Laboratoire 2

Introduction aux Réseaux

Diagnostic et Analyse avec WSL

Concepts abordés :

Adresses IP • Modèles OSI & TCP/IP
Protocoles ARP, DNS, ICMP
Wireshark • Scripts Bash

Systèmes d'exploitation

Collège de Maisonneuve

DEVOIR NOTÉ

Date de remise : **Jeudi 13 février 2026**
Pondération : **100 points**

Nom :	
Prénom :	
Groupe :	

Barème de notation

Répartition des points – Total : 100 points

Section	Points	Note
Partie 1 : Questions théoriques (20 pts)		
Question 1 : Modèle OSI	5	/5
Question 2 : Adresses IP	5	/5
Question 3 : Protocoles (ARP, DNS, ICMP)	10	/10
Partie 2 : Commandes réseau (25 pts)		
Exercice 1 : Configuration réseau	10	/10
Exercice 2 : Tests de connectivité (ping)	8	/8
Exercice 3 : Table ARP et DNS	7	/7
Partie 3 : Analyse Wireshark (25 pts)		
Exercice 4 : Capture ICMP	10	/10
Exercice 5 : Capture DNS	8	/8
Exercice 6 : Capture ARP	7	/7
Partie 4 : Script Bash (30 pts)		
Fonctionnalité du script	15	/15
Qualité du code (commentaires, lisibilité)	8	/8
Gestion des erreurs	7	/7
TOTAL	**100**	**/100**

Attention

Critères de pénalité :

- Retard : **-10 points par jour** de retard
- Plagiat : **Note de 0** et signalement au département
- Captures d'écran manquantes : **-5 points** par capture manquante
- Script non fonctionnel : **-15 points**

Table des matières

1 Introduction

Objectifs du laboratoire

À la fin de ce laboratoire, vous serez capable de :

- Comprendre les modèles de référence OSI et TCP/IP
- Identifier et analyser des adresses IP
- Utiliser les commandes réseau de base (ping, arp, nslookup)
- Capturer et analyser le trafic réseau avec Wireshark
- Créer un script Bash de diagnostic réseau sous WSL

1.1 Environnement requis

Pour ce laboratoire, vous devez avoir :

1. **WSL (Windows Subsystem for Linux)** – Ubuntu de préférence
2. **Wireshark** – Installé sur Windows
3. **Un éditeur de texte** – nano, vim, ou VS Code
4. **Une connexion Internet** – Pour les tests

2 Rappels théoriques

Information

Lisez attentivement cette section. Les questions théoriques porteront sur ces concepts.

2.1 Le modèle OSI (Open Systems Interconnection)

Le modèle OSI est un modèle de référence qui divise les communications réseau en **7 couches**. Chaque couche a un rôle spécifique.

Les 7 couches du modèle OSI

N°	Couche	Rôle	Exemples
7	Application	Interface utilisateur	HTTP, FTP, DNS
6	Présentation	Format des données	SSL/TLS, JPEG
5	Session	Gestion des connexions	NetBIOS
4	Transport	Fiabilité de transmission	TCP, UDP
3	Réseau	Routage et adressage	IP, ICMP
2	Liaison	Accès au média	Ethernet, Wi-Fi, ARP
1	Physique	Transmission binaire	Câbles, signaux

2.2 Le modèle TCP/IP

Le modèle TCP/IP est le modèle **pratique** utilisé sur Internet. Il comporte **4 couches** :

Modèle TCP/IP vs OSI

TCP/IP	Équivalent OSI	Protocoles
Application	Couches 5, 6, 7	HTTP, DNS, FTP, SMTP
Transport	Couche 4	TCP, UDP
Internet	Couche 3	IP, ICMP
Accès réseau	Couches 1, 2	Ethernet, Wi-Fi, ARP

2.3 Adresses IP

Une adresse IP (Internet Protocol) est un **identifiant unique** attribué à chaque appareil sur un réseau.

2.3.1 IPv4

- Format : 4 octets séparés par des points (ex: `192.168.1.100`)
- Chaque octet va de 0 à 255
- Divisée en partie **réseau** et partie **hôte**

Adresses privées (réservées aux réseaux locaux)

- Classe A : `10.0.0.0` – `10.255.255.255`
- Classe B : `172.16.0.0` – `172.31.255.255`
- Classe C : `192.168.0.0` – `192.168.255.255`

2.4 Protocoles clés

2.4.1 ICMP (Internet Control Message Protocol)

Protocole utilisé pour envoyer des messages de **diagnostic** et d'**erreur**. La commande `ping` utilise ICMP (Echo Request / Echo Reply).

2.4.2 ARP (Address Resolution Protocol)

Protocole qui fait le lien entre les **adresses IP** (couche 3) et les **adresses MAC** (couche 2).

2.4.3 DNS (Domain Name System)

Protocole qui traduit les **noms de domaine** en **adresses IP**.

