

Création automatisée d'un schéma de réseau

Travail de Bachelor réalisé en vue de l'obtention du Bachelor HES

par :

Coralie CHEVALLEY

Conseiller au travail de Bachelor :

André SEYDOUX, maître d'enseignement HES

Carouge, le lundi 22 août 2022

Haute École de Gestion de Genève (HEG-GE)

Filière Informatique de gestion

Déclaration

Ce travail de Bachelor est réalisé dans le cadre de l'examen final de la Haute école de gestion de Genève, en vue de l'obtention du titre Bachelor of Science HES-SO Informatique de gestion.

L'étudiant a envoyé ce document par email à l'adresse remise par son conseiller au travail de Bachelor pour analyse par le logiciel de détection de plagiat URKUND, selon la procédure détaillée à l'URL suivante : <https://www.orkund.com>.

L'étudiant accepte, le cas échéant, la clause de confidentialité. L'utilisation des conclusions et recommandations formulées dans le travail de Bachelor, sans préjuger de leur valeur, n'engage ni la responsabilité de l'auteur, ni celle du conseiller au travail de Bachelor, du juré et de la HEG.

« J'atteste avoir réalisé seule le présent travail, sans avoir utilisé des sources autres que celles citées dans la bibliographie. »

Fait à Carouge, le 22 août 2022

Coralie CHEVALLEY



Remerciements

Je souhaite remercier monsieur André SEYDOUX pour avoir été investi par le suivi de mon mémoire, de l'aide et ses conseils apportés.

Résumé

Ce travail est une application qui permettra aux enseignant·e·s du cours de réseau de la filière Informatique de Gestion, ainsi qu'à ses élèves de créer des schémas / diagrammes de réseau selon les normes de Cisco et de la Haute École de Gestion de Genève.

Table des matières

Déclaration.....	i
Remerciements	ii
Résumé	iii
Liste des tableaux	vi
Liste des figures.....	vi
1. Introduction.....	1
2. Étude des logiciels existants.....	2
2.1 Présentation de Diagrams.net (drawio)	2
3. Analyse de la viabilité de l'application	3
3.1 Pseudo code via le logiciel Diagrams.net	3
3.2 Module Python N2G	4
4. Langage de programmation	4
5. Librairies	4
5.1 Librairies pour l'interface graphique	4
5.1.1 Présentation des librairies tkinter	5
5.1.2 Présentation de la librairie customTkinter.....	5
5.2 Librairies pour l'algorithme	6
6. Désirs pour l'application.....	7
6.1 Compatibilité	7
6.2 Langues	7
6.3 Couleurs	7
6.4 Style de l'application.....	7
6.5 Documentation de l'application	8
7. Fonctionnement de l'application.....	8
8. Interface graphique de l'application	9
8.1 Fenêtre principale	9
8.2 Fenêtre d'ajout de lien	11
8.3 Fenêtre de modification de lien.....	12
9. Développement de l'algorithme	13
9.1 Contraintes de validation	13
9.1.1 Classe de l'adresse réseau	13
9.1.2 Adresse 10.136.0.0 détenant un routeur	14
9.1.3 Nombre de machines pour le réseau.....	14
9.1.4 Type d'EIGRP	14
9.1.5 Adresse se trouvant dans une plage d'une autre adresse de réseau	15

9.1.6	Adresse privée et rassemblement des contraintes du chapitre 9.1.1 au chapitre 9.1.5	16
9.1.7	Saisie de l'adresse de réseau	16
9.1.8	Adresse de réseau est le même que la machine liée	17
9.1.9	Deux PC liés	17
9.1.10	Liaison d'interface sérial	17
9.1.11	Machine détient le bon type et nom que ces prédécesseurs.....	17
9.1.12	Routeur détenant l'adresse de réseau	18
9.1.13	Interface déjà utilisée.....	18
9.1.14	Liaison de machine avec elle-même.....	19
9.2	Méthodes de recherches	19
9.2.1	Recherche du type de machine selon le nom de la machine.....	19
9.2.2	Recherche des adresses de réseau selon le nom de la machine	19
9.3	Exécution des valeurs de la liaison	20
9.4	Modification des valeurs de la liaison	20
9.5	Ajout des valeurs de la liaison	21
9.6	Suppression des valeurs de la liaison.....	21
9.7	Importation de fichier avec extension .csv	22
9.8	Exécution du diagramme (exportation en fichier drawio et csv)	23
9.8.1	Attribution des adresses IPv4 aux machines	23
9.8.2	Exportation du fichier drawio	23
9.8.3	Exportation du fichier avec extension .csv.....	24
9.8.4	Exportation du fichier en image	24
10.	Création de la documentation	24
11.	Création du manuel d'utilisation	25
12.	Développement de l'application .py en .exe.....	26
13.	Améliorations potentielles.....	27
13.1.1	Contrainte de validation pour les liaisons bidirectionnelles	27
13.1.2	Contrainte de validation pour l'importation de fichier csv	27
13.1.3	Importation du manuel d'utilisation dans l'application	27
13.1.4	Ajout des bons liens	28
13.1.5	Ajout de nouvelles machines	28
13.1.6	Ajout des protocoles	28
13.1.7	Ajout d'une prévisualisation du diagramme.....	30
13.1.8	Exportation du diagramme en image	30
14.	Conclusion	31
Annexe 1 :	Manuel d'utilisation.....	37

Liste des tableaux

Tableau 1 : Tableau des librairies utilisées	6
Tableau 2 : En-tête du fichier csv	22

Liste des figures

Figure 1 : Exemple de sortie du pseudo code.....	3
Figure 2 : Illustration chronologique des librairies pour l'interface graphique	5
Figure 3 : Logo Cisco	7
Figure 4 : Fenêtre principale au démarrage de l'application	9
Figure 5 : Fenêtre principale après ajout de lien.....	10
Figure 6 : Fenêtre pour l'ajout d'un lien	11
Figure 7 : Fenêtre pour la modification d'un lien	12
Figure 8 : Exemple de schéma avec des protocoles.....	29

1. Introduction

Au sein de la Haute École de Gestion de Genève (HEG), dans la filière Informatique de gestion (IG), des cours de réseau sont donnés et ont pour objectif clé de concevoir un ou des réseaux d'entreprise. Pour mener à bien ce but, les élèves devront effectuer un plan d'adressage adéquat et modelé, selon les normes établies par la HEG et de Cisco.

Présentement, les schémas de réseau sont effectués manuellement et prennent un temps considérable tant pour les professeur·e-s, que pour les élèves. Le but étant d'automatiser ce processus. Pour ce faire, le travail de Bachelor (TB) sera dédié à la création d'exercice d'un diagramme de réseau Cisco normé, grâce au développement d'une logiciel interagissant avec l'utilisateur·rice. Il·Elle lui sera demandé·e, via une interface graphique, le type (routeurs, commutateurs ou PC) et le nom de la machine, l'adresse de réseau, la liaison avec la machine, ainsi que l'interface par laquelle elle sera liée. Chaque liaison entre deux machines crée par l'usager·ère sera visible dans un tableau et aidera à la création automatisée du diagramme.

2. Étude des logiciels existants

Actuellement sur le marché, tout laisse à croire que les logiciels permettant la création de diagramme automatisée demandent à l'utilisateur de créer lui-même son diagramme en glissant-déposant chaque élément, à savoir les machines, les liens bidirectionnels et les labels. Parmi les sept meilleurs logiciels déclarés, ceux-ci peuvent être sous licence payante comme Microsoft Visio, détenir des versions gratuites limitées qui peuvent, par un ou des paiements, fournir de meilleures prestations comme Lucidchart ou être gratuite comme Dia.

Durant le cursus scolaire, un logiciel a été utilisé pour un autre cours appelé « Modélisation » permettant de développer principalement des modèles UML s'appelant Diagrams.net, anciennement drawio.

En poussant les recherches, deux logiciels permettent de créer un diagramme à partir de fichier avec l'extension .csv :

- Microsoft Visio,
- Diagrams.net (drawio).

Après réflexion, le choix s'est porté sur Diagrams.net, car ce logiciel est gratuit et est connu par les élèves de la filière IG.

2.1 Présentation de Diagrams.net (drawio)

Diagrams.net, anciennement drawio, est un logiciel de création de divers diagrammes. Le logiciel est entièrement gratuit et open source et est développé en HTML5¹ et en JavaScript². Il peut être utilisé via une application web à l'adresse : <https://www.diagrams.net/> ou télécharger directement sur son poste de travail. Le logiciel est compatible avec Windows, macOS, Linux, Google Chrome OS. Il peut être intégré dans des services cloud (Dropbox, OneDrive, Google Drive, GitHub et GitLab), mais également être utilisé en plugin pour des applications web sur les plateformes comme NextCloud, MediaWiki, Notion, Atlassian Confluence et JIRA, ainsi que des packages sur des environnements de développement comme PyCharm, IntelliJ IDEA, Visual Studio Code, atom.io, etc. Il propose différents types d'importation, d'intégration et d'exportation qui vont être essentiels pour ce projet.

¹ HTML5 (HyperText Markup Language 5) est un langage de programmation ayant comme dernière version majeure, la cinquième version, et permet, grâce à des balises, de définir le visuel des pages web.

² JavaScript est un langage de programmation utilisé pour le développement d'application web.

