

Explication du Diagramme de Flux

1. **Début** : Le programme commence par la construction de l'arbre de routage.
2. **Construire l'arbre de routage** : L'arbre est construit à partir des sous-réseaux et des routeurs définis dans `TreeBuilder.py`.
3. **Afficher l'arbre initial** : L'arbre est affiché dans le terminal grâce à `TreeVisualizer.py`.
4. **Gérer les réseaux injoignables** : L'utilisateur est invité à entrer un réseau injoignable. Si un réseau est marqué comme injoignable, il est supprimé de l'arbre.
5. **Rechercher une route dans l'arbre** : L'utilisateur entre une adresse IP à rechercher. Le programme vérifie si le résultat est déjà dans le cache.
6. **Utiliser le cache** : Si le résultat est dans le cache, il est chargé directement. Sinon, le programme utilise les algorithmes de recherche pour trouver la route.
7. **Comparer les méthodes de recherche** : Les algorithmes de recherche (préfixe, infixe, suffixe) sont comparés pour trouver la méthode la plus rapide. En parallèle, la méthode BFS est utilisée pour trouver le chemin commençant par la racine vers le réseau recherché.
8. **Trouver la méthode la plus rapide** : La méthode la plus rapide est sélectionnée, et le résultat est sauvegardé dans le cache ainsi que le chemin depuis la racine.
9. **Afficher le chemin trouvé** : Si le chemin est trouvé, la route identifiée par la méthode la plus efficace sera affichée à l'utilisateur, ainsi que le chemin parcouru et le chemin depuis la racine.
10. Liste
11. **Envoyer un paquet** : L'utilisateur est invité à envoyer un paquet sur le réseau. Si oui, le paquet est simulé avec une latence aléatoire.
12. **Simuler l'envoi du paquet** : Le programme simule l'envoi du paquet en affichant chaque saut et la latence totale.
13. **Fin** : Le programme se termine après l'envoi du paquet ou si l'utilisateur choisit de ne pas envoyer de paquet.

Fonctionnalités Couvertes par le Diagramme

- **Construction de l'arbre** : Gérée par `TreeBuilder.py`
- **Gestion des réseaux injoignables** : Gérée par `RouteManager.py`
- **Recherche de routes** : Gérée par `SearchAlgorithms.py` et `CacheManager.py`
- **Envoi de paquets** : Géré par `PacketSender.py`
- **Visualisation de l'arbre** : Gérée par `TreeVisualizer.py`

Ce diagramme de flux résume le déroulement logique du programme et montre comment les différents modules interagissent pour offrir les fonctionnalités principales.