'une attaque de type B (des faux intérêts), le routeur vérifiera le FL et trouvera les intérêts en attente correspondants de PIT. Sinon (la raison est une attaque A ou C, i.e intérêts valides), le routeur ne peut pas distinguer le trafic légitime du trafic incriminé, par conséquent le routeur vérifiera tous les intérêts en attente actuels sous le préfixe P. À partir de ces intérêts en attente, le routeur obtiendra des informations exactes sur leurs interfaces entrantes et effectuera le poussée précise.

2. Selon le type d'attaque, le routeur calculera un poids pour chacune de ces interfaces en fonction du nombre d'intérêts entrants via l'interface. Ensuite, le routeur utilisera le poids et la tolérance/capacité totale attribuée dérivée du champ T/C du NACK pour obtenir une tolérance/capacité pondérée pour chaque interface.
3. Après cela, le routeur envoie à chacune de ces interfaces, un nouveau NACK portant la tolérance/capacité pondérée en tant que T/C et un FL élagué si la raison R est une attaque B. De cette façon, tous les routeurs le long du chemin d'envoi d'intérêt recevront et généreront de nouveaux NACK qui seront propagés aux routeurs en aval supplémentaires.
4. Enfin, la demande de limitation de débit provenant du serveur arrivera à tous les routeurs passerelles des clients suspects.Plus précisément, pour une interface client dont la tolérance/capacité pondérée est supérieure à zéro, le routeur supprimera aléatoirement les paquets d'intérêt sous le préfixe P afin de contrôler le débit d'envoi de l'interface pour qu'il soit inférieur ou égal à la tolérance pondérée/ capacité. Tout en limitant le débit, les routeurs passerelles surveilleront également le comportement de ses clients. Après avoir reçu un NACK du routeur passerelle, les clients légitimes se conformeront et réduiront leur taux d'envoi des intérêts sous le préfixe P, alors que les bots peuvent ne pas obéir aux règles. En conséquence, le routeur peut effectuer limitation sélective du débit : le routeur peut remarquer les individus qui se comportent mal et restreindre davantage la limite, tandis que pour les clients légitimes, le routeur peut assouplir la limite.

En PAP, tous les routeurs maintiennent un RevertTimer, et les routeurs de passerelle maintiendront aussi un temporisateur supplémentaire appelé RateLimitTimer. Notez que les deux temporisateurs n'affectent pas le résultat final du Pushback et que différents routeurs peuvent régler ces deux temporisateurs différemment en fonction de leurs propres besoins.

- RevertTimer décide combien de temps un routeur doit conserver les intérêts suspects (DDoS). Chaque fois qu'un nouveau NACK arrive, le routeur doit vérifier s'il existe un RevertTimer pour le Pushback avec la même raison et le même préfixe. Si oui, le routeur doit mettre à jour le minuteur ; sinon, le routeur doit créer un nouveau temporisateur.
- RateLimitTimer décide combien de temps il faut à un routeur pour décider si un individu se comporte bien ou non. Après le RateLimitTimer, le routeur passerelle supprimera la limite de ses clients légitimes et renforcera la limite des clients incriminés (attaquants) et réinitialisera le RateLimitTimer. Par conséquent, cette minuterie est périodique jusqu'à ce que toutes les limites soient supprimées.

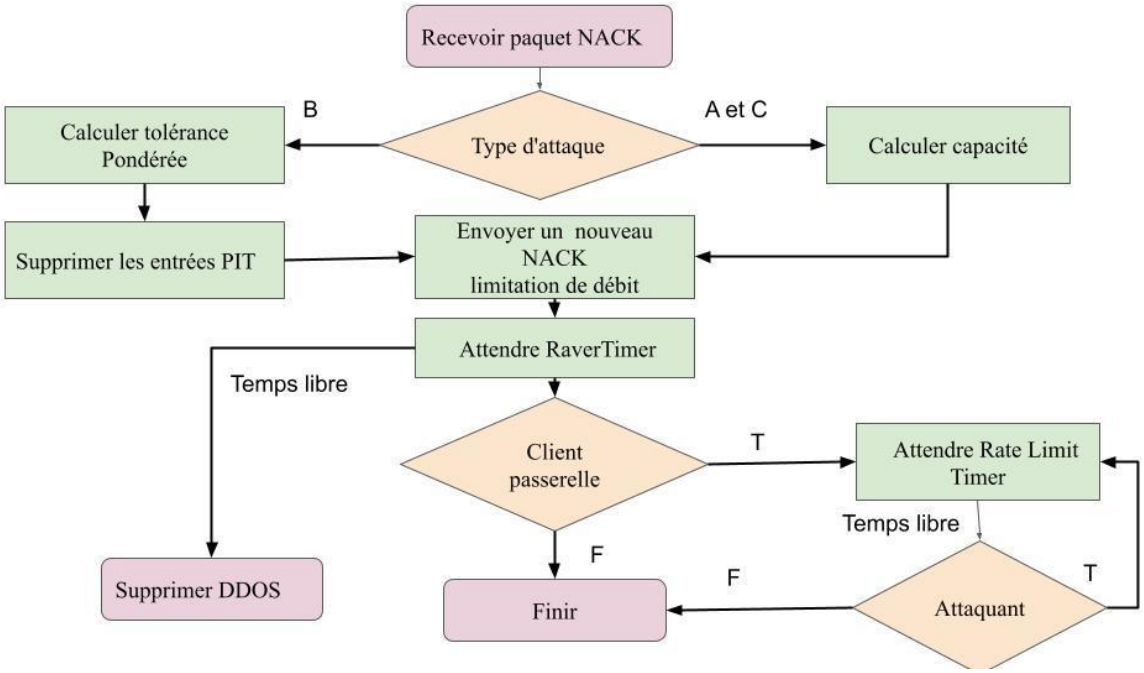

Figure II.9– Algorithme de PAP[Zvj 18].

5.4. Discussion

Le tableau suivant récapitule les différents critères de comparaison pour les trois algorithmes décrits ci-dessus :

	Traceback	Poséidon	PAP
Vitesse d'exécution	Lent	Moyenne	Rapide
Précision	76%	84%	99%
Taux d'erreur	24%	16%	1%
Paquetage de programmation	CCN lite	CCN lite	NdnSim
Type d'approche de contre mesure	Basé sur les statistiques du routeur	Push/Back	Push/Back
Atténuation progressive d'attaquant	Oui	Oui	Oui
Possibilité de blocage d·un client légitime	Oui	Oui	Non
Blocage définitif d'une interface	Oui	Non	Possible[3]

[3] L'algorithme ne spécifie pas le mécanisme de suppression de DDoS ; on peut bloquer définitivement ou momentanément.

Configuration de chaque interface (chemin) reliépar un nœud de manière différente que les autres interfaces.	Oui	Oui	Oui
Chaque interface (chemin) d'un nœud a un niveau de sensibilité que les autres interfaces.	Oui	Oui	Oui
Installation dans chaque nœud CCN	Oui	Oui	Oui
Il fait des mises à jour de PIT	Oui	Non	Oui (sélective)
Si on fait une mise à jour sur le réseau, il faut faire une nouvelle configuration sur la contre-mesure	Oui	Oui	Non
Collaboration entre routeurs	Non	Oui	Oui
Le nœud de déclencheur de l'algorithme	Routeur	Routeur	Serveur victime

Tableau II.1– Comparaison entre les principaux algorithmes de contre-mesure.

De plus, le tableau suivant résume les points forts et les points faibles de chaque algorithme :

	Points forts	Points faibles
Traceback	- MAJ de PIT - la méthode de traçabilité des intérêts est efficace atténue les attaques DDoS.	- Difficulté de définir le bon seuil d'espace mémoire occupé pour d'activer le mécanisme de trace. - Risque de libérer les intérêts satisfaits dans la table PIT - Assure une défense plus agressive (Possibilité de bloquer l'interface de façon définitive) - Possibilité de bloquer un client légitime (faux positive) et de ne pas bloquer un attaquant (faux négatif) - Pas de collaboration entre les routeurs - Consommation exhaustive de la bande passante du réseau en envoyant les paquets falsifiés

Poséidon	- Détection d'IFA en se basant sur plusieurs paramètres calculables - Collaboration entre les routeurs en envoyant des messages d'alerte - Limitation de débit sur les interfaces incriminées. - une meilleure amélioration de leur performance et diminution du taux d'erreurs (surtout en termes de faux positifs).	- Possibilité de bloquer un client légitime (faux positive)et ne pas bloquer un attaquant (faux négatif) - Pas de mise à jour dans la table PIT - Possibilité de bloquer l'interface de façon définitive
PAP	- MAJ de PIT - Collaboration entre les routeurs - Possibilité d'intégration de plusieurs mécanismes de suppression de DDoS - Limitation sélective du débit provenant du serveur victime jusqu'à tous les routeurs passerelles des clients suspects - Utilisation des temporisateurs pour conserver les intérêts suspects et décider si un individu se comporte bien ou non.	- Déclenchement de l'algorithme par le serveur victime. - Difficile à mettre en œuvre (création des nouveau paquets NACK, plusieurs calculs à effectuer, MAJ les temporisateurs)

Tableau II.2– Les points faibles et forts des algorithmes de contre-mesure.

