REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR

ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE SAAD DAHLAB DE BLIDA 1

FACULTE DES SCIENCES

PROJET DE FIN D'ETUDE

POUR L'OBTENTION DU DIPLOME DE MASTER EN INFORMATIQUE

SPECIALITE : SECURITE DES SYSTEMES D'INFORMATION

THEME

Proposition d'un protocole de sécurité contre les attaques du cache pour les réseaux centrés contenus

Mémoire réalisé par :

Dahmani Mohamed

Zouaoui Farouk

Promotrice : ARKAM Merièm

Année Universitaire :2019- 2020

Résumé

Aujourd'hui, les réseaux IP sont dépassés, et ne répondent plus aux exigences des usagers à savoir la disponibilité, la vitesse et la confidentialité, pour cela plusieurs chercheurs ont mis en place un nouveau modèle de réseau, un modèle centré sur l'information. Ce réseau appelé ICN pour est doté d'une nouvelle architecture, et un nouveau mécanisme de communication assurant la disponibilité de l'information et la vitesse de transmission, toutefois, comme les réseaux IP, il reste très vulnérable aux attaques. Dans notre travail, on s'intéresse aux attaques liées aux données et plus exactement au cache (emplacement des données au niveau des routeurs), et on a proposé un mécanisme de sécurité inspiré du protocole « Kerberos » afin de garantir la transmission de données de manière plus sécurisée.

Mot clé : CCN ICN Cache Usurpation d'identité Kerberos Chiffrement, Déchiffrement, Signature électronique

اصبحت اليوم شبكات IP قديمه جدا ولا تلبي احتياجات المستخدمين من حيث السرعه و السرية والتوافر للمعلومه لذلك كثف الباحثون عملهم الاختراع شبكه جديده اسمها شبكه ICN وقاموا بتفعيل ميكانيزمات جديده لهذه الشبكه حيث تضمن سرعه فائقه وتوفير للمعلومه ايا كانت لكن الجانب سريه المعلومه ليس محفوظ مئه بالمئه لذا في عاملنا هذا قمنا بالتخصص في جانب واحد الا وهو Cache حيث تستلم المعلومه فيه، فلقد اقترحنا ميكانيزم ماخوذ من بروتوكول اسمه Kerberos لي زمان جوده عاليه في نقل المعلومات من حيث السريه

الكلمات المفتاحية : ICN , CCN تخزين ,إنتحال شخصية ,كربروس ,تشفير ,فك التشفير, توقيع الرقمي

Today, IP networks are outdated, and no longer require to users demands in availability, speed and confidentiality so because of that, some researchers have implemented a new network model, a model centered on information. This network called ICN has a new architecture, and a new communication mechanism ensuring the availability of information and the speed of transmission, however, like IP networks, it's still vulnerable to attacks.

In our work, we are interested in attacks linked to data and exactly to the cache (location of data in routers cache), and we suggested a new security mechanism inspired from the "Kerberos" protocol in order to guarantee more security for the transmission of data

Key words: CCN, ICN, cache content, KERBEROS, cryptography, electronic signature, Identity theft.

Remerciement

Merci à Allah le miséricordieux de sa grâce, source de notre force et courage tout au long de nos études universitaires. La réalisation de ce mémoire a été possible grâce au concours de plusieurs personnes à qui nous voudrons témoigner toute notre gratitude. Nous voudrons tout d'abord adresser toute notre reconnaissance et notre profonde gratitude à MMe.ARKAM de nous avoir encadrés dans notre mémoire de fin d'études ainsi que pour sa patience, sa disponibilité et surtout ces judicieux conseils. Et aussi pour ses encouragements. Nous désirons aussi remercier les professeurs de l'université de Blida, qui nous ont fourni durent ces 5 dernières années leurs savoir. Nous remercions les membres du jury pour avoir accepté d'évaluer notre travail. Nous voudrions exprimer notre sincère reconnaissance envers nos familles pour leur soutien aussi bien moral que financier et pour leurs sacrifices. Un grand Merci aux amis et collègues qui nous ont apporté leurs soutiens moral et intellectuel tout au long de notre démarche. Nous ne pourrons terminer sans remercier tous ceux qui ont participé de près ou de loin dans l'élaboration de ce projet de fin d'études.

Merci !!!

Dédicaces

Je dédie ce travail qui n'aura jamais pu voir le jour sans les soutiens indéfectibles et sans limite de mes chers parents qui ne cessent de me donner avec amour le nécessaire pour que je puisse arriver à ce que je suis aujourd'hui. Je dédie également ce modeste travail :

~ A mes très chères frères et sœurs pour m'avoir soutenu et aidé tout au long de ce projet.

~ A toute ma famille : mes grands-parents, mes tantes et mes cousins, en particulier Souhail, Bila.

~ A la personne qui m'a soutenu toute l'année, mon binôme et frère de cœur « Mohamed », merci.

~ A tous les enseignants de l'université de Blida.

~ A mes amis : Sofiane, Silia, Meriem, Seif Eddine.

~ Enfin, a tous les étudiants de la promotion 2019 / 2020 SSI.

FAROUK

Sommaire

Liste des figures

Liste des tableaux

Introduction Générale

Dans les prochaines décennies peut être les réseaux IP vont à peine garantir la disponibilité du réseau dans le trio des exigences des utilisateurs (Disponibilité, Vitesse, Confidentialité). Effectivement le développement technologique a vu la possibilité de faire des interventions médicales à distance, la création de villes intelligentes…, mais seul un problème y pose, la vitesse de transmission d'information, c'est pour cela que les chercheurs ont établi une nouvelle architecture de réseaux appelée les Réseaux Centrés Informations, ces derniers garantissent une vitesse de transmission importante et une meilleure disponibilité de contenu vu leur architecture et leur nouveaux protocoles de routage.

Les Réseaux Centrés Information (ICNs) ont introduit des nouveaux concepts et idées dans le domaine de la recherche des protocoles de routage de prochaine génération, proposant une approche alternative à la suite de protocoles TCP/IP bien connue et consolidée. Un ICN envisage un réseau de périphériques de mise en cache intelligents qui transmettent non seulement des bits d'un endroit à l'autre, mais aussi un support du réseau pour fournir aux utilisateurs finaux ce qu'ils sont vraiment intéressés : les données nommées.

Cependant, bien qu'une grande partie de la littérature existante souligne les avantages de ce nouveau paradigme de réseau, toutefois les ICNs sont très vulnérables côté sécurité, sachant que la sécurité du contenu est très importante vu les attaques déjà recensées dans la littérature à savoir : Dénis de service, Blocage mobile, interception etc., nous nous intéressons dans ce travail de proposer un protocole de sécurité afin de résoudre ce problème et plus particulièrement la sécurité du cache des routeurs centrés contenus. Dans ce travail, Nous faisons une étude d’art sur les ICN, les différentes attaques ainsi vulnérabilités qui touchent le cache des routeurs, proposer un mécanisme de sécurité avant de passer à la simulation en de plusieurs scénarios d’attaques en utilisant le package CCN-lite sous le simulateur OMNeT++.

Pour ce faire, nous allons partager notre manuscrit en quatre grand chapitres qui sont :

- **Chapitre I :** Nous présenterons ici les principes de base d'ICN et nous expliquerons l'architecture de l'ICN et ses différentes approches.
- **Chapitre II :** Dans ce chapitre nous présenterons les différents type d'attaquent dans réseau ICN et comment ils mené, ensuite les contres mesures appliqué pour chaque type d'attaque.
- **Chapitre III :** Ce chapitre sera dédié aux attaques PIT ainsi que les différents scenarios possibles.
- **Chapitre IV :** Nous discutons dans ce chapitre-là contre mesure proposée de l'attaque PIT.

Chapitre I :
Les Réseaux Centrés sur l’Information

I.1 Introduction

La demande croissante à une nouvelle architecture de l'internet pour satisfaire les besoins des clients est devenu très intéressant par de la communauté de recherche sur les réseaux en développant une nouvelle architecture au lieu de l'ancienne qui se base sur le modèle bout-en-bout, c'est-à-dire établir des tunnels de communications dans le réseau entre un utilisateur et à un autre.

Au fil des années et avec le nombre d'appareils exponentiellement élevé se trouvant connecté dans le réseau internet, l'architecture classique est devenue de plus en plus complexe pour gérer cette expansion de terminal.

Des chercheurs dans le domaine de communication ont proposé une nouvelle architecture appelée Information-Centric Networking (ICN), cette dernière est considérée aujourd'hui comme étant une nouvelle solution d'internet basée sur le nommage des objets au lieu de la localisation des serveurs.

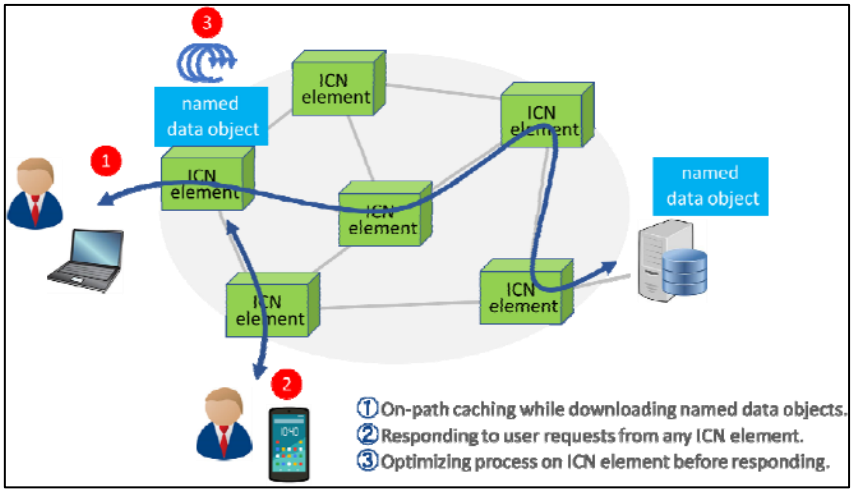
Cette solution a attiré l'attention de plusieurs organisations et communauté spécialisée dans les réseaux informatiques et leurs normalisations.

Nous nous intéressons dans ce chapitre à expliquer ce nouveau concept d'ICN, son contenu et les différentes architectures basées sur le nommage des données existantes dans la littérature, notons ici que nous nous intéressons plus particulièrement à l'architecture CCN que nous détaillerons par la suite.

I.2. Les Réseaux Centrés sur l'information

Le concept ICN est né au moment où les utilisateurs orientent de plus en plus leurs intérêts vers le contenu recherché lui-même plutôt que l'emplacement ou le serveur où le contenu est stocké (rechercher un contenu et non le fournisseur du contenu) (Figure I.1).

Le comportement centré sur le contenu des applications utilisateur a rendu le paradigme de communication point à point des réseaux IP inefficace. En nommant le contenu dans la couche réseau, ICN prend en charge nativement les mécanismes de mise en cache dans le réseau. Donc les utilisateurs envoient leurs demandes comportant le nom du contenu recherché et reçoivent le contenu demandé à partir du routeur ICN le plus proche stockant une copie du contenu dans son cache, au lieu d'acheminer la requête jusqu'au serveur principal (le fournisseur ou le propriétaire du contenu).



FigureI.1 : Fonctionnement de base du ICN **[4]**

I.3. Les concepts de base de ICN [8]

Les Réseaux ICNs ont été initialement proposés pour révolutionner l'architecture de l'internet avec une nouvelle conception de réseau qui prend en charge l'accès aux objets de données nommés. En raison de l'accès aux objets de données, la mise en cache et la réplication en réseau peu coûteuses, ce qui veut dire une meilleure efficacité et une meilleure évolutivité en terme de distribution de données et utilisation de la bande passante du réseau.

Donc les défis de mise en place de l'ICN ont été identifiés : comment nommer les données des objets pour les identifier de manière unique (Nommage), Comment localiser les objets de données nommés et les transmettre aux clients (Routage), comment sécuriser des objets de données nommés circulant dans le réseau (sécurité) et comment soutenir la mobilité dans un environnement centré sur l'information et indépendant du lieu (mobilité) et enfin la mise en cache de données.

Dans cette section nous expliquerons en détail ces différents concepts.

I.3.1 Le Nommage

Le nommage des objets de données est un nouveau paradigme et concept clé pour l'ICN, il permet d'identifier les objets de données indépendamment de leur emplacement, c'est ce qui fait la déférence entre l'architecture actuelle de l'internet. Les noms des objets sont uniques pour cela deux approches existent : le schéma de nommage hiérarchique ou plat.

- ✓ Le nommage hiérarchique est similaire aux URLs utilisés donc le nommage contenant

des segments de chaine séparées par « / », exemple : "univ-blida1/presentation.mp4", il permet l'agrégation des informations de routage et améliore l'évolutivité d'un schéma de routage et les mises à jour. Il ne peut être mis en place que par l'éditeur lui-même.

- ✓ La deuxième approche est le nommage plat qui utilise une valeur hachée de contenu ou son nom par exemple : 0x85520005, il est de langueur fixe et illisible par le client, ces noms sont auto-certifiés.

Ces deux méthodes ont été mise au point pour arriver à donner à chaque contenu un nom unique indépendamment de son emplacement.

I.3.2 Le Routage

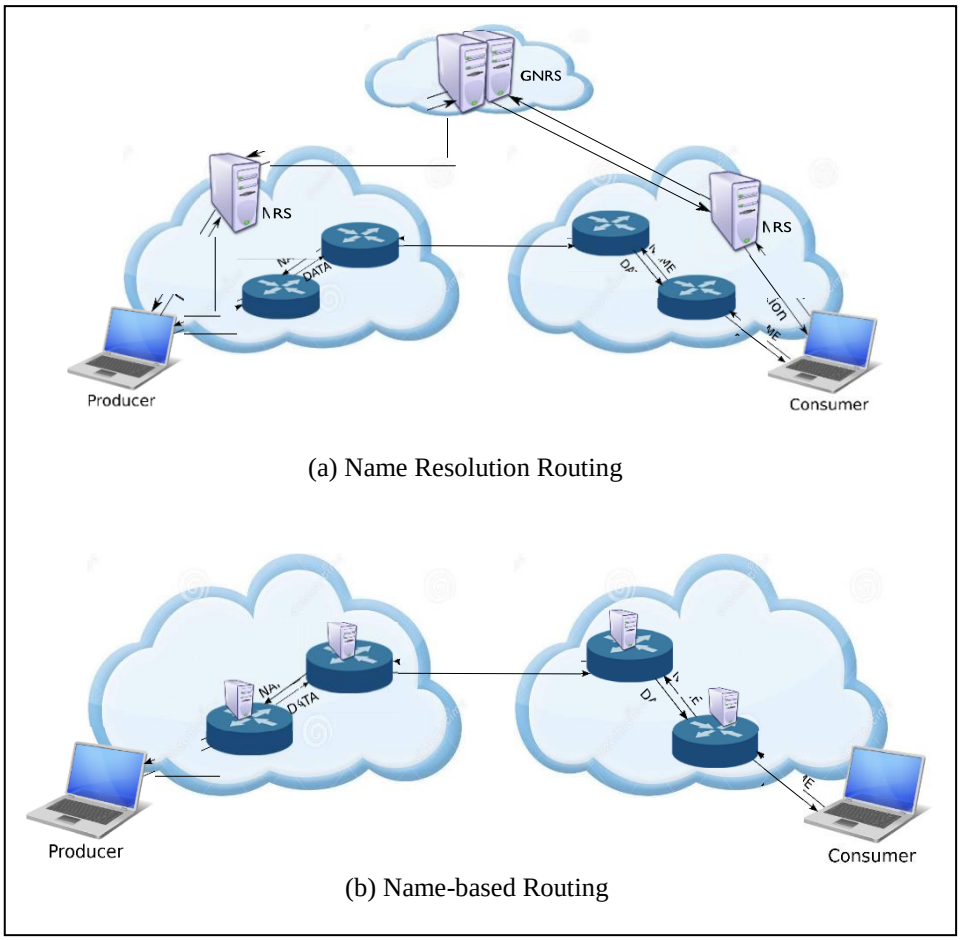
Le routage est une méthode pour répondre à la question de comment acheminer les demandes entre les fournisseurs les clients et dans les ICNs connait deux approches comme montre la Figure I.2 et nous présentons ces deux approches comment fonctionnent dans un réseau CCN **[8]**

La première approche « Name Resolution Routing » :

1. Le nom du contenu est traduit dans un ou plusieurs localisateurs s'il existe.
2. Les localisateurs topologiques peuvent s'agir de l'adresse du fournisseur ou d'un nœud de cache qui conserve une copie de contenu demandé.
3. L'entité qui stocke ces informations de localisation est appelée système de résolution de noms (NRS).
4. Le NRS stocke une liaison entre les noms de contenu et leur localisateur.
5. Les NRS sont organisés de manière hiérarchique et chacun couvre une zone spécifique du réseau, par conséquent, le consommateur doit rediriger la demande à son NRS dédié pour récupérer les informations de localisation.
6. Dans le cas où le NRS ne dispose pas de l'informations de localisation, il redirige la demande vers le NRS global du niveau supérieur.
7. Différents NRS dans la topologie sont peuplées grâce aux messages de signalisation provenant des fournisseurs pour annoncer une disponibilité du contenu.

Le deuxième approche « Name-Base Routing » :

1. Elle se repose sur une seule étape pour récupérer un contenu.
2. Cette approche est basée sur la hiérarchie des noms.
3. Elle achemine les demandes et les transmettre directement aux fournisseur sans qu'il soit nécessaire de résoudre les noms des localisateurs.
4. La demande est envoyée à l'interface suivante qui correspond au préfixe le plus long, ça signifie que chaque nœud doit être conscient d'une partie des informations de routage.
5. Une fois que le fournisseur a reçu la demande, les données sont renvoyées au client via la demande chemin inverse.



(a) Name Resolution Routing

(b) Name-based Routing

Figure I.2 : Approche de routage **[8]**

I.3.3 La Mobilité

Même si le contenu change d'emplacement (de serveur), il sera toujours accessible car son nom est indépendant de l'emplacement physique. La mobilité dans les ICNs peut être classée en mobilité des clients et mobilité des fournisseurs **[12]**.

Donc la mobilité permet aux clients une accessibilité au contenu sans perturbation en raison de modification de l'emplacement et les informations de routage doivent être mise à jour dans chaque nœud du chemin de la requête lorsqu'une mobilité se produit.

I.3.4 La mise-en-cache

Comme vu dans la section précédente, la mobilité est un principe de base de l'ICN et l'emplacement même s'il change les contenus seront toujours accessibles au client et ce principe là (caching) garantie qu'une copie d'un contenu est stockée dans des routeurs dans le réseau.

