

# **Rapport de projet : Surveillance du réseau avec Wireshark**

**Nom** : Abdoul-Rachid Bawa

**Niveau** : Étudiant en 2<sup>e</sup> année d'Informatique de Gestion

**Période** : Janvier 2025

**Outil utilisé** : Wireshark

**Objectif** : Observer et analyser en temps réel le trafic réseau de différents protocoles courants.

---

## **1. Présentation de l'outil**

**Wireshark** est un analyseur de paquets réseau qui permet de capturer, filtrer et inspecter les données circulant sur un réseau informatique.

Il est utilisé pour le diagnostic, la surveillance de trafic, la sécurité réseau ou encore l'apprentissage des protocoles.

## **2. Objectifs pédagogiques**

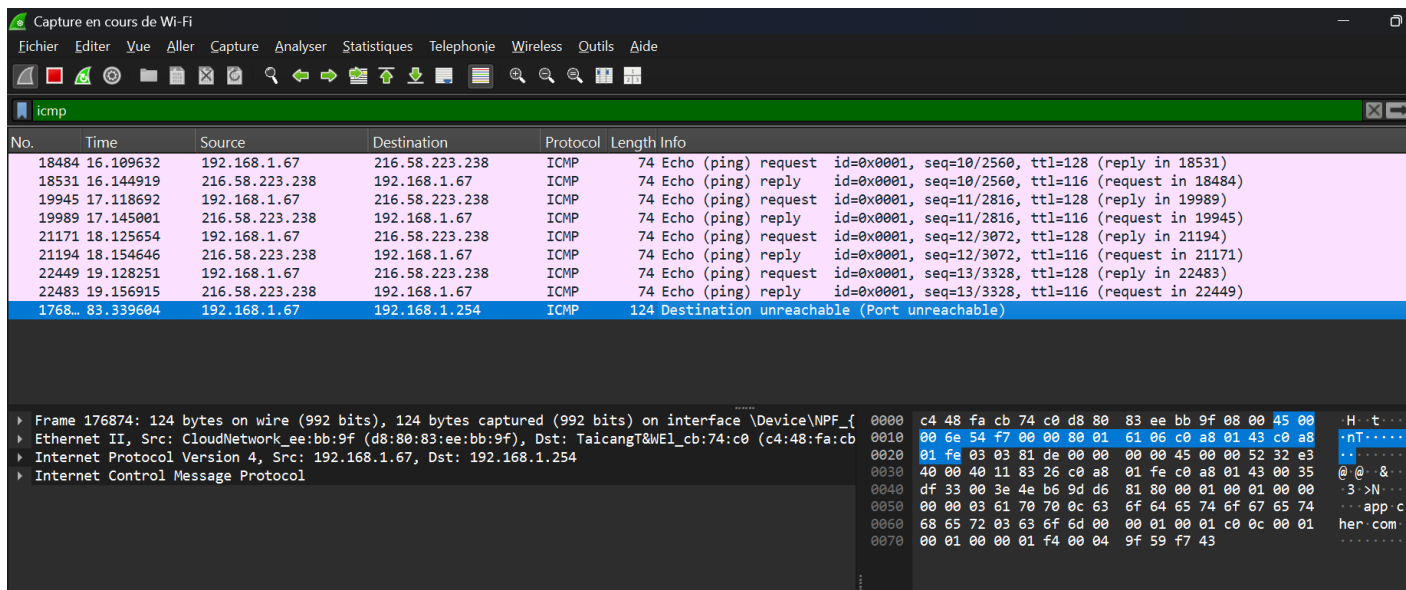
- Se familiariser avec les paquets réseau (couches OSI)
- Identifier des protocoles courants (ICMP, DNS, HTTP)
- Comprendre comment les données circulent entre client et serveur