Ces comparaisons nous donnent un aperçu des avantages et des inconvénients de chaque algorithme. Nous constatons que les contre-mesures selon l'approche Push/Back sont meilleurs avec une précision dépassant 84% (pour Poséidon) jusqu'à 99% (pour PAP) contre une précision de76% pour l'algorithme de Traceback basé sur les statistiques des routeurs, ce qui nous laisse choisir les solutions de l'approche Push/back, sur avec l'algorithme de PAP 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 nôtre, nous allons choisir le PAP comme nouvel algorithme à implémenter et à tester qui va nous permettre de détecter rapidement les attaques IFA de manière efficace en repoussant le trafic vers les attaquants.

6. Conclusion

Dans ce chapitre, nous avons présenté une classification globale des attaques ICN avant de se concentrer sur celles basé sur le routage centré contenu, notamment les attaques DDoS. Ensuite, nous avons expliqué ce type d'attaque en mettant l'accent sur les attaques d'inondation par des faux intérêts lancés par des zombies. Puis, nous avons étudiés les approches de contre- mesures des attaques IFA et les principaux algorithmes. Enfin, nous avons terminé avec une discussion pour justifier notre choix à l'algorithme de contre mesure PAP. Notre propre version développée de cet algorithme sera présenté dans le chapitre suivant.

Chapitre III : Déploiement d’une solution allégée de Producer-AssistedPushback (PAP4FIFA)

1. Introduction

La simulation des réseaux est une technique par laquelle on modélise le comportement d'un réseau à l'aide d'un simulateur. Une fois la topologie mise en place, la simulation permet de voir le comportement des nœuds et leur fonctionnement, ainsi que la répétition des expériences. Cette modélisation a pour avantages le gain de temps et la variation des paramètres, ce qui est à l'inverse d'un déploiement réel qui nécessite plus de temps, de ressources et sur lequel c'est difficile de tester toutes les possibilités.

Le déploiement d'une architecture CCN est quasiment impossible pour l'instant vu les ressources matériels et logiciels nécessitant cette mise en place. Pour cela, l'utilisateur d'un simulateur est primordial. Afin de faire fonctionner notre réseau CCN sur une petite topologie et tester l'impact réel de la contre mesure retenue dans le chapitre précédent à savoir l'algorithme Producer-AssistedPushback (PAP) qui, même si c'est une solution très puissante et efficace, est très lourde et difficile à implémenter. Nous allons commencer par tester le code source du PAP, dont une partie de lui a été déjà proposé et implémentée par [Vas 18] sur les réseaux NDN simulé sur le simulateur NS3, afin d'essayer de l'adapter et de le simplifier pour notre réseau CCN sous le simulateur « OMNET++ ». Ce choix de simulateur a été exigé pour nous permettre de comparer les résultats obtenus avec ceux déjà réalisés par [**Raw 18**] pour les contres mesures POSEIDON et TRACEBACK, et donc il est nécessaire de déployer notre solution dans les mêmes conditions déployées.

Pour réaliser notre travail, nous allons présenter dans la première section, notre première simulation qui est la reproduction de l'environnement de travail où le protocole PAP a été implémenté et expliquer les raisons de l'échec de ce code source existant. Puis, nous passerons à la deuxième simulation où nous présenterons le simulateur « OMNET++ » que nous avons utilisé dans notre travail, ainsi que le package CCN-lite qui nous permet de simuler les CCN et de faire un scénario d'attaque d'interception avec les solutions possibles. Dans cette partie, nous proposerons une solution allégée de notre contre mesure PAP pour une meilleure facilité d'implémentation vu la complexité et la lourdeur de la solution qui traite plusieurs types d'attaques et l'adapter seulement pour bloquer les attaques d'inondation par des faux intérêts.

2. Première implémentation sous le simulateur NS3

[**Drb20**] a implémenté le protocole PAP pour la protection des réseaux NDN en utilisant le simulateur ndnSIM de NS3. En ce qui suit, nous allons reproduire le même environnement de travail pour tester si le code source proposé est fonctionnel ou non, et puis son impact.

2.1. Présentation du Simulateur ndnSIM

NdSIM est un simulateur open source basé sur NS3 qui implémente fidèlement les composants de base d'un réseau NDN de manière modulaire, la conception de ndnSIMa les objectifs suivants :

- Etre un package open source pour permettre à la communauté de recherche d'exécuter des expérimentations sur une plateforme de simulation commune.
- Etre capable de simuler fidèlement toutes les opérations de base du protocole NDN.
- Être capable de supporter des expériences de simulation à grande échelle.
- Faciliter les expérimentations de la couche réseau avec le routage, la mise en cache des données, le transfert de paquets et la gestion de la congestion.

La figure suivante présente la structure globale de l'environnement ndnSIM:

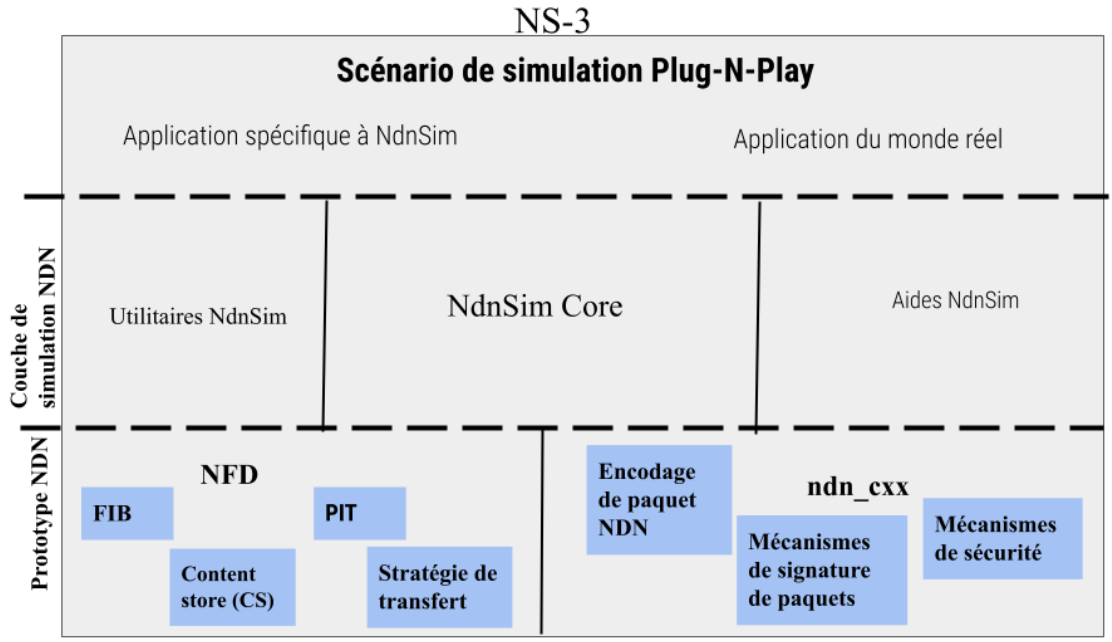

Figure III.1–Structure de simulation ndnSIM.

2.2. Test du code source PAP proposé par [Zhi 18]

Dans cette première partie de notre travail, nous avons choisi de travailler avec NDN, car nous avons trouvé qu'une partie du code source du protocole sur lequel nous travaillons existait déjà avec le simulateur NDN. Malheureusement, nous avons trouvé des difficultés de travailler avec ce simulateur pour les raisons suivantes :

1. Problèmes dans l'installation du simulateur : Il faut installer les bonnes versions du système linux et de langage python, chose qui n'est pas mentionné clairement dans les étapes d'installation.
2. Ce n'est pas facile de travailler avec, car il ne fonctionne qu'avec la ligne de commande.
3. Problème de compilation de code source disponible dû au manque d'information de la version de python. A chaque compilation, il affiche les erreurs de la figure suivante :