Et comme montre la figure 3, le client 2 envoie une demande pour récupérer le contenue (X) et cette requête arrivée au sommet de l'hiérarchique et ensuite les nœud 2 et 6 viennent de stocker une copie de ce contenu, Alors si le client 3 ou 1 vont envoyer une requête pour demander le contenu X, ils seront satisfaits par le nœud 2 ou 6 et pas besoin d'aller jusqu'à la source de contenu [**8]**

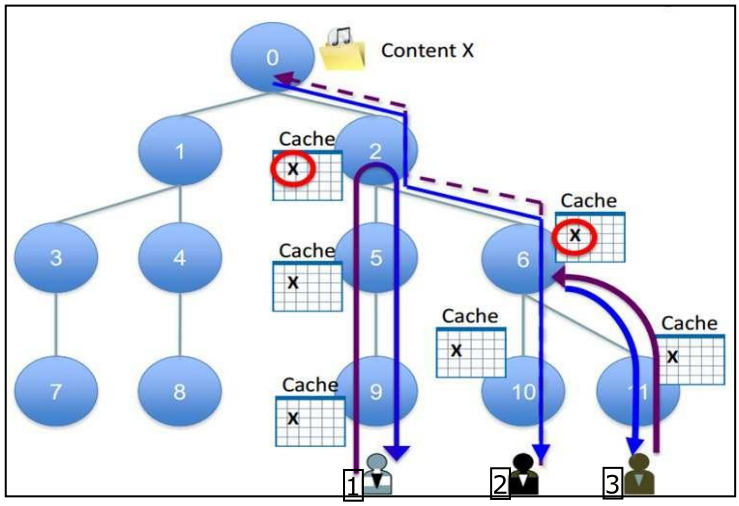


Figure.I.3 : Cache réseau dans ICN **[8]**

I.4 Architecture des ICNs

Dans cette section, nous illustrons les architectures les plus usuelles de l'ICN à savoir : DONA, PSIRP, NetInf et CCN.

Nous présentons ces initiatives en fonction de leur ordre chronologique de publication, comme illustré dans la figure. I.4

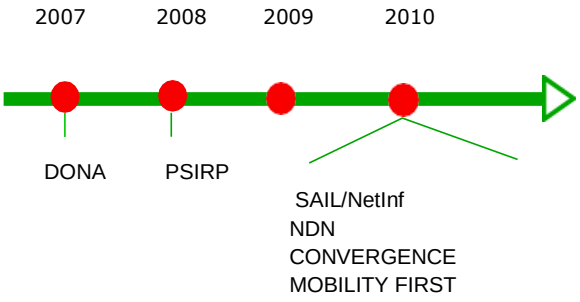


Figure I.4 : Chronologie de développement d'ICN [**8]**

I.4.1 Data-Oriented Network Architecture (DONA)

L'architecture de réseau orientée données (DONA) [Koponen 2007] conçu en 2007 et proposé pour remplacer le nom des URL hiérarchiques ou bien le DNS en noms plats ou auto-certifié.

Les contenus sont publiés dans le réseau par les sources, les nœuds sont autorisés à obtenir de contenu uniquement en s'inscrivant auprès des RH Poignées de résolution (RH - Resolution Handles) au lieu de serveurs DNS.

Le nommage dans DONA vient avec < **P : L** >, le Principal (c'est un contenu publié)

- P : le hach cryptographique de la clé publique de principal
- L : étiquette choisit par le principal qui doit être unique

Format du paquet <**`Data, Public Key, Signature >`**

Le client vérifie les données en comparant le hash de la clé publique avec celui de principal ou bien le contenu publié , c'est la même clé qui génère la signature.

Les RH ont une structure hiérarchique qui peut décider de mettre en cache ou non un contenu, dans cette architecture les opérations principales supportées sont REGISTER (P : L) FIND (P : L), demande de paquets, nommer, rechercher les paquets, ces opérations sont acheminées par leur nom vers la RH appropriée.

Les données de paquet sont renvoyées via le chemin inverse permettant la mise en cache.

Le contenu de DONA doit d'abord être publié ou enregistré pour permettre sa récupération, pour le faire, les fournisseurs de contenu doivent envoyer les messages REGIST ER à son RH local. Ces messages configurent la table d'enregistrement sur chaque RH qui fournit des informations sur le prochain saut et la distance à la copie.
Une fois qu'un contenu donné est enregistré, des demandes ou des messages FIND, peut y être efficacement acheminé. Une fois le contenu localisé, les paquets sont échangés avec le demandeur utilisant le routage IP standard

Les noms dans DONA sont auto-certifiés, les clients reçoivent dans le paquet de données la clé publique du propriétaire et une signature des données elles-mêmes. Ils peuvent vérifier que les données reçues correspondent au nom demandé.

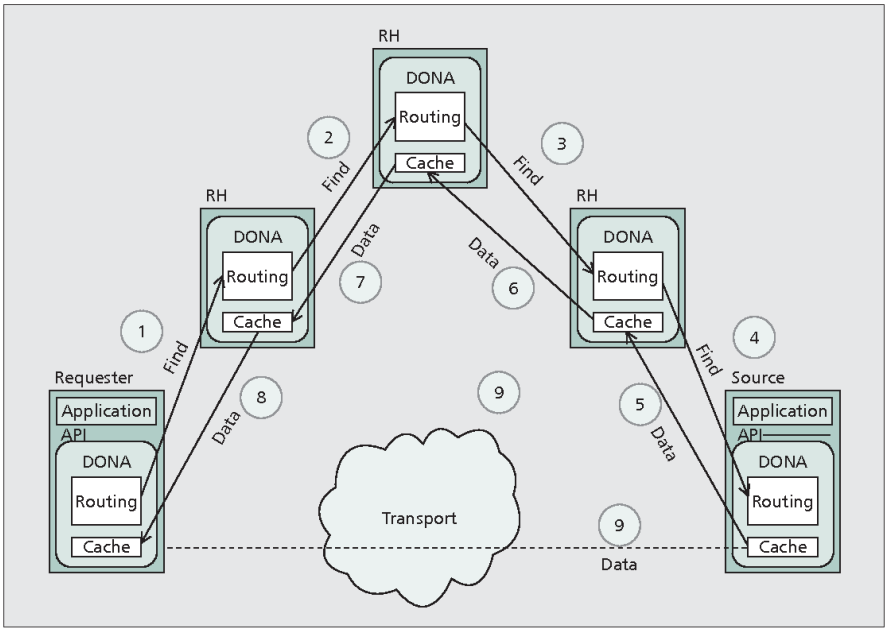


Figure 2. *DONA overview when caching on all resolution handlers (RHs).*

Figure I.5: Vue ensemble DONA **[3]**

1.4.2 Publish-Subscribe Internet Routing Paradigm (PSIRP)

PSIRP est un EU FP7 a été conçu en 2008, il a connu un succès par Publish-Subscribe Internet Technology (PURSUIT) et puis a été introduit comme PSIRP, il propose une nouvelle architecture de l'internet, basé sur PSIRP utilise un système pour chaque donnée

identifiée de manière unique par une paire d'identifiants : un identifiant de rendez-vous (RId) et un identifiant de portée (SId). Le RId est un élément d'information doit être unique dans une étendue, tandis que le SId désigne l'étendue à laquelle appartient un élément d'information.

Le contenu doit d'abord être publié pour en permettre l'accès et pour publier le contenu, l'éditeur doit connaître le SId ainsi que créer un RId pour la publication qui est ensuite transmis au nœud de rendez-vous du réseau de rendez-vous SId.

Un client informe le RId aussi le SId d'une information souhaitée et émet un message d'abonnement vers le point de rendez-vous approprié. Une fois ce message reçu par le point de rendez-vous, un chemin de transfert est créé entre l'éditeur et l'abonné. Chaque publication active possède un identifiant de transfert (FId) qui indique le chemin de transfert à suivre.

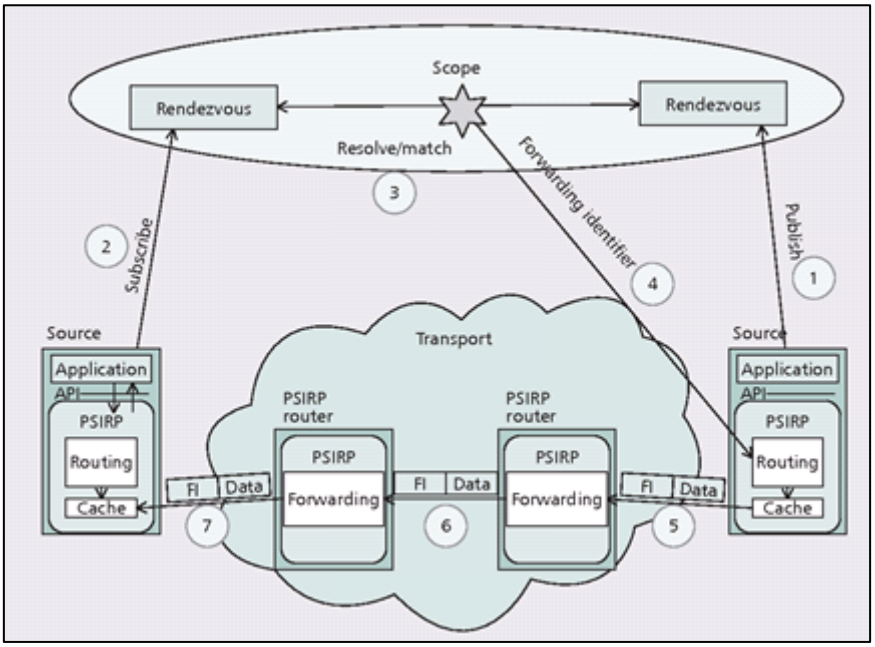


Figure I.6 : Vue d'ensemble PSIRP **[3]**

1.4.3 4WARD NetInf

Network of Information (NetInf) est un projet de FP7 4WARD qui a ensuite évolué dans le cadre du projet FP7 SAIL. Peut-être implémenté en superposition au-dessus de Infrastructure Internet actuelle. Il se concentre sur les questions d'information de niveau

supérieur et représente le contenu sous la forme d'objets dits d'information (IO).

Comme montre la figure I.7 un exemple de routage basé sur la résolution de nom

Les étapes de 5 à 8 les nœuds de NetInf envoient une requête GET jusqu'à trouver une copie de mise en cache ou aller jusqu'au serveur pour la chercher.

Au retour, l'objet peut être mis en cache dans des nœuds traversés ensuite le demandeur initial peut interroger un NRS (étapes 1 et 2) via un message GET pour résoudre le nom de l'objet dans un ensemble d'indices de routage, dans cette exemple c'est la couche inférieure de hot Localisateurs. Par la suite, les indications de routage sont utilisées pour récupérer l'objet via le réseau de transport sous-jacent

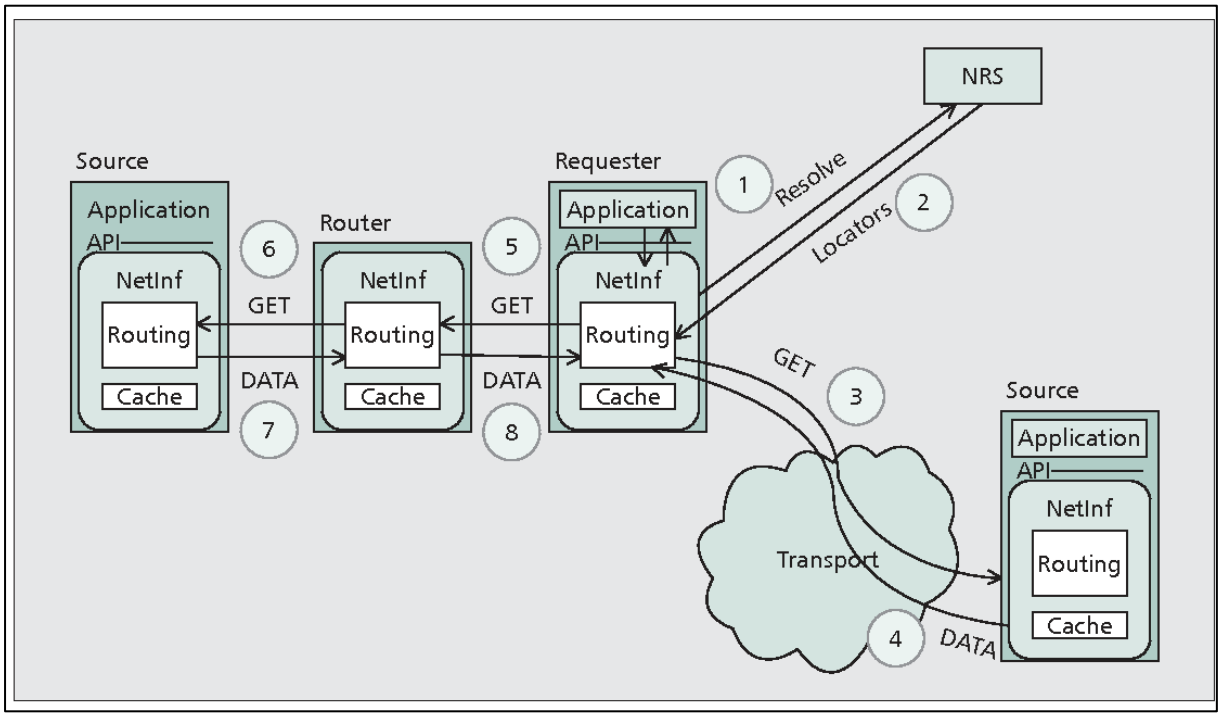


Figure I.7: Vue ensemble NetInf **[3]**

I.4.4 Content-Centred Networking (CCN)

CCN est un Réseaux ICN basé sur le contenu, introduit un schéma de dénomination unique dans lequel chaque contenu est identifié par un ensemble de paires attribut / valeur au lieu de noms hiérarchiques ou d'étiquettes plates. Une demande d'utilisateur est un prédicat de sélection qui est une disjonction logique de conjonctions de contraintes élémentaires sur les valeurs d'attributs individuels.

L'appariement de la publication et de l'abonnement est un processus de recherche pour trouver des contenus qui correspondent aux prédicats de sélection déclarés par les consommateurs. Comme CCN se différencie de toutes les architectures ICN susmentionnées par ses différentes sémantiques, le routage et le transfert nécessitent des approches totalement différentes.

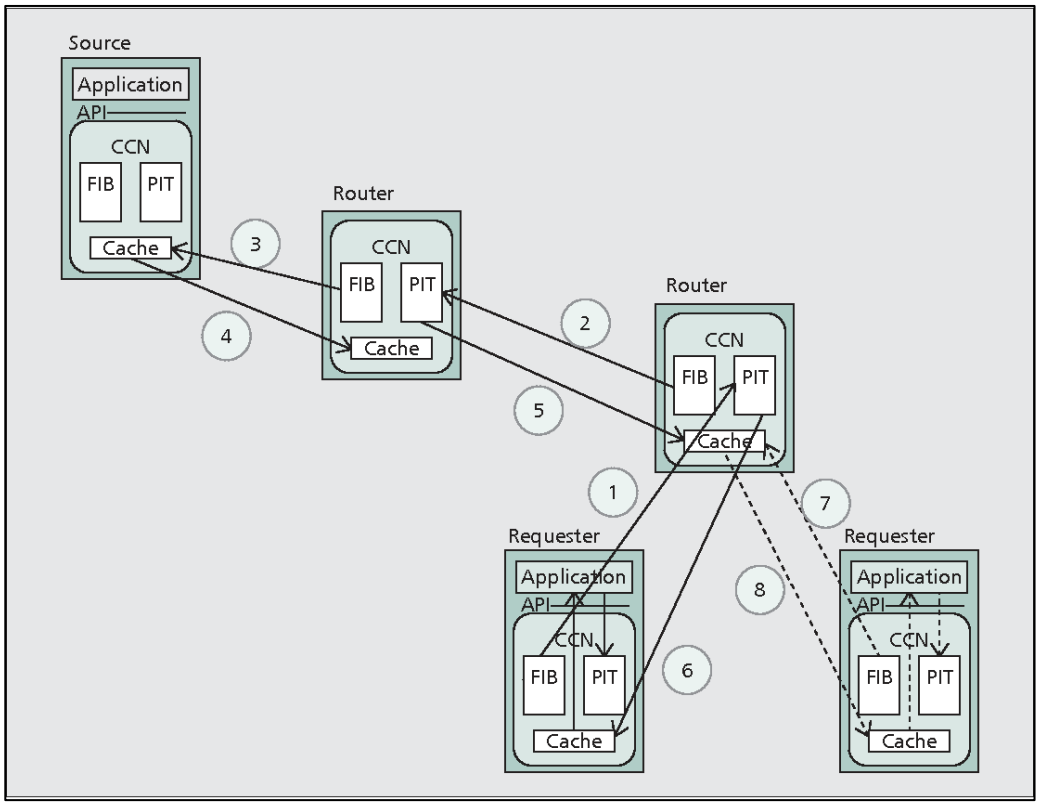


Figure I.8 : Vue d'ensemble CCN **[3]**

I.4.5 Comparaison entre les 4 approches de ICN

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

Tab I.1 : Table de comparaison entre les principes de base d'ICN **[3]**

1.5 Content-Centric Networking

Contrairement à l'architecture Internet basée sur IP, orientée hôte, le réseau centré sur le contenu (CCN) met l'accent sur le contenu en le rendant directement adressable et routable aussi les points de terminaison comme : les pc, les téléphones … etc. communiquent sur la base de données nommées au lieu d'adresses IP.

CCN se caractérise par l'échange de base de messages de demande de contenu (Appelés « intérêts ») et de messages de retour de contenu (appelés « objets de contenu »).

Elle est considérée comme une architecture de réseau centré sur l'information (ICN).

Les objectifs de CCN sont de fournir un réseau plus sécurisé, flexible et évolutif, répondant ainsi aux exigences modernes d'Internet pour une distribution de contenu sécurisée surtout rapide à grande échelle vers un ensemble diversifié de terminaux. CCN incarne un modèle de sécurité qui sécurise explicitement des éléments de contenu individuels plutôt que de sécuriser la connexion. Il offre une flexibilité en utilisant des noms de données au lieu de noms d'hôtes (adresses IP). De plus, le contenu nommé et sécurisé réside dans des caches distribués automatiquement remplis à la demande ou sélectivement pré remplis. À la demande de son nom, CCN fournit le contenu nommé à l'utilisateur à partir du cache le plus proche, traversant moins de sauts de réseau, éliminant les demandes redondantes et consommant globalement moins de ressources.

1.5.1 L'architecture du réseau CCN

L'architecture CCN (Content-Centric Networking) **[7]** est l'une des structures de base du ICN. Dans CCN, les noms sont lisibles et hiérarchiques sur trois éléments du système **[2]** :

- ***PIT*** **:**

  (Pending Interest Table) Il conserve les paquets d'intérêt jusqu'à ce qu'ils soient satisfaits ou que leur durée de vie soit expirée.

  Ensuite les paquets peuvent suivre les chemins inverses pour arriver aux clients demandeurs.

  Lorsqu'un contenu spécifique a plusieurs paquets d'intérêt, le premier paquet d'intérêt est transmis, pendant que tous les autres sont en attente dans PIT et attendent le paquet de données correspondant pour qu'ils soient satisfaits.

CCN Pending Interest Table	
Nom de contenu	**Interfaces**
ccnx:/youtube.com/news/baby_born/video1	face308, face321
ccn:/google.fr/	face103
ccn:/orange.fr/news/meteo/page.html	face201
. . .	. . .

Tab I.2 : Table CCN des intérêts en attente [**6]**

- ***FIB* :**

  (Forwarding Information Base) Il stocke des informations sur les interfaces du paquet d'intérêt et les transmet vers des sources ayant les contenus recherchés, Il maintient les préfixes de noms et les interfaces correspondantes au fournisseur, qui peuvent avoir le contenu demandé ou bien une copié.

CCN Forwarding Information Base	
Nom de domaine /prefixes	**Interface**
ccn:/youtube.com/	face101, face102
ccn:/google.fr	face103
ccn:/orange.fr/news/meteo/	face201
. . .	. . .