3. Analyse de la viabilité de l'application

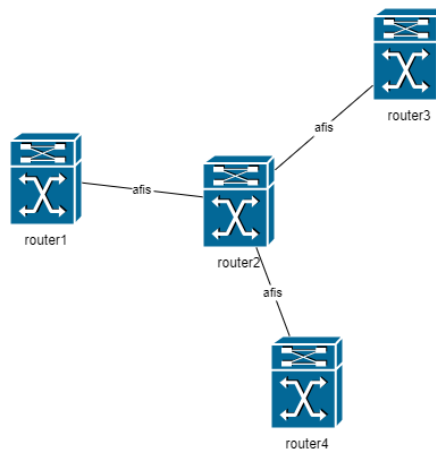
3.1 Pseudo code via le logiciel Diagrams.net

Diagrams.net propose l'importation des données d'un diagramme à partir de fichier texte, CSV, SQL et Mermaid. Une première familiarisation avec le pseudo code via l'importation par des fichiers csv a été effectué. Cela aurait été possible de le faire ainsi, mais n'était pas facile d'utilisation, car ce dernier était strict et complexe.

Les données étaient inscrites dans un fichier texte (extension en .txt) et ne pouvaient pas être modulables pour un diagramme de réseau. En voici un exemple :

Figure 1 : Exemple de sortie du pseudo code

```
# label: %local_pe%
# identity: id
# style: shape=shape;html=1;pointerEvents=1;dashed=0;fillColor=#036897;strokeColor=#ffffff;strokeWidth=2;verticalLabelPosition=bottom;verticalAlign=top;align=center;outlineConnect=0;
# namespace: csvimport
# connect: [{"from": "remote_pe", "to": "local_pe", "invert": true, "label": "afis", "style": "curved=0;endArrow=none;endFill=0;"}]
# width: 64
# height: 82
# padding: 15
# nodespacing: 40
# levelspace: 100
# edgespacing: 40
# layout: auto
# parent: rparent
# parentstyle: shape=rect;whiteSpace=wrap;html=1;horizontal=1;horizontalStack=0;resizeParent=1;fillColor=#f8f8ff;strokeColor=#666666;perimeter=rectanglePerimeter;container=1;recursiveResize=1;collapsible=0;
## CSV starts under this line
id,local_pe,local_as,shape,afis,remote_pe
1,router1,as-101,mxgraph.cisco.routers.atm_tag_switch_router,vpnv4,
2,router2,as-201,mxgraph.cisco.routers.atm_tag_switch_router,vpnv6,router1
2,router2,as-201,mxgraph.cisco.routers.atm_tag_switch_router,vpnv6,router4
3,router3,as-301,mxgraph.cisco.routers.atm_tag_switch_router,ipv6 unicast,router2
4,router4,as-401,mxgraph.cisco.routers.atm_tag_switch_router,ipv4 unicast,
```



Malheureusement, cette première option n'a pas pu aboutir. En effet, le modèle se rapprochait de celui voulu, mais avait ses défauts. Deux étaient marquants :

- Un label par liaison était possible, or trois devaient être présents.
- Les périphériques intermédiaires et finaux proposés par Diagrams.net n'étaient pas les mêmes utilisés qu'aux cours de réseau et devait être remplacé par des icônes importées.

3.2 Module Python N2G

Une seconde option fut possible en utilisant le module Python N2G. Ce dernier permet de développer un fichier texte structuré dans le format XML utilisé par Diagrams.net. Après quelques tests, cette option était assurément la meilleure. Elle était facile d'utilisation, pouvait être utilisée avec des importations de données en dictionnaire, en liste, en xml, en csv et en code brut et fournissait toutes les contraintes voulues pour l'exécution de la future application.

4. Langage de programmation

Le langage de programmation choisi pour ce logiciel est Python. Ce choix a été pris, car le module N2G est utilisable en Python. Ce langage de programmation a l'avantage d'être utilisé pour une programmation impérative structurée, fonctionnelle et orientée objet. Il n'y a donc pas besoin d'utiliser d'autres langages de programmation complémentaire.

5. Librairies

5.1 Librairies pour l'interface graphique

Afin de créer l'interface graphique, il a fallu effectuer des recherches sur une librairie Python adéquate. Les recherches ont été concluante et deux librairies ont été découvertes.

La première étant PyQt, mais était sous une licence payante. Cette dernière apportait un GUI (Graphical User Interface) intégré qui détient une utilisation semblable à celui de Microsoft Visual Studio pour le langage de programmation C#. Il n'aurait suffi que de disposer les widgets en les glissant-déposant à notre convenance dans la fenêtre dédiée.

La seconde était tkinter et celle-ci fut choisie. Après des premiers essais pour la création d'interfaces, ces dernières étaient faciles d'utilisation, car elles détiennent une documentation détaillée. Dans un premier temps, le visuel ne paraissait pas moderne et ressemblait, à titre de comparaison, à l'interface de Windows XP. De nos jours, la majorité des applications détiennent un graphisme épuré, affiné et avec des fenêtres et des widgets au bord arrondi. Il a fallu donc trouver une alternative à cela, afin de rendre l'application moderne. De ce fait, une librairie complémentaire ou une remplaçante devaient être trouvées. Le résultat de ces recherches fut concluant et une librairie complémentaire appelée customTkinter fut décelée.

5.1.1 Présentation des librairies tkinter

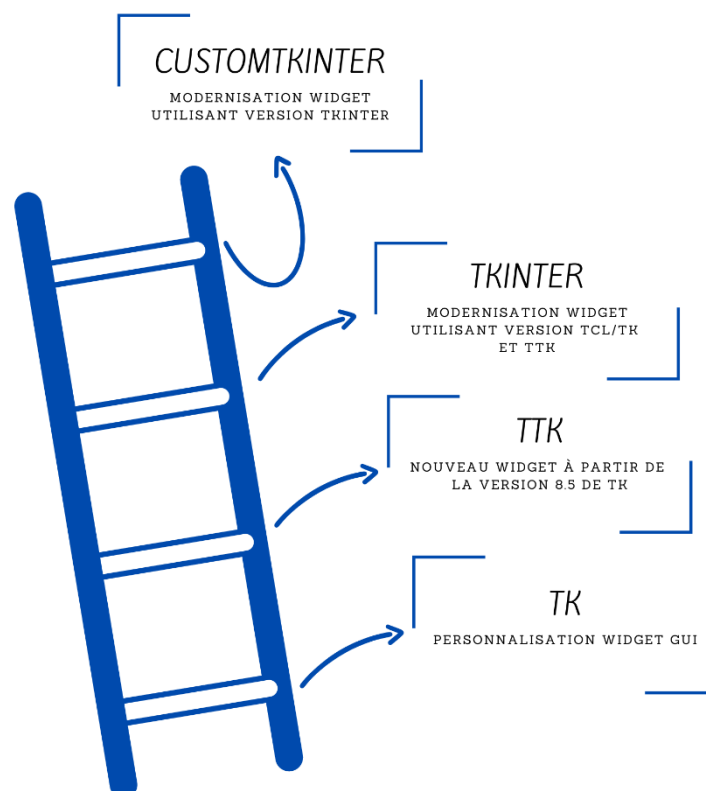
Tkinter est une librairie libre qui permet de créer une interface graphique sur Python. Tkinter vient à l'origine de « Tk interface ». Cette dernière est une version adaptée de la librairie Tk GUI qui a permis de le rendre plus moderne lors de sa création. Il est compatible avec les plateformes Unix, impliquant macOS et le Windows.

5.1.2 Présentation de la librairie customTkinter

CustomTkinter a été créé par TomSchimansky et détient un GitHub : <https://github.com/TomSchimansky/CustomTkinter>. L'interface graphique est d'autant plus moderne que son prédécesseur tkinter et convient au visuel du marché actuel. Les bords des widgets (boutons, curseurs, radios boutons, etc.) sont arrondis. Elle propose également un mode sombre ou clair.

Afin de se représenter la hiérarchie du logiciel, une illustration est attachée présentement :

Figure 2 : Illustration chronologique des librairies pour l'interface graphique



(Illustration créée avec canva.com)

Cette échelle représente la chronologie de Tk jusqu'à CustomTkinter. Toutes les librairies utilisent le langage de programmation Tcl.

5.2 Bibliothèques pour l'algorithme

Pour effectuer l'algorithme, plusieurs bibliothèques ont dû être utilisées. Pour faciliter la compréhension de celles-ci, un tableau illustratif est mis à disposition ci-après :

Tableau 1 : Tableau des bibliothèques utilisées

Nom de la bibliothèque	Description	Utilisation
PIL	PIL (Python Imaging Library) permet de manipuler des images. Elle est utilisée avec les extensions PPM, PNG, JPEG, GIF, TIFF et BMP.	Elle a été utilisée pour l'importation des trois images de titre.
csv	Csv (Comma Separated Values) permet l'importation et/ou l'exportation d'un fichier csv. Un fichier csv contient des données séparées par un délimiteur.	Csv a été exploitée pour l'importation (lecture d'un fichier csv) et l'exportation (écriture d'un fichier csv).
Ipaddress	Ipaddress permet de pratiquer plusieurs manipulation ou vérification d'une adresse IPv4 ou IPv6.	Elles ont été employées pour la comparaison entre deux adresses, des vérifications telle que si l'adresse est privée, dans quelle classe, elle appartient ou à quel réseau, elle dépend.
N2G	N2G permet de créer des graphes à partir de données.	N2G a été servie pour générer un diagramme consultable sur Diagrams.net.
os	Os permet d'exploiter les capacités qui dépendent du système d'exploitation. Il est possible de lire et/ou d'écrire un fichier avec sa méthode open() ou encore de os.path pour manier les chemins.	Dans ce projet, la méthode open() est utilisé pour la lecture et pour l'écriture du fichier csv. La méthode os.path a également été utilisée pour la recherche du chemin de draw.io.exe, afin de transformer le diagramme exporté en image, mais ce dernier ne marche que si le chemin de draw.io.exe se trouve au chemin désigné. Il a été mis dans un try / exception pour qu'une erreur ne se déclenche pas.