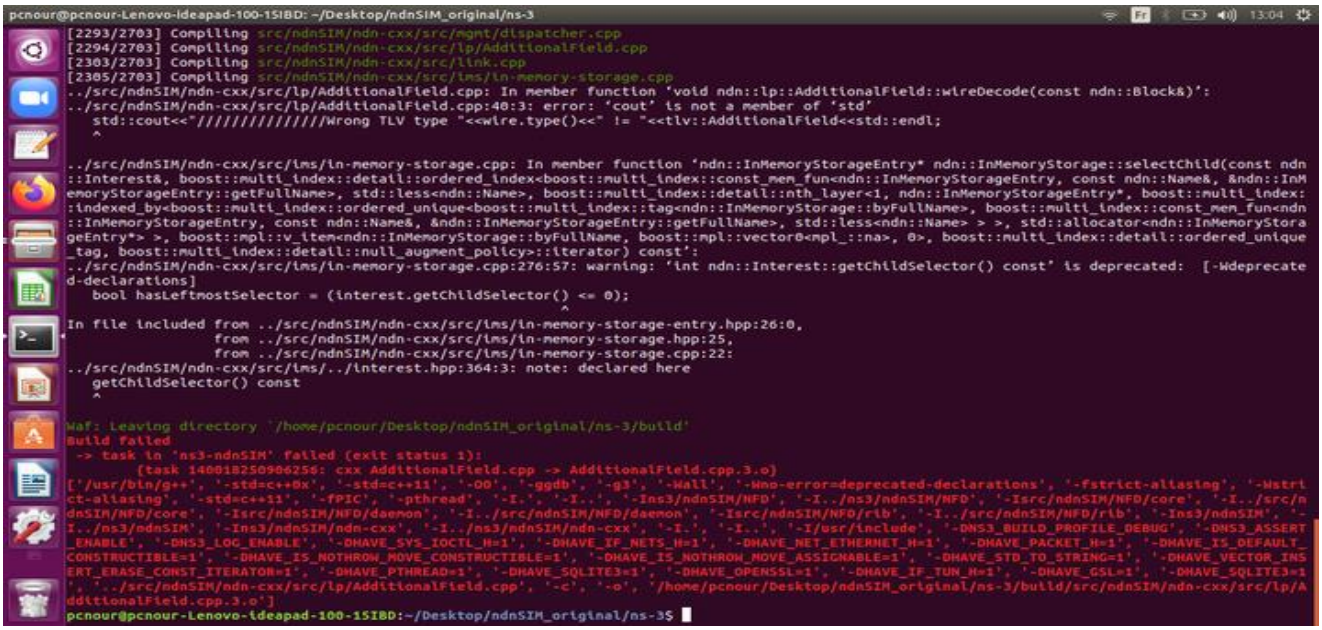

Figure III. 2–Erreur de compilation.

Nous avons donc trouvé qu'il est préférable de travailler avec Omnet++ qu'avec le simulateur NDN et d'écrire notre propre code du protocole qu'utiliser le code source disponible qui manque des commentaires et d'aide de déploiement.

3. Proposition d'une version simplifiée du protocole PAP « PAP4FIFA » pour OMNET++

PAP repousse le trafic DDoS vers des entités qui se comportent mal, à une granularité beaucoup plus fine que les mécanismes de défense DDoS existants. Cependant, le protocole tel proposé par ses auteurs **[Zha 18]** traite plusieurs types d'attaques IFA sur les paquets de données inexistants et les données dynamiques, ceci a rendu le protocole trop lourd à exécuter et trop complexe à implémenter. Puisque, nous nous intéressons seulement aux attaques d'inondation par des faux intérêt ou FIFA (False InterestFloodingAttacks), nous proposons de simplifier le protocole et garder la partie qui traite ce type d'attaque ce qui nous permettra d'atteindre deux objectifs :

1. Implémenter la solution dans des délais réalistes surtout qu'aucune implémentation pour les CCN n'a été effectuée.
2. Simplifier le protocole de sorte à garder les avantages de robustesse de la solution et la rendre plus légère et plus rapide à répondre à une attaque.

3.1. Présentation de PAP4FIFA

La figure III.3 comporte l'organigramme du protocole PAP simplifié que nous appelons PAP4FIFA : PAP for False InterestFloodingAttacks. Notre solution permettra la détection rapide des attaques FIFA en utilisant les commentaires explicites et réaction efficace en repoussant le trafic vers les clients/attaquants qui se comportent mal.

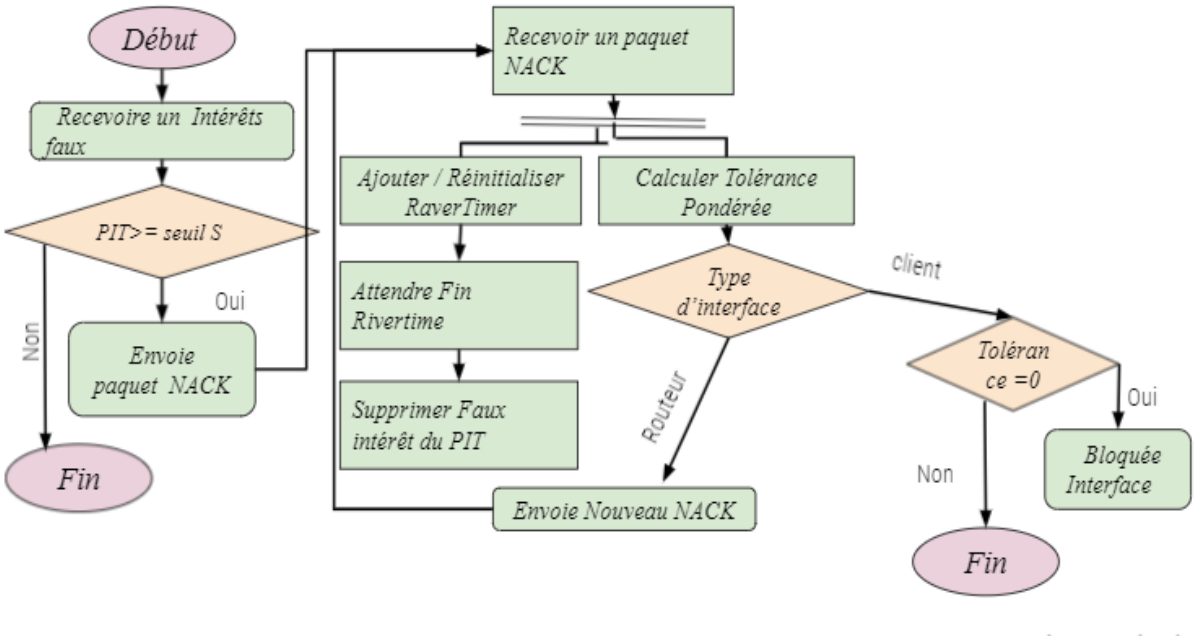

Figure III. 3– L'algorithme de notre solution « PAP4FIFA ».

Tout d'abord, les attaquants et les clients légitimes envoient des intérêts (demandes de contenu), et s'il s'agit de faux intérêt (le contenu est inexistant) dans les nœuds intermédiaires, ils seront acheminés vers le serveur victime, qui va les enregistrer dans la table PIT et vérifie la charge de sa table PIT (le nombre de ses entrées). Une fois cette dernière est supérieure à un seuil prédéfini « S = 60% », l'algorithme de sécurité PAP simplifié se déclenche :

i. Le serveur crée et envoi un paquet appelé NACK aux routeurs avals (premiers routeurs liés directement au serveur)
ii. Chaque routeur aval récupère le paquet NACK, calcule la tolérance pondérée de chaque interface d'entrée, et renvoie le nouveau NACK aux autres routeurs intermédiaires après avoir ajouté la valeur RaverTimer. Cette dernière est la variable qui décide combien de temps un routeur doit conserver les enregistrements suspects d'être des attaques FIFA, et ainsi de suite jusqu'à arriver aux routeurs d'extrémité (c'est les routeurs liés directement aux clients).
iii. Les routeurs passerelles réceptionnent les NACK, recalculent la tolérance pondérée pour chaque interface d'entrée. Si cette tolérance « T » est égale à 0, donc le nœud d'extrémité est un nœud suspect d'être un attaquant puisque le réseau ne tolère plus d'autres intérêts envoyés par ce dernier, le routeur bloquera l'interface suspecte. Sinon la tolérance T est supérieure à 0 alors le routeur passerelle ne fait rien puisqu'aucune interface n'est considérée comme suspecte.

3.2. Phase de détection (Seuil de déclenchement)

Nous supposons que le serveur victime peut déclencher le protocole PAP et envoyer le NACK quand l'espace mémoire disponible est surchargé et 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 en termes de nombres

des entrées insérées. Après avoir détecté une occupation de mémoire anormale, le processus est déclenché, et le paquet NACK est généré pour les entrées qui sont restées longtemps insatisfaites. Afin d'exploiter au maximum l'espace mémoire disponible pour le PIT, il faut définir le seuil pour que le Pushback soit activé, à S% de la mémoire occupée.

3.3. Paquet NACK

Après que la mémoire PIT a occupé plus de S%, le serveur envoie le paquet NACK au routeur amont qui contient les champs suivants :

- **R :** Type d'attaque de l'intérêt (dans notre cas, c'est toujours le Type C).
- **P :** La liste des préfixes du nœud que l'attaque à traverser.
- **T :** Le pourcentage des faux paquets est accepté par interface.

3.4. Calcul de la tolérance

Après que le routeur amont a reçu le **NACK**, il commence à calculer le poids et la tolérance pondérée de même que PAP**[Zvj 18]**pour chaque interface « i » comme suit :

- Ti=TNACK*Wi

- Où TNACK : Valeur de tolérance dérivée du PAP NACK.
- Wi : Poids pour le i ème interface qui est calculé par l'équation suivante :
- $W_i = \frac{1}{NL} \sum_{j=1}^{NP} \frac{1}{NFj}$
- NP : Nombre de toutes les entrées PIT correspondantes.
- NFj : Nombre d'interfaces entrant dans la jième entrée PIT.