Tab I.3 : Table FIB de CCN **[6]**

- ***CS* :**

  (Content Store) Sa fonctionnalité de base est d'optimiser le temps de récupération du contenu, la latence de livraison et d'économiser bande passante.

  Installé dans nœuds CCN, Il est utilisé comme cache temporaire pour les paquets de données car plusieurs utilisateurs peuvent demander le même contenu.

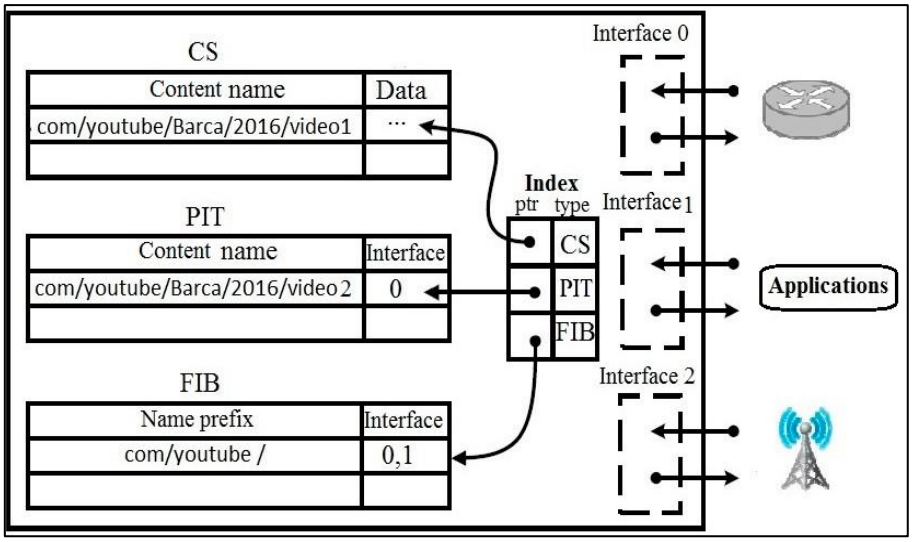


Figure I.9 : Structure du nœud CCN **[2]**

Un client transmet un message d'*intérêt* pour demander des objets d'information.
La demande est acheminée par nom.

CS agit comme un cache donc le CCN vérifie si ce dernier dispose contenu demandé. S'il ne dispose pas de contenu recherché donc c'est un cas d'échec de cache, le paquet d'intérêt est envoyé au saut suivant en fonction de la FIB, il vérifie dans la table PIT s'il reçut un message d'*intérêt* de même contenu demandé via une interface s'il ne trouve alors il stock dans la table PIT le nom du contenu dans le paquet *intérêt* avec l'interface par laquelle le message a été reçu ,ensuite selon la table FIB il transmet le message d'*intérêt* dans le réseau et si une fois il trouve que dans sa table PIT qu'elle contient déjà le nom du contenu correspondant à une interface, il ne va pas transmettre le message reçu et il ne fait que la mise à jour de la table PIT.

Donc c'était pour le cas où cache CS ne dispose pas de contenu et s'il dispose de contenu recherché, il le transmet par l'intermédiaire de la face sur laquelle la message d'*intérêt* a été reçu au client et en fin par l'intermédiaire d'autres nœuds.

NDN prend en charge la mobilité des clients Cependant, si un fournisseur change de l'emplacement donc toutes les FIB doivent être mises à jour et aussi le NDN envoie de nouveaux messages d'*intérêt* à partir de l'emplacement actuel.
Concernant la sécurité, les messages de données en NDN contiennent une signature et la clé publique du signataire. Ainsi, lorsqu'un client reçoit des données, il peut vérifier leur intégrité.

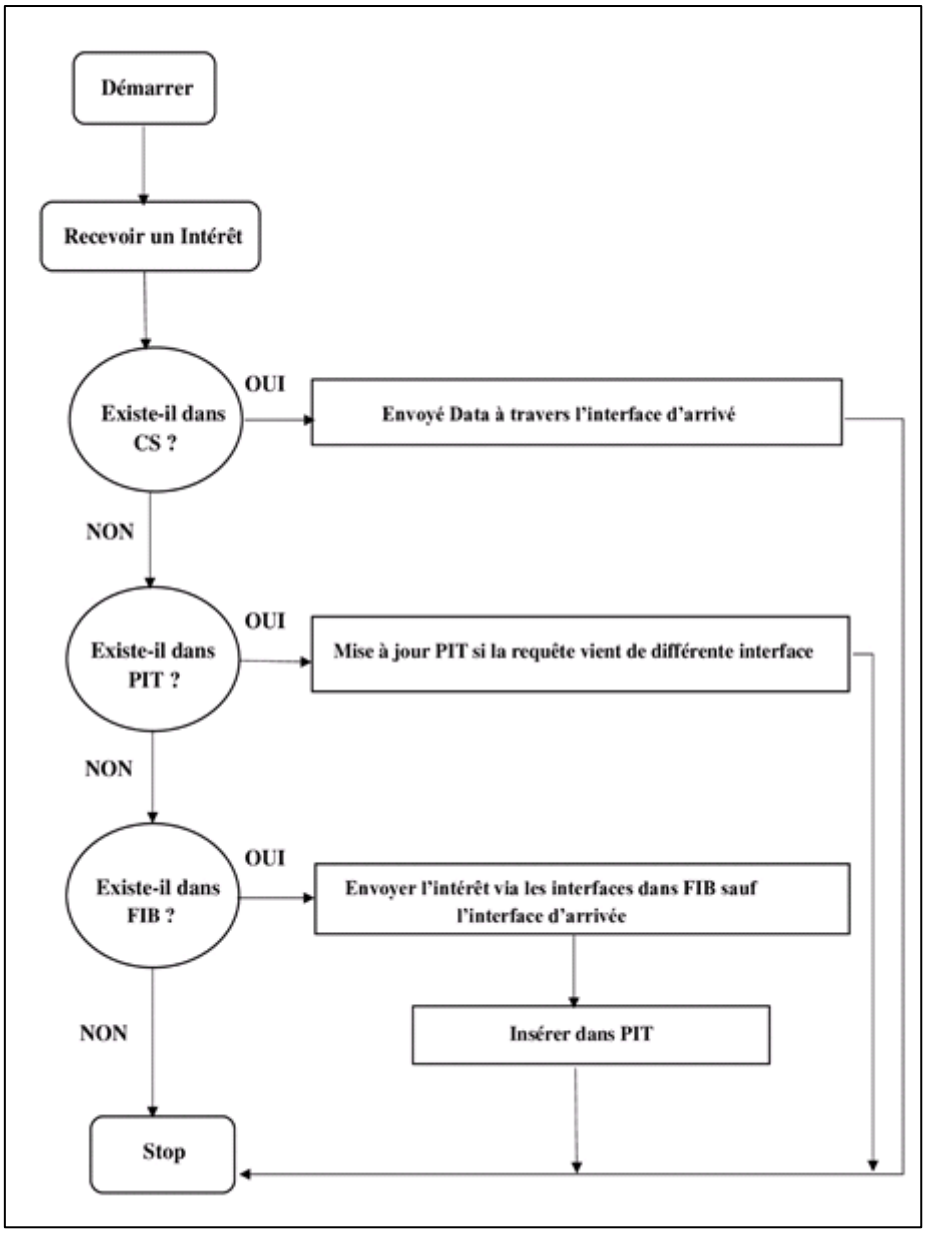


Figure I.10 : Traitement d'un paquet d'intérêt dans CCN

1.5.2 CCN Naming

Le CCN propose d'utiliser des noms hiérarchiques composés de nombres arbitraires des composants, le nommage est similaire aux URL. Par exemple : « /www/youtube/com/video1/001 » peut être un nom de données se référant au premier morceau d'une vidéo youtube et aux composants de nom sont séparés par le caractère /. L'une des motivations à utiliser des noms hiérarchiques est de réutiliser Protocoles de routage IGP / BGP à partir du réseau IP.

Les noms sont censés avoir un sens pour les couches supérieures. En particulier, le composant du nom de famille est classiquement le numéro de séquence utilisé par la couche de transport CCN / NDN (c'est-à-dire similaire au numéro de séquence de TCP ACK dans Fonctionnalité).

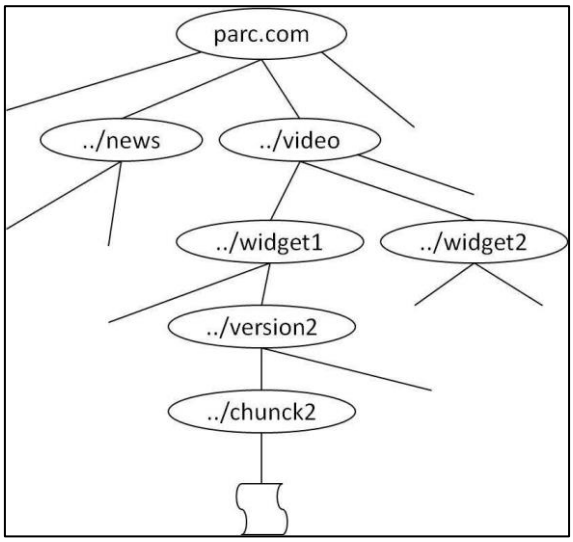


Figure I.11 : Nommage dans CCN [**9]**

1.5.3 CCN Routing

Dans CCN, seuls les paquets d'intérêt sont routés. Le réseau utilise un routage basé sur le nom pour acheminer les intérêts. Bien que le routage soit toujours un sujet de recherche ouvert dans le CCN, il reste par défaut est un routage hiérarchique similaire aux protocoles IGP / BGP utilisés dans les réseaux IP avec la personnalisation nécessaire pour un routage basé sur le nom. **[9]**

1.6 Conclusion

Dans ce chapitre, nous avons présenté les principes de base d'ICN, nous avons expliqué ses différentes architectures trouvées dans la littérature, ainsi que leurs évolutions au fil des années. En plus nous avons représenté dans un tableau ces différentes approches en faisant une comparaison à plusieurs contraintes, ensuite nous avons discuté les différentes attaques liées au cache et quelques solutions proposées jusqu'à maintenant.

Chapitre II :

La sécurité dans ICN

II.1 Introduction

La sécurité dans les réseaux CCN était et est toujours une question importante à cause de plusieurs problèmes majeur de confidentialité trouvé, c'est pour cela jusqu'à aujourd'hui nombreuses études et recherches sont en cours pour trouver des solutions efficaces et optimisés pour régler le problème de confidentialité qui a été touché ou bien la sécurité en général**.** Dans ce chapitre nous discutons trois grands domaines : Les vulnérabilités existantes dans réseau CCN, les trois type d'attaques les plus exploités par les pirates, en fin nous terminons avec des solutions proposées afin de mettre fin à ces type d'attaques ou au moins diminuer les risques. Dans la littérature il y a trois types d'attaques lié au cache : (*Nommage, routage, cache**).***

Les sous-sections suivantes, nous discutons chacune de ces types en détail.

II.2 Les attaques

Le CCN connait de nombreux problèmes de vulnérabilités à résoudre, et par conséquent il y a de nombreuses attaques qui se produisent dans ses environnements.

Ces attaques sont classifiées en quatre catégories : *nommage, routage, mise en cache et autres attaques connexes*.

II.2.1 Nommage

L'architecture CCN offre plus d'accès aux demandes des utilisateurs, ce qui est une bonne occasion pour les pirates d'en profitez en essayant de contrôler les informations d'une manière facile avec des techniques qu'ils utilisent aussi de blocage des informations.

Dans les attaques liées au nommage dans CCN, un attaquant tente d'empêcher la distribution d'un contenu spécifique en bloquant la transmission de ce contenu et / ou en détectant qui demande ce contenu **[13], [14]**. Du coup il existe deux type d'attaque de nommage :

A. Watchlist [1] :

L'attaquant surveille le réseau pour effectuer un filtrage en temps réel des contenus souhaités, ce filtrage est basé sur une liste prédéfinie par l'attaquant. Il peut supprimer la demande et / ou enregistrer les informations du demandeur. De plus, il peut essayer de supprimer lui-même le contenu correspondant. Tel que représenté dans la figure I.1, l'attaquant capture les demandes des utilisateurs pour filtrer et enregistrer qui a demandé quoi, ensuite filtre et enregistre également le retour contenu, qui contient des informations sur l'éditeur et les données.

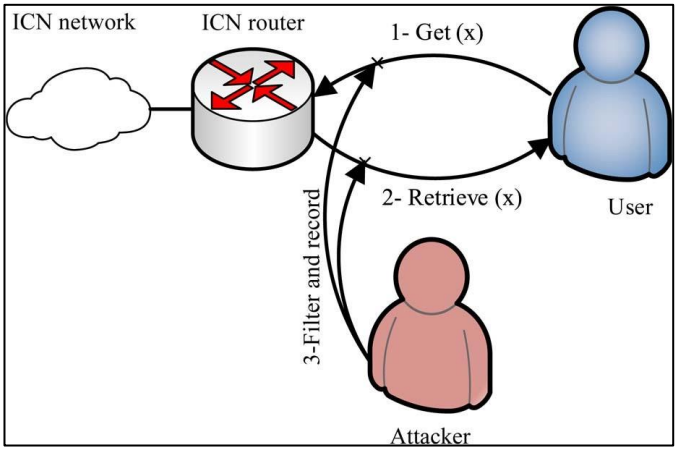


Figure II.1 : Attaque Watchlist **[1]**

Scénario de l'attaque :

1- Un utilisateur demande un contenu ICN nommé (x).

2- L'utilisateur doit récupérer le contenu (x).

3- L'attaquant peut filtrer et enregistrer les requêtes et / ou le contenu en fonction de sa liste prédéfinie.

B. Sniffing [1] :

Le reniflement est le processus de surveillance et de capture de tous les paquets passant par un réseau donné à l'aide d'outils de renfilage tel que les logiciels. Il est également appelé écoute électronique appliquée aux réseaux informatiques. Et contrairement à la liste prédéfinie dans l'attaque de la liste de surveillance, l'attaquant surveille le réseau pour vérifier les données s'il doit être marqué afin de le filtrer ou de l'éliminer.

La principale différence est que l'attaquant n'a pas de liste prédéfinie, mais il / elle fait quelques analyses sur les demandes ou sur le contenu donc il fait des tentatives jusqu'à arriver à obtenir des contenus qui peuvent être utiles pour lui ou les utiliser pour son intérêt.

II.2.2 Routage

La cohérence de synchronisation dans le réseau dépond à quel point le system il peut satisfaire les demandes des clients en temps réel et résister aux attaques liée routage, ce genre d'attaque peuvent être classées en deux types : attaques par déni de service distribué (DDoS) et par usurpation d'identité (Spoofing).

A. DDoS

Est une attaque de déni de service distribué qui est classée comme une attaque d'épuisement des ressources de synchronisation. Et peut être classé en infrastructure, source, blocage mobile et attaque par inondation, il peut survenir en raison de ces dernières attaques, il est également connu que cette attaque entraine à supprimer des requêtes à cause des délais expirés des flux de trafic indésirables et / ou un déni de service en échouant la cohérence de synchronisation dans le réseau.

- Infrastructure

Un attaquant envoie de nombreuse requêtes pour un contenu peu importe s'il existe ou pas dans le cache réseau. Pendant que les nœuds de CCN sont entrain de transmettre la copie la plus proche de contenu souhaité, en ce moment les requêtes envoyées vers la source entrainent des conditions de surcharge. Par conséquent un déni de service est déclenché en cas le nombre de demande envoyées est très élevé.

Dans le temps réel cette attaque ne peut pas arriver car les clients envoient des demandes dans un temps spécifique et normalement ça engendre pas une amplification toutefois les CCN est doté d'un mécanisme très utile pour diminuer cette amplification des requêtes parce que le routage se fait vers plusieurs endroits **[1]**.

Comme illustré sur la figure II.2, le scenario de l'attaque est comme suit :

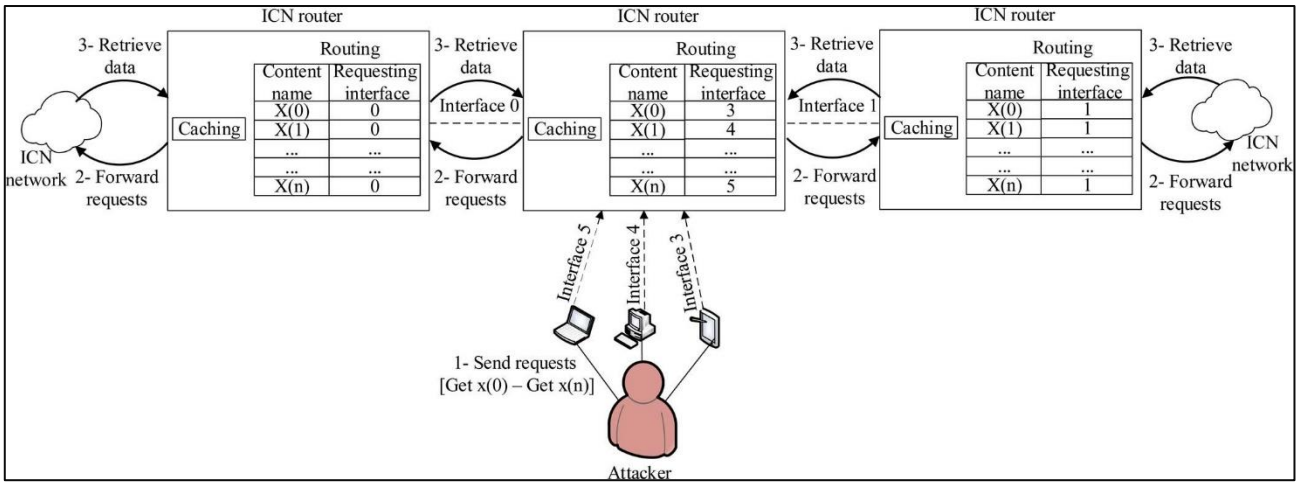


Figure II.2 : Attaque infrastructure **[1]**

1- Un attaquant, qui contrôle de nombreux systèmes d'extrémité, envoie un grand nombre de requêtes aux routeurs CCN.
2- Les routeurs attaqués transmettent leurs demandes aux routeurs voisins, et à leur tour, ils les envoient à leurs routeurs voisins et ainsi de suite.
3- ICN commence à récupérer ces grandes quantités de données depuis différents chemins et le renvoie aux emplacements demandés.

- Source

Dans les réseaux CCN, viser à attaquer une source, ça peut également entrainer aux conditions de surcharge de l'infrastructure de routage. Le scénario d'attaque est similaire au scénario d'attaque d'infrastructure : Un attaquant envoie nombreuse de demande d'un contenu à une source spécifique pour créer une défaillance de performances. Ensuite, cette attaque augmente le temps de réponse de la transmission de contenu pour ce contenu source ou son routeur d'accès. En plus de cet effet, l'attaque peut réduire le taux de retour des données et affecter les demandes de tous les nœuds dans les chemins vers les récepteurs **[1]**.

- Blocage mobile

Un attaquant mobile vise à surcharger toute une région avec plusieurs techniques qu'il dispose à travers la surcharge des routeurs d'accès mobile, tout en l'envoi d'un grand nombre de demandes de contenu pour faire dépasser le délai d'expiration de l'état qui entraine à un

blocage de la disponibilité régionale réseaux. Le scénario d'attaque présenté à la figure II.3 est similaire au scénario d'attaque d'infrastructure. La différence entre les deux attaques c'est que l'attaquant mobile envoie un très grand nombre de requêtes vers les réseaux voisins, alors que l'attaquant traverse le réseau de manière circulaire et continue **[1]**.