## **3. Expériences réalisées**

### **Test 1 : Ping ICMP vers Google**

- **Protocole observé** : ICMP
- **Commande utilisée** : ping google.com
- **Filtre Wireshark** : icmp
- **Observations** :
  - 4 requêtes Echo (ping) Request envoyées
  - 4 réponses Echo (ping) Reply reçues
  - Adresse IP de Google identifiée (ex : 142.250.x.x)
  - Temps de réponse affiché dans les paquets

## Capture :



The image shows a Wireshark packet capture of ICMP Echo (ping) requests and replies. The packet list table is as follows:

No.	Time	Source	Destination	Protocol	Length	Info
18484	16.109632	192.168.1.67	216.58.223.238	ICMP	74	Echo (ping) request id=0x0001, seq=10/2560, ttl=128 (reply in 18531)
18531	16.144919	216.58.223.238	192.168.1.67	ICMP	74	Echo (ping) reply id=0x0001, seq=10/2560, ttl=116 (request in 18484)
19945	17.118692	192.168.1.67	216.58.223.238	ICMP	74	Echo (ping) request id=0x0001, seq=11/2816, ttl=128 (reply in 19989)
19989	17.145001	216.58.223.238	192.168.1.67	ICMP	74	Echo (ping) reply id=0x0001, seq=11/2816, ttl=116 (request in 19945)
21171	18.125654	192.168.1.67	216.58.223.238	ICMP	74	Echo (ping) request id=0x0001, seq=12/3072, ttl=128 (reply in 21194)
21194	18.154646	216.58.223.238	192.168.1.67	ICMP	74	Echo (ping) reply id=0x0001, seq=12/3072, ttl=116 (request in 21171)
22449	19.128251	192.168.1.67	216.58.223.238	ICMP	74	Echo (ping) request id=0x0001, seq=13/3328, ttl=128 (reply in 22483)
22483	19.156915	216.58.223.238	192.168.1.67	ICMP	74	Echo (ping) reply id=0x0001, seq=13/3328, ttl=116 (request in 22449)
1768...	83.339604	192.168.1.67	192.168.1.254	ICMP	124	Destination unreachable (Port unreachable)

The packet details pane for the selected packet (No. 1768...) shows the following structure:

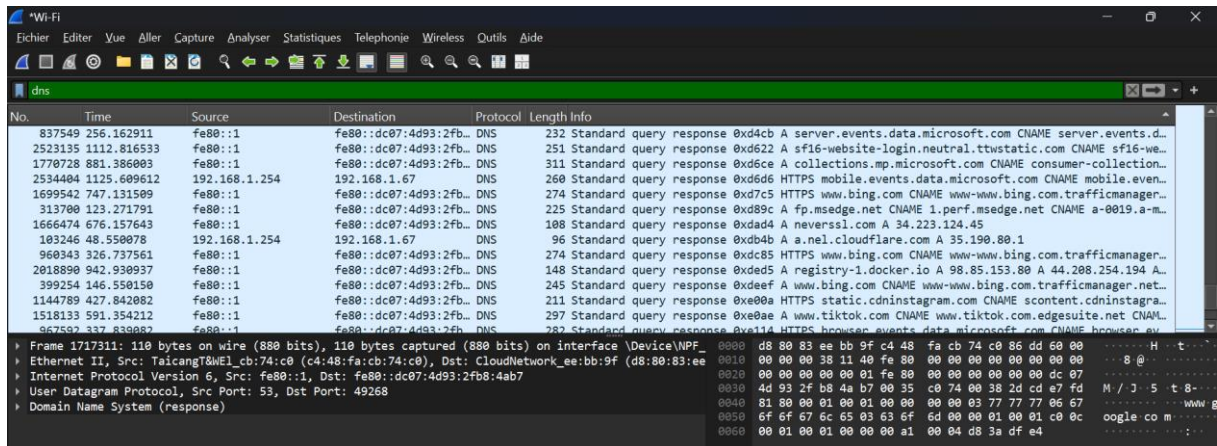
- Frame 176874: 124 bytes on wire (992 bits), 124 bytes captured (992 bits) on interface \Device\NPF\_{...}
- Ethernet II, Src: CloudNetwork\_ee:bb:9f (d8:80:83:ee:bb:9f), Dst: TaicangT&WEl\_cb:74:c0 (c4:48:fa:cb:74:c0)
- Internet Protocol Version 4, Src: 192.168.1.67, Dst: 192.168.1.254
- Internet Control Message Protocol

The packet bytes pane shows the raw data in hexadecimal and ASCII format.