Après cela, le routeur envoie à chacune de ces interfaces un nouveau NACK portant la tolérance pondérée en tant que T. De cette façon, tous les routeurs amont du chemin d'envoi d'intérêt recevront et généreront de nouveaux NACK qui seront propagés aux routeurs en passerelles supplémentaires.

3.5. Phase de réaction

Après avoir calculé la tolérance pondérée pour chaque interface, le routeur envoi de nouveaux paquets **NACK** à chaque interface d'un routeur connecté à lui. Si l'interface s'agit d'une interface d'un client dont la tolérance est égale à zéro, cette interface est bloquée (clients suspects).

3.6. Temporisateur RaverTime

Comme dans la version originale de **PAP**, tous les routeurs, dans notre version, maintiennent un **RevertTimer** qui décide combien de temps un routeur doit conserver les faux intérêts. Chaque fois qu'un nouveau **NACK** arrive, le routeur doit vérifier s'il existe un **RevertTimer** pour le mécanisme Pushback avec la même raison et le même préfixe. Si oui, le routeur doit mettre à jour le temporisateur. Sinon, le routeur doit créer un nouveau temporisateur. Ainsi, différents routeurs peuvent régler ce temporisateur différemment en fonction de leurs propres besoins.

4. Implémentation de la solution PAP4FIFA

Dans cette partie, nous commençons par présenter les outils de développement utilisés dans l'implémentation de notre algorithme. Ensuite, nous expliquons les étapes de mise en place que nous avons suivi tout le long de son développement.

4.1. Outils de développement

Pour développer notre solution, nous allons utiliser le simulateur des réseaux CCN (CCN-lite) sur le simulateur réseau OMNET++ avec le Framework INET. Ces outils sont décrits dans les sections suivantes.

4.1.1. Simulateur OMNET++

OMNET++ est un simulateur à événements discrets orienté objet, basé sur C++. Il a été conçu pour simuler les systèmes réseaux de communication, les systèmes multiprocesseurs, et d'autres systèmes distribués. OMNET++ est un projet open source dont le développement a commencé en 1992 par Andras Vargas à l'université de Budapest. Actuellement, ce simulateur est utilisé par des dizaines d'universités pour la validation de nouveaux matériels et logiciels, ainsi que pour l'analyse de performance et l'évaluation de protocoles de communication tel que **:**

- La modélisation des protocoles de communications.
- La modélisation des réseaux filaires et sans fil.
- La modélisation des systèmes répartis.
- Les architectures Hardware.
- En général, il peut être utilisé pour n'importe quel système à évènements discrets pouvant être modélisé selon des entités communiquant par envoient de messages.

L'avantage de OMNET ++ est sa facilité d'apprentissage, d'intégration de nouveaux modules et la modification de ceux déjà implémentés. Nous introduisons dans la suite l'architecture du simulateur OMNET++ et les librairies Mobility Framework et INET

4.1.2. CCN-lite :

Le simulateur OMNET++ n'est pas spécialisé pour les réseaux centrés Contenu (CCN). Pour cela, il existe plusieurs extensions, plateformes et simulateurs basés sur OMNET++ qui essayent d'introduire la notion du CCN dans OMNET++. La plus utilisée et la plus facile à manipuler est "CCN-Lite".

La motivation initiale en 2011 pour la création de CCN-lite était que le logiciel de routeur CCNx de PARC était devenu énorme. CCN-lite fournit une alternative allégée à des fins éducatives et comme tremplin. C'est pour ceux qui veulent un simple logiciel pour l'expérimentation ou le développement personnel, mais qui n'ont pas besoin de toutes les fonctionnalités de l'ensemble. CCN-lite est une implémentation ICN légère appartenant à la licence ISC permissive qui est développée 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 Mac Os,cela couvre :

1. Le protocole de réseau centré sur le contenu de PARC.
2. Le Projet Named-Data Networking (NDN).
3. Le projet Naming-Function Networking.

4. Un encodage expérimental et compact pour les environnements IOT.

4.1.3. Framework INET

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 pour les chercheurs et les étudiants travaillant avec des réseaux de communication. INET est particulièrement utile lors de la conception et de la validation de nouveaux protocoles ou de l'exploration de scénarios nouveaux ou exotiques.

INET est construit autour du concept de modules qui communiquent par passage de messages. Les agents et les protocoles réseau sont représentés par des composants, qui peuvent être librement combinés pour former des hôtes, des routeurs, des commutateurs et d'autres périphériques réseau. De nouveaux composants peuvent être programmés par l'utilisateur et les composants existants ont été écrits de manière à être faciles à comprendre et à modifier.

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 à partir de la ligne de commande.

4.2. Etapes de mise en place

La figure suivante résume les étapes de mise en place que nous avons suivi tout le long du processus d'implémentation :

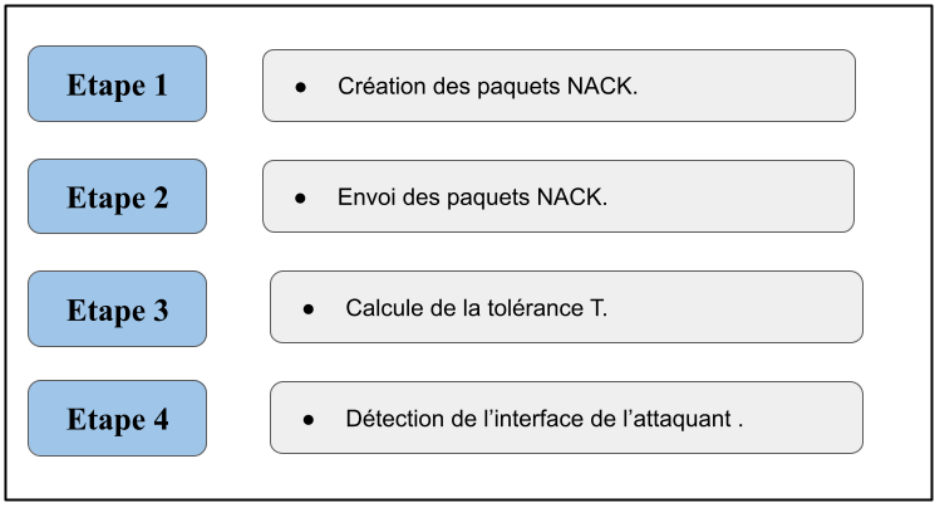

Figure III. 4– Étapes de travail.

4.2.1. Etape 1 : Création de Paquet NACK :

Tout d'abord, nous avons essayé de créer un nouveau paquet que nous appelons NACK et qui contient que la variable T (Le pourcentage des faux paquets accepté par interface) étant donné que la variable R (type d'attaque de l'intérêt) est toujours fixé à C).

Pour ce faire, nous avons essayé plusieurs méthodes qui ne se sont pas révélées positives. Une de ses méthodes est la suivante :

1. Ajouter une fonction dans la classe **CCn.h** et la déterminer dans **CCn.c** qui s'appelle **'sendNACKfct'** pour envoyer le paquet NACK .

```
int addToCacheDummy(const char *contentName, const int startChunk,
        const int numChunks);
bool sendBatchInterests(const char *contentName, int startChunk,
        int numChunks);
int sendNACKfct(const char *contentName, const int startChunk,
            const int numChunks);
bool addFwdRule(const char *contentName, FaceType faceTp, const char *dst,
        int aux);
```

Figure III. 5– L'ajout du fonction "sendNACKfct" dans CCn.h.

2. Ajouter un nouveau type " **MSG_TYPE_NACK**" dans la fonction **CCNLiteRElay**de la classe CCnCore.cc.

```
assert(typeIorC == MSG_TYPE_INTEREST || typeIorC == MSG_TYPE_CONTENT || typeIorC ==MSG_TYPE_NACK );
assert(ctrlBlock);
active_relay = (struct ccnl_omnet_s *) ctrlBlock;

switch (suite) {
case SUITE_CCNx0:
```

```
switch (typeIorC) {
case MSG_TYPE_INTEREST:
    pktInfo += "I]";
    break;
case MSG_TYPE_CONTENT:
    pktInfo += "C]";
    break;
case MSG_TYPE_NACK:
        pktInfo += "N]";
        break;
```

Figure III.6– L'ajout "MSG_TYPE_NACK".

3. Déclarer les variables nécessaires.

```
 */
class CcnPacket : public ::ByteArrayMessage
{
  protected:
    long cacheId_var;
    char suiteType_var;
    char msgPayloadType_var;
    long T;
    long C;


int CcnPacket :: getT()
{
    return T;
}

void CcnPacket::setT(int T)
{
    this->T=T;
}
int CcnPacket  :: getC()
{
    return C;
}

void CcnPacket::setC(int C)
{
    this->C=C;
}
```

Figure III.7– Déclaration de variable.