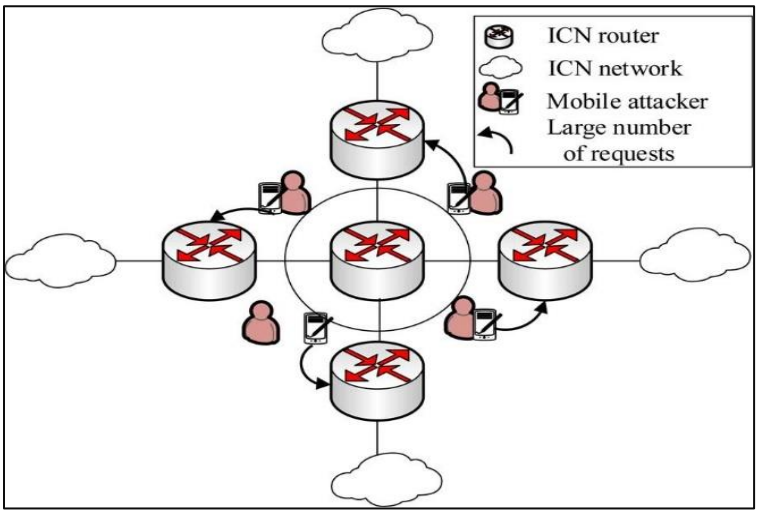


Figure II.3 : Attaque Blocage mobile **[1]**

- Inondation (Flooding)

Dans un réseau homogène, l'attaquant a les mêmes capacités que tous les autres utilisateurs du réseau. Cependant, si l'attaquant a plus de puissance que ses victimes, l'attaque de "Flooding" devient performante.

Pour effectuer une attaque d'inondation, un attaquant tente d'envoyer un grand nombre de requêtes dépassant le nombre de requêtes limité. A cause de CCN qu'est centrée sur le contenu, il n'y a pas d'identifiant d'hôte pour limiter le taux de demandes alors les nœuds du réseau acceptent un certain nombre de requêtes, dès qu'ils atteignent la limite, ils ignorent les demandes suivantes. Par conséquent, l'attaquant réussit à surcharger l'infrastructure globale du réseau et nuit à tous les utilisateurs. Le scenario de l'attaque tout simplement est basé sur l'envoie d'un nombre important de requêtes qui dépasse les limites du nœuds CCN et donc CCN va automatiquement négliger tous les reste des demandes envoyé par les client légitime.

- Timing

Cette d'attaque est similaire au scénario d'attaque d'infrastructure, dans ce cas l'attaquant envoie nombreuse de requêtes via une ou plusieurs routes de sorte que le routage et la transmission de données font des délais plus longs que normal pour augmenter le délai d'expiration des requêtes des utilisateurs légitimes, pour but de casser la cohérence entre la publication asynchrone de CCN.

B. Spoofing

En autre terme c'est « L'usurpation d'identité électronique » et plus précisément, c'est l'usurpation d'adresse IP dans l'architecture bout en bout, dans le réseau CCN peut être utilisé de trois manière ou bien via trois attaque tel que : Jamming, Hijacking et l'interception

-Jamming

Cette attaque peut être très dangereuse car elle peut être menée par une personne non authentifiée et inconnu dans le réseau. Un attaquant peut perturber le réseau en utilisant des capteurs de "Jamming" qui mettent des capteurs hors de service. Cette attaque est simple et efficace. Un nœud de CCN envoie un grand nombre de requêtes malveillantes de contenu inutile. L'attaquant qui se prend pour un abonné de réseau envoie les demandes malveillantes pour perturber le flux d'informations dans le réseau. CCN répond aux demande envoyé et le contenu est transmet à la destination sans un récepteur. Comme le montre la figure II.4, l'attaquant envoie des requêtes à un nœud partagé, ensuit il le transmet aux nœuds voisins.

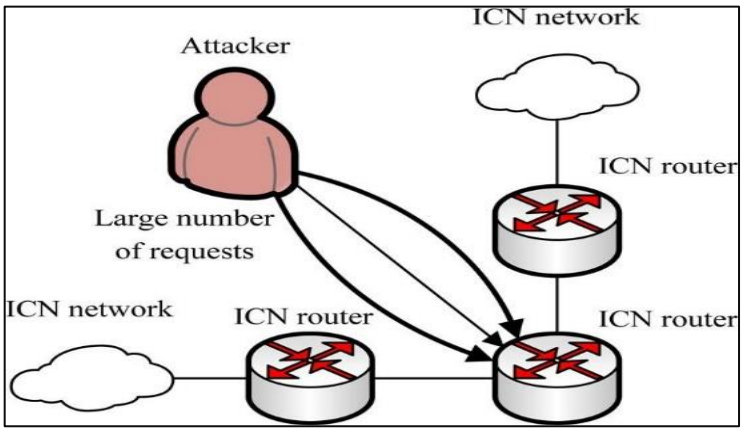


Figure II.4 : Attaque Jamming **[1]**

-Hijacking

Le détournement est un type d'attaque de sécurité réseau dans lequel l'attaquant prend le contrôle d'une communication. L'application de cette attaque dans le réseau CCN, Un attaquant se prend pour un éditeur de confiance, il peut à tout moment de transmission annoncer une erreur itinéraire pour tout contenu. A cette effet d'annonce, tous les demande de contenu qui sont à sa proximité seront dirigés vers ces invalides itinéraires. Par conséquent, ces demandes resteront sans réponse, ce qui conduire à un DoS. Vu que le mécanisme de routage dans CCN tente d'acheminer vers plusieurs emplacements dans le réseau, là l'attaquant peut réussir à mener ces attaques.

Comme montre la figure II.5, l'attaquant annonce un routes invalide pour certains contenus afin d'attirer les demandes des utilisateurs et quand les utilisateurs légitimes envoient des requêtes pour l'une de ces routes malveillantes, Les nœuds CCN transmettent ces demandes aux nœuds malveillants. Par conséquent, l'utilisateur légitime ne reçoit pas de réponse.

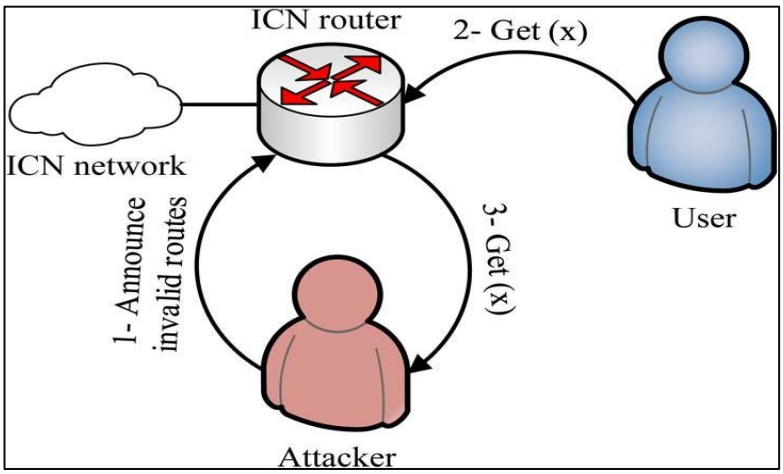


Figure II.5 : Attaque Hijacking **[1]**

Scénario de l'attaque :

1- Un attaquant annonce des routes invalides pour certains contenus comprenant (x).

2- Un utilisateur demande un contenu CCN nommé (x).

3- routeur CCN redirige les requêtes de l'utilisateur vers les routes malveillantes de l'attaquant et par conséquent, l'utilisateur n'obtient aucune réponse.

-Interception

Cette attaque similaire à l'attaque « homme de milieu ». Un attaquant qui se prend pour un éditeur de confiance annonce des itinéraires invalides, à travers ses tentatives d'attaque tous les demandes de contenu peuvent être capturées et envoyées à l'emplacement approprié. Cette fois les clients reçoive le contenu normalement sans aucun problème de perturbation, par conséquent l'attaquant peut savoir facilement le contenu demandé. Comme il illustré dans la Fig II.6, l'attaquant annonce des itinéraires invalides pour certains contenus à attirer les demandes de l'utilisateur, ensuite les utilisateurs légitimes envoient des demandes de contenu aux routes malveillantes, les nœuds ICN transmettent ces requêtes au nœud malveillant de l'attaquant qui a demandé ce contenu, puis le transmet pour obtenir les données réelles. Une fois les données réelles arrivent au nœud de l'attaquant, il retourne le contenu au nœud ICN demandé, qui à son tour le transmet à l'utilisateur légitime.

A la fin de la transmission, il semble au client que tout est normal et il y a aucun problème de disponibilité mais en vrai, l'attaquant a touché la confidentialité de l'utilisateur.

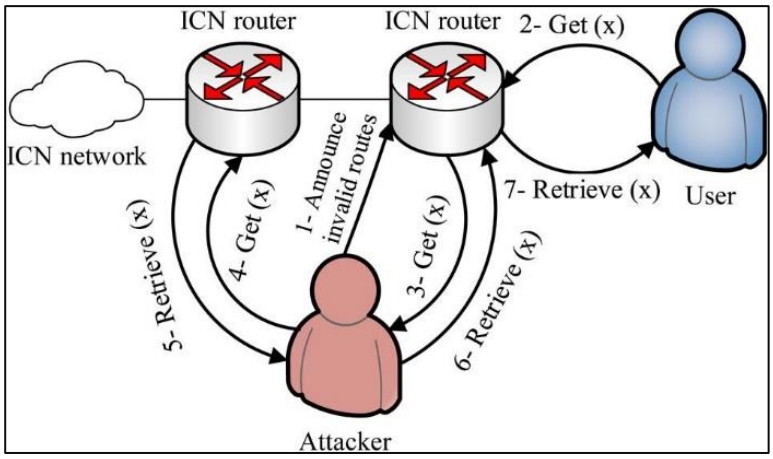


Figure II.6 : Attaque Interception **[1]**

Scénario de l'attaque :

1 - Un attaquant annonce des itinéraires invalides pour certains contenus comprenant (x).

2 - Un utilisateur demande un contenu CCN nommé (x).

3– routeur CCN redirige la demande de l'utilisateur vers les routes malveillantes de l'attaquant.

4 - L'attaquant transmet la demande pour obtenir le contenu réel.

5 - L'attaquant récupère le contenu (x).

6 - L'attaquant transmet le contenu à l'utilisateur demandeur.

7 - L'utilisateur récupère le contenu (x).

II.2.3 Mise en Cache

Nous nous intéressons pour notre étude aux attaques de mise en cache (Caching) car ce dernier représente la disponibilité de contenu dans le réseau CCN en plus le cache contenu est un pilier majeur qui vise à fournir la copie la plus proche disponible dans un nœud de CCN à un client, du coup le cache contenu est très exposé au différente attaque sachant que l'ICN est prouvé vulnérable à ce genre d'attaque. Toutefois les attaque qui vise CS inclus trois sous-catégories : *Time Analysis, Bogus Announcements, pollution*. En autre terme on peut dire que l'attaque de mise en cache peut être classée selon le temps analyse, annonces fictives ou attaques de pollution de cache **[5]**.

A. Temps d'analyse :

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.

Comme représenté sur la figure II.7, T1 est le temps nécessaire pour envoyer la demande et recevoir des données entre la source de contenu et le routeur le plus proche à l'utilisateur ou à l'adversaire, et T2 est le temps nécessaire pour envoyer la demande et recevoir des données entre l'utilisateur ou l'adversaire et le routeur le plus proche.

Scénario de l'attaque :

1 - Un utilisateur demande un contenu ICN nommé (x).

2 et 3 : les routeurs ICN essaient de trouver le contenu (x).

4 et 5 - Les routeurs ICN transfèrent le contenu (x) à l'utilisateur demandé.

6 - L'utilisateur récupère le contenu (x) en temps total T1 + T2, l'adversaire utilise ce décalage horaire pour savoir si un utilisateur proche a demandé ce contenu avant ou non.

7 - Un adversaire demande le contenu (x).

8- L'adversaire récupère le contenu (x) dans le temps T2 uniquement, car il existe déjà une copie en cache de le contenu.

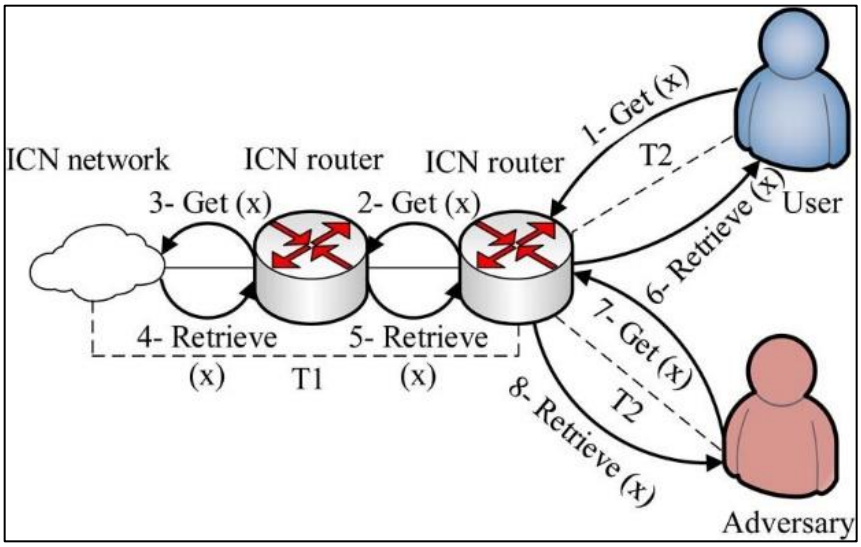


Figure II.7 : Attaque temps d'analyse **[1]**

B. Annonces fictives :

Le système de mise en cache est un pilier majeur de la partie architecture CCN, Donc le scénario d'attaque comme suit : Un utilisateur demande du contenu CCN nommé (x), tandis qu'un attaquant envoyer de nombreuses mises à jour d'annonces pour le contenu ou la copie en cache à une fréquence qui dépasse le temps de convergence du routage de demande de contenu local y compris (x) pour but de violer les systèmes de mise en cache et de routage

Et par conséquent les routeurs CCN ne pourront pas mettre à jour sa table de routage à cause de ces fausses annonces, qui entraînent une récupération de contenu incomplète ou fausse, comme illustré sur la Figure II.8.

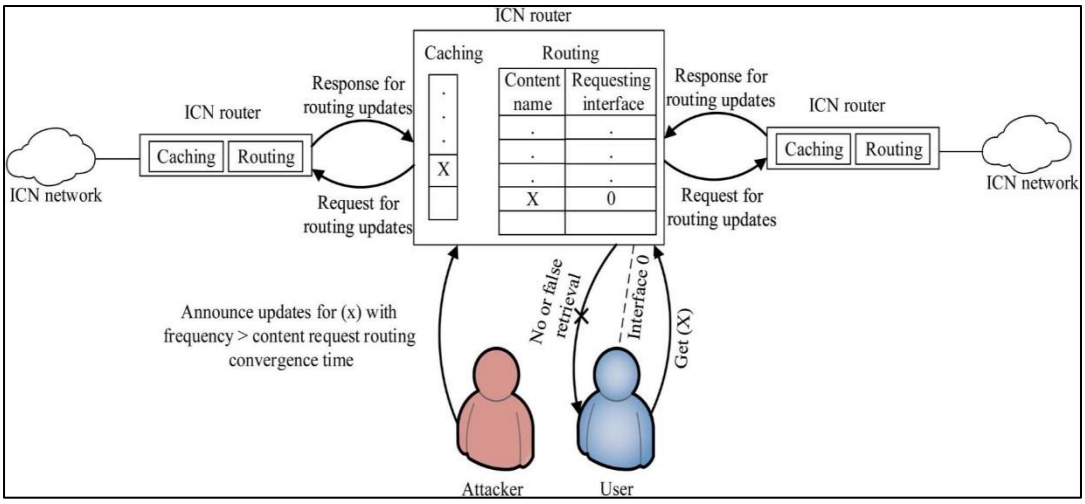


Figure II.8 : Attaque d'annonces fictives **[1]**

C. Pollution :

Il y a deux types d'attaque de pollution :

1- Demande aléatoire :

Un attaquant vise à endommager le système de mise en cache de CCN et à modifier la popularité du contenu.

L'attaquant force les caches CCN remplir les caches avec des contenus invalides ou bien des contenus impopulaires en envoyant des requêtes aléatoires.

Un contenu impopulaire fait référence à un contenu qui n'est pas fréquemment demandé par les utilisateurs du réseau, donc c'est l'occasion pour l'attaquant où il peut demander de faux contenus pour remplir les caches.

Un contenu est faux s'il est modifié ou n'atteint pas de la source prévue, ou s'il ne s'agit pas du contenu demandé par l'utilisateur. Comme le montre la figure II.9, dans le cas normal, si un deuxième utilisateur demande une copie mise en cache, il l'obtient du retour le plus proche et en même temps les routeur CCN mettent en cache chaque contenu qui les traverse. Comme le montre la figure II.10, dans le cas d'un attaquant, ce dernier il envoie un nombre massif de demandes aléatoires pour gâcher le système de mise en cache. Dans ce dernier cas si l'attaquant réussit son attaque et en autre coté si le deuxième utilisateur demande le même contenu alors en raison de la dernière attaque sa demande prend le chemin complet en tant que premier utilisateur.

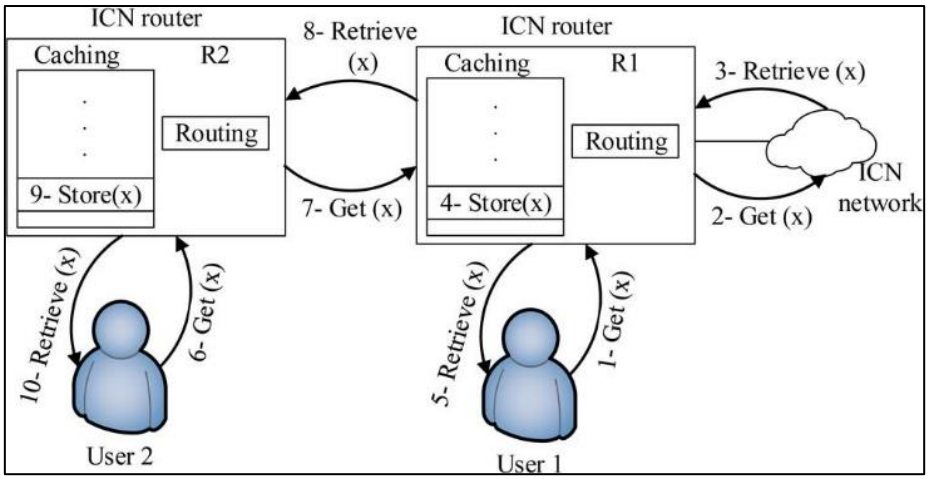


Figure II.9 : Attaque demandes aléatoires (cas normal) **[1]**

Scénario de l'attaque :

1- Demandes de contenu CCN de l'utilisateur 1 nommées (x).

2- Le routeur R1 essaie de trouver le contenu (x).

3- R1 récupère le contenu du réseau CCN.

4- R1 stock en cache le contenu (x).

5- L'utilisateur 1 récupère le contenu (x).

6- L'utilisateur 2 demande le même contenu (x) via le routeur R2.

7- R2 essaie de trouver la copie la plus proche, qui existe dans le routeur R1.