## Test 2 : Requête DNS vers Wikipedia

- **Protocole observé :** DNS
- **Commande utilisée :** nslookup wikipedia.org
- **Filtre Wireshark :** dns
- **Observations :**
  - Requête Standard query A wikipedia.org
  - Réponse contenant l'adresse IP de Wikipedia
  - Serveur DNS utilisé visible dans les métadonnées

## Capture :



The image shows a Wireshark capture of DNS traffic. The top pane displays a list of captured packets, with the 'dns' filter applied. The middle pane shows the details of the selected packet (No. 1717311), which is a DNS response. The bottom pane shows the raw packet data in hexadecimal and ASCII.

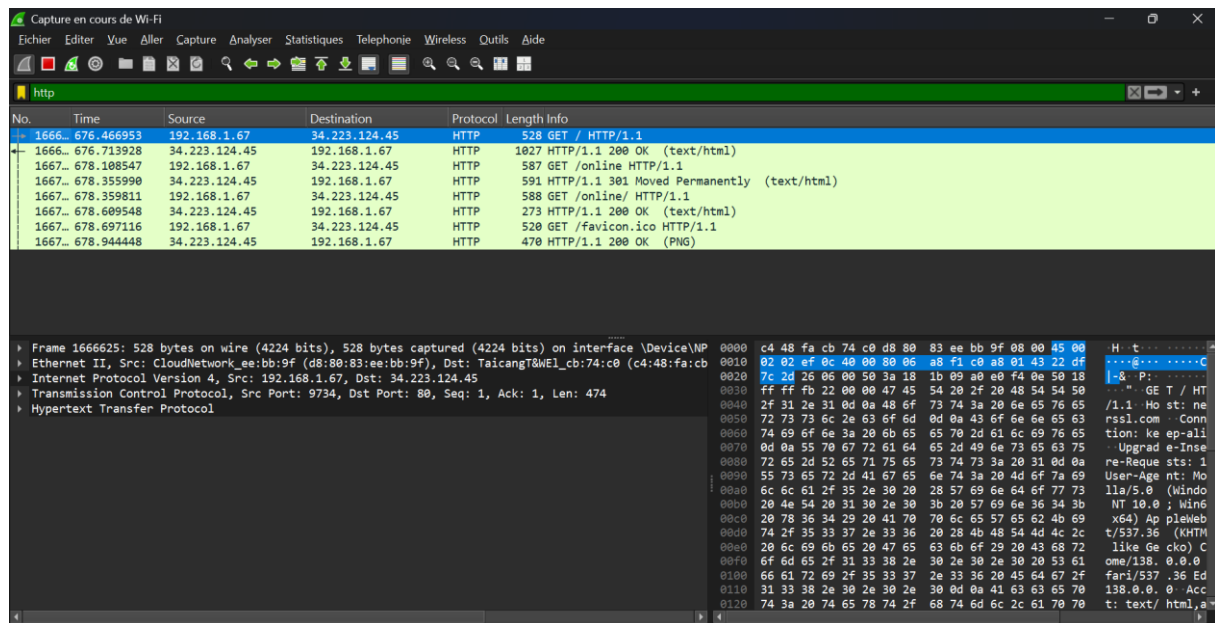
No.	Time	Source	Destination	Protocol	Length	Info
837549	256.162911	fe80::1	fe80::dc07:4d93:2fb...	DNS	232	Standard query response 0xd4cb A server.events.data.microsoft.com CNAME server.events.d...
2523135	1112.816533	fe80::1	fe80::dc07:4d93:2fb...	DNS	251	Standard query response 0xd622 A sf16-website-login.neutral.ttwstatic.com CNAME sf16-we...
1770728	881.366803	fe80::1	fe80::dc07:4d93:2fb...	DNS	311	Standard query response 0xd6ce A collections.mp.microsoft.com CNAME consumer-collection...
2534404	1125.609612	192.168.1.254	192.168.1.67	DNS	260	Standard query response 0xd6de HTTPS mobile.events.data.microsoft.com CNAME mobile.even...
1699542	747.131509	fe80::1	fe80::dc07:4d93:2fb...	DNS	274	Standard query response 0xd7c5 HTTPS www.bing.com CNAME www-www.bing.com.trafficmanager...
313700	123.271791	fe80::1	fe80::dc07:4d93:2fb...	DNS	225	Standard query response 0xd89c A fp.msedge.net CNAME 1.perf.msedge.net CNAME a-0019.a-m...
1666474	676.157643	fe80::1	fe80::dc07:4d93:2fb...	DNS	108	Standard query response 0xdad4 A neverssl.com A 34.223.124.45
103246	48.550078	192.168.1.254	192.168.1.67	DNS	96	Standard query response 0xdb4b A a.nel.cloudflare.com A 35.190.80.1
960343	326.737561	fe80::1	fe80::dc07:4d93:2fb...	DNS	274	Standard query response 0xdc85 HTTPS www.bing.com CNAME www-www.bing.com.trafficmanager...
2018890	942.930937	fe80::1	fe80::dc07:4d93:2fb...	DNS	148	Standard query response 0xdec5 A registry-1.docker.io A 98.85.153.80 A 44.208.254.194 A...
399254	146.550150	fe80::1	fe80::dc07:4d93:2fb...	DNS	245	Standard query response 0xdeef A www.bing.com CNAME www-www.bing.com.trafficmanager.net...
1144789	427.842082	fe80::1	fe80::dc07:4d93:2fb...	DNS	211	Standard query response 0xe00a HTTPS static.cdninstagram.com CNAME scontent.cdninstagra...
1518133	591.354212	fe80::1	fe80::dc07:4d93:2fb...	DNS	297	Standard query response 0xe0ae A www.tiktok.com CNAME www.tiktok.com.edgesuite.net CNAME...
967502	337.830082	fe80::1	fe80::dc07:4d93:2fb...	DNS	282	Standard query response 0xe114 HTTPS browser.events.data.microsoft.com CNAME browser.ev...