Puisque cette technique a échoué et c'est très difficile, voire quasiment impossible de créer un nouveau paquet dans CCN-Lite, nous sommes passées à une deuxième solution qui consiste à utiliser les paquets CCNs existants pour intégrer les informations de NACK. Au fait, Dans le CCN, il y a 2 paquets INTEREST et DATA. Le paquet INTEREST sera envoyé par les clients alors que le paquet DATA sera envoyé par le serveur. Etant donné que nous nous intéressons aux faux intérêts qui ne correspondent au contenu inexistant, donc nous n'aurons pas besoin d'envoyer les paquets DATA. Pour cela, nous avons opté à l'adaptation de paquet DATA afin de jouer le rôle de paquet NACK. Pour ce faire, nous n'avons déclaré que la variable T (Tolérance) dans les classes **" ccnPacket_m.c"** et **" ccnPacket_m.h"** comme suit :

```
CcnPacket::CcnPacket(const char *name, int kind) : ByteArrayMessage(name,kind)
{
    this->setByteLength(6);

    this->cacheId_var = 0;
    this->suiteType_var = 0;
    this->msgPayloadType_var = 0;
    this->T_var = 0;
```

```
void CcnPacket::parsimPack(cCommBuffer *b)
{
    ByteArrayMessage::parsimPack(b);
    doPacking(b,this->cacheId_var);
    doPacking(b,this->suiteType_var);
    doPacking(b,this->msgPayloadType_var);
    doPacking(b,this->T_var);

}

void CcnPacket::parsimUnpack(cCommBuffer *b)
{
    ByteArrayMessage::parsimUnpack(b);
    doUnpacking(b,this->cacheId_var);
    doUnpacking(b,this->suiteType_var);
    doUnpacking(b,this->msgPayloadType_var);
doUnpacking(b,this->T_var );

}
```

```
int CcnPacket::getT() const
{
    return T_var;
}

void CcnPacket::setT(int T)
{
    this->T_var = T;
}
```

Figure III. 8– Déclaration de variable.

4.2.2. Étape 2. Envoi de paquet NACK

Nous avons ajouté T dans la fonction "**fromMACFace**" qui est une méthode importante pour le passage d'un paquet entre les nœuds (il faut toujours vérifier les adresses mac toujours) : elle définit le processus de l'arrivée d'une trame de la couche 2 vers un paquet de la couche 3.

```
obj_info->packet_bytes = (unsigned char *) data;
obj_info->packet_len = len;
obj_info->T=ccnPkt->getT();
```

Figure III. 9– Modification dans la fonction "fromMACFace".

Nous avons également ajouté la variable "T" dans des fonctions différentes de trois classes différentes en raison de l'accessibilité à la variable, qui sera envoyée avec le paquet tout au long des fonctions :

1. Dans la fonction **"ccnl_app_RX"** de la classe "**ccnl-uapi.c"**.

```
io->packet_bytes = c->pkt->buf->data;
io->packet_len = c->pkt->buf->datalen;
io->name = ccnl_prefix_to_path(c->pkt->pfx); // TODO: should
io->chunk_seqn =
        (c->pkt->pfx->chunknum) ?
                ((int) *(c->pkt->pfx->chunknum)) : -1;
io->T= relay->T;
```

Figure III.10–Fonction "ccnl_app_RX".

2. Dans la structure "**info_data_s**" de la classe "**ccnl-uapi .h**".

```
// base type object
struct info_data_s {
    struct info_data_vt_s const * vtable;
    const char * name;    // User can provide the whole name (prefix) as a string
                          //   and we take care of componentisation, .. or ...
    const char * name_components[]; // The name components as a set of '\0'-terminated strings
                                    //   (useful for nfn when the component delimiter is not at
                                    //   every "/") -- last component ptr MUST be NULL
    int chunk_seqn;         // Optionally provided chunk num
                            //   Since chunk naming is somewhat arbitrary (e.g. ../cN,
                            //   ../N, ../c_N, etc), and left to a sender/receiver contract,
                            //   to reduce ambiguity here we assume that the chunk seq num
                            //   SHOULD be present whenever the last component provided in
                            //   "name" or "name_components" members is NOT identifying a
                            //   chunk. The implication of this is that if a chunk_seqn is
                            //   not provided (-1 value) the "name" or "name_components" may
                            //   still encode a chunk no (appl specific convention), but
                            //   ccn-lite will be oblivious to this (therefore not being able
                            //   to handle top-level segmentation internally, only segmentation
                            //   of segments at a 2nd level).
    unsigned char * packet_bytes;       // Buffer holding the packet
    unsigned int packet_len;         // Packet buffer len
    int T;
};
```

Figure III.11– La classe ccnl-uapi.h.

3. Dans la fonction "**endService"** de la classe "**ccnInet.cc"**.

```
DBGPRT(EVAUX, Info, this->getFullPath())
        << "Received CcnPacket packet over Ethernet from "
        << macCtx->getSrc() << " at " << macCtx->getDest() << "T: " << ccnPkt->getT()
        << std::endl;
```

Figure III. 12– La classe "ccnInet.cc".

4.2.3. Étape 3. Calcul de la tolérance

Dans la classe "**CcnCore .cc**" précisément dans la fonction "**toMACface**", nous avons défini le processus d'arrivée d'un paquet de la couche 3 vers une trame de la couche 2. On commence par la création du paquet, et on ajoute une variable de type String T dans les paramètres de la fonction "**ToMacface"**.

Nous avons également dû utiliser différentes fonctions dans notre code, que nous avons expliquées dans les lignes suivantes :

- ***ccnCtx->getSrcAddress802()*** : Prend la valeur de l'adresse mac de l' interface source, et elle la compare avec l'adresse prédéfinie et fixe du serveur.
- Si la taille du PIT du serveur est supérieure à la valeur seuil max du PIT "S", on calcule la tolérance et puis on l'envoie au routeur passerelle.

```
 if(  strcmp(ctx->getSrcAddress802().str().c_str(), servermac.c_str())!=0

if(Taille_PIT>S)
{
Tnack= T*(a/NL*a/NF);
            stringstream inter1;
            string tnack;

            inter1<<Tnack;
            inter1>>tnack;
pktInfo += "   T Pondree: " + tnack;

nbr_Content++;
EV<<"nbr_Contenu \n";
          EV << nbr_Content;}
```

Figure III.13– Calcul de la tolérance.

- Au niveau des routeurs c'est le même calcul.

```
if(   strcmp(ctx->getDstAddress802().str().c_str(), Mac.c_str())!=0){

Tnack= T*(1/NL*1/NF);
            stringstream inter1;
            string tnack;

            inter1<<Tnack;
            inter1>>tnack;
pktInfo += "   T Pondree: " + tnack;
```

Figure III.14– Mise à jour de T au niveau routeur.

4.2.4. Étape 4. Détection de l'interface de l'attaquant

Si, la valeur de T est égale à 0, donc c'est un attaquant sinon c'est un client légitime

- ***typeIorC*** == MSG_TYPE_CONTENT : Elle compare si le paquet est de type "contenu"

```
if ( typeIorC== MSG_TYPE_CONTENT){
if (Tnack =0)
{
    EV<<"c'est  un attaquant \n ";


}else {            EV<<"ce n'est pas  un attaquant \n ";}
```

Figure III. 15– Détection de l'attaquant.

5. Simulation

Une fois que nous avons implémenté notre algorithme PAP4FIFA, nous avons passé aux tests via des simulations de quelques scénarios. Dans cette partie, nous commençons par décrire l'environnement de simulation et les étapes de mise en marche de simulation. Ensuite, nous décrivons la topologie utilisée ainsi que les scénarios à simuler. Enfin, nous présenterons les résultats de notre simulation.

5.1. Environnement du travail

Notre simulation a été faite en utilisant CCN-lite sous le simulateur OMNET++. Ces outils ont été installés sur Ubuntu en utilisant un ordinateur portable. La configuration de l'environnement de travail est résumée dans le tableau suivant :

Matériel	**CPU**	2.6 GHZ
	RAM	8 GO
	Disque Dur	1 TB
Logiciels	**OS**	Ubuntu 16.04
	Omnet++	IDE OMNet ++ V 4.4
	Inetframework	INET Framework V 2.6
	CCN-lite	CCN-Lite 0.3.0

Tableau III.1– Environnement de travail.

5.2. Les étapes de mise en marche du simulateur

Ces étapes sont importantes dans cet ordre pour que ccn-lite peut s'exécuter dans le simulateur Omnet++ :

1. Confirmation des versions des logiciels: Ubuntu 16.04, omnet4.4,inet 2.6,ccn-lite.
2. Importation de inet, et la création du projet.
3. Importation de ccn-lite, et le nettoyage du local et du projet.
4. L'accès au fichier ccn-lite → simulations → exemple1. Ensuite, le cliquer avec la droite et l'exécuter avec Run.