6. Désirs pour l'application

Avant le développement de l'application, des choix déterminés ont été établis comme la compatibilité du logiciel, les langues, les couleurs, le design et la documentation.

6.1 Compatibilité

La décision de permettre à la compatibilité ascendante et descendante, ainsi qu'à plusieurs systèmes d'exploitation (Unix, macOS et Windows) était une partie essentielle à ce projet. En effet, tant les professeur·e·s que les élèves peuvent être utilisateur·rice·s de l'application et détiennent des ordinateurs de toute marque et de toute génération.

6.2 Langues

Concernant l'interface graphique de l'application, ce dernier est en français. Ce choix vu évident, car le logiciel sera utilisé pour les cours de réseau qui sont donnés en français.

Pour le script du logiciel, celui-ci est en anglais. Comme enseigné à la HEG, le français doit être évité à tout prix pour ne pas avoir de problème de compréhension et de confusion. De ce fait, cette décision fut prise.

6.3 Couleurs

Les couleurs utilisées dans le logiciel n'ont pas été choisies par hasard. Elle dispose, comme couleur dominante, le bleu. Le bleu signifie la communication et le renouveau. Les bleus présents sont le ton et sous-ton du logo de Cisco. Le logo ressemble à ceci :

Figure 3 : Logo Cisco



(Wikipédia)

La seconde couleur est le blanc. Sa signification est portée sur l'élégance, la puissance et la pureté. Cette dernière a l'avantage de s'adapter à toutes les autres. Elle est apparente pareillement comme seconde couleur dominante sur le site de Cisco.

6.4 Style de l'application

Il a été demandé que le logiciel soit fonctionnel, ergonomique et plaisant à utiliser. C'est pour cela que l'esthétique de l'application est épurée et ne détient pas d'éléments superficiels. Une normalisation des designs, qui sont présentement sur le marché, a été établie. Elle se décrit comme étant minimaliste, épurée, illustrée et assignée de la douceur avec des bords arrondis tant pour les fenêtres que pour les widgets.

6.5 Documentation de l'application

Il n'est pas sensé sans savoir que Python est un langage de programmation qui ne détient que peu de contrainte. Afin d'être en cohérence avec ce qui a été appris à la HEG, toutes les variables de l'application sont données avec leur type d'objet de variable et chaque méthode détient son type d'objet de retour accompagné de sa docstring qui explique l'utilité de celle-ci, des explications sur ses paramètres et de sa valeur retournée. Cela a servi de créer en aval une documentation complète de l'application dans un fichier .html se trouvant dans le dossier « Documentation » de l'application.

7. Fonctionnement de l'application

L'application détient un tableau sur la fenêtre principale qui aidera à l'exécution du diagramme. Ce tableau dispose d'un en-tête et de lignes contenant les informations d'une liaison unidirectionnelle entre deux machines. Ces lignes disposent les informations suivantes :

- le type de la machine (routeur 1921 et 1841, commutateur 3660 et 2960 et PC),
- le nom de la machine,
- l'adresse de réseau et son préfixe,
- la liaison avec une autre machine
- l'interface affiliée par la machine.

Les colonnes du tableau peuvent être triées par ordre alphabétique si l'utilisateur·rice clique sur un des en-têtes.

Pour ajouter des lignes dans le tableau, l'usager·ère a la possibilité d'importer un fichier .csv contenant des données compatibles au tableau ou ajouter un lien, grâce à un formulaire s'ouvrant sur une seconde fenêtre.

II-Elle peut également modifier ou supprimer une ligne en la sélectionnant et en cliquant sur le bouton voulu (« Modifier un lien » ou « Supprimer un lien »). Lors de la modification, une fenêtre similaire à celle de l'ajout de lien s'ouvre.

L'utilisateur·rice devra être attentif·ve à insérer les liaisons de façon bidirectionnelle, c'est-à-dire de créer deux liaisons unidirectionnelles, pour le bon fonctionnement de l'application.

Lorsque celui·elle-ci est satisfait·e de sa base de données, il·elle exportera le diagramme à l'emplacement voulu. L'algorithme effectuera des vérifications de champs avant de créer et enregistrer les deux fichiers. Ces fichiers sont :

- le fichier contenant le diagramme avec l'extension .drawio,
- le fichier contenant les données du tableau avec l'extension .csv.

Si le chemin défini par draw.io.exe se trouve à l'emplacement « C:\Program Files\Draw.io\draw.io.exe », dans ce cas, un troisième fichier sera enregistré avec l'extension .png.

8. Interface graphique de l'application

Il y a dans ce logiciel trois fenêtres apparentes :

- la fenêtre principale,
- la fenêtre pour l'ajout d'un lien,
- la fenêtre pour la modification d'un lien.

8.1 Fenêtre principale

La fenêtre principale détient un tableau contenant les informations pour chaque liaison unidirectionnelle, mais également des boutons tels que :

- « Importer un fichier »,
- « Ajouter un lien »,
- « Modifier un lien »,
- « Supprimer un lien »,
- « Créer le schéma ».

Lors de l'exécution de l'application, la fenêtre ressemble à ceci :

Figure 4 : Fenêtre principale au démarrage de l'application



À l'exécution de l'application, seuls les boutons « Importer un fichier » et « Ajouter un lien » sont présents. Lors de l'importation de fichier, une fenêtre s'ouvre, afin que l'utilisateur·rice sélectionne le fichier csv correct et conforme au tableau se trouvant dans la fenêtre principale. Les données sont ensuite retranscrites dans le tableau de la fenêtre principale.

Dès qu'un lien est ajouté, la fenêtre aura cet aspect :

Figure 5 : Fenêtre principale après ajout de lien



Le bouton « Importer un fichier » n'est plus apparent et laisse place au bouton « Créer le schéma » permettant l'exécution de l'algorithme pour la création du diagramme pour l'ouverture sur Diagrams.net et les données du tableau transférées dans un fichier csv. Si le chemin du logiciel de Diagrams.net se trouve être « C:\Program Files\Draw.io\draw.io.exe », alors un troisième fichier avec l'extension .png sera créé. Les fichiers seront sauvegardés à l'emplacement voulu de l'utilisateur·rice.

Dès lors qu'un lien est sélectionné dans le tableau, deux boutons apparaissent : « Supprimer un lien » et « Modifier un lien ». Si l'adresse choisi est une adresse se trouvant dans une salle et est différente du réseau 10.136.0.0/22, la mention « Vous êtes dans la salle [...] » se situera en haut à gauche du tableau.

Lors de la suppression, si la machine se trouvant dans la ligne à supprimer se trouve être la dernière, cela engendra également une suppression dans la colonne « Liaison » ce nom de machine dans les autres lignes.

8.2 Fenêtre d'ajout de lien

Lorsque l'utilisateur clique sur le bouton « Ajouter un lien », la seconde fenêtre apparaît :

Figure 6 : Fenêtre pour l'ajout d'un lien

Ajouter un lien

Ajouter
UN LIEN

Machine Routeur 1921 Nom de la machine R1

Adresse de réseau / préfixe 10.136.8.0/26 Liée à la machine

Interface Ajouter

Elle pourra y indiquer le type, le nom, l'adresse réseau, l'interface de la machine et sa liaison avec une autre machine. Lorsque l'utilisateur cliquera sur le bouton « Ajouter », ce dernier déclenchera une série de contraintes de vérification qui sera énoncée plus en détail au [chapitre 9.1](#).

Pour les types de machines, l'utilisateur peut choisir entre un routeur (1921 ou 1841), un commutateur (3660 ou 2960) ou un PC.

Le nom de la machine inconnu sera stocké dans une liste de machines détenue dans l'algorithme. Une vérification est effectuée sur son équivalence, sans la prise en compte des majuscules/minuscules, avec une machine dans la liste de machines. Si le nom est le même, mais est écrit différemment, un changement de nom est établi.

Exemple : Si auparavant, l'utilisateur avait nommé une machine « r1 » et veut à présent la nommée « R1 », le changement se fait automatiquement. Cela ne pourrait pas fonctionner si l'utilisateur ajoute une machine s'appelant « Routeur-1 ». Dans ce cas précis, deux machines seront inscrites dans la liste de machines.

Concernant l'adresse de réseau, celle-ci devra être indiquée par son adresse suivie de son préfixe séparée par une barre oblique (/). De nombreuses vérifications sont effectuées pour l'adresse de réseau. Pour en nommer quelques-unes, il y a un contrôle

sur la mise en forme de l'adresse qui doit être compris entre 0.0.0.0 et 255.255.255.255 et que son préfixe est contenu entre 22 et 30, un contrôle pour savoir si l'adresse est privée ou publique, si l'adresse est normée par les EIGRP 65100 et 65200 de la HEG, que le nombre de machines assigné au réseau ne dépasse pas la quantité utilisée.