`www.google.com` ⟶ `142.250.xxx.xxx`

3 Partie 1 : Questions théoriques (20 points)

Question 1 – Modèle OSI (5 points)

a) Complétez le tableau en indiquant le numéro et le nom de la couche OSI correspondante :

Protocole/Élément	N° Couche	Nom de la couche
HTTP		
Adresse IP		
Câble Ethernet		
TCP		
Adresse MAC		

b) Quelle est la principale différence entre le modèle OSI et le modèle TCP/IP ?

Question 2 – Adresses IP (5 points)

a) Pour chaque adresse IP, indiquez si elle est **privée** ou **publique** :

Adresse IP	Privée / Publique
192.168.1.50	
8.8.8.8	
10.0.0.1	
172.20.5.100	
200.100.50.25	

b) Qu'est-ce qu'un masque de sous-réseau ? À quoi sert-il ?

Question 3 – Protocoles (10 points)

a) Expliquez le fonctionnement du protocole ARP. Pourquoi est-il nécessaire ? (3 pts)

b) Quelle est la différence entre une requête DNS de type A et de type AAAA ? (2 pts)

c) Expliquez ce que fait la commande `ping` au niveau du protocole ICMP. Quels types de messages sont échangés ? (3 pts)

d) Sur quel port et avec quel protocole de transport fonctionne DNS par défaut ? Pourquoi ce choix ? (2 pts)

4 Partie 2 : Commandes réseau (25 points)

Attention

Pour cette partie, utilisez le terminal WSL (Ubuntu). Incluez des **captures d'écran** de vos commandes et résultats.

Exercice 1 : Configuration réseau (10 points)

a) Exécutez la commande appropriée pour afficher votre configuration réseau et répondez :

- Quelle commande avez-vous utilisée ?
- Quelle est votre adresse IP ?
- Quel est votre masque de sous-réseau ?
- Quel est le nom de votre interface réseau principale ?

b) Trouvez et notez l'adresse de votre passerelle par défaut (quelle commande utilisez-vous ?) :

c) Affichez les serveurs DNS configurés sur votre système. Quelle commande utilisez-vous et quels serveurs DNS sont configurés ?

Capture d'écran 1 : Incluez une capture montrant votre configuration réseau

Exercice 2 : Tests de connectivité avec ping (8 points)

Effectuez les tests suivants et notez les résultats :
a) Ping vers localhost (127.0.0.1) – 4 paquets :

- Commande exacte utilisée : _____
- Résultat (succès/échec) : _____
- Temps moyen de réponse : _____

b) Ping vers votre passerelle – 4 paquets :

- Résultat (succès/échec) : _____
- Temps moyen de réponse : _____

c) Ping vers 8.8.8.8 (serveur DNS Google) – 4 paquets :

- Résultat (succès/échec) : _____
- Temps moyen de réponse : _____

d) Si le ping vers 8.8.8.8 fonctionne mais pas vers google.com, quel serait le problème probable ?

Capture d'écran 2 : Incluez une capture de vos tests ping

Exercice 3 : Table ARP et résolution DNS (7 points)

a) Affichez votre table ARP :

- Commande utilisée : _____
- Combien d'entrées voyez-vous ? _____
- Notez une entrée (IP et MAC) : _____

b) Effectuez une requête DNS pour `www.collegemaisonneuve.qc.ca` :

- Commande utilisée : _____
- Adresse IP obtenue : _____

c) Utilisez la commande `dig` pour obtenir des informations détaillées sur `github.com`. Quel est le TTL (Time To Live) de l'enregistrement ?

Capture d'écran 3 : Incluez une capture de la table ARP et d'une requête DNS

5 Partie 3 : Analyse Wireshark (25 points)

Attention

Lancez Wireshark **sur Windows**. Sélectionnez l'interface réseau active pour capturer le trafic.

Exercice 4 : Capture et analyse ICMP (10 points)

1. Démarrez une capture dans Wireshark
2. Appliquez le filtre d'affichage : `icmp`
3. Dans WSL, exécutez : `ping -c 4 8.8.8.8`
4. Arrêtez la capture

Analysez un paquet "Echo (ping) request" et complétez :

Information	Valeur observée
Adresse MAC source	
Adresse MAC destination	
Adresse IP source	
Adresse IP destination	
Type ICMP (numéro)	
Code ICMP	

Question : Quelle est la différence entre le Type ICMP d'un "Echo Request" et d'un "Echo Reply" ?

Capture d'écran 4 : Capture Wireshark montrant les paquets ICMP

Exercice 5 : Capture et analyse DNS (8 points)

1. Démarrez une nouvelle capture
2. Appliquez le filtre : `dns`
3. Dans WSL, exécutez : `nslookup www.github.com`
4. Arrêtez la capture

Analysez la requête et la réponse DNS :

Information	Valeur observée
Port source (requête)	
Port destination (requête)	
Protocole de transport	
Type de requête DNS	
Adresse IP dans la réponse	

Capture d'écran 5 : Capture Wireshark montrant la requête et réponse DNS

Exercice 6 : Capture et analyse ARP (7 points)

1. Démarrez une capture avec le filtre : `arp`
2. Dans WSL, videz la table ARP : `sudo ip neigh flush all`
3. Pingez votre passerelle : `ping -c 1 <adresse_passerelle>`
4. Observez les paquets ARP

Complétez le tableau pour un échange ARP observé :

Information	ARP Request	ARP Reply
Adresse MAC source		
Adresse MAC destination		
Adresse IP recherchée		

Question : Pourquoi l'adresse MAC de destination dans l'ARP Request est-elle `ff:ff:ff:ff:ff:ff` ?