8- R1 envoie le contenu au routeur R2.

9- R2 met en cache le contenu (x).

10- L'utilisateur2 récupère le contenu (x).

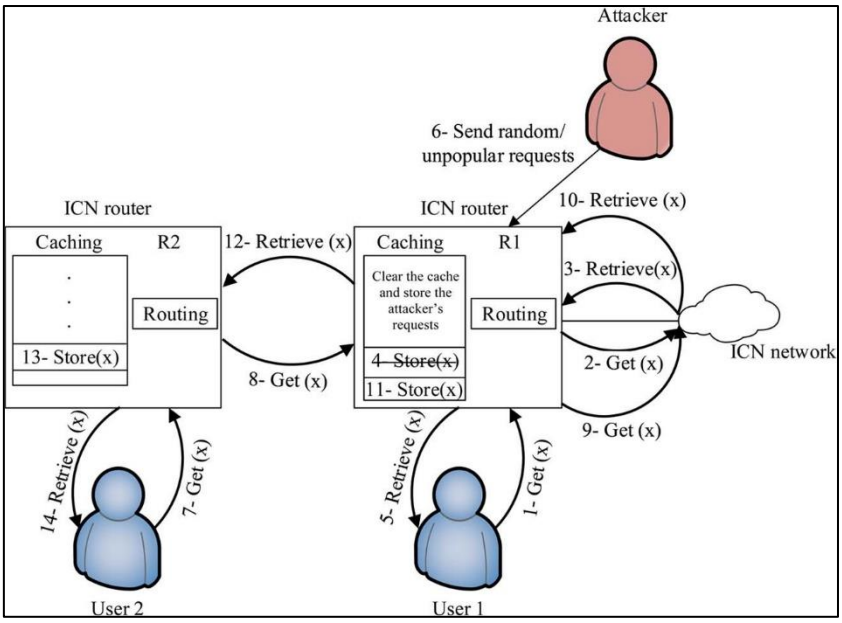


Figure II.10 : Attaque demandes aléatoires (cas d'un attaquant) **[1]**

Scénario de l'attaque :

1- Demandes de contenu CCN par l'utilisateur 1 nommées (x).

2- Le routeur R1 essaie de trouver le contenu (x).

3- R1 récupère le contenu du réseau CCN.

4- R1 met en cache le contenu (x).

5- L'utilisateur 1 récupère le contenu (x).

6- Un attaquant envoie un grand nombre de requêtes aléatoires pour violer le cache.

7- L'utilisateur 2 demande le même contenu (x) via le routeur R2.

8- R2 essaie de trouver la copie la plus proche et envoie une demande à R1.

9- Le routeur R1 essaie de trouver le contenu (x).

10- R1 récupère le contenu du réseau CCN.

11- R1 met en cache le contenu (x).

12- R1 envoie le contenu à R2.

13- R2 met en cache le contenu (x).

14- L'utilisateur2 récupère le contenu (x).

2- Demande impopulaire :

Pour faire ce type d'attaque il faut savoir qu'ils sont les contenu popularité pour que l'attaquant puisse deviner et puis demande les contenus impopulaires afin de gâcher la mise en caches de réseau CCN et peut-être modifier la popularité des contenus et ce genre d'attaque ont un impact sur les éléments suivants :

- ***Confidentialité :*** La mise en cache dans CCN est un mécanisme, connait des vulnérabilités pouvant toucher la confidentialité des utilisateur Comme dans l'analyse du temps attaque, l'adversaire peut savoir si un utilisateur proche a déjà demandé ce contenu ou non et cela viole la confidentialité de l'utilisateur.
- ***Déni de service :*** Les fausses annonces provoquent de nombreuses mises à jour de contenus qui conduisent à des informations incomplètes ou erronées. Par conséquent, les utilisateurs ne récupèrent pas le contenu demandé.
- ***Cache la pollution :*** Tout utilisateur dans CCN peut envoyer plusieurs et / ou des demandes impopulaires qui provoquent une pollution du cache.

II.3 Les contres mesures existantes [1]

II.3.1 Nommage

Il existe quelque solution fiable pour les attaques liées au nommage, telles que comme mix-nets **[15]**, Tor **[16]**, Freedom, Anonymizer, Freenet **[17]** et le chiffrement refusable **[18]**.

Ils sont des solutions théoriques mais le problème c'est que ces solutions ne peuvent pas être appliqué directement dans ICN, car ils nécessitent des conditions qui ne conviennent pas au ICN actuel comme une infrastructure de taille utilisateur ainsi que les informations partagées entre l'éditeur et l'utilisateur, et infrastructure de stockage.

La solution de sécurité ICN doit obtenir la confidentialité, doit également être simple en termes de calcul pour que les utilisateurs peuvent récupérer les contenus, coûteux en calcul.

Arianfar et coll. **[14]** présentent une solution générique pour les attaques liées au nommage qui ne nécessitent pas de clés partagées entre les éditeurs et les consommateurs.

Cette solution fait plusieurs hypothèses qui peuvent ne pas être applicables dans ICN et n'offre pas une confidentialité idéale, elle ne convient que lorsqu'il y a un grand nombre d'utilisateurs. Ion et coll. **[19]** ont conçu un chiffrement basé sur les attributs et schéma de confidentialité de routage pour ICN afin de soutenir la confidentialité des données. L'idée de

base est d'appliquer les politiques de contrôle d'accès distribuées au contenu et spécifier ces politiques en termes de contenu. Ce schéma prend en charge environnements à grande échelle sans avoir besoin de partager des clés. Il est aussi testé uniquement sur l'architecture NDN, il doit donc être testé dans les autres architectures d'ICN tel que CCN le sujet de notre étude.

II.3.2 Routage :

Ils existent des solutions pour les attaques liées au routage, donc les solutions généralement vont se diriger vers la limitation de débit par utilisateur final.

Sachant que l'ICN n'offre pas l'identifiant d'hôte dans son architecture ce qui permet à l'attaquant de créer nombreuse de demande. Du coup il existe plusieurs solutions proposées par expert dans le domaine de réseau spécifiquement dans les réseaux ICN tel que :

Gasti et coll. **[20]** qui proposent une classification de haut niveau des attaques DDoS et de leurs solutions dans l'architecture NDN.

Fotiou et coll. **[21]** suggèrent un algorithme de classement des contenus ICN pour lutter contre le spam, basé sur les classements des éditeurs et des abonnés.

Compagno et coll. **[22]** présenter le concept d'inondation des demandes pour les contenus indisponibles. De nombreux articles classifient les attaques DDoS et leur détection / mécanismes de prévention **[23] - [24**]. Les contre-mesures largement discutées pour DDoS dans l'architecture Internet sont la trace IP **[25]**, filtrage de paquets **[26]** et limitation de débit **[27]**. Celles-ci Les techniques ne peuvent pas être utilisées dans ICN car elles dépendent des adresses IP des points d'extrémité, en fin ça reste un sujet ouvert et à développer pour les chercheurs.

II.3.3 Mise en Cache :

Les pirates tentent toujours de trouver des méthodes pour polluer le cache plus facilement, plus de paquets d'intérêt pour un contenu spécifique signifie plus de demande, ce qui forcera un autre contenu, donc dans ICN, il est donc difficile de filtrer les demandes qui créent une fausse demande. De plus, les nœuds malveillants peuvent générer du faux contenu et d'autre méthode afin de polluer le cache ou voler les informations.

Les solutions existantes contre les attaques du cache sont conçues pour un seul serveur de cache et ne conviennent pas pour les CCN ni pour les ICN en général, car la mise en cache dans ICN touche tous les contenus de tous les nœuds. La solution de sécurité ICN devrait

réduire les effets de ces attaques sur la mise en cache et ne stocker que le contenu le plus fréquemment demandé.

La solution Cacheshield gère de façon aléatoire et impopulaire demandes de mise en cache ICN. La pertinence de Cacheshield dans d'autres architectures de l'ICN et l'évolutivité du système doivent être évaluée. Mohaisen et Al. **[10]** propose un mécanisme de protection de la vie privée pour l'attaque d'analyse temporelle. Le mécanisme ne prend pas en compte les différentes politiques de mise en cache et suppose que l'adversaire est proche de l'utilisateur attaqué. Ghali et Al. **[11]** adresse l'empoisonnement du contenu pour la mise en cache dans l'architecture CCN. Ils présentent un algorithme de classement basé sur les commentaires des consommateurs, ce qui permet aux routeurs de distinguer entre des contenus valides et malveillants. Il existe également de nombreux solutions pour les attaques d'empoisonnement du cache comme dans les extensions de sécurité du système de noms de domaine (DNSSEC) et la solution de sécurité pour stopper les attaques d'empoisonnement de cache dans la hiérarchie DNS (SDNS). Ces schémas dépendent des adresses IP des points d'extrémité et ne sont donc pas adaptés aux ICNs.

La majorité des solutions qu'ont été proposées, elles sont pour but de mettre fin à l'empoisonnement du cache en conséquent ça renforce la disponibilité de contenu mais d'un autre côté de sécurité, l'intégrité n'a pas été traité également comme la disponibilité comme un principe important pour garantir la sécurité de cache contenu. Du coup nous avons vu que la table PIT est un pilier de ICN et qui peut être subit une modification ou suppression d'information par une attaque de mise en cache, ce que n'était jamais traité auparavant par les chercheurs dans leurs études pour sécuriser le cache contenu.

II.4 Conclusion

Dance ce chapitre nous avons présenté plusieurs type d'attaque liées aux trois aspects fameux dans les réseaux CCN : Nommage, Routage et Cache contenu, ensuite nous avons discuté les solutions de sécurité offert par l'architecture ICN en général et CCN en particulier.

Au final nous avons choisi d'étudier les attaques qui sont liées au cache contenu en proposant une nouvelle vulnérabilité situé dans la table PIT qui peut être exploité par les utilisateurs malveillants en changeant les informations introduits lors de la transmission des paquets d'intérêt ou bien les supprimer, mais cette fois si le pirate exploite cette vulnérabilité, ils ne va pas toucher uniquement la disponibilité de contenu comme les autres attaques de cache étudié mais l'intégrité aussi.

Chapitre III : ATTAQUE PIT

III.1 Introduction

Chaque système informatique est exposé à des vulnérabilités du système, du coup ils sont exploitables par les gens malveillants tel que les pirates pour réaliser des attaques à différentes motivations en implémentant différents type d'attaque connu ou non-connu.

Dans le cadre de notre travail, nous essayons de produire une nouvelle attaque à travers la modification de la table PIT sachant que ce dernier est une fonctionnalité très importante dans l'architecture d'ICN notamment les routeurs, cette attaque a pour but de détourner les informations circulant dans le réseau ICN afin de voler les données transmit et aussi toucher à la vie privée des utilisateurs légitimes du réseau.

III.2 Attaque PIT

Dans les nouvelles architectures d'ICN, la table PIT sert à mémoriser temporairement les messages d'intérêt et permet aussi aux paquets d'intérêt de suivre le chemin inverse en sauvegardant l'intérêt envoyé par l'émetteur ansi son interface. Pour l'attaquant, la table PIT est un trésor qu'il peut utiliser pour profiter de ces informations afin de réaliser son attaque.

Nous avons imaginé deux scenarios différents d'attaque PIT, qui sont décrits comme suit :

1er scenario :

1- Identification de routeur le plus proche de l'attaquant
2- L'utilisateur envoie demande un contenu nommé (x)
3- Le routeur voisin cherche le contenu (x) dans cache s'il existe
4- S'il n'existe pas, l'attaquant est toujours en écouteur et en ce moment le routeur crée le nouvel intérêt pour l'ajouter à la table PIT, l'attaquant tente de modifier l'interface d'utilisateur émetteur par son propre interface avant que l'intérêt soit ajouté à liste d'attente
5- L'attaquant reçois le contenu (x) qui été prévu envoyer au client concerné par la demande

2eme scenario :

1- Si le routeur trouve le contenu dans son cache, produit un nouveau paquet pour renvoyer le contenu au client concerné ou bien le récepteur
2- Avant la création de nouveau paquet, l'attaquant tente de changer l'interface de réel récepteur par son interface

Afin de simuler une attaque sur notre réseau CCN selon les deux scénarios proposés, nous avons modifié l'algorithme de de déroulement de PIT, FIB et CS utilisé dans le package CCN-Lite installé dans l'émulateur Omnet++ illustré ci-après :

```
Function MAJ_PIT_FIB_CS
i = Interest.Name();
if Data = CS.Find(i) then
        return Data
endif
if  PIT.find(i) then
        PIT.AddFace(i,q)
else
        PIT.CreateFace(i,q)
        if  FIB.find(i) then
        Find the node j with the highest utility for content type i in FIB and GNU table
                if  j ϵ E then
                        Forward interest message to node j
                else
                        Store interest message waiting for the next contact
                endif
        else  Drop Interest message
```

L'attaque d'usurpation d'identité n'a pas été effectuée sur le réseau, nous avons ciblé directement les routeurs concernés et modifié le contenus de leur table PIT s'il le faut ou encore récupérer directement le contenu depuis le CS, ces deux cas de figures sont illustrés dans les scénarios précédents et ont été mis au point dans deux fonctions exécutables sur la plateforme de simulation, ci-après les deux fonctions avec leurs explications:

1er scenario:

```
Function Ccnl_interest_append_pending (i , from )
struct ccnl_pendit_s  pi

if id_router == 2 then
        from = my_face
endif
x=0
for pi = i.pending ; pi ; pi.next
        if pi.face == from then
                "we found a matching interest"
                update_time(now)
                x=1
        endif
end
if x=0 then
        pi.face = from  ;  pi.last_used = ccnl_now() ; i.pending = pi
end.
```

La fonction **ccnl_interest_append_pending** est chargée de mettre à jour la table PIT en ajoutant le nouvel intérêt crée par la fonction **ccnl_interest_new**

- La première condition dans la fonction vérifie si on est dans le routeur cible (celui dont on veut modifier la table) ou non, en comparant son ID à une valeur fixe, si c'est le routeur visé, l'attaquant remplace l'interface de client légitime dans le paramètre de la fonction par son propre ID.
- Quelque-soit le routeur (cible ou non), une boucle sera placée pour vérifier si l'intérêt existe déjà ou non dans la table PIT, s'il existe la date d'envoi de l'intérêt sera mise à jour, sinon l'intérêt sera inséré dans la table PIT.

2*eme* ***scenario:***

```
Function ccnl_fwd_handleInteres (i, from)
 if CS is not null then
        for (c = i; c; c = c. next) then
                if i is found in CS. LOOKUP(i) then
                        data ← c;
                endif
        end
 endif
 if data matching content =1 then
      return SEND (data, my_face)
endif
```

La fonction **ccnl_fwd_handleInteres** a comme paramètres deux attributs :

- Le premier « i » est le contenu demandé qui peut être ciblé par l'attaquant
- Le deuxième « from » est l'interface du client récepteur.
- La première condition a pour but de tester si le cache contenu est vide ou pas
- En suite après vérification, le programme exécute la boucle pour chercher dans le CS, la variable « c » joue le rôle de pointeur navigant dans le cache routeur après il fera un autre teste afin de voir la correspondance entre le contenu demandé et les contenus existant dans le cache contenu, et il est sauvegardé dans la variable « data. »
- Apres l'exécution de la boucle, le programme exécutera une dernière teste pour voir si le contenu souhaiter est trouvé, à la fin la fonction retournera le contenu data a l'interface qui correspond et c'est là que l'attaquant « **my_face** » peut exploiter cette faille en mettant son propre interface afin de recevoir le paquet qui était destiner à l'interface « from »

Pour mieux comprendre les deux algorithmes et leur déroulement nous avons réalisé une conception sur l'organigramme comme présente les figures III.1 et III.2 :