La liaison avec une autre machine est une liste déroulante qui contient la liste des machines. Des contraintes de validations sont effectuées pour vérifier, par exemple, que le nom de la machine et la machine liée n'est pas identique, que deux PCs ne soient pas liés, que les machines liées appartiennent à la même adresse, etc.

Le dernier champ est actif que lorsque la liaison est sélectionnée. La liste d'interfaces est automatiquement générée selon le type de machine choisi. Dès lors qu'une interface a déjà été utilisée ultérieurement pour la machine nommée, un message d'erreur se déclenche.

8.3 Fenêtre de modification de lien

En sélectionnant un lien dans le tableau de la fenêtre principale et en pressant le bouton « Modifier un lien », une fenêtre secondaire s'ouvre :

Figure 7 : Fenêtre pour la modification d'un lien

Modifier un lien

Modifier UN LIEN

Machine Routeur 1921 Nom de la machine R1

Adresse de réseau / préfixe 10.136.8.0/26 Liée à la machine
Préfixe compris entre 22 et 30
ex: 10.136.8.0/26

Interface Modifier

Les champs sont les mêmes que ceux de la fenêtre pour l'ajout de lien. L'utilisateur·rice pourra modifier à sa guise l'ensemble des champs, excepté le champ de type de machine. Le champ est inactif, étant donné que des erreurs entre liaisons peuvent survenir et dysfonctionner le processus de l'application.

Lors de changement de nom de machine, celui-ci sera modifié pour l'ensemble des liens détenant cette machine. Si le nom de la machine se trouve être semblable, le nom de la machine sera suivi par « (copie) », afin de distinguer les 2 machines. En effet, cette modification pourrait engendrer des erreurs sur les liaisons en ayant, par exemple, plusieurs interfaces identiques ou un nombre supérieur de machine sur un réseau.

Supposé que l'adresse soit changée, un test de champ sera actif pour vérifier que la liaison détient la même adresse.

Les contrôles de validation des champs sont identiques à celle de la fenêtre pour l'ajout de lien.

9. Développement de l'algorithme

La partie la plus proéminente de ce projet était l'algorithme. À cet effet, des recherches ont été faites pour la création de contraintes de validation pour les champs des fenêtres secondaires d'ajout et de modification de lien. La majorité des contrôles s'exécute, dès lors que les boutons « Ajouter » ou « Modifier » sont cliqués.

9.1 Contraintes de validation

Il est présenté dans ce chapitre, les 14 contraintes de validations qui représentent dans le script python des méthodes. Elles ont été créées, afin de permettre de vérifier les champs des fenêtres d'ajout ou de modification de liaison.

9.1.1 Classe de l'adresse réseau

Cette première validation aidera à savoir si l'adresse de réseau inscrite est de la classe A, B ou C. Si elle ne se trouve dans aucune de ces classes, un message d'erreur se déclenchera.

Pour rappel, les adresses de la classe A sont normées par la HEG et est contenue entre l'adresse 10.136.0.0 et 10.136.231.255 et sont soumis à l'EIGRP 65100.

Concernant les adresses de la classe B, elles sont comprises entre l'adresse 172.16.0.0 et 172.31.255.255.

Finalement, pour la classe C, ce sont les adresses incluses entre 192.168.0.0 et 192.168.255.255.

Durant cette validation, l'adresse de réseau est ajoutée à la liste de réseaux. Si l'adresse appartient à un tableau de réseau d'une des salles de la HEG, cela aura pour conséquence d'inscrire dans la fenêtre principale l'inscription « Vous êtes dans la salle »

précédée du nom de la salle. À contrario, un message sera affiché dans une fenêtre contextuelle.

Le message est le suivant :

« L'adresse n'appartient à aucune classe privée. Pour rappel, les classes sont :

- Classe A : [voir plage disponible par salle]
- Classe B : 172.16.0.0 à 172.31.255.255
- Classe C : 192.168.0.0 à 192.168.255.255 »

9.1.2 Adresse 10.136.0.0 détenant un routeur

D'après les normes de la HEG, l'adresse 10.136.0.0 doit détenir uniquement des routeurs. C'est pour cela que le contrôle de ceci a été créé.

Un ajout de l'adresse dans la liste des adresses de réseau se fera lorsque la validation sera correcte.

Si une erreur est découverte, ce message apparaîtra dans un pop-up : « L'adresse 10.136.0.0/22 ne peut accepter que des routeurs, selon la norme HEG. »

9.1.3 Nombre de machines pour le réseau

Afin de rester en cohérence avec le nombre total d'adresse disponible pour un réseau, une vérification a été établie.

Pour ce faire, il a fallu parcourir l'ensemble des lignes du tableau de la fenêtre principale et comparer l'adresse de réseau ajoutée ou modifiée avec l'adresse de réseau de la ligne. Si tel est le cas, un compteur sera incrémenté. Ce dernier doit être inférieur au nombre de machine disponible du réseau.

Dans le cas contraire, le message « La plage de réseau choisie n'a plus de place disponible. » sera visible dans une fenêtre.

9.1.4 Type d'EIGRP

La HEG détient une norme concernant les adresses de réseau. Ces dernières appartiennent à l'EIGRP 65100 ou l'EIGRP 65200. Afin d'être cohérent avec la règle établie, il a fallu confirmer que l'adresse découle du même EIGRP que les adresses déjà inscrites dans le tableau.

S'il s'avère que l'adresse ne soit pas dans le même EIGRP, alors deux messages possibles peuvent être déclenchés :

- « L'adresse de réseau n'appartient pas à l'EIGRP 65100. »
- « L'adresse de réseau n'appartient pas à l'EIGRP 65200. »

9.1.5 Adresse se trouvant dans une plage d'une autre adresse de réseau

Cette vérification permet de contrôler si l'adresse est contenue dans une plage de réseau existante dans la liste d'adresse de réseau.

Pour ce faire, il a fallu parcourir l'ensemble de cette liste et de vérifier, dans un premier temps, que le préfixe soit différent des préfixes des adresses de réseau de la liste.

La seconde condition est de savoir si une salle était assignée aux adresses déjà connues. Si le cas se présente, une dernière condition fut de connaître si l'adresse se trouve dans le tableau de réseau de la salle. L'ajout de l'adresse dans la liste des adresses de réseau fut également établi.

Si la seconde condition n'était pas concluante, il a été vérifié que l'adresse de réseau ne se trouve pas être 10.136.0.0/22. Dans le cas où la condition fut positive, deux contraintes ont été établies en parallèle. Elles permettent de savoir si l'EIGRP est le 65100 ou l'EIGRP 65200 et que l'adresse est de cette EIGRP avec la contrainte citée au [chapitre 9.1.4](#), « Type d'EIGRP ». La première contrainte parcourait l'ensemble des salles pour savoir si l'adresse se trouvait dans l'une d'elle et gardait en mémoire le nom de la salle et ajoutait l'adresse à la liste des adresses. Pour la seconde contraintes, une évaluation est effectuée en émettant la clause que l'adresse du réseau ne se trouve pas dans les adresses de réseau de la liste. Un ajout est exécuté si l'adresse ne s'y trouve pas.

Dans le cas où l'adresse de réseau serait le 10.136.0.0/22, une vérification avec la méthode du [chapitre 9.1.2](#) est effectuée.

Si aucune des conditions antérieures n'a été établie, deux messages d'erreur peuvent apparaître selon si celui-ci est de l'EIGRP 65100 ou non. Dans l'éventualité que cela le soit, le message est le suivant : « Cette adresse n'appartient pas à cette classe ou n'est pas assignée à l'EIGRP 65100. ». À contrario, cela sera : « Cette adresse n'appartient pas à l'EIGRP 65200. ».

9.1.6 Adresse privée et rassemblement des contraintes du chapitre 9.1.1 au chapitre 9.1.5

Une initiale condition permet de vérifier si l'adresse de réseau est une adresse privée. Lorsque celle-ci est publique, un message d'erreur est émis : « L'adresse de réseau choisie n'est pas privée. ».

Si l'adresse est privée, trois contrôles sont effectués sur l'adresse. Dans le premier cas, il est vérifié que la liste de réseaux soit vide, puis si celle-ci est l'adresse 10.136.0.0/22. Si c'est le cas, il sera contrôlé que celle-ci est un routeur ([chapitre 9.1.2](#)). À l'opposé, une vérification de la provenance de l'adresse sera effectuée, afin de savoir si elle appartient à la même classe que les autres ([chapitre 9.1.1](#)).

Dans un second temps, il sera vérifié que l'adresse déjà existante dans la liste d'adresse, puis de vérifier le nombre de machines ([chapitre 9.1.3](#)). le type d'EIGRP ([chapitre 9.1.4](#)). Après cela, si l'adresse est le 10.136.0.0/22, alors il sera contrôlé également que celle-ci soit un routeur ([chapitre 9.1.2](#)).

Dans un troisième temps, lorsqu'aucune des deux autres conditions sont remplie, il sera évalué si la plage a déjà été utilisée ([chapitre 9.1.5](#)), puis le type d'EIGRP ([chapitre 9.1.4](#)). Finalement, comme précédemment, il est vérifié que l'adresse est le 10.136.0.0/22 et que celle-ci ne soit pas un routeur ([chapitre 9.1.2](#)).

