



Projet de réseaux informatique

Modélisation d'un protocole de
routage à vecteurs de distances



Faculté
des Sciences

UMONS
Université de Mons



Contrat

1 Objectif

L'objectif de ce projet est d'approfondir votre compréhension du fonctionnement des protocoles de routage, en particulier ceux basés sur des vecteurs de distances. Pour atteindre cet objectif, vous implémenterez un simulateur capable de modéliser une topologie quelconque et de simuler l'apprentissage des routes via les échanges de messages entre routeurs.

2 Implémentation

Vous devez fournir une implémentation, sous forme d'un programme informatique écrit en **Python**, réalisant l'objectif décrit dans la section précédente. Vous utiliserez la notion de classe pour modéliser les différents composants du réseau (routeurs, liens, paquets, ...).

Simulateur

Votre simulateur doit prendre en charge ou implémenter les fonctionnalités suivantes :

- Il doit permettre de représenter une topologie quelconque de routeurs interconnectés par des liens, chaque lien étant caractérisé par une longueur, une vitesse de transmission et une vitesse de propagation.
- Chaque routeur possède un identifiant (entier > 0).
- L'échange de paquets entre deux routeurs voisins doit être simulé en prenant en compte les délais liés aux caractéristiques des liens (distance, vitesse de propagation, débit de transmission) et à la taille des paquets.
- Chaque lien est associé à un coût administratif configurable¹. Ces coûts seront utilisés par votre protocole de routage pour déterminer les routes optimales.
- Les vecteurs de distances doivent être transportés dans des paquets, soumis aux délais liés aux liens et à la taille des paquets. Ces vecteurs étant échangés uniquement entre routeurs voisins, il n'est pas nécessaire de simuler le *forwarding* des paquets. La taille d'un paquet doit être proportionnelle à la longueur du vecteur de distances qu'il transporte.
- Chaque routeur doit conserver :
 - Les destinations, *i.e.* les identifiants, des routeurs connus² ;
 - Le coût des routes associées ;
 - Le voisin utilisé pour atteindre chaque destination ; et
 - Toutes autres informations nécessaires au bon fonctionnement du protocole de routage.

¹Le coût des liens est symétrique, c'est-à-dire qu'il est identique quelle que soit la direction considérée.

²Les routeurs dont l'existence a été apprise par votre protocole de routage.

- Le simulateur doit permettre de modifier le coût administratif et l'état³ d'un lien à tout moment pendant la simulation.
- Les routeurs doivent mettre à jour leurs informations en fonction des messages reçus de leurs voisins ou des modifications des coûts des liens. Ils doivent notifier leurs voisins des changements lorsque nécessaire.
- À la fin de la simulation, chaque routeur doit disposer des routes les plus courtes pour atteindre tous les autres routeurs du réseau.
- Votre simulateur doit générer des logs permettant de suivre l'apprentissage des routes par les routeurs. Ces logs doivent inclure les vecteurs de distances reçus et envoyés par chaque routeur. De plus, si une route est modifiée, la raison de cette modification doit apparaître dans les logs.

Affichage & Simulation

Un des objectifs du projet est d'utiliser votre implémentation pour simuler divers scénarios décrits dans la section **Simulations**. Pour chaque scénario, vous créerez un fichier JSON modélisant la topologie et paramétrant les éventuels changements de coûts.

Pour faciliter la correction, le format JSON ci-dessous est imposé. L'attribut `links` contient la liste des liens de la topologie, définis par leur vitesse de propagation (en m/s), leur vitesse de transmission (en bps), leur longueur (en m), leur coût, ainsi que les identifiants des routeurs qu'ils relient⁴. L'attribut `events` regroupe les événements programmés. Dans le cadre de ce projet, seul le type `cost_change` est pris en charge : il permet de modifier le coût d'un lien à un instant donné, spécifié par l'attribut `time` (moment de déclenchement, en millisecondes depuis le début de la simulation). **Le respect strict du format et des unités spécifiés est obligatoire, sous peine de pénalisation.**

```
{
  "links": [
    {
      "propagation_speed": int, # en m/s
      "transmission_speed": int, # en bps
      "distance": int, # en m
      "cost": int,
      "endpoints": [int, int] # identifiants des routeurs
    },
    ...
  ],
  "events": [
    {
      "type": "cost_change",
      "time": int, # déclenchement de l'événement en ms
      "link": [int, int], # identifiants des routeurs
      "new_cost": int
    },
    ...
  ]
}
```

³Un lien peut cesser de fonctionner, auquel cas son coût administratif devient $+\infty$.

⁴Les routeurs ne sont donc pas explicitement déclarés.

Votre implémentation doit être capable de lire ces fichiers JSON, modéliser la topologie correspondante, programmer les événements et simuler l'apprentissage des routes. Lors de l'exécution, les logs doivent permettre de suivre la simulation.

