

Projet IN403

Compte Rendu du Projet : "Algorithmique de Graphes – Tables de Routage"

Ce projet a été réalisé par Ayoub Ourimi, Mohamed Knis et Bilal Ourif, étudiants respectivement des groupes de TD 3 et TD 2. Nous avons choisi d'utiliser Python pour le développement en raison de ses robustes bibliothèques de manipulation de données et de visualisation graphique.

1. Introduction

Dans un monde numérique où l'efficacité des réseaux est cruciale, comprendre et optimiser les tables de routage est fondamental. Ce projet de simulation de réseau de 100 nœuds vise à établir et visualiser les tables de routage, permettant de tracer les chemins les plus courts pour la transmission des données. En utilisant des algorithmes de graphes et une interface utilisateur interactive, ce projet démontre les applications pratiques des principes théoriques, essentiels pour améliorer la gestion des réseaux informatiques.

2. Cadre du Projet

Le projet s'appuie sur des concepts fondamentaux des réseaux informatiques tels que :

- **Backbone (Tier 1)** : Cœur du réseau, comprenant des nœuds hautement connectés essentiels pour le transfert de données à grande échelle.
- **Opérateurs de transit (Tier 2)** : Nœuds facilitant le transfert de données entre le backbone et les utilisateurs finaux, jouant un rôle de médiateurs.
- **Opérateurs de niveau 3 (Tier 3)** : Situés en périphérie, ces nœuds distribuent les données localement aux utilisateurs finaux.

3. Création du Réseau

Le réseau de 100 nœuds a été généré aléatoirement en utilisant la bibliothèque NetworkX, en respectant des probabilités spécifiques :

- **Tier 1** : 10 nœuds, 75% de chance de connexion, temps de communication de 5 à 10 unités.
- **Tier 2** : 20 nœuds connectés aléatoirement à 1-2 nœuds de Tier 1 et 2-3 nœuds de Tier 2, avec des temps de 10 à 20 unités.

- **Tier 3** : 70 nœuds, chacun connecté à 2 nœuds de Tier 2, avec des temps de 20 à 50 unités.

4. Vérification de la Connexité

Pour assurer que le réseau soit entièrement connexe, nous avons implémenté une procédure de vérification utilisant un parcours en profondeur (Depth-First Search, DFS). Cette méthode récursive explore tous les sommets d'un graphe à partir d'un nœud de départ et marque chaque nœud visité, garantissant que tous les nœuds sont accessibles depuis n'importe quel autre nœud du réseau. Si un graphe généré n'est pas connexe, le processus de génération est relancé jusqu'à obtenir une structure entièrement accessible.

5. Détermination des Tables de Routage

L'algorithme de Dijkstra a été utilisé pour calculer le chemin le plus court depuis chaque nœud vers tous les autres nœuds, formant ainsi les bases des tables de routage nécessaires à la navigation efficace des paquets dans le réseau.

6. Visualisation des Chemins

Notre interface utilisateur, développée avec Tkinter, permet une interaction visuelle pour la sélection des nœuds et la visualisation des chemins de routage grâce aux tables établies, illustrant clairement le parcours des messages à travers le réseau.

Conclusion

En conclusion, ce projet a mis en évidence l'importance cruciale des algorithmes de graphes dans la gestion efficace des réseaux informatiques. Par l'élaboration de tables de routage pour un réseau de 100 nœuds, nous avons appliqué des concepts théoriques pour optimiser les chemins de transmission des données, garantissant ainsi rapidité et efficacité.

L'interface utilisateur développée facilite non seulement la visualisation de ces processus complexes, mais rend également les stratégies de routage plus accessibles et compréhensibles pour les utilisateurs. Ce projet confirme le rôle essentiel de l'algorithmique des graphes dans le domaine des technologies de réseau et jette les bases pour de futures avancées dans ce secteur.