Table des matières

[1. Informations de base 1](#_Toc148625295)

[2. Compréhension du protocole ARP 2](#_Toc148625296)

[3. Simulation ARP 5](#_Toc148625297)

[4. Analyse de Traces 6](#_Toc148625298)

[5. Programmation Scapy 6](#_Toc148625299)

# Informations de base

## Objectifs

* Se familiariser avec le protocole ARP et l’acheminement des données grâce à lui.
* Analyser des paquets ARP dans une trace de trafic avec Wireshark
* Programmer à l’aide une analyse de paquet à l’aide de Scapy

## Matériel fourni

* Une trace de trafic arp\_et\_ping.pcapng
* Une trace de trafic arp\_et\_ping\_capture\_2.pcapng

## Modalités

* Durée : Le reste du cours
* Individuel
* Utilisez ce document pour remplir le travail

## À faire

1. Complétez l’exercice en remplissant chaque rectangle vide
2. Remettez-le-moi sur Léa dans Exercice 13

# Compréhension du protocole ARP

## Pouvez-vous me faire une analogie simple qui expliquerait le concept d’ARP?

Ex. : vous pouvez prendre un livre dans une bibliothèque, etc.

|  |
| --- |
|  |

## Selon le scénario suivant :

Résolution ARP pour une communication entre A et B. remplissent les endroits vides selon vos recherches et votre compréhension du protocole.

### Configuration initiale :

* **Ordinateur A (192.168.1.2) souhaite envoyer un ping à l'ordinateur B (192.168.1.3).**
* **Les deux ordinateurs sont dans le même réseau local.**
* L’ordinateur B vient tout juste de se connecter sur le [ \_\_\_\_\_\_\_\_\_\_\_\_ ].
* Ordinateur A connaît l'adresse IP de B ([ \_\_\_\_\_\_\_\_\_\_\_\_ ]), mais ne connaît pas son adresse [ \_\_\_\_\_\_\_\_\_\_\_\_ ].
* La table ARP de l'ordinateur A est [ \_\_\_\_\_\_\_\_\_\_\_\_ ].

### Étape 1 - Ordinateur A envoie une requête ARP :

Avant d'envoyer le ping, l'ordinateur A doit découvrir l'adresse MAC correspondant à l'adresse [ \_\_\_\_\_\_\_\_\_\_\_\_ ]de B.

L'ordinateur A envoie une requête ARP (ARP Request) en [ \_\_\_\_\_\_\_\_\_\_\_\_ ]sur le réseau local. Cette requête ARP contient :

* Adresse IP source : [ \_\_\_\_\_\_\_\_\_\_\_\_ ] (A)
* Adresse MAC source : L'adresse MAC de [ \_\_\_\_\_\_\_\_\_\_\_\_ ]
* Adresse IP de destination : [ \_\_\_\_\_\_\_\_\_\_\_\_ ] (B)
* Adresse MAC de destination : [ \_\_\_\_\_\_\_\_\_\_\_\_ ] (broadcast)

### Étape 2 - Ordinateur B répond avec une réponse ARP :

Lorsque l'ordinateur B reçoit la requête ARP, il vérifie que l'adresse IP de destination ([ \_\_\_\_\_\_\_\_\_\_\_\_ ]) correspond à la sienne.

L'ordinateur B répond avec une réponse ARP (ARP Reply) en indiquant son adresse [ \_\_\_\_\_\_\_\_\_\_\_\_ ]. Cette réponse ARP contient :

* Adresse IP source : [ \_\_\_\_\_\_\_\_\_\_\_\_ ] (B)
* Adresse MAC source : L'adresse MAC de l'[ \_\_\_\_\_\_\_\_\_\_\_\_ ]
* Adresse IP de destination : [ \_\_\_\_\_\_\_\_\_\_\_\_ ] (A)
* Adresse MAC de destination : L'adresse MAC de l'[ \_\_\_\_\_\_\_\_\_\_\_\_ ]

### Étape 3 - Ordinateur A reçoit la réponse ARP :

Lorsque l'ordinateur A reçoit la réponse ARP de B, il met à jour sa [ \_\_\_\_\_\_\_\_\_\_\_\_ ] avec la correspondance IP-MAC de B.

