

# Rapport de Synthèse – Réseaux et Systèmes Distribués

Mohamed Reda Hamdi / Abdelmoiz Bensbai

## Table des matières

<b>1</b>	<b>Résumé du Semestre (Travaux Pratiques)</b>	<b>2</b>
1.1	Routage Statique et Dynamique . . . . .	2
1.2	Services Réseau sous Linux . . . . .	2
1.3	Projet Réseau Universitaire . . . . .	2
<b>2</b>	<b>Systèmes Centralisés vs Distribués</b>	<b>3</b>
<b>3</b>	<b>Caractéristiques des Systèmes Distribués</b>	<b>3</b>
<b>4</b>	<b>Middleware et RPC</b>	<b>3</b>
<b>5</b>	<b>Sérialisation et Désérialisation</b>	<b>3</b>
<b>6</b>	<b>Docker et Conteneurisation</b>	<b>4</b>
<b>7</b>	<b>Architecture Réseau – Projet Clusters</b>	<b>4</b>
<b>8</b>	<b>Conclusion</b>	<b>5</b>

# 1 Résumé du Semestre (Travaux Pratiques)

## 1.1 Routage Statique et Dynamique

- **Routage statique** : routes configurées manuellement, adapté aux petits réseaux.
- **RIP (Routing Information Protocol)** : mise à jour périodique par distance, facile à configurer.
- **OSPF (Open Shortest Path First)** : routage par état de liens, adapté aux grands réseaux.

## 1.2 Services Réseau sous Linux

- **DHCP Server / Relay** : configuration de l'attribution dynamique d'IP.
- **DNS (BIND)** : configuration des zones directes et inverses, tests avec dig/nslookup.
- **Serveur Web (Apache)** : configuration de site virtuel, accès à distance.

## 1.3 Projet Réseau Universitaire

- **Architecture multi-bâtiment** avec VLANs : profs, étudiants, visiteurs.
- **Sécurité réseau** : ACL, sécurité de ports.
- **OSPF** pour l'interconnexion.

## 2 Systèmes Centralisés vs Distribués

- **Centralisé** : système unique, une seule unité centrale.
- **Distribué** : plusieurs composants répartis, accès local/distant, plus fiable et évolutif.

## 3 Caractéristiques des Systèmes Distribués

- **Transparence** : l'utilisateur ignore l'emplacement réel des données.
- **Ouverture** : basé sur des standards d'interopérabilité.
- **Tolérance aux pannes** : le système continue en cas de défaillance.
- **Concurrence** : accès simultané aux ressources.
- **Migration/Translation** : déplacement transparent des ressources.

## 4 Middleware et RPC

### Middleware

- Masque l'hétérogénéité des composants.
- Types : messagerie, base de données, serveur d'application.

### Architecture Client-Serveur et RPC

- Client demande, serveur répond.
- Communication via **messages** : requête → réponse.
- **RPC** : appel distant de procédure comme s'il était local.

### Fonctionnement de RPC

1. Déclaration des méthodes et paramètres.
2. Sérialisation des paramètres (transfert).
3. Stub client et squelette serveur.
4. Exécution distante et retour de résultat.

### Limitations de RPC

- Communication synchrone.
- Paramètres passés par valeur.
- Localisation fixe du serveur.

## 5 Sérialisation et Désérialisation

- **Sérialisation** : conversion d'un objet pour transmission/réseau.
- **Désérialisation** : reconstitution d'objet depuis un flux.

## 6 Docker et Conteneurisation

### Définition

Docker permet de créer, déployer et gérer des applications dans des conteneurs légers et portables.

### Avantages

- Isolation des applications.
- Portabilité sur tous les environnements.
- Optimisation des ressources.
- Intégration dans les pipelines CI/CD.

### Composants Docker

- Docker Engine, Docker CLI
- Images, conteneurs, Docker Hub
- Docker Compose

### Commandes Docker Utiles

- `docker ps`, `docker run <image>`, `docker stop <id>`
- `docker build -t mon_image .`, `docker-compose up`
- `docker exec -it <id> bash`

## 7 Architecture Réseau – Projet Clusters

### Topologie

- Routeur principal + switch central
- 3 Clusters indépendants (chaque cluster avec 3 machines)

### Adresses IP des machines

- Cluster 0 : 192.168.1.1 à 1.3
- Cluster 1 : 192.168.2.1 à 2.3
- Cluster 2 : 192.168.3.1 à 3.3

### Table de Routage Exemple

Destination	Masque	Passerelle	Interface
192.168.1.0	255.255.255.0	-	Fa0/1
192.168.2.0	255.255.255.0	192.168.1.1	Fa0/2
192.168.3.0	255.255.255.0	192.168.1.1	Fa0/3

## 8 Conclusion

Ce rapport présente l'ensemble des connaissances acquises : du routage au déploiement d'applications conteneurisées, en passant par les systèmes distribués, l'architecture client-serveur, et la virtualisation. Il fournit une base solide pour des projets professionnels dans le domaine des systèmes informatiques modernes.