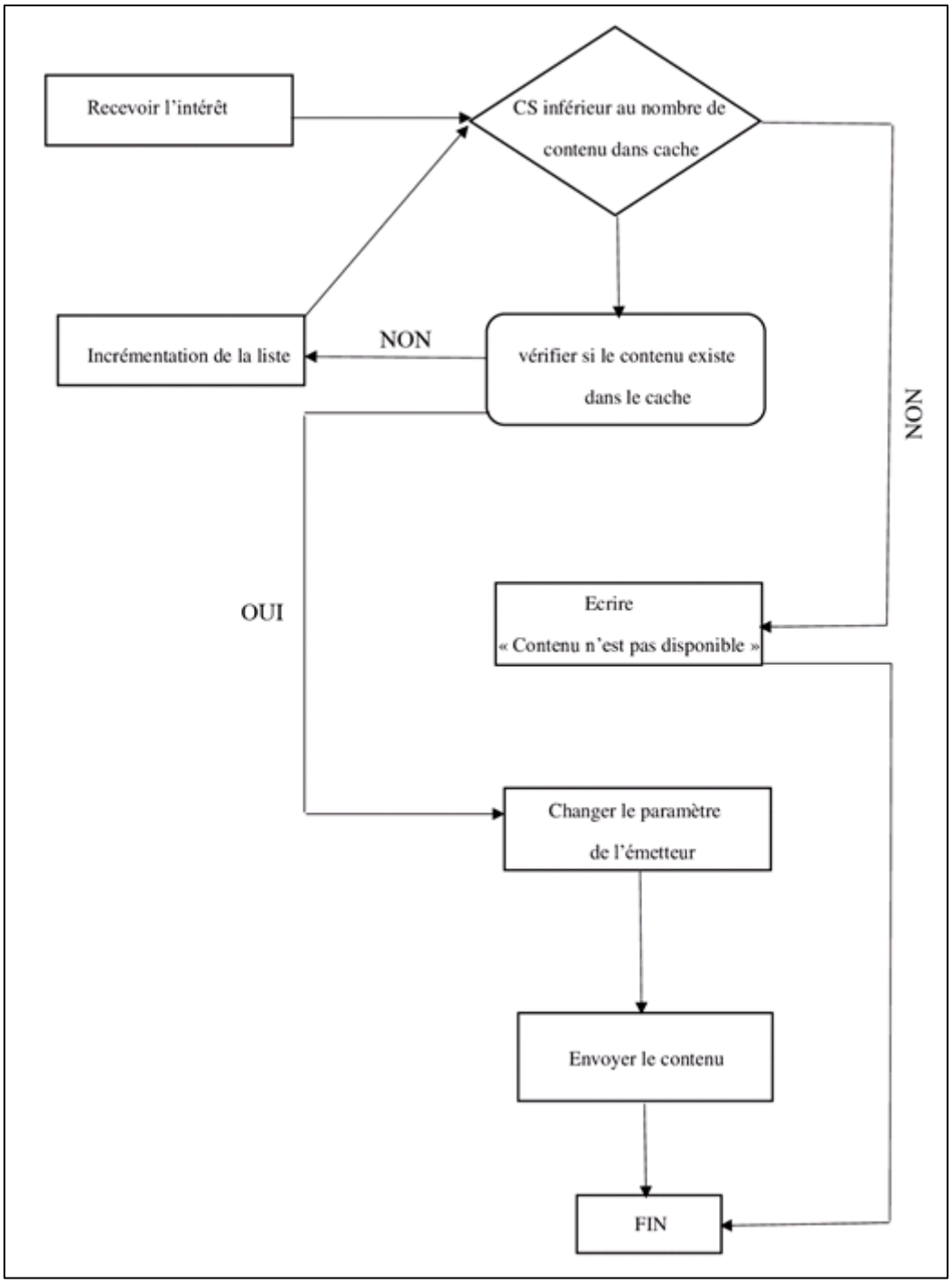


Figure III.1 : Organigramme de ccnl_interest_append_pending

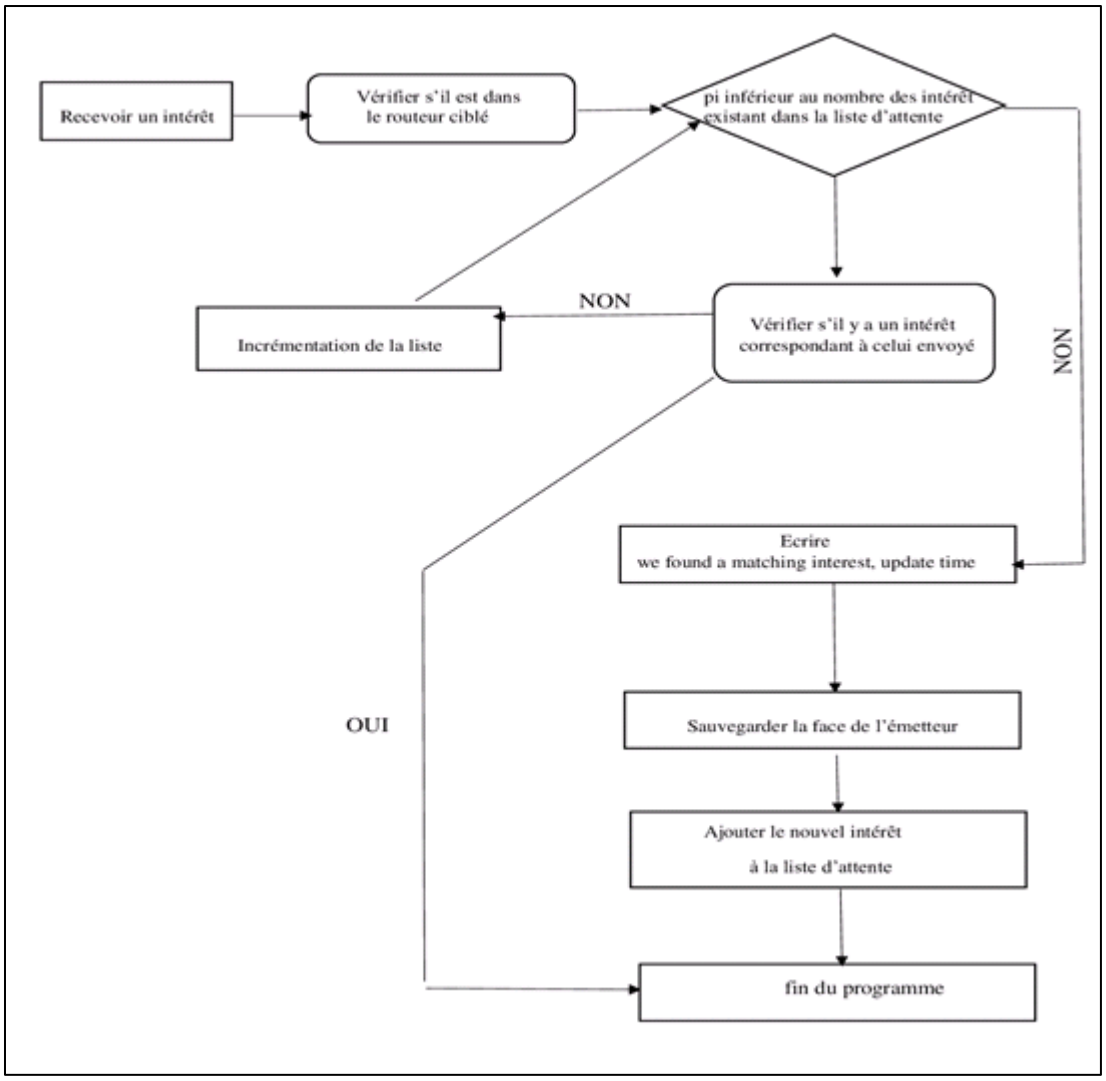


Figure III.2 : Organigramme de ccnl_fwd_handleInteres

III.3 Déploiement de l'environnement de travail

Pour bien illustré notre problème nous nous dirigeons vers la simulation qui est en général basée sur la reproduction d'un fait informatique et de nœuds qui fonctionne dans un environnement dédié aux réseaux et cela nous fait gagner en temps et en coup, car il est très difficile de reproduire l'adressage des systèmes complexes voire impossible dans quelques cas comme la simulation réel des CCNs ,Dans notre travail nous présentons quelques outils que nous avons choisi pour simuler ce problème il s'agit du simulateur OMNET++, et nous avons utilisé le package CCN lite afin de reproduire des attaques léger sur les réseaux CCNs , en dernière nous utiliserons ces outils la a fin de simuler des attaques et voir les différentes failles.

III.3.1 Omnet++

OMNet++ est un simulateur d'événements basé sur le langage C++, destiné principalement à simuler les protocoles réseau et les systèmes distribués. Il est totalement programmable, paramétrable et modulaire. C'est une application open source et sous licence GNU, développée par Andras Varga, chercheur à l'université de Budapest.

OMNet++ est destiné avant tout à un usage académique et est l'intermédiaire entre des logiciels de simulation comme NS, destiné principalement à la recherche et OPNET qui est une alternative commerciale de OMNet++

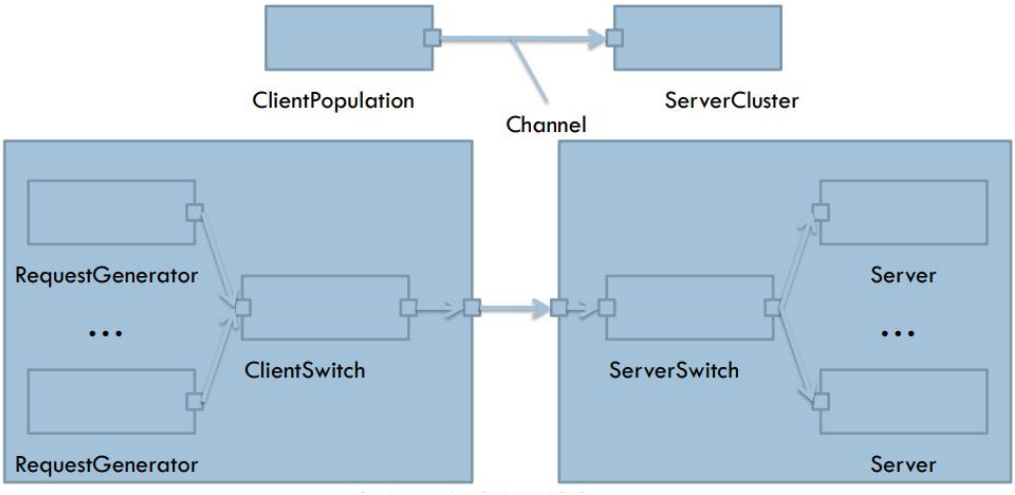


Figure III.3 : schéma expliquant le fonctionnement de OMENT++

1- Protocoles sont disponibles sur ces outils de simulation

Omnet++ gère nativement le TCP/IP, le SCSI et le FDDI. Le logiciel gérant le plus large panel de protocole est OPNET (notamment TCP/IP, ATM, Ethernet, etc.…). CLASS par exemple ne gère que les réseaux ATM.

2- Construction d'un modèle de simulation

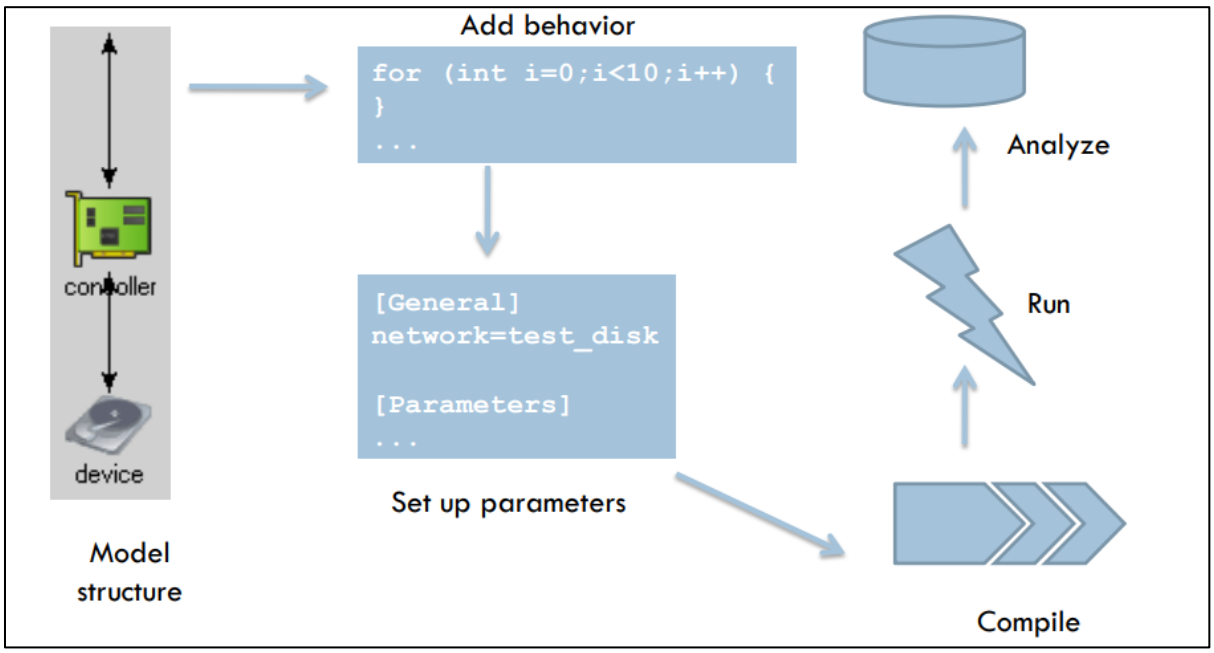


Figure III.4 : les phases de l'exécution d'un programme OMENT++

3- Séparation entre topologie et fonctionnement

Les outils de simulation réseaux considèrent tous qu'un réseau est un ensemble de nœuds connectés entre eux. Quelques outils, comme Parsec et C++Sim, ne fournissent pas de moyens explicites de description des topologies. Dans Parsec, il faut programmer une "entité pilote" qui initialisera le réseau en créant les nœuds nécessaires et en les interconnectant. Cette solution n'encourage pas la séparation entre topologie et fonctionnement et les possibilités de réutiliser des modèles existant sont peu nombreuses. D'autres outils comme NS et CLASS ne gèrent pas la hiérarchie dans les réseaux, ce qui entraine moins de flexibilité dans l'architecture du modèle. OPNET, comme OMNet++, permet la gestion de hiérarchie dans les modèles, mais il y a tout de même quelques restrictions. La principale différence entre OMNet++ et OPNET est que les modèles OPNET utilise uniquement des topologies fixes alors qu'OMNet++, avec l'éditeur graphique NED, permet de créer des topologies entièrement paramétrables. L'éditeur de OPNET propose une librairie de modèles sous forme de fichiers binaires au format propriétaire.

4- Les performances de ce genre d'application

Le facteur déterminant le temps d'exécution de la simulation est le langage de programmation utilisé. Les simulations sous OMNet++ peuvent être programmées en C ou en C++. C'est également le cas pour NS, Parsec, OPNET, C++SIM, NetSim++, SMURPH, Ptolemy. Ce sont ces langages qui fournissent les meilleurs temps d'exécution.

5- Les fichiers sources des applications

OMNet++ est totalement open-source mais certains logiciels commerciaux comme Parsec ne mettent pas leurs sources à disposition. L'accès au code source ne permet pas uniquement de modifier le moteur de simulation mais aide également au débogage des modèles de simulation.

III.3.2 CCN-lite

CCN-lite est une implémentation ICN légère appartenant à la licence ISC permissive qui est développé à l'Université de Bâle en suisse Transitaire au format multi-paquets : NDN, CCNx, etc. CCN-lite fonctionne sur plusieurs plateformes comme x86 / 64 sur Linux, BSD et MacOs, Module de noyau pour Linux, Android, Arduino ARM Cortex série A RIOT (par exemple ARM Cortex série M)

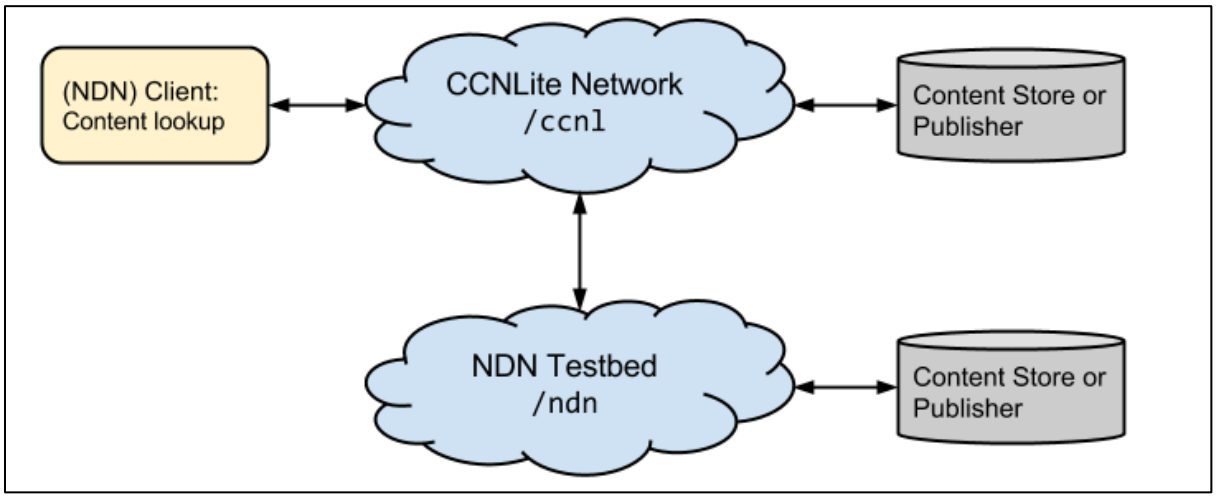


Figure III.5 : réaction du ccn-lite avec la requête

Caractérisation

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

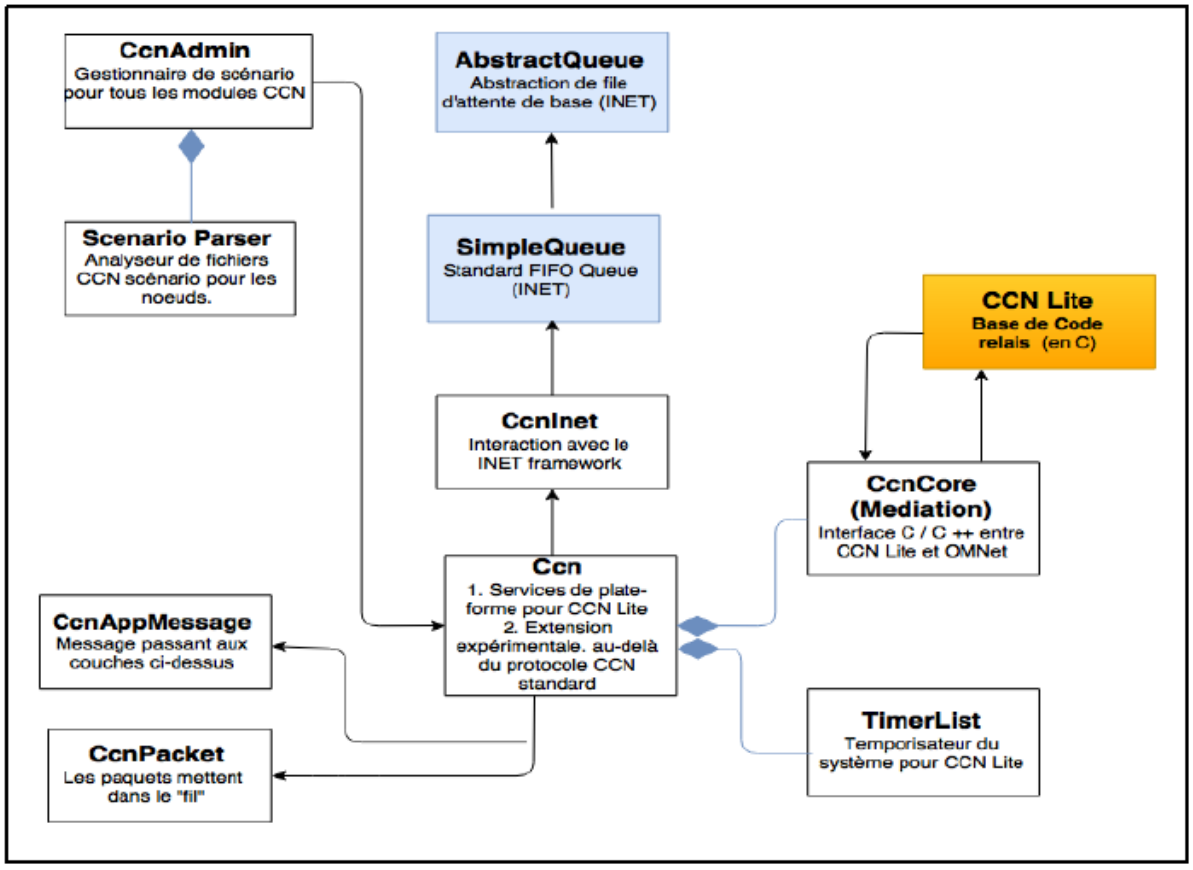


Figure III.6 : les différentes classes de bases dans OMNET++

III.4 Les matériel et les logiciels utilisés :

Une telle architecture réseau déploie en général un matériel assez important, toutefois dans le cadre de notre master et en manque d'infrastructure et matériel nécessaires, nous nous sommes contenté d'une simple topologie avec les moyens de bord, le tableau suivant résume les capacités des ordinateurs utilisés.

CPU	RAM	DIQUE DUR	OS	OMNET++	INET FRAMEWORK	CCN-lite
Intel Core i5-6300U 2,4 GHz - Turbo : 3,00 Ghz - DMI : 4 GT/s - Cache : 3 Mo - Socket FCBGA1356.	12GB DDR4-SDRAM	256 GB HDD 7200 tr/min	Ubuntu 16.04.4 desktop amd64	OMNET++ 4.5	INET 2.6	CCN-lite 0.3.0

III.5 Topologie utilisée :

Nos deux attaques visent à cibler le cache contenu d'un routeur pour cela ne nous allons pas besoin de faire une topologie très complexe alors nous avons besoin de créer 5 seul clients récepteur et émetteur d'intérêt, 6 routeurs, 2 attaquants, un serveur et un admin qui a pour rôle de donner à chaque composant de la topologie son rôle spécifique et sa configuration et en plus de cela il remplit le serveur de contenu et au final il déconnecte les appareils pour terminer la simulation.