### Étape 4 - Envoi du ping :

Maintenant que l'ordinateur A connaît l'adresse MAC de B, il peut envoyer le [ \_\_\_\_\_\_\_\_\_\_\_\_ ].

L'ordinateur A crée un paquet [ \_\_\_\_\_\_\_\_\_\_\_\_ ] (ping) avec l'adresse IP de destination 192.168.1.3 et l'adresse MAC de destination de B.

Le paquet ICMP est envoyé à B via le réseau [ \_\_\_\_\_\_\_\_\_\_\_\_ ].

### Étape 5 - Réponse de B :

L'ordinateur B reçoit le paquet ICMP de A, traite la demande de ping, et renvoie une réponse ICMP à [ \_\_\_\_\_\_\_\_\_\_\_\_ ].

### Résumé, dans vos mots, du scénario

|  |
| --- |
|  |

## Scénario à résoudre

*La Machine A ne répond plus aux requêtes ARP*

**Configuration initiale :**

* Machine A (192.168.1.2) souhaite communiquer avec Machine B (192.168.1.3).
* Machine B connaît l'adresse IP de Machine A, mais ne connaît pas son adresse MAC. La table ARP de Machine B est vide.

**Problème** :

* Machine A cesse de répondre aux requêtes ARP de Machine B.
* La table ARP de Machine B reste vide.

### Diagnostiquer pourquoi Machine A ne répond plus aux requêtes ARP de B.

|  |
| --- |
|  |

### Proposer des solutions pour rétablir la connectivité entre A et B.

|  |
| --- |
|  |

# Simulation ARP

Selon le diagramme suivant :

Une image contenant texte, diagramme, capture d’écran, ligne

Description générée automatiquement

## Capture d’écran de votre simulation fonctionnelle

1. Refaite ce diagramme dans Packet Tracer
2. Rendez la simulation fonctionnelle
3. Prenez une capture d’écran et mettez vos initiales en bas à droite.

|  |
| --- |
|  |

## **Table ARP**

Compléter la table ARP complète de votre réseau simulé

|  |  |
| --- | --- |
| **Adresse IP** | **Adresse MAC** |
| 192.168.1.2 |  |
| 192.168.1.3 |  |
| 192.168.1.4 |  |
| 192.168.1.5 |  |

# Analyse de Traces

Détermination des requêtes ARP nécessaires à une commande PING. Dans la trace Wireshark trafic **arp\_et\_ping.pcapng**, regardez le trafic ICMP (pour les commandes PING).

## Déterminez les requêtes PING effectuées lors de l’enregistrement

(sous la forme PING X : IP Source -> IP Destination, où le X est un chiffre pour numéroter vos ping) *Indice : Il y en a trois* 😉

|  |
| --- |
|  |

## Déterminez les réponses ARP effectuées

(sous la forme ARP X : IP Source -> IP Destination)

|  |
| --- |
|  |

## Associez vos réponses ARP aux bonnes requêtes PING

(sous la forme PING X => ARP X - ARP Y)

|  |
| --- |
|  |

# Programmation Scapy

1. Pour analyser la trace de trafic de façon automatique, faites un petit programme avec Python et Scapy pour lister les réponses ARP. Pour ce faire, vous devrez vérifier :

Pour chaque paquet saisi :

* Si le protocole utilisé est ARP et si c’est une réponse ARP
* Si les deux conditions sont remplies, vous affichez ce paquet

1. Une fois la liste des réponses ARP récupérée, faites une correspondance entre l’adresse IP et l’adresse MAC de chaque paquet. Votre fonction devrait retourner un dictionnaire, avec les adresses IP comme clé et les adresses MAC comme valeur.
2. Finalement, faites une fonction qui confirme votre analyse des commandes PING, c’est-à-dire, une fonction qui liste toutes les commandes ping effectuées dans la trace.
3. **EXPERT ONLY :** Si vous terminez à l’avance, vous pouvez compléter ce programme avec une fonction qui vérifie si une réponse est reçue pour chaque commande effectuée.

* Utilisez la 2e trace fournie (arp\_et\_ping\_capture\_2) pour vérifier votre programme. Un PING dans la capture n’a pas de réponse.

## Copier votre code Python dans l’encadré suivant

|  |
| --- |
|  |