9.1.7 Saisie de l'adresse de réseau

Pour la vérification de la saisie de l'adresse de réseau, la librairie regex a été mis en avant. Celle-ci a permis de standardiser l'entrée de l'utilisateur·rice dans le champ de l'adresse de réseau en étant dans le format suivant X.X.X.X/X. Il est aussi testé que les adresses soient contenues entre 0.0.0.0 et 255.255.255.255 et que les préfixes soient entre 0 et 32.

Par la suite, le dictionnaire contenant les premières adresses de réseau est parcouru, afin d'exiger que l'adresse appartienne à l'une des adresses de réseau selon son préfixe.

Finalement, pour s'assurer d'avoir une bonne adresse, la vérification basée sur l'ensemble des contraintes vu jusqu'à maintenant soit correcte en appelant la contrainte mentionnée au [chapitre 9.1.6](#).

Un message d'erreur apparaît à travers une fenêtre supplémentaire :

« L'adresse de réseau n'est pas valide ou est vide.

Pour rappel, le champ est accepté dès lors qu'il détient une adresse de réseau valide avec son préfixe compris entre 22 et 30. ».

Il s'affiche si aucune condition n'est correcte.

9.1.8 Adresse de réseau est le même que la machine liée

Lorsqu'une machine est liée à une autre, une vérification qui détermine si l'adresse inscrite est la même que celle liée est réalisée. Pour ce faire, l'adresse sera comparée à l'ensemble de la liste d'adresse détenue par la machine liée. Une méthode a été créée, afin d'en faciliter la recherche d'adresses de la machine liée. Elle sera expliquée dans le [chapitre 9.2.2](#). Le message d'erreur pour celui-ci est : « La machine de type XXX ne peut pas être liée avec une machine d'un autre réseau. ».

9.1.9 Deux PC liés

Dans ce test, est vérifié que lorsque la machine est un PC, ainsi que sa liaison, celui-ci déclenche une erreur :

« 2 PCs ne peuvent pas être liés.

Veuillez choisir un autre type de liaison ou type de machine. »

Pour ce faire, une première condition contrôle que le type de machine est un PC. Si tel est le cas, un parcours sur l'ensemble des lignes du tableau des liaisons sera effectué. Finalement, dès lors qu'une correspondance avec le nom de la machine liée est la même qu'une ligne et que le type est un PC, alors l'erreur se déclenche. Dans le cas contraire, rien ne se passe.

9.1.10 Liaison d'interface sérial

Le routeur 1841 détient des câbles serials. Pour rappel, ces derniers ne peuvent pas être reliés qu'avec d'autres interfaces serials. C'est pour cela, que cette vérification est réalisée. Pour ce faire, il est testé que l'interface soit serial0/1/0 ou serial 0/1/1 et que le type de machine et la recherche du type de la machine liée (voir [chapitre 9.2.1](#)) sont des routeurs 1841. Si le résultat de la vérification est négatif, un message d'erreur se déclenche : « Les câbles serial ne peuvent être liés qu'entre eux. ».

9.1.11 Machine détient le bon type et nom que ces prédécesseurs

Il n'est vérifié que la machine ajoutée ou modifiée, ayant un même nom qu'une machine ajouté antérieurement dans le tableau de liaison, détient le même type de machine. À cette fin, un parcours sur la liste des machines est effectué et vérifie que le nom de la machine et le nom est identique à une de la liste. Si tel est le cas, un second parcours sur le tableau de liaison est effectué et contrôle que le type de la machine et le nom de

la machine parcouru dans la liste des machines ne soit pas identique à qui sont trouvés dans le tableau de liaison.

Si tel est le cas, un message d'erreur est provoqué :

« Ce nom de machine existe déjà sous un autre type de machine.

Le type de machine est : XXX ».

À contrario, rien ne se passe.

9.1.12 Routeur détenant l'adresse de réseau

Les routeurs peuvent détenir plusieurs adresses de réseau, mais ne peuvent avoir qu'une adresse de réseau par interface. C'est pourquoi, il a fallu créer une contrainte de validation pour cette situation.

Dans un premier temps, il sera vérifié s'il s'agit d'une modification ou de création de liaison. Dans le cas de la modification, il est testé que l'adresse de la liaison sélectionnée et la nouvelle adresse soit différente et que le type de machine est un routeur (1921 ou 1841). À cet effet, l'adresse est contrôlée qu'elle ne soit pas dans la liste des adresses de la machine. Concernant un ajout de liaison, il est vérifié que la machine soit un routeur, puis que la liste de la machine ne soit pas vide pour pouvoir tester que l'adresse est contenue dans cette liste.

Finalement, si la vérification n'est pas concluante, une erreur se déclenche :

« Le routeur XXX est lié avec une autre machine par cette même adresse.

Veuillez choisir une autre adresse de réseau. ».

9.1.13 Interface déjà utilisée

Cette contrainte de validation permet de savoir si l'interface a déjà été assignée par la machine. Pour ce faire, il est contrôlé que ce soit un ajout de liaison et que l'interface ne soit pas dans la liste des interfaces de la machine. Dans l'hypothèse que cela soit incorrect, plusieurs messages différents peuvent être possibles. S'il ne reste plus d'interface disponibles, le message est « Aucune interface est disponible pour cette machine. ». S'il ne reste qu'une interface, cela est « L'interface est déjà utilisée pour cette machine.

Une dernière interface est disponible :

- XXX .»

S'il en reste plus d'un, alors cela est « L'interface est déjà utilisée pour cette machine.

X interfaces sont disponibles :

- XXX
- XXX
- ... ».

9.1.14 Liaison de machine avec elle-même

Une machine ne peut pas être liée avec elle-même, c'est pour cela qu'avec cette contrainte de validation, il a été vérifié que lorsqu'un ajout est effectué que le nom de la machine et sa liaison ne soit pas identique.

Si tel est le cas un message d'erreur apparaît : « La machine ne peut pas être liée avec elle-même ».

9.2 Méthodes de recherches

Pour faciliter les recherches, deux méthodes ont été créées.

9.2.1 Recherche du type de machine selon le nom de la machine

Pour effectuer la recherche du type de machine selon son nom, il a fallu explorer l'ensemble de la liste de machines et y comparer avec le nom de la machine, afin d'obtenir le même nom inscrit dans le tableau, puis parcourir sur l'ensemble du tableau de liaisons, afin que dès que le nom dans le tableau soit identique à celui tiré de la liste de machines.

Si rien n'est trouvé, le résultat est vide.

9.2.2 Recherche des adresses de réseau selon le nom de la machine

Concernant la recherche des adresses de réseau selon son nom, le principe est semblable à celui de la recherche ci-dessus. Avant toute chose, une liste d'adresse est initialisée à vide. Par la suite, il faut parcourir la liste de machines pour obtenir la bonne syntaxe du nom de la machine, puis y boucler sur la liste du tableau de liaisons et comparer les noms. Si la comparaison coïncide, alors un ajout dans la liste d'adresses est effectué. Finalement, le résultat de la recherche est cette liste.

9.3 Exécution des valeurs de la liaison

Tant pour la modification que pour l'ajout de valeurs, des modifications et des stockages de valeurs doivent être exécutés, afin de ne pas avoir une interruption de l'application ou des potentiels erreurs. À cet effet, lorsque la machine liée n'est pas ou plus indiqué, la valeur du champ de l'interface est automatiquement vidée.

Dans un second temps, les valeurs sont stockées dans une variable.

Dans le cas où l'adresse de réseau serait 10.136.0.0/22, la valeur de la machine liée est modifiée par « EIGRP 65100 ». Pour son interface, l'utilisateur·rice devra y inscrire ultérieurement en modifiant cette entrée.

Si le nom de la salle est connu, une phrase apparaîtra dans la fenêtre principale. Le message est le suivant : « Vous êtes dans la salle » suivi du nom de la salle. Dans le cas contraire, aucun élément ne sera visible sur l'application.

L'affichage de boutons réapparaît avant la fermeture de la fenêtre secondaire. Lorsque plus de deux éléments se trouvent dans le tableau des liaisons, le bouton « Créer le schéma » apparaît. Dans tous les autres cas, le bouton « Ajouter un lien » est présent sur l'application et le bouton « Importer un fichier » est supprimé.

9.4 Modification des valeurs de la liaison

Des modifications de valeurs sont à effectuer, afin qu'il n'y est pas d'incohérence et d'erreur dans le tableau des liaisons.

Avant que l'exécution de l'ajout ou de la modification de valeur ne se déclenche, la ligne sélectionnée est enregistrée dans une variable. Elle permet dans un premier temps d'être utilisé pour la validation du [chapitre 9.1.12](#).

Le dictionnaire comportant la liste des interfaces utilisées par machine est modifié lorsque l'interface de la ligne sélectionnée est remplacée. Par la suite, si le type de la machine est un routeur, un remplacement dans la liste des adresses par routeur se trouvant dans un dictionnaire est également établi lorsque l'adresse est modifiée.

Dans l'éventualité que l'utilisateur·rice, lors de la modification, indique un nom semblable à une autre machine, le mot « (copie) » sera suivi du nom de la machine. Cela permet de ne pas avoir de problème avec les interfaces précédemment utilisées par les deux machines, ainsi que les adresses de réseau.

Lorsqu'un nom de machine est modifié, toutes les lignes comprenant celui-ci changera également. Pour ce faire, il faut parcourir l'ensemble des lignes et de faire un changement de nom lorsqu'il est identique au nom de machine de la ligne sélectionnée.