À l'issue de la simulation⁵, votre implémentation affichera dans la console, pour chaque routeur, une liste des routeurs distants connus, le coût associé à chaque route et le routeur voisin permettant d'atteindre chaque destination, sous le format texte suivant :

```
-- Router 6 :
  dest    cost    next-hop
    3      11         1
    9      35         1
    7       3         4
    ...
```

3 Simulations

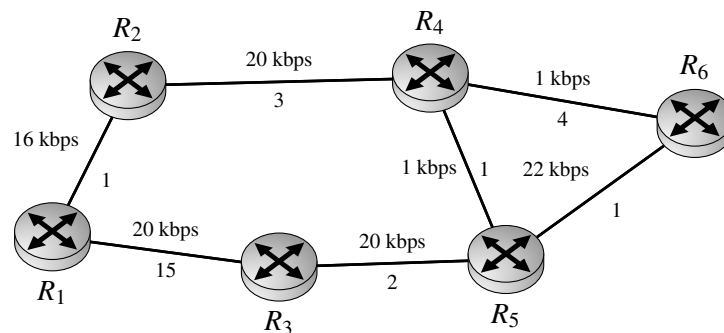
Après avoir implémenté votre simulateur, utilisez-le pour simuler les scénarios décrits ci-dessous. Sauf indication contraire, vous êtes libre de définir la topologie, les coûts administratifs et les caractéristiques des liens. Dans votre rapport, présentez une illustration de la topologie choisie et les résultats obtenus pour chaque scénario.

Validation de votre implémentation

Concevez une topologie composée de 5 routeurs interconnectés par 7 liens. Simulez votre protocole de routage avec cette topologie. Dans votre rapport, décrivez la topologie choisie à l'aide d'une illustration. Présentez ensuite les routes optimales obtenues par votre simulateur, en les comparant aux routes calculées manuellement⁶. Expliquez également, de manière concise, comment déterminer la route empruntée par un paquet entre deux routeurs en utilisant uniquement les tables de routage générées par votre simulateur.

Impact des délais

Considérez la topologie ci-dessous, où le débit et le coût administratif des liens sont indiqués⁷.



⁵Dans le cas où le protocole de routage se stabilise, et donc où la simulation se termine.

⁶Vous n'êtes pas tenu de justifier dans votre rapport pourquoi vos routes sont optimales.

⁷La longueur et la vitesse de propagation des liens sont identiques pour tous les liens.

Utilisez votre simulateur pour modéliser cette topologie et simuler l'apprentissage des routes. Dans votre rapport, concentrez-vous uniquement sur les routes menant au routeur A. Décrivez en détail, à l'aide des logs, comment les routes annonçant le routeur A se propagent dans la topologie. Déterminez et justifiez, dans votre rapport, si les délais influencent la propagation des routes et s'ils influencent les routes finales calculées.

Comptage à l'infini

Utilisez votre simulateur pour reproduire une situation où un comptage à l'infini se produit. Dans votre rapport, décrivez les causes et les conséquences de ce phénomène. Appuyez vos explications à l'aide d'une illustration de la topologie et des logs générés par le simulateur, en identifiant les logs qui mettent en évidence le comptage à l'infini.

Solution du problème de comptage à l'infini

Proposez une solution, différente du *poison reverse*, pour résoudre le problème de comptage à l'infini. Pour cela, documentez-vous sur les solutions existantes et sélectionnez-en une que vous présenterez dans votre rapport⁸.

Dans votre rapport, expliquez le principe de fonctionnement de la solution choisie ainsi que les raisons pour lesquelles elle permet d'éviter le comptage à l'infini. Illustrez à l'aide d'un exemple de comptage à l'infini, et montrez comment ce problème est résolu grâce à la solution proposée. Si la solution ne provient pas de vous, veuillez à citer vos sources, en privilégiant les articles scientifiques.

⁸Vous ne devez pas implémenter la solution dans votre simulateur.

Délivrables

Ce projet est à réaliser par **groupe de deux étudiants**. Le projet doit être soumis sur la plateforme Moodle pour le **dimanche 1 juin 2025 à 23h59**. **Au plus tard** à cette date, vous devez avoir rendu les livrables du projet : le rapport ainsi que le code source de votre implémentation. Une fois la date limite passée, plus aucun projet ne sera accepté. Veuillez suivre scrupuleusement les consignes indiquées ci-dessous.

Le rapport doit être **exclusivement** fourni au **format PDF** et ne doit pas faire plus de **10 pages**. Il doit mentionner clairement les noms, prénoms et matricules de chaque membre du groupe ainsi que le numéro du groupe. Votre rapport devra inclure une brève section intitulée *Exécution*, détaillant la structure de votre projet et les fichiers à exécuter pour reproduire les scénarios décrits dans la section **Simulations** de l'énoncé. Une autre partie du rapport sera consacrée à l'*Implémentation*, où vous aborderez brièvement les éventuelles difficultés rencontrées, ainsi que l'état final de l'implémentation. Enfin, une section *Simulations* apportera les réponses aux questions posées dans la section correspondante de l'énoncé.

Le rapport et les sources (.py) de votre implémentation devront être fournis **dans une archive**. L'archive devra être nommée "projet-reseaux-2025-grX" où X est remplacé par le numéro de votre groupe. Votre archive sera fournie au format "zip" ou "tar.gz" et doit respecter l'arborescence de répertoires suivante :

```
projet-reseaux-2025-grX/  
  rapport.pdf  
  src/  
    ensemble des fichiers sources (.py)
```

Tout non-respect des consignes précédentes implique un projet **non recevable et donc une **note nulle**.**