Capture d'écran 6 : Capture Wireshark montrant l'échange ARP

6 Partie 4 : Script de diagnostic réseau (30 points)

Exercice 7 : Création d'un script Bash de diagnostic

Vous devez créer un script Bash nommé `diagnostic_reseau.sh` qui effectue un diagnostic complet du réseau.

Exigences fonctionnelles (15 points)

Votre script doit obligatoirement :

1. **Afficher les informations système** (3 pts)
   - Nom de l'hôte
   - Date et heure actuelles
   - Version du système
2. **Afficher la configuration réseau** (4 pts)
   - Adresse IP locale
   - Adresse de la passerelle par défaut
   - Serveurs DNS configurés
3. **Effectuer des tests de connectivité** (5 pts)
   - Test de localhost (127.0.0.1)
   - Test de la passerelle
   - Test d'Internet (8.8.8.8)
   - Test de résolution DNS (google.com)
   - Afficher clairement si chaque test réussit ou échoue
4. **Afficher la table ARP** (1 pt)
5. **Effectuer des résolutions DNS** (2 pts)
   - Résoudre au moins 2 domaines différents
   - Afficher les adresses IP obtenues

Critères de qualité (15 points)

- **Commentaires** (4 pts) : En-tête avec votre nom et description, commentaires explicatifs dans le code
- **Lisibilité** (4 pts) : Code bien structuré, indentation correcte, noms de variables explicites
- **Affichage clair** (4 pts) : Titres de sections, messages compréhensibles pour l'utilisateur
- **Gestion des erreurs** (3 pts) : Le script doit gérer les cas d'échec (ex: si un ping ne répond pas, afficher un message d'erreur approprié au lieu de planter)

Information

Conseils :

- Utilisez des conditions `if` pour vérifier le succès des commandes
- L'option `-c` de ping limite le nombre de paquets
- L'option `-W` de ping définit un timeout (utile pour éviter d'attendre trop longtemps)
- Testez votre script plusieurs fois avant de le remettre
- Rendez votre script exécutable avec `chmod +x`

7 Consignes de remise

À remettre

Format : Remettre sur Teams le reponse.md

Contenu du dépôt GitHub :

1. **Le document complété** (reponse.md) avec toutes les réponses aux questions.
2. **Votre script** `diagnostic_reseau.sh`
3. **Dossier "captures"** contenant :
   - Capture 1 : Configuration réseau
   - Capture 2 : Tests ping
   - Capture 3 : Table ARP et DNS
   - Capture 4 : Wireshark ICMP
   - Capture 5 : Wireshark DNS
   - Capture 6 : Wireshark ARP
   - Capture 7 : Exécution de votre script

Date limite : Jeudi 13 février 2026, 23h59

Méthode de remise : Via la plateforme du cours (Teams)

Attention

Rappels importants :

- Les captures d'écran doivent être **lisibles**
- Le script doit être **exécutable** et **fonctionnel**
- Tout plagiat sera sanctionné par la note de **0**
- En cas de problème technique, contactez votre enseignant **avant** la date limite

Annexe : Aide-mémoire des commandes

Commande	Description
`ip addr show`	Affiche la configuration IP de toutes les interfaces
`ip route`	Affiche la table de routage
`ping -c N host`	Envoie N paquets ICMP vers host
`ping -W T host`	Définit un timeout de T secondes
`arp -a`	Affiche la table ARP
`ip neigh show`	Affiche la table ARP (commande moderne)
`nslookup domain`	Effectue une requête DNS
`dig domain`	Effectue une requête DNS détaillée
`dig +short domain`	Affiche uniquement l'IP résolue
`cat /etc/resolv.conf`	Affiche les serveurs DNS configurés
`hostname`	Affiche le nom de l'hôte
`hostname -I`	Affiche l'adresse IP de la machine

Filtre Wireshark	Description
`icmp`	Affiche uniquement le trafic ICMP
`dns`	Affiche uniquement le trafic DNS
`arp`	Affiche uniquement le trafic ARP
`ip.addr == X.X.X.X`	Filtre par adresse IP
`tcp.port == 80`	Filtre par port TCP

Structure Bash	Exemple
Shebang	`#!/bin/bash`
Variable	`MA_VAR="valeur"`
Affichage	`echo "Message"`
Condition	`if [ condition ]; then ... fi`
Commande dans variable	`IP=$(hostname -I)`
Tester succès commande	`if ping -c 1 host > /dev/null; then ...`
Rendre exécutable	`chmod +x script.sh`
Exécuter	`./script.sh` ou `bash script.sh`