Si l'adresse est modifiée avec le 10.136.0.0/22, alors la liaison sera avec l'EIGRP 65100.

9.5 Ajout des valeurs de la liaison

À propos de l'ajout des valeurs de la liaison, un premier test est fait pour savoir si le nom de la machine est déjà connu de la liste. Si tel est le cas, un parcours sur cette liste est effectué et une comparaison entre le nom de la machine et la liste est effectué sans prendre en compte une syntaxe similaire au nom de la machine dans la liste, afin de modifier les lignes du tableau selon le nouveau nom assigné.

Comme pour la modification, lorsque l'adresse 10.136.0.0/22 est mise, alors sa liaison sera automatiquement l'EIGRP 65100.

9.6 Suppression des valeurs de la liaison

Pour la suppression d'une ligne dans le tableau des liaisons, cette dernière ne se présente pas comme l'ajout ou la modification de liens. Aucune fenêtre secondaire ne s'ouvre lors de l'exécution de la suppression. Seule la sélection de la ligne à effacer et une pression sur le bouton « Supprimer un lien » sont nécessaire.

Dans un premier temps, un stockage de la ligne sélectionnée est fait, afin de pouvoir modifier des données avant sa suppression définitive. À cela, le tableau de liaison est parcouru, afin de savoir si la machine est la dernière ou si d'autres liaisons détiennent encore celle-ci. Dans le cas où ce serait la dernière, toutes les liaisons contenant le nom de la machine doivent également être effacées, afin de garder une cohérence et d'éviter des erreurs qui pourraient interrompre l'application. La disparition du nom de la machine sera également établie sur la liste des machines, ainsi que la ligne sélectionnée dans le tableau des liaisons. Pour l'interface graphique, les boutons « Supprimer un lien » et « Modifier un lien » sont automatiquement enlevés. Ils apparaîtront à nouveau lorsqu'une ligne sera sélectionnée.

Dans un second temps, le nombre de ligne dans le tableau des liaisons est vérifié. Lorsque le tableau ne comporte aucune ligne, le bouton « Importer un fichier csv » apparaît à nouveau. Les variables, comme celle de la salle actuelle et de l'EIGRP, sont réinitialisées à vide.

Dans le cas où il ne reste qu'une ligne dans le tableau des liaisons, le bouton « Créer un schéma » disparaît. La variable comprenant le nom de la salle est également réinitialisée si l'adresse de la dernière ligne est 10.136.0.0/22.

Il est également vérifié que lorsque le nom de la salle n'est plus connu de la variable la contenant, alors le label affichant le nom de celle-ci dans la fenêtre principale ne soit plus visible.

Lorsque la machine n'est plus connue de la liste des machines et que le dictionnaire comprenant la liste des interfaces par machine est existant, une suppression par le nom de la machine est effectuée dans le dictionnaire.

Dans l'éventualité que la ligne sélectionnée est de type routeur, deux vérifications sont possibles. La première étant que lorsque cette ligne ne détient pas le dictionnaire comportant la liste des adresses utilisées par le routeur n'est pas vide ou que le nom de la machine n'est plus contenu dans la liste des machines et que le dictionnaire n'est pas vide, la suppression par le nom du routeur est effectuée dans le dictionnaire.

La seconde étant que si le nom de la machine est encore connu de la liste des machines et que la liste d'adresses du dictionnaire est remplie, alors il est supprimé que l'adresse de la ligne sélectionnée.

9.7 Importation de fichier avec extension .csv

L'importation du fichier csv se fait à travers le bouton « Importer un fichier » et permet d'implémenter les données dans le tableau des liaisons. Le fichier csv doit répondre à la norme de l'application, à savoir détenir un tableau semblable à un tableau de liaisons. Il devra comporter sur la première ligne un en-tête comme ceci :

Tableau 2 : En-tête du fichier csv

Type de machine	Nom de la machine	Adresse réseau	Interface	Liaison
-----------------	-------------------	----------------	-----------	---------

Pour l'importation du fichier csv, celle-ci a été effectuée, grâce l'importation de filedialog de la librairie tkinter. Elle a servi à obtenir le chemin du fichier ce qui avec la méthode open de la librairie os a permis de lire le fichier. Il a été bien entendu assuré avec l'encodage UTF-8 pour les éventuels accents ou caractères spéciaux du fichier csv.

Par la suite, un parcours sur l'ensemble des lignes du fichier csv a été fait en ignorant la première ligne d'en-tête. Cela permet, dans un premier temps, de remplir les lignes du tableau des liaisons dans la fenêtre principale de l'application, mais également de remplir toutes les listes, les dictionnaires et les variables en backend de l'application.

Toute la phase d'ouverture du fichier se trouve dans le bloc de try/except, à l'intention que lorsqu'une exception est déclenchée le message d'erreur « Le fichier n'a pas pu être importé. » est déclenché.

L'affichage ou non du label « Vous êtes dans la salle XXX » est également fait.

9.8 Exécution du diagramme (exportation en fichier drawio et csv)

9.8.1 Attribution des adresses IPv4 aux machines

L'attribution des adresses IPv4 pour chaque machine a été établie. Pour ce faire, chaque ligne du tableau des liaisons a été parcourue et chaque adresse IPv4 des adresses de réseau est stockée dans un dictionnaire, afin de faciliter l'utilisation. À chaque machine est assignée une adresse IPv4 selon l'adresse de réseau, puis la valeur du dictionnaire est incrémentée d'un, à l'exception des commutateurs.

Toutes les données importantes pour la création du diagramme est stocké dans un dictionnaire qui prend comme type de valeur :

- l'interface,
- l'adresse IPv4,
- adresse de réseau,
- le nom de la machine liée,
- le type de la machine.

9.8.2 Exportation du fichier drawio

Avant l'exportation, la création du schéma doit être établie avec la librairie N2G. Pour ce faire, il a été choisi de créer à partir d'un dictionnaire. Il contient une liste de nœuds qui représente les machines et une liste de liaison qui représente les liens entre deux machines.

Un parcours sur l'ensemble des lignes du tableau de liaison a été fait, afin que, dans un premier temps, le style du nœud soit affilié, ainsi que celui du lien. Pour ajouter les bons libellés, il a été ajouté en parcourant sur le dictionnaire obtenu, grâce à la méthode du [chapitre 9.6.1](#). Il a été utilisé deux fois. La première pour le nom de la machine et la seconde pour la machine liée. Pour l'EIGRP 65100, donné pour les adresses IPv4 du réseau 10.136.0.0/22, il a fallu créer également le nœud de l'EIGRP 65100 qui est représenté graphiquement par un nuage.

Vient ensuite la création du diagramme, grâce au dictionnaire créé au début de cette méthode. L'algorithme choisi pour la sortie du diagramme a été kk (algorithme dirigé par

la force de Kamada-Kawai). Il a pour l'avantage d'être un algorithme rapide et est basé sur l'algorithme de Fruchterman-Reingold.

Finalement, il est demandé à l'utilisateur·rice de choisir le répertoire d'enregistrement des fichiers et le nom des fichiers, grâce au `filedialog` de la librairie `tkinter` le chemin du fichier sera stocké dans une variable. En appelant la méthode `diagram.dump_file(filename=XXX, folder=XXX)`, il sera sauvegardé à l'endroit voulu.

9.8.3 Exportation du fichier avec extension .csv

La création du fichier csv utilise, comme l'importation de fichier csv, la méthode `open` de la librairie `os`. Un encodage est également établi en UTF-8. La première ligne est celle de l'en-tête et est la même que celle présente sur le tableau des liaisons de machine. Pour le reste des lignes, un stockage par ligne est effectué en utilisant la méthode `csv.writer(XXX).writerow(XXX)`.

9.8.4 Exportation du fichier en image

Concernant l'exportation du fichier en image .png, cela ne marche que si `draw.io.exe` se trouve sur le chemin « `C:\Program Files\Draw.io\draw.io.exe` », car grâce à la librairie `os`, il a été possible de créer une ligne de commande qui appelle l'application `Diagrams.net` et établie un export en format png accompagné du chemin de l'enregistrement et qui est la suivante :

```
os.system('"C:\Program Files\Draw.io\draw.io.exe' --export --format png --uncompressed " + folder + "/" + file_path_split[-1])
```

Après avoir réussi à créer l'image, elle est automatiquement ouverte après la fermeture de l'application.

Ce code est inséré dans un bloc `try/except`, afin qu'il ne déclenche pas d'exception qui pourrait interrompre l'application.

Malheureusement, une meilleure solution ne s'est pas annoncée lors de ce projet, mais sera pour l'avenir de ce logiciel, une fonctionnalité à développer.

10. Création de la documentation

La création de la documentation a été choisie décidée avant la création de l'application. Elle a permis de décrire au mieux les différentes méthodes de l'application et d'avoir une meilleure représentation des milliers de lignes de code. Pour parvenir à un tel résultat, des recherches ont été établies, afin de connaître les modules pouvant être utilisés pour la génération de la documentation. Une API a été trouvée et est très répandue. Celle-ci

s'appelle pdoc. Il a été utilisé en ligne de commande dans l'invitation de commande de Windows. Il a fallu tout d'abord installer pdoc et cela a été fait par la ligne de commande :

pip install pdoc.

La ligne de commande pour l'exécution de la documentation est la suivante :

pdoc 'chemin du script python' -- output-dir 'chemin du dossier de l'enregistrement de la documentation'.