Frame 1717311: 110 bytes on wire (880 bits), 110 bytes captured (880 bits) on interface \Device\NPF...  
Ethernet II, Src: TaicangT&WEL\_cb:74:c0 (c4:48:fa:cb:74:c0), Dst: CloudNetwork\_ee:bb:9f (d8:80:83:ee...)  
Internet Protocol Version 6, Src: fe80::1, Dst: fe80::dc07:4d93:2fb8:4ab7  
User Datagram Protocol, Src Port: 53, Dst Port: 49268  
Domain Name System (response)

## Test 3 : Connexion HTTP simple

- **Protocole observé : HTTP**
- **Site visité : http://neverssl.com**
- **Filtre Wireshark : http**
- **Observations :**
  - Requête GET / envoyée au serveur
  - En-têtes visibles : Host, User-Agent, Accept, etc.
  - Réponse 200 OK contenant la page HTML du site

## Capture :



## 4. Conclusion générale

Ce mini projet m'a permis de :

- Visualiser concrètement le fonctionnement des protocoles réseau
- Comprendre l'utilité de chaque protocole dans la communication client-serveur
- M'initier à l'analyse de trafic pour le diagnostic et la cybersécurité

**Wireshark** est un outil précieux pour tout technicien réseau ou développeur voulant comprendre comment ses applications interagissent sur le réseau.

**Fait à Lomé, en Janvier 2025**

*Abdoul-Rachid Bawa*