5.3. Topologie utilisée

Pour une meilleure comparaison avec les travaux connexes **[Bof 20]**, nous avons utilisé une topologie contenant (Figure III.16) :

- 2 clients légitimes.
- Faux négative (Client légitime qui peut envoyer des faux intérêts par erreur).
- Utilisateur malveillant (attaquant).
- 3 routeurs.
- Un serveur qui contient des contenus nommés.
- Un administrateur.

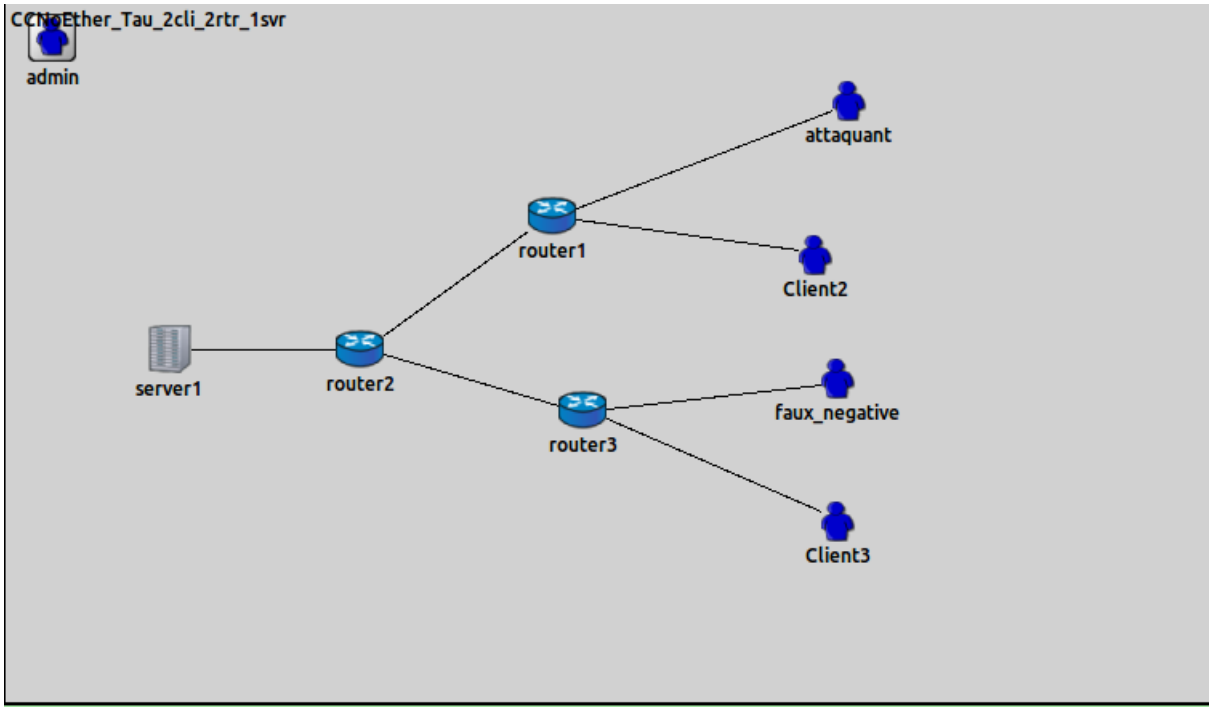

Figure III. 16–Topologie simulé dans le réseau CCN.

La topologie est représentée dans notre simulateur par un fichier de l'extension. NED, qui contient le code source de la topologie, ainsi que 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 est de 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.

```
connections:

    attaquant.ethg[0] <--> fastEthernet <--> router1.ethg[0];
    Client2.ethg[0] <--> fastEthernet <--> router1.ethg[1];

    router1.ethg[2] <--> fastEthernet <--> router2.ethg[0];
    router2.ethg[1] <--> fastEthernet <--> server1.ethg[0];
    Client3.ethg[0] <--> fastEthernet <--> router3.ethg[0];
    faux_negative.ethg[0] <--> fastEthernet <--> router3.ethg[1];
    router3.ethg[2] <--> fastEthernet <--> router2.ethg[2];
```

Figure III. 17– Les connexions fast Ethernet entre les nœuds dans le fichier de topologie. NED.

5.3.1. Routage et chargement des données :

Comme les réseaux IP, les réseaux ICN ont besoin de routage et de chargement de données. Pour cela, nous avons besoin d'un fichier de configuration sous extension **.cfg**. Ce fichier est composé de trois parties principales et une partie de commentaires.

- **[eInterestMode]** : C'est le PIT du nœud CCN, qui contient les intérêts nommés demandés par le nœud source (les données nommées), avec les attributs :
  - **ContentName** : Nom du contenu.
  - **StartChunk** : Début du segment.
  - **ChunksCount** : Nombre des segments.
  - **Request Time** : Temps de réponse.
- **[ePreCacheMode]** : Représente le CS (cache) pour stocker les contenus et 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 est équivalente à la table FIB, et elle contient les informations de routage suivant :
  - **ContentPrefix** : Le préfix du nom de contenu.
  - **NextHop**: L'interface de nœud suivant.
  - **Access From** : L'interface de sortie.
  - **Update Time** : Le temps de mise à jour.
- **[eCommentsMode]** : Là on peut insérer des commentaires.

```
[eInterestMode]

ContentName = /D/L/C/C2 , StartChunk = 0 , ChunksCount = 1 , RequestTime = 100/*s*/
ContentName = /D/L/C/C2/faux , StartChunk = 0 , ChunksCount = 1 , RequestTime = 100/*s*/
ContentName = /D/L/C/C2/faux , StartChunk = 0 , ChunksCount = 1 , RequestTime = 100/*s*/
ContentName = /D/L/C/C2/faux , StartChunk = 0 , ChunksCount = 1 , RequestTime = 100/*s*/



[ePreCacheMode]

[eFwdRulesMode]

ContentPrefix = /D/L/C , NextHop = router1.eth[0] , AccessFrom = attaquant.eth[0] , UpdateTime = 0/*s*/
ContentPrefix = /D/L/sc , NextHop = router1.eth[0] , AccessFrom = attaquant.eth[0] , UpdateTime = 0/*s*/


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


ContentPrefix = /D/L/C , NextHop = router1.eth[0] , AccessFrom = attaquant.eth[0] , UpdateTime = 0/*s*/
ContentPrefix = /D/L/R , NextHop = router1.eth[0] , AccessFrom = attaquant.eth[0] , UpdateTime = 0/*s*/
ContentPrefix = /D/L/sa , NextHop = router1.eth[0] , AccessFrom = attaquant.eth[0] , UpdateTime = 0/*s*/
ContentPrefix = /D/L/M , NextHop = router1.eth[0] , AccessFrom = attaquant.eth[0] , UpdateTime = 0/*s*/
ContentPrefix = /D/L/sc , NextHop = router1.eth[0] , AccessFrom = attaquant.eth[0] , UpdateTime = 0/*s*/
ContentPrefix = /D/J , NextHop = router1.eth[0] , AccessFrom = attaquant.eth[0] , UpdateTime = 0/*s*/
ContentPrefix = /D/Di/En , NextHop = router1.eth[0] , AccessFrom = attaquant.eth[0] , UpdateTime = 0/*s*/
ContentPrefix = /D/Di/S/bio , NextHop = router1.eth[0] , AccessFrom = attaquant.eth[0] , UpdateTime = 0/*s*/
ContentPrefix = /D/Di/S/g, NextHop = router1.eth[0] , AccessFrom = attaquant.eth[0] , UpdateTime = 0/*s*/
ContentPrefix =  /D/Di/S/nom , NextHop = router1.eth[0] , AccessFrom = attaquant.eth[0] , UpdateTime = 0/*s*/
ContentPrefix = /D/E/al, NextHop = router1.eth[0] , AccessFrom = attaquant.eth[0] , UpdateTime = 0/*s*/
ContentPrefix = /D/E/CO , NextHop = router1.eth[0] , AccessFrom = attaquant.eth[0] , UpdateTime = 0/*s*/
```

Figure III.18– Fichier de configuration d'un nœud de type attaquant.

5.3.2. Liaison et configuration des nœuds

Après la création de la topologie, ainsi que la configuration des composants du réseau ICN, nous avons forcément besoin de faire une liaison entre les fichiers de configuration et les appareils connectés. Pour cela, nous disposons d'un fichier qui s'appelle **.INI**, et ce dernier a pour rôle de lier la configuration avec le nœud qui le correspond.

Comme nous pouvons le constater dans la Figure **III.19,** le routeur 1 correspond au fichier appelé **router1_ccn.cfg**, et ainsi de suite pour les autres nœuds.

```
*.attaquant.net.ccnScenarioFile = "attaquant_ccn.cfg"
*.Client2.net.ccnScenarioFile = "client2_ccn.cfg"
*.Client3.net.ccnScenarioFile = "client3_ccn.cfg"
*.faux_negative.net.ccnScenarioFile = "faux_negative.cfg"
*.router1.net.ccnScenarioFile = "router1_ccn.cfg"
*.router2.net.ccnScenarioFile = "router2_ccn.cfg"
*.router3.net.ccnScenarioFile = "router3_ccn.cfg"
*.server1.net.ccnScenarioFile = "server1_ccn.cfg"
```

Figure III.19– Fichier INI de notre topologie.