Il comporte le nom de l'API utilisé (pdoc), suivi du chemin du script python en .py, puis de -- output-dir avec le chemin du dossier d'enregistrement qui détermine le dossier auquel la sauvegarde est voulue. À la fin de l'exécution, un fichier en .html est créé détenant les méthodes de l'application, sa description, ses paramètres et la visualisation du code. D'autres fichiers comme search.js et index.html sont également conçus.

Les fichiers énumérés ci-dessus ont été supprimés, car pour ce projet, ils n'étaient pas nécessaires. Le fichier .html a été modifié, afin de répondre au besoin de la documentation voulue. Pour ce faire, la méthode de recherche a été enlevée et les titres ont été changés. Le nom du fichier a également été modifié par index.html. Le résultat de la documentation se trouve dans le dossier Documentation du dossier de l'application.

11. Création du manuel d'utilisation

Un manuel d'utilisation a été créé, afin que l'utilisateur·rice puisse connaître le fonctionnement de l'application. Il est énuméré dans celui-ci la marche à suivre pour la création d'un schéma de réseau. Il se trouve dans le dossier Documentation du dossier de l'application sous le nom de manuel_utilisation.pdf. Il comporte les étapes suivantes :

- Importation d'un fichier csv
- Ajout d'un lien
- Modification d'un lien
- Suppression d'un lien
- Enregistrement du diagramme
- Modification du diagramme dans le fichier *.drawio
- Exportation sous un format spécifique (PNG, PDF, ...)

Chaque étape est illustrée par des images de l'application et indiquée par des encadrés rouges.

Des messages d'avertissement sur des erreurs pouvant survenir durant l'utilisation de l'application et pouvant l'interrompre sont indiqués avec le symbole \triangle et détiennent un texte en gras et en rouge.

12. Développement de l'application .py en .exe

Pour le développement de l'application en .exe (« executable »), des recherches sur la manière de le réaliser ont été établies. Plusieurs modules étaient possibles. Le choix s'est porté sur auto-py-to-exe qui est une interface graphique simplifiée et utilise PyInstaller de Python.

Il a fallu tout d'abord l'installer dans le PC et c'est fait en ligne de commande dans l'invite de commande de Windows. La commande est : *pip install auto-py-to-exe*.

Par la suite, la commande *auto-py-to-exe.exe* a dû être établie, afin d'ouvrir le logiciel.

Une fenêtre s'ouvre et le chemin du script python de l'application est demandé, dans le cadre de ce projet, ce fut app.py. Ensuite, il est demandé si l'application devra être en un dossier ou un fichier. Le choix du dossier devait être fait étant donné l'utilisation de la librairie customTkinter qui a la contrainte de ne pas pouvoir permettre la création d'un logiciel en un fichier. Ceci est expliqué dans la documentation de la librairie à l'adresse <https://github.com/TomSchimansky/CustomTkinter/wiki/Packaging>. L'ajout du dossier de la librairie de customTkinter doit également être ajouté dans les fichiers/les dossiers additionnels. Elle se trouve dans le dossier de Python dans le PC au chemin suivant :

../Programs Files/Python/PythonXXX/Lib/site-packages/customtkinter/

D'autres dossiers additionnels ont été ajoutés, à savoir le dossier documentation, exemples, image et styles.

L'icône de l'application a été jointe. Il est demandé de mettre le chemin de celle-ci.

Dans les paramètres avancés, le nom de l'application a été mis. Ce dernier est « Créateur de schéma de réseau ».

Puis, l'exportation de l'application est demandée en indiquant le chemin dans la section des paramètres.

Finalement, la conversion peut être lancée en cliquant sur le bouton « Convert .py to .exe ».

13. Améliorations potentielles

L'ampleur du projet est parue plus conséquente que prévu, étant donné le nombre de contraintes trouvées pour le logiciel. En conséquence, les heures à effectuer (recherches, développement de l'application et écrit) pour le travail de Bachelor n'étaient pas suffisantes. Pour un futur proche de l'application, plusieurs fonctionnalités ont été trouvées et sont citées ci-après.

13.1.1 Contrainte de validation pour les liaisons bidirectionnelles

Le logiciel ne détient pas au moment de la création du schéma, une contrainte permettant de connaître si les liaisons sont bidirectionnelles. Cette évaluation permettrait d'éviter les oublies et les erreurs avant l'exécution finale du diagramme.

13.1.2 Contrainte de validation pour l'importation de fichier csv

Actuellement, l'importation de fichier csv ne détient pas de vérification des données. Il pourrait être imaginé de définir les mêmes contraintes que celle pour l'ajout ou la modification de liaison.

En parcourant l'ensemble du fichier, il pourrait être envisagé que l'erreur de la ligne soit indiquée avec son type d'erreur, lorsqu'une contrainte de validation est déclenchée, pour que l'utilisateur·rice puisse les connaître et les corriger avant une nouvelle importation. Mais aussi de vérifier, si les éléments du tableau correspondent aux listes des champs de l'application. Il pourrait être, par exemple, pour les types de machine et ses interfaces qui sont immuables.

Un contrôle pourrait également s'avérer être utile pour définir la connaissance des liaisons de machines avec une liste des noms de machine connus.

Une vérification de mise en forme du tableau peut être également conçue, afin que le fichier soit aux normes de l'application et qu'aucune erreur soit provoquée. Il serait pensé de connaître si le fichier est vide ou si un en-tête est présent dans le tableau et que les éléments soient bien séparés par des virgules ou des points-virgules.


13.1.3 Importation du manuel d'utilisation dans l'application

Un manuel d'utilisation a été créé pour mieux connaître les fonctionnalités et pour faciliter l'utilisation de l'application. Ce choix, d'importer le manuel d'utilisation dans l'application, a été motivé pour faciliter l'accès de la documentation, car actuellement, la documentation se trouve dans le dossier de l'application dans le dossier documentation.

Un premier essai non-concluant de l'importation du manuel d'utilisation en format pdf a été tenté. Ce dernier était encapsulé dans un label cliquable, appelé « Comment

fonctionne l'application ? », et devait être ouvert dans une fenêtre secondaire. Lorsque le lien était cliqué, la fenêtre s'ouvrait, mais des erreurs pour l'ouverture du pdf fut déclenché par l'interpréteur de Python. Plusieurs tentatives avec diverses librairies ont été testées, mais sans succès. Le code se trouve dans la méthode *create_hypertext_wiki* qui a été commenté pour ne pas déclencher d'erreur.

13.1.4 Ajout des bons liens

Présentement, l'application offre un schéma correct à l'exception des liens. En effet, les câbles serial sont représentés en rouge et linéaire, or comme vu à la HEG sur packet tracer et en classe, ces câbles sont figurés par une ligne brisée, plus connu sous le nom de zigzag. Malheureusement, aucun style de lien de ce genre n'a été trouvé dans Diagrams.net. Il faudra donc le créer. Un lien se rapprochant du lien voulu ressemble à un S avec une flèche et s'appelle Curve. Il ressemble à ceci : .

Il pourrait également être imaginé que l'on puisse distinguer les câbles croisés, des câbles droits en ajoutant leur couleur respective, à savoir le rouge pour les câbles croisés et le bleu pour les câbles droits. Pour ce faire, il faudra générer un algorithme permettant de comparer les deux types de machine liées et d'en assigné le bon câble.

13.1.5 Ajout de nouvelles machines

L'application détient des machines, dites simples, à savoir les routeurs, les commutateurs de couche 2 et les ordinateurs. Il pourrait être imaginé d'ajouter aussi un AP (Access Point) pour les cours de réseau sans fil, un commutateur de couche 3, un hub et tout autre appareil utilisé durant les cours de réseau. Pour chacun, il faudra en assigner les styles des machines, mais également ceux des liens. Il nécessitera également de créer de nouvelles contraintes de validation pour que l'application soit cohérente et que toutes les erreurs soient gérées.

13.1.6 Ajout des protocoles

Plusieurs protocoles ont été abordés durant les cours de réseaux. Parmi ceux-ci, il est étudié :

- Protocole PPPoE (Point-to-Point over Ethernet)
- Tunnel GRE (Generic Routing Encapsulation)
- Protocole BGP
- ACL et ACL étendues (Access Control List) IPv4
- Protocole EIGRP IPv4
- Routage static
- STP (Spanning-tree Protocol)

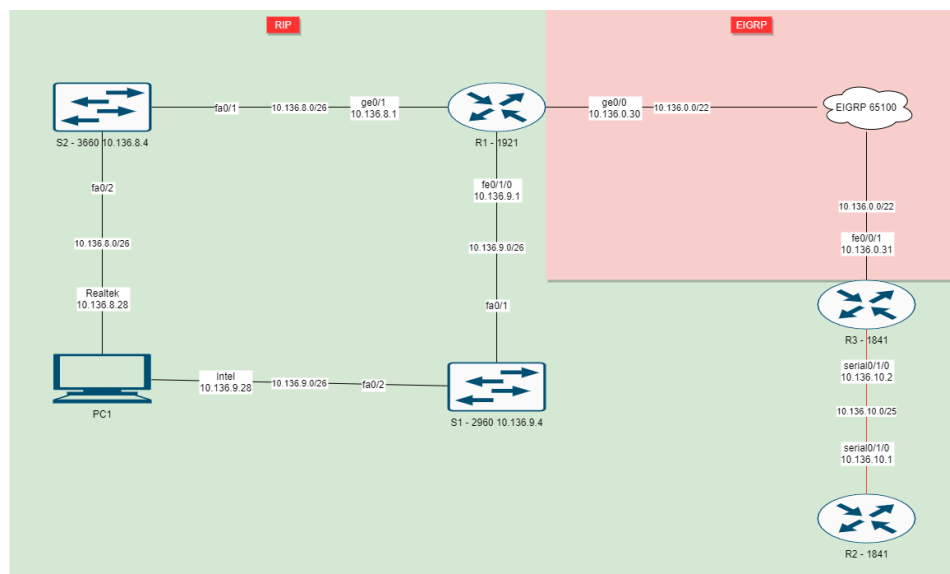
- Protocole HSRP (Hot Standby Router Protocol)
- Protocole OSPF (Open Shortest Path First)
- Etherchannel
- Protocole RIP (Routing Information Protocol)
- NAT (Network Address Translation) et PAT (Port Address Translation)

Ils pourraient être ajoutés dans l'application, afin que celle-ci soit complète. Selon les types de protocole, des encadrés de couleur peuvent être faits pour les distinguer. Pour illustrer ce cas, l'exemple sera narratif :

Un élève apprend les protocoles de routage EIGRP et RIP. Les ensembles de réseaux contenus dans les différents protocoles pourraient être dans des encadrés de couleur, afin de les différencier des uns des autres.