Nous avons conçu cette topologie pour essayer de reproduire au maximum les faits réels

Le fichier NED contient le code source qui a pour but de créer cette topologie on y retrouve les paramètres <Delay> qui représente le retard des messages envoyés sur le canal et généralement il est égal à 0.5us, le deuxième <Datarate> il représente le nombre de flux de données passant par le canal dans une seconde (le débit/s) et il est égale à 100mbs. Pour finir connecte les nœuds entre par un fast Ethernet dans la partie connexion dans le fichier

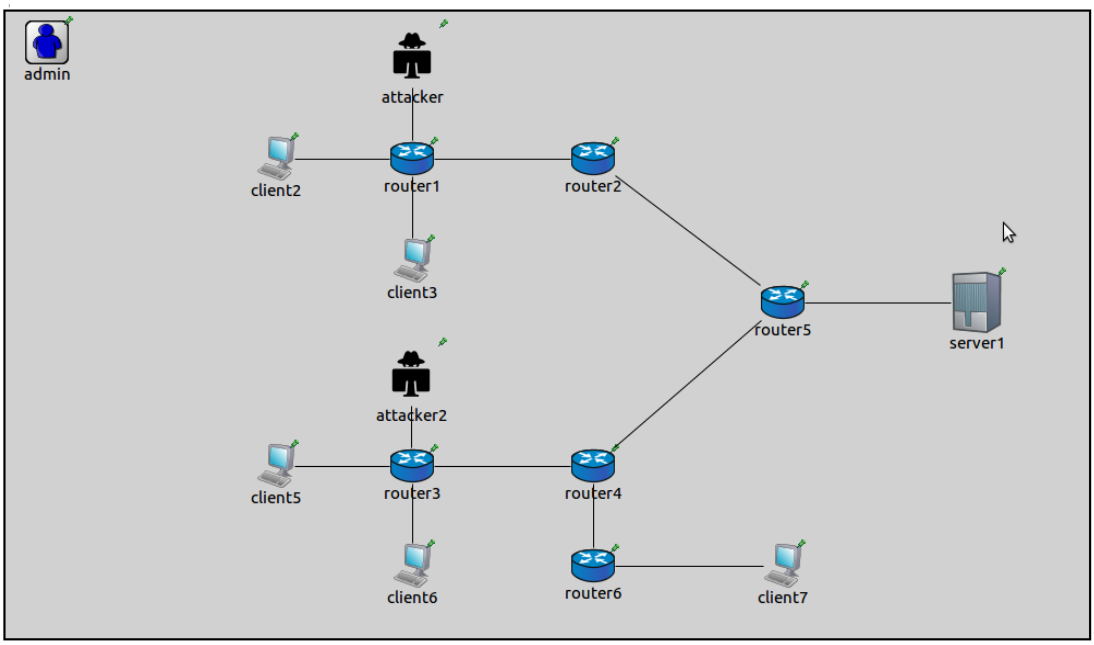


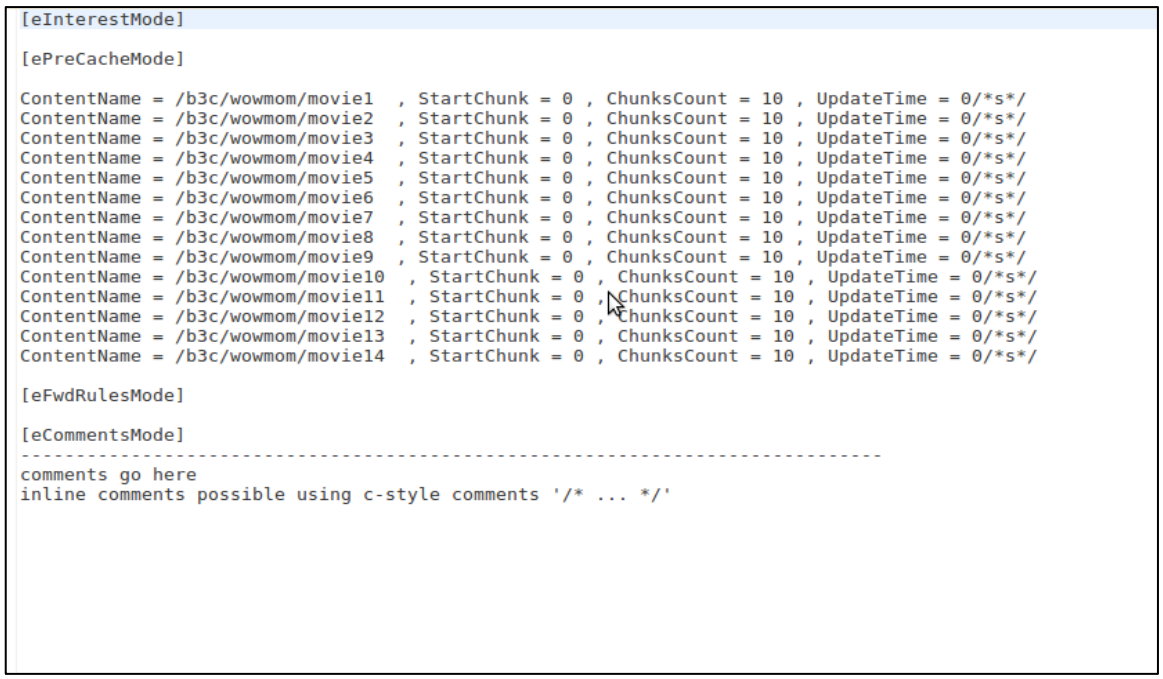
Figure III.7 Topologie simulé dans le réseau CCN

III.5.1 Routage et chargement des donnés :

Comme les réseaux IP les réseaux ICN ont besoin de routage et de chargement de données et pour cela on a besoin d'un fichier de configuration, ce dernier est sous extension .cfg plus un fichier qui gère, l'intérêt en première partie du code chaque nœud émetteur et son temps de réponse se nomme <eInterestMode>

La deuxième partie concerne le routage de ce réseau et il est représenté par la table FIB, s'appelle <eFwdRulesMode>

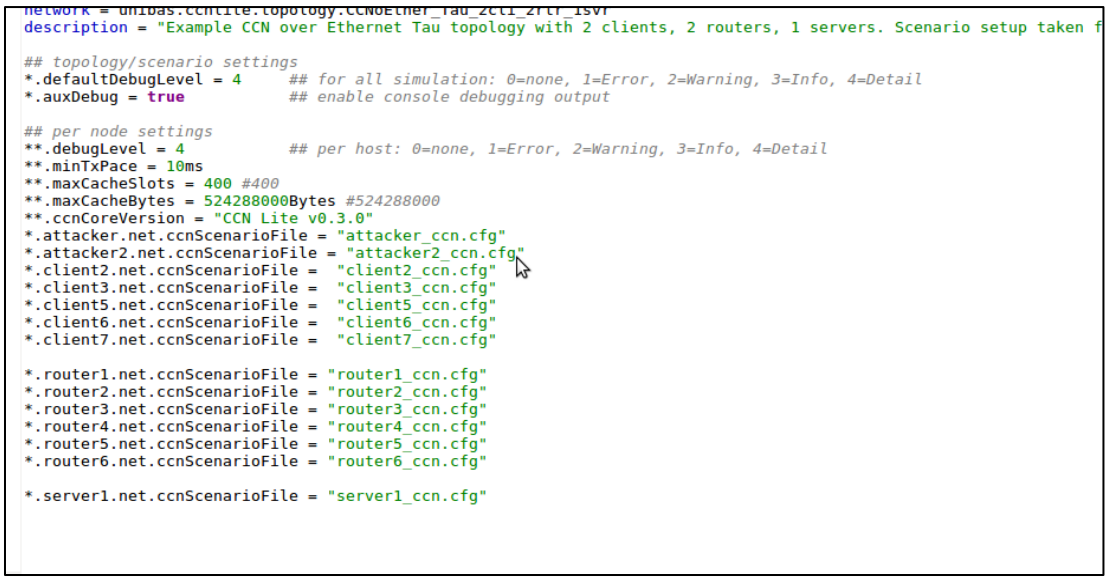
La dernière partie du code géré l'espace de stockage de l'information et le nœud qui utilisent ce fichier est le serveur, le nom du fichier est <ePrecacheMode>



```
[eInterestMode]

[ePreCacheMode]

ContentName = /b3c/wowmom/movie1  , StartChunk = 0 , ChunksCount = 10 , UpdateTime = 0/*s*/
ContentName = /b3c/wowmom/movie2  , StartChunk = 0 , ChunksCount = 10 , UpdateTime = 0/*s*/
ContentName = /b3c/wowmom/movie3  , StartChunk = 0 , ChunksCount = 10 , UpdateTime = 0/*s*/
ContentName = /b3c/wowmom/movie4  , StartChunk = 0 , ChunksCount = 10 , UpdateTime = 0/*s*/
ContentName = /b3c/wowmom/movie5  , StartChunk = 0 , ChunksCount = 10 , UpdateTime = 0/*s*/
ContentName = /b3c/wowmom/movie6  , StartChunk = 0 , ChunksCount = 10 , UpdateTime = 0/*s*/
ContentName = /b3c/wowmom/movie7  , StartChunk = 0 , ChunksCount = 10 , UpdateTime = 0/*s*/
ContentName = /b3c/wowmom/movie8  , StartChunk = 0 , ChunksCount = 10 , UpdateTime = 0/*s*/
ContentName = /b3c/wowmom/movie9  , StartChunk = 0 , ChunksCount = 10 , UpdateTime = 0/*s*/
ContentName = /b3c/wowmom/movie10  , StartChunk = 0 , ChunksCount = 10 , UpdateTime = 0/*s*/
ContentName = /b3c/wowmom/movie11  , StartChunk = 0 , ChunksCount = 10 , UpdateTime = 0/*s*/
ContentName = /b3c/wowmom/movie12  , StartChunk = 0 , ChunksCount = 10 , UpdateTime = 0/*s*/
ContentName = /b3c/wowmom/movie13  , StartChunk = 0 , ChunksCount = 10 , UpdateTime = 0/*s*/
ContentName = /b3c/wowmom/movie14  , StartChunk = 0 , ChunksCount = 10 , UpdateTime = 0/*s*/

[eFwdRulesMode]

[eCommentsMode]
-------------------------------------------------------------------------------
comments go here
inline comments possible using c-style comments '/* ... */'
```

Figure III.8 : fichier de configuration d'un nœud de type client légitime

III.5.2 Liaison et configuration des nœuds :



```
network = unibas.ccnlite.topology.CCNoEther_Tau_2cl_2rtr_1svr
description = "Example CCN over Ethernet Tau topology with 2 clients, 2 routers, 1 servers. Scenario setup taken f

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

## per node settings
**.debugLevel = 4            ## per host: 0=none, 1=Error, 2=Warning, 3=Info, 4=Detail
**.minTxPace = 10ms
**.maxCacheSlots = 400 #400
**.maxCacheBytes = 524288000Bytes #524288000
**.ccnCoreVersion = "CCN Lite v0.3.0"
*.attacker.net.ccnScenarioFile = "attacker_ccn.cfg"
*.attacker2.net.ccnScenarioFile = "attacker2_ccn.cfg"
*.client2.net.ccnScenarioFile =  "client2_ccn.cfg"
*.client3.net.ccnScenarioFile =  "client3_ccn.cfg"
*.client5.net.ccnScenarioFile =  "client5_ccn.cfg"
*.client6.net.ccnScenarioFile =  "client6_ccn.cfg"
*.client7.net.ccnScenarioFile =  "client7_ccn.cfg"

*.router1.net.ccnScenarioFile = "router1_ccn.cfg"
*.router2.net.ccnScenarioFile = "router2_ccn.cfg"
*.router3.net.ccnScenarioFile = "router3_ccn.cfg"
*.router4.net.ccnScenarioFile = "router4_ccn.cfg"
*.router5.net.ccnScenarioFile = "router5_ccn.cfg"
*.router6.net.ccnScenarioFile = "router6_ccn.cfg"

*.server1.net.ccnScenarioFile = "server1_ccn.cfg"
```

Figure III.9 fichier INI de notre topologie

Après la création de la topologie ensuite la configuration des composants de ce réseau ICN, nous avons besoin forcément de faire une liaison entre les fichiers de configuration et les appareils connecté et pour cela on dispose d'un fichier qui s'appelle INI, ce dernier a pour rôle

de lier la configuration avec le nœud qui correspond. Comme par exemple si on voit la figure5 nous remarquons que le routeur 1 correspond au fichier qui s'appelle router_ccn.cfg ainsi de suite pour les autres nœuds.

III.6 Les résultats des deux attaques planifiées :

Après avoir fait plusieurs tests et simulation d'attaque sur le cache contenu du routeur 1 avec les deux scénarios nous constatons le résultat suivant :

Scénario 1 :

Temps en seconde	**Acteur (statut)**	**Contenu**	**Résultat**
1.000000	Attacker (émetteur)	Movie1	Envoyer un intérêt
1.000164	Attacker (récepteur)	Movie1	Contenu est reçu
1.500000	Attacker (émetteur)	Movie2	Envoyer un intérêt
1.500164	Attacker (récepteur)	Movie2	Contenu est reçu
2.000000	Client2 (émetteur)	Movie4	Envoyer un intérêt
2.000164	Attacker (récepteur)	Movie4	Recevoir le contenu du CS de routeur1
2.500000	Client2 (émetteur)	Movie1	Envoyer un intérêt
2.500006	Client2 (récepteur)	Movie1	Contenu est reçu
3.000000	Client3 (émetteur)	Movie7	Envoyer un intérêt
3.000164	Attacker (récepteur)	Movie7	Recevoir le contenu du CS de routeur1
4.000000	Client3 (émetteur)	Movie4	Envoyer un intérêt
4.000006	Client3 (récepteur)	Movie4	Contenu est reçu
5.000000	Attacker2 (émetteur)	Movie10	Envoyer un intérêt
5.000210	Attacker2 (récepteur)	Movie10	Contenu est reçu
6.000000	Client5 (émetteur)	Movie1	Envoyer un intérêt
6.000116	Attacker2 (récepteur)	Movie1	Recevoir le contenu du CS de routeur2
7.000000	Client6 (émetteur)	Movie1	Envoyer un intérêt
7.000006	Client6 (récepteur)	Movie1	Contenu est reçu
9.000000	Client7 (émetteur)	Movie1	Envoyer un intérêt

9.000010	Client7 (récepteur)	Movie1	Contenu est reçu
10.50000	Client7 (émetteur)	Movie2	Envoyer un intérêt
10.50015	Client7 (récepteur)	Movie2	Contenu est reçu

Tab III.4 : Déroulement du premier scénario

Dans ce scénario là les client envoie un intérêt qui n'existe pas dans le cache contenu du routeur voisin alors l'intérêt contenue son chemin jusqu'au serveur ou routeur qui contient le contenu souhaité ensuite le contenu est envoyé dans le chemin inverse du coup les vrai émetteurs ou bien demandeurs de contenue reçoivent aucune réponse , parce que les attaquants qui interceptent les contenus .Mais dans un autre temps nous remarquons que les attaquants n'arrivent pas à recevoir les contenus lorsqu'ils existe déjà dans le cache routeur voisin du coup on en déduit que le premier scénario a fonctionné avec sucée à 100%.

Scénario 2 :

Temps en seconde	**Acteur (statut)**	**Contenu**	**Résultat**
1.000000	Attacker (émetteur)	Movie1	Envoyer un intérêt
1.000164	Attacker (récepteur)	Movie1	Contenu est bien reçu
1.500000	Attacker (émetteur)	Movie2	Envoyer un intérêt
1.500164	Attacker (récepteur)	Movie2	Contenu est bien reçu
2.000000	Client2 (émetteur)	Movie4	Envoyer un intérêt
2.000164	Client2 (récepteur)	Movie4	Contenu est bien reçu
2.500000	Client2 (émetteur)	Movie1	Envoyer un intérêt
2.500006	Attacker (récepteur)	Movie1	Recevoir le contenu du CS de routeur1
3.000000	Client3 (émetteur)	Movie7	Envoyer un intérêt
3.000164	Client3 (récepteur)	Movie7	Contenu est bien reçu
4.000000	Client3 (émetteur)	Movie4	Envoyer un intérêt
4.000006	Attacker (récepteur)	Movie4	Recevoir le contenu du CS de routeur1
5.000000	Attacker2 (émetteur)	Movie10	Envoyer un intérêt

5.000210	Attacker2 (récepteur)	Movie10	Contenu est reçu
6.000000	Client5 (émetteur)	Movie1	Envoyer un intérêt
6.000116	Client5 (récepteur)	Movie1	Contenu est bien reçu
7.000000	Client6 (émetteur)	Movie1	Envoyer un intérêt
7.000006	Attacker2 (récepteur)	Movie1	Recevoir le contenu du CS de routeur2
9.000000	Client7 (émetteur)	Movie1	Envoyer un intérêt
9.000010	Client7 (récepteur)	Movie1	Contenu est reçu
10.50000	Client7 (émetteur)	Movie2	Envoyer un intérêt
10.50015	Client7 (récepteur)	Movie2	Contenu est reçu

Tab III.5 : Déroulement du deuxième scénario

Dans ce dernier déroulement le cas où le contenu existe déjà dans le cache contenu du routeur voisin, les clients envoient des intérêts aux routeurs les plus proches, après la recherche au niveau de leur CS , ils transmettent les contenus aux attaquants et non pas aux vrais clients émetteurs, c'est-à-dire que les attaquants ont réussi à intercepter les contenus et cette fois-ci nous remarquons que lorsque le contenu n'est pas disponible dans CS du routeur cible , les vrais émetteurs reçoivent leur réponses sans aucune intervention malveillante des attaquants , cela veut dire que le deuxième scenario fonctionne dans tous les cas et plusieurs test à 100%.

Scénario 3 :

Temps en seconde	**Acteur (statut)**	**Contenu**	**Résultat**
1.000000	Attacker (émetteur)	Movie1	Envoyer un intérêt
1.000164	Attacker (récepteur)	Movie1	Contenu est reçu
1.500000	Attacker (émetteur)	Movie2	Envoyer un intérêt
1.500164	Attacker (récepteur)	Movie2	Contenu est reçu
2.000000	Client2 (émetteur)	Movie4	Envoyer un intérêt
2.000164	Attacker (récepteur)	Movie4	Recevoir le contenu du CS de routeur1
2.500000	Client2 (émetteur)	Movie1	Envoyer un intérêt

2.500006	Attacker (récepteur)	Movie1	Recevoir le contenu du CS de routeur1
3.000000	Client3 (émetteur)	Movie7	Envoyer un intérêt
3.000164	Attacker (récepteur)	Movie7	Recevoir le contenu du CS de routeur1
4.000000	Client3 (émetteur)	Movie4	Envoyer un intérêt
4.000006	Attacker (récepteur)	Movie4	Recevoir le contenu du CS de routeur1
5.000000	Attacker2 (émetteur)	Movie10	Envoyer un intérêt
5.000210	Attacker2 (récepteur)	Movie10	Contenu est reçu
6.000000	Client5 (émetteur)	Movie1	Envoyer un intérêt
6.000116	Attacker2 (récepteur)	Movie1	Recevoir le contenu du CS de routeur2
7.000000	Client6 (émetteur)	Movie1	Envoyer un intérêt
7.000006	Attacker2 (récepteur)	Movie1	Recevoir le contenu du CS de routeur2
9.000000	Client7 (émetteur)	Movie1	Envoyer un intérêt
9.000010	Client7 (récepteur)	Movie1	Contenu est reçu
10.50000	Client7 (émetteur)	Movie2	Envoyer un intérêt
10.50015	Client7 (récepteur)	Movie2	Contenu est reçu

Tab III.6 : Déroulement du troisième scénario

Dans ce déroulement nous faisons, l'attaque du premier scenario et le deuxième en même temps et nous remarquons que les attaquants ont intercepté tous les réponses qui était prévus destiné aux autres clients du coup nous en déduit que cette attaque a réussi dans plusieurs tests à 100%.