5.4. Les résultats obtenus

Après le lancement de la simulation (voir la figure III 20), les nœuds transmettent les paquets entre eux, et ces paquets sont de trois types :

1. Paquets qui sont un contenu demandé.
2. Les vrais intérêts qui retournent un résultat (un contenu).
3. Les faux intérêts qui demandent un contenu inexistant.

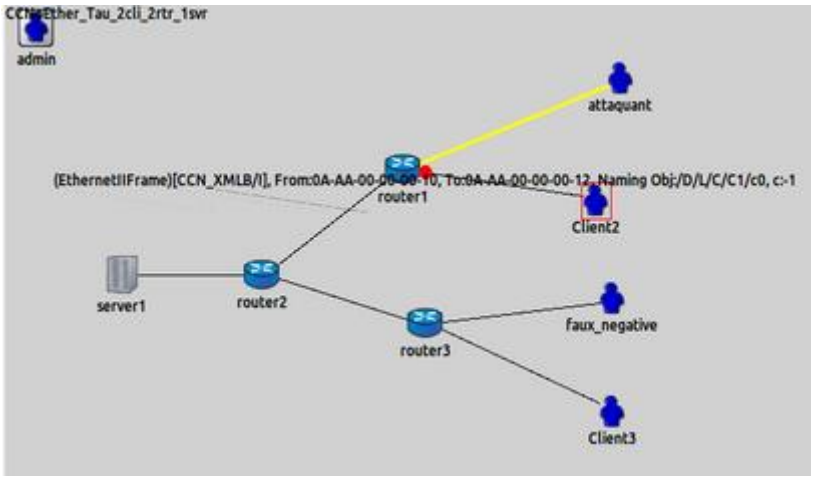

Figure III.20– Simulation.

Depuis ce traitement, nous pouvons détecter si l'interface est en train de recevoir une attaque ou pas :

Temps (en ms)	Entré	Nombre d'intérêts	Nombre des faux intérêts	Détection
10.426	Attaquant	20	20	il a détecté que c'est un attaquant
154.564	Client 2	20	0	il a détecté que ce n'est pas un attaquant
9.236	Faux négatif	15	5	il n'a pas détecté que c'est un attaquant
155.514	Client 3	10	0	il a détecté que ce n'est pas un attaquant

Tableau III. 2– Déroulement de la simulation avec intérêt.

6. Comparaison entre les principaux algorithmes de contre mesure

Notre algorithme PAP4FIFAest une solution simplifiée de l'algorithme principal PAP qui rentre, dans l'ensemble des contre-mesures des attaques FIFA (False InterestFloodingAttack) tel Poséidon et Traceback objet de notre étude comparative.

Pour mieux se positionner avec ces travaux connexes déjà comparés dans la section 5.4 du chapitre 2, nous essayerons de trouver pour chaque critère de similarité avec quel autre algorithme (Traceback, Poséidon ou PAP) PAP4FIFA est plus proche. Le rapprochement est illustré dans le tableau III.2.

Critères de similarité	**PAP4FIFA**	**Algorithme(s) Similaire(s) parmi Poseidon, PAP et TraceBack**
Vitesse d'exécution	Rapide	PAP
Précision	100%	Aucun
Taux d'erreur	0 %	Aucun
Paquetage de programmation	CCN-lite	Poséidon et Traceback
Type d'approche de contre mesure	Push/Back	Poseidon et PAP
Atténuation progressived'attaquant	Oui	Tous
Possibilité de blocage d'unclient légitime	Non	PAP
Blocage définitif d'uneinterface	Oui	Traceback
Configuration de chaqueinterface (chemin) reliée parun nœud de manière différenteque les autres interfaces.	Oui	Tous
Chaque interface (chemin)d'un nœud a un niveau desensibilité que les autresinterfaces.	Oui	Tous
Installation dans chaque nœud CCN	Oui	Tous
Il fait des mises à jour de PIT	Oui	Traceback et PAP
Si on fait une mise à jour surle réseau, il faut faire unenouvelle configuration sur lacontre-mesure	Non	PAP
Collaboration entre routeurs	Oui	Poséidon et PAP
Le nœud de déclencheur de l'algorithme	Serveur	PAP

	victime	

Tableau III. 3– Comparaison de notre algorithme avec les principaux algorithmes de contre-mesure.

La comparaison présentée précédemment ne peut être décisive ou exhaustive puisque à cause du manque de temps, nous n'avons pas implémenté l'algorithme PAP pour tester ses performances dans le même environnement et sur la même topologie que les autres algorithmes. Aussi, le nombre de tests effectués sur PAP4FIFA sont très limités.

Par ailleurs, nous pouvons dire que notre solution peut être prometteuse puisqu'elle englobe tous les avantages de PAP alors qu'elle est plus légère car elle ne se concentre que sur un seul type d'attaque, les avantages et faiblesses de notre solution sont présentée dans le tableau III.3.

	Points forts	Points faibles
PAP4FIFA	- Collaboration entre les routeurs - une meilleure amélioration de leur performance et diminutiondu taux d'erreurs (surtout en termes de faux positifs). - Possibilité d'intégration de plusieurs mécanismes de suppression de DDoS - Utilisation de temporisateurs pour supprimer les intérêts suspects. - Mise à jour dans la table PIT.	- Difficulté de définir le bon seuil d'espace mémoire occupé pour d'activer le mécanisme de trace. - Possibilité de bloquer l'interface de façon définitive - Déclenchement de l'algorithme par le serveur victime.

Tableau III. 4– Les points faibles et les points forts des algorithmes de contre-mesure.

Même si notre solution apporte beaucoup d'avantages telle la collaboration entre les différents routeurs pour la détection des utilisateurs malveillances, point non traité dans Poséidon et Traceback, avoir le choix entre l'utilisation de solutions de blocage définitif ou temporaire des attaquants et donc englobant les deux techniques des deux algorithmes précédents, toutefois laisser le serveur déclencher le mécanisme lorsqu'il est attaqué et ne pas bloquer les attaques plutôt diminue considérablement dans les performances du serveur, le temps de réponse, la surcharge de ce dernier est un point faible pour un réseau tel CCN qui se veut très rapide et efficace.

7. Conclusion

Pour la simulation des réseaux plusieurs techniques de modélisation ont été créées, qui ont pour but de voir le comportement des nœuds et leur fonctionnement. Le fait que dans les réseaux CCNs la simulation réelle du réseau est impossible pour l'instant, nous avons travaillé avec le simulateur « NS3 », malheureusement la solution n'a pas abouti à un résultat nous nous sommes orientés vers « OMNET++ ». Nous avons présenté en premier lieu l'environnement OMNET++ ensuite nous avons fait une description pour le package CCN-lite. Nous avons proposé une solution simplifiée de l'algorithme PAP appelée PAP4FIFA, que nous avons déployée sur l'environnement virtuel. Ceci nous a permis de tester la robustesse de notre solution que nous avons comparée à la fin avec d'autres contre-mesures.

Conclusion Générale& Perspectives

Conclusion Générale& Perspectives

Le réseau centré sur l'information ICN regroupe une nouvelle famille de protocoles réseau qui cherchent à améliorer les communications réseau étant donné l'usage actuel que l'on fait d'Internet pour rechercher des contenus spécifiques. En se basant sur le nom de contenu au lieu des adresses IP des hôtes, ICN a montré ses avantages dans la résolution du problème de nommage, distribution et routage du contenu ainsi que le problème de mobilité des utilisateurs. Cependant, les problèmes de sécurité persistent toujours et doivent être résolus afin de faire progresser cette architecture prometteuse. Cette dernière fait apparaître, en plus des attaques héritées, des nouvelles attaques de sécurité liées au nommage, au routage, au cache et autres. Dans le cadre de ce projet, notre objectif était de développer une nouvelle solution contre les attaques zombies (liées au routage) dans CCN permettant de détecter rapidement ce type des attaques et de réagir de manière efficace en repoussant le trafic vers les attaquants.

Pour atteindre cet objectif, nous avons commencé par une étude bibliographique dans le but d'avoir une compréhension globale de l'architecture d'ICN, ses fonctionnalités et ses projets de déploiement, notamment les projets CCN et NDN. Ensuite, nous nous sommes concentrés sur les attaques ICN liés au routage, en particulier les attaques d'inondation des intérêts qui peuvent être lancées par un réseau de zombies en provoquant un déni de service (Dos). Enfin, nous avons terminé cette étude bibliographique par présenter les approches de contre mesure et les principaux algorithmes existants comme Traceback, Poséidon et PAP. Ce qui nous a permis d'opter pour l'algorithme de PAP comme une nouvelle solution à développer et la comparer avec les autres solutions déjà implémentées. Au fait, cet algorithme peut d'abord repousser de manière efficace le trafic d'attaque en quelques secondes vers des entités qui se comportent mal, à une granularité beaucoup plus fine que les mécanismes existants, ensuite supprimer l'attaque.

