Université de Versailles - Saint Quentin
"Algorithmique de Graphes"
Le projet : *Tables de routage*

1 Cadre

Ce projet est à réaliser en groupes de 3 étudiants voire exceptionnellement en binôme.

A la fin du projet (début mai), il conviendra :

— de faire une démonstration (d'une quinzaine de minutes environ) de votre programme devant l'enseignant de votre groupe de TD (si votre groupe de projet est composé d'étudiants de différents groupes de TD, vous pouvez choisir à quel enseignant vous avez le plus envie de faire cette magnifique demonstration).

— de lui rendre (ou de lui envoyer par mail) un rapport **au format pdf** décrivant le travail effectué, les structures de données choisies, les principales procédures, ... Le code du programme sera impérativement mis en annexe de ce rapport et il ne doit donc y avoir **qu'un seul fichier** envoyé. Il convient dans ce rapport de bien expliquer ce que vous avez fait, l'enseignant doit comprendre votre travail sans avoir à se plonger dans le détail de vos lignes de code.

2 Travail à réaliser

Le but du projet est de réaliser une application qui permette principalement d'établir la table de routage de chaque noeud d'un réseau de 100 noeuds. Le programme pourra être réalisé dans le langage de votre choix (C, C++, Python, Java, ...). Plus précisément, le travail attendu peut être décomposé en 5 tâches principales.

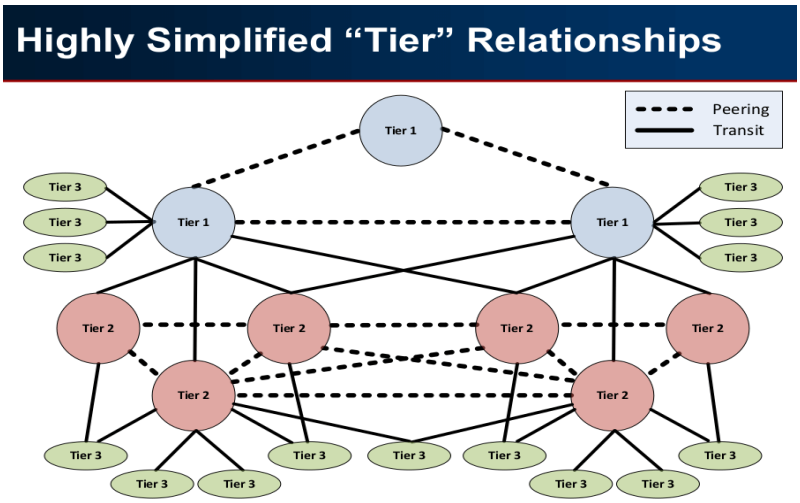
2.1 Description du cadre du projet

Il s'agit ici de produire un petit texte de 10 - 15 lignes décrivant et expliquant les principaux termes techniques utilisés dans ce projet (transit IP, backbone, niveau des opérateurs (Tier), peering, ...). Vous êtes vivement encouragés à utiliser pour ce faire un agent conversationnel (ChatGPT ou un autre chatbot). Si le texte produit pouvait être un peu humoristique, ce serait un plus.

2.2 La création aléatoire d'un réseau réaliste

Il convient ici de créer la topologie d'interconnexion d'un réseau de 100 noeuds de la forme indiquée sur le schéma figurant sur la page suivante. Les liens et les temps de communication sur chacun de ces liens vont être créés de manière aléatoire selon les règles suivantes :

— le *backbone* ou coeur du réseau (Tier1 sur le schéma) composé de 10 noeuds très connectés entre eux par des lignes de communication à très haut débit. On va considérer que pour chaque lien possible entre ces noeuds (non orienté), il y a 75% de chance qu'il existe. Par ailleurs, si le lien existe, il sera valué par une valeur tirée aléatoirement entre 5 et 10 (représentant le temps unitaire moyen de communication sur cette ligne).



— les opérateurs de niveau 2 (Tier 2) souvent appelés opérateurs de transit. 20 noeuds qui seront chacun connectés à 1 ou 2 noeuds du backbone tiré(s) aléatoirement et à 2 ou 3 noeuds de niveau 2 (tirés également aléatoirement). Chacun de ces liens sera valué par une valeur comprise entre 10 et 20.

— les opérateurs de niveau 3 (Tier3). 70 noeuds reliés chacun à 2 noeuds de niveau 2. Les liens seront valués par une valeur comprise entre 20 et 50.

Il faudra également dans cette partie déterminer la structure de données qui vous parait la plus adéquate pour représenter et mémoriser le graphe correspondant à ce réseau.

2.3 La vérification de la connexité du réseau

Même s'il est très probable que le réseau créé soit connexe, le tirage aléatoire des liens peut faire qu'exceptionnellement il ne le soit pas. Le but de cette partie est de développer une procédure pour vérifier que tous les sommets peuvent bien être atteints à partir d'un sommet de départ quelconque. Si ce n'est pas le cas, il convient de relancer la création d'un nouveau réseau.

2.4 La détermination de la table de routage de chaque noeud

La table de routage d'un noeud doit indiquer pour chaque destination possible (les 99 autres noeuds) à quel voisin il convient de router un message (un paquet) compte tenu de la destination finale. C'est donc le prochain noeud sur un plus court chemin vers cette destination car le message doit suivre le cheminement le plus court en temps de communication (les temps de routage en chaque noeud intermédiaire sont considérés comme negligeables). Il convient donc dans cette partie de développer l'algorithme qui calcule ces 100 tables de routage.

2.5 La reconstitution du chemin entre 2 noeuds

Il faut dans cette partie permettre à l'utilisateur de saisir 2 noeuds : un noeud émetteur de message et un noeud destinataire et à l'aide des tables de routage (il ne faut pas refaire ici le calcul de plus court chemin

mais juste utiliser les tables de routage établies dans la partie précédente) reconstituer le chemin que va suivre ce message. Ce chemin devra être indiqué (affiché) à l'utilisateur.

2.6 Une belle interface

C'est une partie optionnelle qui peut remplacer la tâche précédente.

Le réseau s'affiche à l'écran et l'utilisateur clique directement sur le noeud émetteur et le noeud destinataire. Le chemin que va suivre le message s'affiche directement sur l'écran (toujours calculé à partir des tables de routage).