III.8 Conclusion

Après avoir effectué à multiples reprises aussi depuis le résultat obtenu en voyant aussi que le contenu a été dérouté et retransmet à l'attaquant cela signifie qu'il y a une faille dans ce réseau et que le réseau CCN est très vulnérable toutefois nous peuvent constater que le réseau CCN n’est pas assez sécurisé.

Les derniers tests effectués révèlent que le système possède une grave faille et la vulnérabilité est sûr car à chaque essaie le résultat était positif. On conclut donc que le système est vulnérable à chaque attaque visant le contenu qui passe par la table PIT et le niveau de vulnérabilité est de 100 %. Dans le chapitre suivant, nous proposerons une contre mesure à cette attaque.

Chapitre IV : Contre-Mesure Kerberos-CCN

IV-1 Introduction

Les données sont la cible la plus convoitée chez les pirates, un exploit tant désiré par tous les attaquants des réseaux informatiques, d'autant plus dans les réseaux centrés contenu, ou tout l'intérêt est basé sur ça.

Nous avons montré dans le chapitre précédent la facilité de telle attaques dans les réseaux CCN vu leurs vulnérabilité en utilisant une attaque des attaques les plus fréquentes à savoir l'usurpation d'identité pour arriver au cache de n'importe quel routeur et donc aux données des serveurs qui sont multiplié dans des centaines voire des milliers de routeurs dans le monde entier.

Dans ce chapitre, nous allons détailler la solution « Kerberos-CCN » que nous avons proposée comme contre mesure aux attaques d'usurpation d'identité dans le but du vol de contenu, nous commençons notre chapitre par expliquer notre solution, par la suite, nous donnerons les différents scénarios de contre mesure, et nous terminerons par un jeu de test.

IV.2 Notre solution « Kerberos-CCN »

La distribution des tickets dans le protocole Kerberos est un concept principal de son fonctionnement pour garantir une authentification et transmission de flux de donnée sécurisée entre les terminaux et les serveurs du réseau et pour faire cette topologie ils ont conçu une troisième partie qui s'appelle KDC (Key Distribution Center) et ce dernier est chargé de générer les clés à tous les acteurs du réseau.

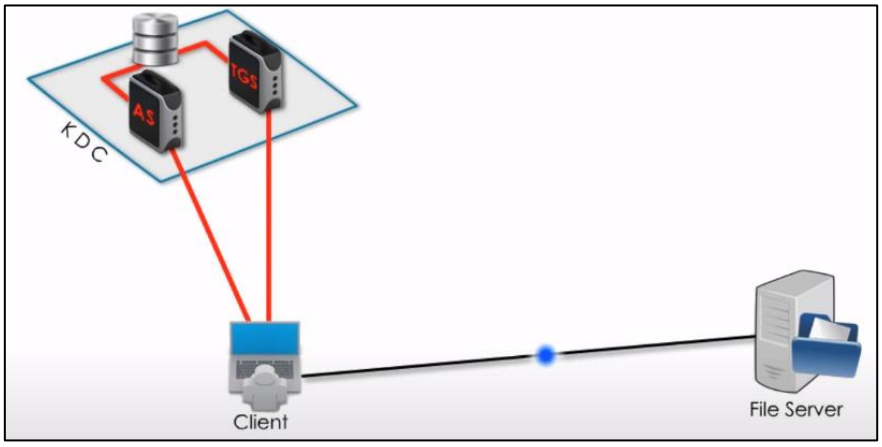


Figure IV.1 : Topologie de Kerberos

Du coup nous avons inspiré de ce protocole pour faire notre travail de contre mesure de l'attaque PIT en ajoutant une troisième partie dans l'architecture de CCN qui s'appelle

distributeur de clé (KD) et par rapport au Kerberos notre troisième partie fonctionne comme suit :

1. Elle est chargée de générer de clé publique et privé, il ne partage que les clés publiques au serveur, routeurs et clients mais elle ne partage les clés privées qu'au serveurs et routeurs
2. Uniquement les routeurs et les serveurs qui sont liées directement au KD
3. Chaque routeur de réseau ICN a une liste d'attente au niveau de cette partie

Et une autre mise à jour pour le paquet d'intérêt, nous avons besoin d'un champ supplémentaire pour sauvegarder des informations importantes lors la circulation de paquet tel que l'identifiant de routeur, la signature de contenu et la clé publique

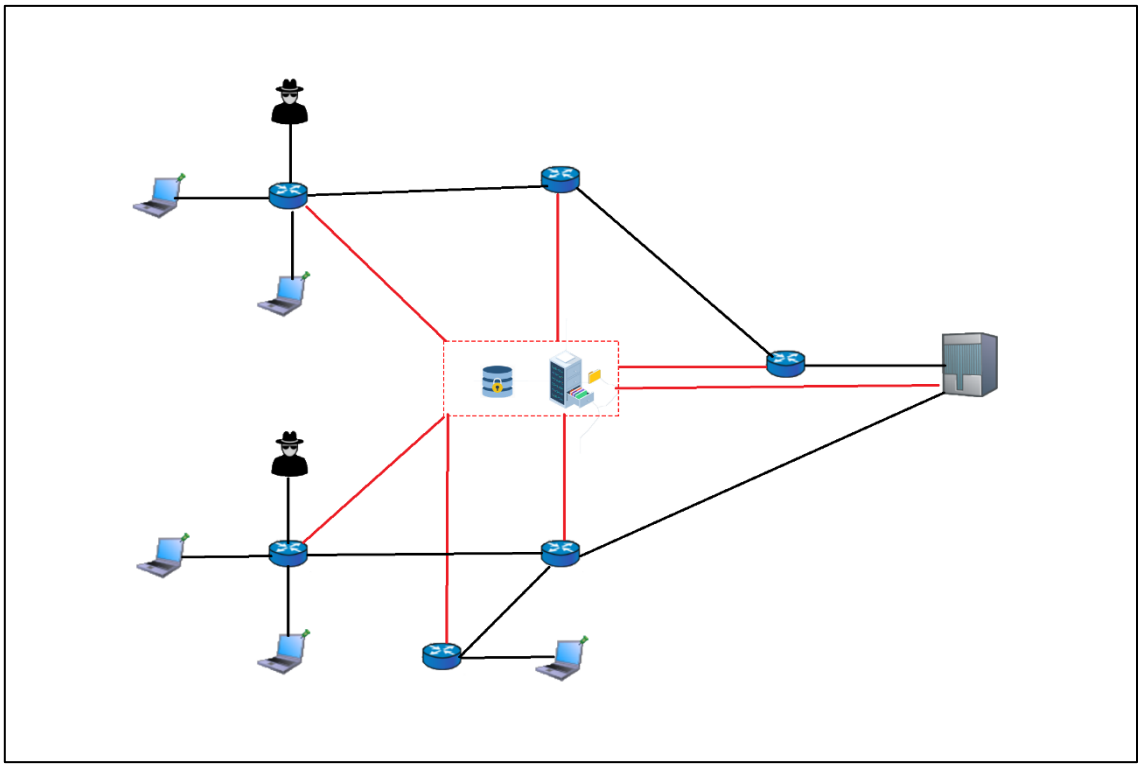


Figure IV.2 : Topologie de notre contre mesure

Dans cette topologie il y a quatre acteurs en principe qui sont : les terminaux (pc, téléphone, ... etc.) Les routeurs, les serveurs et le serveur KD, ce dernier est la troisième partie et un élément essentiel qui garantit la génération des clés publiques et privées des terminaux de réseau CCN, ses routeurs et ses serveurs via l'algorithme de RSA ainsi garantie la distribution de ces clés.

Le but de cette stratégie est de protéger les données ou bien les contenus circulant dans le réseau d'être volé, usurpé ou intercepté par des utilisateurs malveillants ou des utilisateurs espions qui n'ont pas le droit à une information transmit et ainsi avant d'envoyer une information à n'importe quel utilisateur, là au niveau de chaque routeur de CCN ils vérifient si le terminal reçus par la table PIT qui va reçoit la donnée est le vrai émetteur qui a demandé le contenu ou pas, donc dans le cas où ce n'est pas le vrai récepteur de contenu ,le système va automatiquement deviner que ce dernier utilisateur il a détournée les informations d'une manière en modifiant la table PIT.

IV.2 Scenario de contre mesure

1. Client envoie un intérêt d'un contenu (x) et envoie aussi sa clé publique encapsuler dans le paquet d'intérêt.
2. Le retour1 reçoit l'intérêt et la clé publique de client ensuite l'envoie au KD pour la sauvegarder dans sa liste d'attente (FIFO).
3. Avant envoyer l'intérêt au nœud suivant le routeur1 encapsuler son ID dans le paquet d'intérêt
4. L'intérêt de la demande circule dans le réseau pour trouver le contenu selon FIB.
5. Un serveur contenant le contenu (x) reçoit l'intérêt.
6. Ce serveur doit obtenir la clé publique de vrai terminal émetteur et pour l'obtenir il envoie une demande au KD pour récupérer cette clé publique sauvegarder dans la liste d'attente de routeur via son ID qu'est encapsuler dans le paquet d'intérêt
7. Le serveur reçoit la clé publique
8. Le serveur prépare le paquet pour l 'encapsuler et avant l'envoyer, il hash une copie de contenu par l'algorithme MD5 et ensuite le chiffre via la clé publique de client émetteur.
9. Envoie le paquet avec le contenu (x) et la signature électronique de ce contenu encapsuler dans le paquet d'intérêt.
10. Le dernier routeur reçoit la signature et le contenu, avant envoyer le contenu le routeur chercher le privé de client récepteur (selon la table PIT) dans le KD.
11. Le routeur déchiffre la signature par la clé reçue et en fin obtient un hash, il hash en ce moment le contenu venant de la source via MD5 et le compare avec le hash obtenu après de le déchiffrement.

12. Si les deux hash sont identiques le routeur transmet le contenu et ça serait évidemment le vrai client émetteur au début, et s'ils sont pas identique le routeur n'envoie pas le contenu et va devenir que c'est une interception par un client malveillant.

IV.3 Jeu de test

Nous avons effectué 22 tests comportant 6 attaques et 16 intérêts légitimes, les résultats sont mis dans le tableau suivant :

Temps en seconde	**Acteur (statut)**	**Contenu**	**Résultat**
1.000000	Attacker (émetteur)	Movie1	Envoyer un intérêt
1.000164	Attacker (récepteur)	Movie1	Contenu est reçu
1.500000	Attacker (émetteur)	Movie2	Envoyer un intérêt
1.500164	Attacker (récepteur)	Movie2	Contenu est reçu
2.000000	Client2 (émetteur)	Movie4	Envoyer un intérêt
2.000164	Aucun récepteur	/	
2.500000	Client2 (émetteur)	Movie1	Envoyer un intérêt
2.500006	Aucun récepteur	/	
3.000000	Client3 (émetteur)	Movie7	Envoyer un intérêt
3.000164	Aucun récepteur	/	
4.000000	Client3 (émetteur)	Movie4	Envoyer un intérêt
4.000006	Aucun récepteur	/	
5.000000	Attacker2 (émetteur)	Movie10	Envoyer un intérêt
5.000210	Attacker2 (récepteur)	Movie10	Contenu est reçu
6.000000	Client5 (émetteur)	Movie1	Envoyer un intérêt
6.000116	Aucun récepteur	/	
7.000000	Client6 (émetteur)	Movie1	Envoyer un intérêt
7.000006	Aucun récepteur	/	
9.000000	Client7 (émetteur)	Movie1	Envoyer un intérêt
9.000010	Client7 (récepteur)	Movie1	Contenu est reçu
10.50000	Client7 (émetteur)	Movie2	Envoyer un intérêt
10.50015	Client7 (récepteur)	Movie2	Contenu est reçu

Tab IV.7 : Déroulement de la contre-mesure

D’après le résultat de ce déroulement on remarque bien que toutes les attaques ont été bloqué par les routeurs une fois trouver que les récepteurs n’est le vrai alors ces routeurs ont gardé l’information et ont pas risqué de l’envoyer vu qu’ont perdu le vrai émetteur de la demande.

IV.4 Conclusion

Le mécanisme de contre mesure déployé dans ce travail a été efficace à 100%, aucune attaque n’a eu de réponse, toutefois, l’utilisateur légitime ne reçoit aucune réponse, devra renvoyer sa demande à plusieurs fois s’il veut une réponse. Cette faiblesse devra être traitée par un autre mécanisme plus robuste de sécurité.

Conclusion Générale

Depuis la création de l'internet dans les années 70s et jusqu'au jour d'aujourd'hui, le monde ne cesse de se transformer, et les besoins en communication, transmission ne cessent d'augmenter ce qui a rendu l'architecture actuelle de l'internet incapable de satisfaire trop longtemps toute cette demande et ses exigences, ce qui nous oblige à penser à de nouvelles architectures et de nouveaux paradigmes plus adaptés aux nouveaux besoins et contraintes.

D'ailleurs des chercheurs dans le domaine des réseaux ont d'ores et déjà réfléchi et pensé à une nouvelle solution mieux adaptée aux usages actuels, on parle ici des ICN, toutefois cette nouvelle architecture malgré les louanges reste très vulnérable aux attaques, et ne répond pas vraiment au respect de la confidentialité et à la sécurité du contenu.

Dans notre travail nous avons proposé une solution pour maitre fin contre les attaques de vol cache dans les réseaux CCN basées sur l'usurpation d'identité des usagers légitimes.

Après la création d'une topologie adéquate à notre conception pour tester les attaques et mis en place plusieurs scénarios d'attaques et ensuite on a obtenu des résultats de chaque une et son déroulement dans le réseau.

La simulation lors du test de la solution nous a aidé beaucoup à remarquer et voir les vulnérabilités rencontrées dans le système jusqu'à arriver à une contre mesure finale sachant que dans l'informatique tout ce qu'est nouveau est vulnérable alors ça reste une solution efficace en ce moment voyant les résultats obtenus et reste à améliorer toujours.

Bibliographies

[1] E. G. AbdAllah, H. S. Hassanein and M. Zulkernine, "A Survey of Security Attacks in Information-Centric Networking," in *IEEE Communications Surveys & Tutorials*, vol. 17, no. 3, pp. 1441-1454, thirdquarter 2015.

[2] Majed, Al-qutwani & Wang, Xingwei & Yi, Bo. (2019). Name Lookup in Named Data Networking: A Review. Information. 10. 85. 10.3390/info10030085.

[3] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher and B. Ohlman, "A survey of information-centric networking," in *IEEE Communications Magazine*, vol. 50, no. 7, pp. 26-36, July 2012. doi: 10.1109/MCOM.2012.6231276

[4] K. Yu *et al*., "Information-Centric Networking: Research and Standardization Status," in *IEEE Access*, vol. 7, pp. 126164-126176, 2019. doi: 10.1109/ACCESS.2019.2938586

[5] R. Tourani, S. Misra, T. Mick and G. Panwar, "Security, Privacy, and Access Control in Information-Centric Networking: A Survey," in *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, pp. 566-600, Firstquarter 2018.

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

[7] Natalya Rozhnova. Congestion control for Content-Centric Networking. Networking and Internet Architecture [cs.NI]. Université Pierre et Marie Curie - Paris VI, 2015.

[8] Maroua Meddeb. Information-Centric Networking, A natural design for IoT applications? Other. INSA de Toulouse ; Ecole Nationale des Sciences de l'Informatique, 2017.

[9] Xuan Zeng. Towards seamless mobility in ICN: connectivity, security, and reliability. Networking and Internet Architecture [cs.NI]. Sorbonne Université, 2018.

[10] A. Mohaisen, X. Zhang, M. Schuchard, H. Xie, and Y. Kim, "Protecting access privacy of cached contents in information centric networks," in Proc. SIGCOMM, Hong Kong, China, May 2013, pp. 1001–1003.

[11] C. Ghali, G. Tsudik, and E. Uzun, "Needle in a haystack: Mitigating content poisoning in named-data networking," in Proc. SENT, San Diego, CA, USA, 2014, pp. 1–10

[12] Z. Zhou, X. Tan, H. Li, Z. Zhao and D. Ma. MobiNDN: A mobility support architecture for NDN. In Proceedings of the 33rd Chinese Control Conference, pages 5515–5520, July 2014. (Cited on pages 31 and 53.)

[13] C. Dannewitz, J. Golic, B. Ohlman, and B. Ahlgren, "Secure naming for a network of information," in Proc. IEEE INFOCOM, Mar. 2010, pp. 1–6.

[14] S. Arianfar, T. Koponen, B. Raghavan, and S. Shenker, "On preserving privacy in content-oriented networks," in Proc. ACM SIGCOMM Workshop ICN, Aug. 2011, pp. 19–24

[15] D. L. Chaum, "Untraceable electronic mail, return addresses, digital pseudonyms," Commun. ACM, vol. 24, no. 2, pp. 84–90, Feb. 1981.

[16] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The secondgeneration onion router," in Proc. 13th USENIX Security Symp., 2004, p. 21.

[17] I. Clarke, T. W. Hong, S. G. Miller, O. Sandberg, and B. Wiley, "Protecting free expression online with Freenet," IEEE Internet Comput., vol. 6, no. 1, pp. 40–49, Jan./Feb. 2002.

[18] R. Canetti, C. Dwork, M. Naor, and R. Ostrovsky, "Deniable encryption," in Proc. CRYPTO, vol. 1294, Lecture Notes in Computer Science, 1997,

[19] M. Ion, J. Zhang, M. Schuchard, and E. M. Schooler, "Toward contentcentric privacy in ICN: Attribute-based encryption and routing," in Proc. ASIA CCS, Hangzhou, China, Aug. 2013, pp. 513–514.

[20] P. Gasti, G. Tsudik, E. Uzun, and L. Zhang, "DoS & DDoS in named data networking," in Proc 22nd Int. Conf. Comput. Commun. Netw., 2013, pp. 1–7.

[21] N. Fotiou, G. F. Marias, and G. C. Polyzos, "Fighting spam in publish/subscribe networks using information ranking," in Proc. 6th EURO-NF Conf. NGI, Paris, France, Jun. 2010, pp. 1–6.

[22] A. Compagno, M. Conti, P. Gasti, and G. Tsudik, "Poseidon: Mitigating interest flooding DDoS attacks in named data networking," in Proc. IEEE 38th Conf. Local Comput. Netw., Oct. 2013, pp. 630–638

[23] S. T. Zargar, J. Joshi, and D. Tipper, "A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks," IEEE Commun. Surveys Tuts., vol. 15, no. 4, pp. 2046–2069, 2013.

[24] J. Mirkovic and P. Reiher, "A taxonomy of DDoS attack and DDoS defense mechanisms," ACM SIGCOMM Comput. Commun. Rev., vol. 34, no. 2, pp. 39–53, Apr. 2004.

[25] L. Lu, M. C. Chan, and E. C. Chang, “A general model of probabilistic packet marking for IP traceback,” in Proc. ASIACCS, 2008, pp. 179–188.

[26] E. Kline, A. Afanasyev, and P. Reiher, “Shield: DoS filtering using traffic deflecting,” in Proc. 19th IEEE ICNP, 2011, pp. 37–42.

[27] A. Mohaisen, X. Zhang, M. Schuchard, H. Xie, and Y. Kim, “A DoSlimiting network architecture,” ACM SIGCOMM CCR, vol. 35, no. 4, pp. 241–252, Oct. 2005.