En passant au déploiement de cette solution, nous avons rencontré des problèmes de mise en place du code source PAP existant sur les réseaux NDN simulé (ndnSIM) sur le simulateur NS3. Ce qui nous a poussés à utiliser le réseau CCN (CCN-lite) sous le simulateur OMNET++ et ainsi à proposer une version simplifiée du PAP (que nous avons appelé PAP4FIFA), facile à implémenter vu la complexité et la lourdeur de la solution originale de PAP qui traite plusieurs types d'attaques. Notre version de l'algorithme permet de traiter seulement l'attaque d'inondation des faux intérêts par des zombies en bloquant les interfaces des attaquants. Les résultats de simulation montrent que notre solution allégée peut efficacement repousser le trafic d'attaque en quelques secondes vers l'attaquant.

Toutefois, et faute de temps, nous n'avons pas pu arriver à :

- Implémenter la version complète de PAP (mettre en œuvre les autres attaques A et C et d'autres mécanismes de suppression DDoS que le blocage)
- Evaluer les performances de PAP par des simulations approfondies pour confirmer les résultats annoncés dans [Zha 18].
- Evaluer les performances des autres solutions Poséidon, Traceback pour faire une comparaison globale entre les trois algorithmes.

Et comme perspectives, plusieurs améliorations peuvent y être apportées dans le cadre de ce projet de recherche, notamment :

- Etudier et développer d'autres solutions de contre-mesure pour les attaques IFA, voire autres attaques DDoS et de zombies.

- Développement d'un outil de comparaison des solutions de contre-mesure pour les attaques de zombies permettant d'intégrer facilement une nouvelle solution implémentée dans n'importe quel simulateur ICN.

Bibliographies

RéférencesBibliographiques

[Amp 13]Alberto Compagno; Mauro Conti; Paolo Gasti; Gene Tsudik«Poseidon: Mitigating Interest Flooding DDoS Attacks in Named Data Networking», Journal, In 38th Annual IEEE Conference onLocal Computer Networks, Sydney-Australia, 21-24 Oct 2013.

[Api 13]Alexander Afanasyev, Priya Mahadevan, Ilya Moiseenko, ErsinUzun, Lixia Zhang«Interest Flooding Attack and Countermeasures in Named Data Networking»,Universitéde California, California-Los Angeles, 2013.

[Aub 17] Aubry Elian «Protocole de routage pour l'architecture NDN», Thèse de Doctorat,Université de Lorraine,Lorraine-France, 2017.

[Bal 16]BalkisHamdane«Réseaux du futur : sécurité et nommage», Thèse de doctorat, École supérieure des communications, Tunis, 22 Novembre 2016.

[Bda 12]Baugher Mark, Davie Bruce, Ashok Narayanan, David Oran«Verifying names for read only named data»,IEEE Conference on Computer Communications Workshops, Boston-Etats unis, 2012.

[Bof 20]Boukhatache Mohamed Abdelfetah et Fekrini Mohamed Billel«Proposition de Mécanisme de Sécurité Contre Les Attaque De Zombies et interception de contenu dans les CCN», mémoire de master, Université Saad Dahleb Blida,Blida-Algérie, 2020.

[Bou 20]Boubakr Nour «ICN Communication Optimization for Internet of Things», Mémoire Hal, Institut de technologie de Pékin,Pékin-Chinoise, 2020.

[Bpj 11] Bertrand Mathieu, Patrick Truong, Jean-François Peltier, You Wei, Gwendal Simon«Informationcentric networking: currentresearchactivities and challenges», Mémoire HAL, France, 2011.

[Csc 02] Chang-Shen Chen, Shian-ShyongTseng, Chien-Liang Lui«Aunifyingframework for intelligent DNS management», Journal, Université national Chiao-Tung 1001, Hsinchu-Taïwan, 2002.

[Drb 20]Driouch Asma et Boujama Hasna«Développement d'un Système de gestion des urgences sismiques basé sur des communications opportunistes de Smartphones dans les réseaux centrés sur l'internet (ICN)», mémoire de master, Université Saad Dahleb Blida, Blida-Algérie, 2020.

[Ehm 15]Eslam G. AbdAllah, Hossam S. Hassanein, Mohammad Zulkernine«A Survey of Security Attacks in Information-Centric Networking», Journal, IEEE communication Surveys & Tutorials,New York-American, juil2015.

[Eli 17] Elian Aubry«Protocole de routage pour l'architecture NDN»,Thèse de doctorat, Université de Lorraine, Lorraine-Frane, 2017.

[Fab 13] Fabian Oehlmann«Content-Centric Networking »,Journal, Université technique de Munich, allemande, 2013.

[Gup 17] B B Gupta «PredictingNumber of Zombies in DDoS AttacksUsing Pace Regression Model», Journal, Journal de l'informatique et des technologies de l'information, États-Unis d'Amérique, April 2012.

[Hak 17]Hachad Amine et KhanissiBalaoua«Etude et Implémentation des Mécanismes de Sécurité pour le Routage Centré Contenu», mémoire de master, Université Saad Dahleb Blida, Blida-Algérie, 2017.
[Ham 19]Hamza Ben Ammar «Onmodels for performance evaluation and cache resources placement in multicache networks», Thèse de Doctorat, Université de Rennes 1 comue Université Bretagne, France, 2019.
[Hyj 13]Huichen Dai, Yi Wang, Jindou Fan, Bin Liu«Mitigate DDOSAttacks in NDN by Interest Traceback», Journal, Université Tsinghua, Turin-Italy, Apr 2013.

[Mam 14]M. Aamir, M. Zaidi«Denial of-service in content centric (named data) networking : a tutorial and state-of-the-art suvey», Journal, Computer Science , États-Unis d'Amérique, 2014.

[Msr 12]Md. Faizul Bari, Shihabur Rahman Chowdhury, Reaz Ahmed, Raouf Boutaba, Bertrand Mathieu, David R«survey of naming and routing in information centric networks»,Journal, University of Waterloo IEEE Communications Magazine, Lannion-France, 2012.

[Med 17]MeddebMaroua.«Information-Centric Networking, A natural design for IoT applications ?». Thèse de doctorat, INSA de Toulouse, Toulouse-France, 2017.

[Nah 16] NarayanChhetry, H. Kalita«InterestFlooding Attack in Named Data Networking: A Survey», Journal, ADBU-Journal de la technologie de l'ingénierie, États-Unis, 2016.

[Osg 17]OnurAscigil,Sergi René, George Xylomenos, IoannisPsaras, George Pavlou«A keyword-basedICN-IoT platform», Journal, , Université London, London-Angleterre, 2017.

[Pio 17] Piotr PawelLaskowski«Internet Security – Technology and social Awarenessof the dangers», Journal, Université de Bialystok, Białystok-Pologne, 2017.

[Raw 18]Ramla Mohamed et Walid Miloud Dahmane «Etude et Implémentation des Mécanismes de Sécurité pour le Routage Centré Contenu», mémoire de master, Université Saad Dahleb Blida, Blida-Algérie, 2018.
[Rst 17] Reza Tourani, SatyajayantMisra, Travis Mick, GauravPanwar «Security, Privacy, and Access Control in Information-Centric Networking: A Survey», Université d'État du Nouveau-Mexique, California-Los Angeles, Juin 2017.

[Sbm 18]SobiaArshad, Babar Shahzaad, Muhammad Awais Azam, Jonathan Loo, Syed Hassan Ahmed, Saleem Aslam«Hierarchical and Flat based Hybrid Naming Scheme», IEEE internet of things journal, London-Angleterre, 2018.

[Srt 15]Sravya Mundra, Tarik El Taeib«TCP/IP Protocole layering», International Journal of Computer Science and Information Technology Research, Bridgeport-California, January - March 2015.
[Tyg 17] TasnuvaMahjabin, Yang Xiao, Guang Sun, Wangdond Jiang «A survey of distributed denial-of-service attack, prevention, and mitigation techniqus», Journal international des réseaux de capteursdistribués, University of Alabama, Tuscaloosa-États-Unis,2017.

[Wei 14] Wei You «A Content-Centric Networking Node for a Realistic Efficient Implementation and Deployment», Thèse de Doctorat, Université de Rennes 1,Bretagne-France, 2014.

[Xue 17]XueHaoyue, LiYuhong, Rahmani Rahim, Kanter Theo, X Que«A mechanism for mitigating DoS attack in ICN-based internet of things», Journal, Association des Machines Informatiques (ACM),États-Unis, 2017.

[Zhi18]Zhiyi Zhang « Référentiel GitHub : implémentation PAP», 2018.

En ligne https://github.com/Zhiyi-Zhang/NDN-DoS-Simulation[Consulté le 24 juillet 2021].

[Zvj 18]Zhiyi Zhang,Vishrant Vasavada, Jonathan Lin, Siva Kesava Reddy K, Peter Reiher,Lixia Zhang«Producer-Assisted Pushback», Technical Report NDN-0065, California-Los Angles, Aug 2018.