Son énoncé pourrait ressembler à quelque chose comme ceci :

Figure 8 : Exemple de schéma avec des protocoles



L'ajout pourrait se faire sur la version du diagramme dans le format svg, afin de permettre des encadrés de couleurs.

Lorsque l'ajout de protocole sera établi, il ne faudra pas oublier d'ajouter des contraintes de validation tant sur les champs que lors de l'exécution du diagramme pour ne pas d'erreur qui pourrait interrompre le logiciel ou obtenir un affichage correct.

13.1.7 Ajout d'une prévisualisation du diagramme

Pour faciliter la compréhension du futur schéma et l'illustrer, il a été pensé d'effectuer une prévisualisation du diagramme dans une fenêtre secondaire en cliquant sur un bouton cliquable et se trouvant proche du bouton « Créer le schéma ». Le code serait semblable au code pour l'ajout du manuel d'utilisation dans l'application.

13.1.8 Exportation du diagramme en image

Actuellement, les images sont automatisées que lorsque Diagrams.net se trouve dans « C:\Program Files\Draw.io\draw.io.exe ». L'idéal pour cette fonction serait de trouver un moyen de demander au script python de chercher dans l'ordinateur le chemin du lieu de ce logiciel en ayant un temps de recherche rapide. Une solution serait de créer un second script .cmd qui sera exécuté lorsque celui est appelé.

Une autre possibilité serait d'implémenter du code JavaScript avec la librairie JGraph qui est utilisé pour la réalisation des diagrammes dans Diagrams.net et de créer ainsi l'image. Pour ce faire, les données du diagramme devront être transportables entre le script Python et le script JavaScript. Il pourrait être possible d'utiliser un encodage json ou xml pour y faciliter le transport.

14. Conclusion

Le travail de Bachelor était une bonne expérience et a permis de mettre au défi : le savoir-faire et les connaissances instruites à la HEG et de prendre des initiatives de recherches sur ce qui a été faites sur le marché et d'y utiliser à bon escient. Mais également de construire une application unique et fonctionnelle en établissant des recherches sur les différents éléments et fonctionnalités. Malheureusement, les 360 heures sont trop courtes pour les tâches demandées pour le travail de Bachelor et ne laissent pas le temps de développer comme appris lors du GREP qui lui a été fait sur une année. Des étapes de projet étudiées à la HEG sont mises de côté comme les notions économiques de l'application (analyse de marché, analyse SWOT, etc.), sa modélisation et son optimisation.

L'application a un aspect basic et simple, mais sa complexité se repose sur son algorithme qui a permis de créer plusieurs contraintes de validation de champs et d'en modifier les champs nécessaires, mais également de gérer les suppressions.

Le logiciel a pu être grandement développé, mais des points d'améliorations majeures qui a été cité dans le [chapitre 13](#) peuvent encore être apporté.

Bibliographie

7.16. Matching IPv4 Addresses - Regular Expressions Cookbook [Book], [sans date]. [en ligne]. [Consulté le 6 juillet 2022]. Disponible à l'adresse: <https://www.oreilly.com/library/view/regular-expressions-cookbook/9780596802837/ch07s16.html>

ADIRIO, 2017. Answer to « check if ip address belongs to a Class A, B and C in python ». *Stack Overflow*. [en ligne]. 22 février 2017. [Consulté le 9 août 2022]. Disponible à l'adresse: <https://stackoverflow.com/a/42385606>

APPRENDRE LE PYTHON #9 ? INTERFACE GRAPHIQUE (avec Tkinter), 2019. [en ligne]. [Consulté le 20 juin 2022]. Disponible à l'adresse: <https://www.youtube.com/watch?v=N4M4W7JPOL4>

Blog - Insert a diagram from specially formatted CSV data, [sans date]. [en ligne]. [Consulté le 13 juin 2022]. Disponible à l'adresse: <https://www.diagrams.net/blog/insert-from-csv>

CODER, hard, 2017. check if ip address belongs to a Class A, B and C in python. *Stack Overflow*. [en ligne]. 22 février 2017. [Consulté le 9 août 2022]. Disponible à l'adresse: <https://stackoverflow.com/q/42385097>

Color and Style Our Treeview - Python Tkinter GUI Tutorial #118, 2020. [en ligne]. [Consulté le 30 juin 2022]. Disponible à l'adresse: <https://www.youtube.com/watch?v=ewxT3ZEGKAA>

Comment Créer une Interface Graphique avec Python, 2021. [en ligne]. [Consulté le 20 juin 2022]. Disponible à l'adresse: <https://www.youtube.com/watch?v=F8w3RqL1kis>

Create a GUI app with Tkinter - Step by Step Tutorial, 2020. [en ligne]. [Consulté le 20 juin 2022]. Disponible à l'adresse: https://www.youtube.com/watch?v=itRLRfuL_PQ

create csv file python, [sans date]. *pythonspot*. [en ligne]. [Consulté le 4 août 2022]. Disponible à l'adresse: <https://pythonspot.com/files-spreadsheets-csv/>

Create First GUI Application using Python-Tkinter, 2020. *GeeksforGeeks*. [en ligne]. [Consulté le 20 juin 2022]. Disponible à l'adresse: <https://www.geeksforgeeks.org/create-first-gui-application-using-python-tkinter/>

CRIQUI, Robert, 2020. Changing background color of ttk Treeview.Heading. *Stack Overflow*. [en ligne]. 4 août 2020. [Consulté le 5 juillet 2022]. Disponible à l'adresse: <https://stackoverflow.com/q/63239295>

CSTAYYAB, 2022. Answer to « Executing Javascript from Python ». *Stack Overflow*. [en ligne]. 13 juillet 2022. [Consulté le 25 juin 2022]. Disponible à l'adresse: <https://stackoverflow.com/a/72969311>

csv — CSV File Reading and Writing — Python 3.10.6 documentation, [sans date]. [en ligne]. [Consulté le 17 août 2022]. Disponible à l'adresse: <https://docs.python.org/3/library/csv.html>

customtkinter: Create modern looking gui with tkinter and python, [sans date]. [en ligne]. version 0.3. [Consulté le 25 juin 2022 a]. Disponible à l'adresse: <https://github.com/TomSchimansky/CustomTkinter>

customtkinter: Create modern looking gui with tkinter and python, [sans date]. [en ligne]. version 0.2. [Consulté le 28 juin 2022 b]. Disponible à l'adresse: <https://github.com/TomSchimansky/CustomTkinter>

D, Sandeep S., 2021. Answer to « Tk treeview column sort ». *Stack Overflow*. [en ligne]. 23 septembre 2021. [Consulté le 18 août 2022]. Disponible à l'adresse: <https://stackoverflow.com/a/69294086>

DABKOWSKI, Piotr, 2015. Answer to « Executing Javascript from Python ». *Stack Overflow*. [en ligne]. 29 mai 2015. Disponible à l'adresse: <https://stackoverflow.com/a/30537286>

DASH, Ujjwal, 2020. Answer to « Changing background color of ttk Treeview Heading ». *Stack Overflow*. [en ligne]. 4 août 2020. [Consulté le 5 juillet 2022]. Disponible à l'adresse: <https://stackoverflow.com/a/63239388>

Diagrams.net, 2022. *Wikipedia*. [en ligne]. [Consulté le 14 août 2022]. Disponible à l'adresse: <https://en.wikipedia.org/w/index.php?title=Diagrams.net&oldid=1105593415>Page

Version ID: 1105593415

DIENOW, 2018. Answer to « Executing Javascript from Python ». *Stack Overflow*. [en ligne]. 21 mars 2018. [Consulté le 25 juin 2022]. Disponible à l'adresse: <https://stackoverflow.com/a/49402261>

DORATO, Federico, 2020. Answer to « Tk treeview column sort ». *Stack Overflow*. [en ligne]. 29 avril 2020. [Consulté le 18 août 2022]. Disponible à l'adresse: <https://stackoverflow.com/a/61495299>

ECHO, 2019. Answer to « Executing Javascript from Python ». *Stack Overflow*. [en ligne]. 31 octobre 2019. [Consulté le 25 juin 2022]. Disponible à l'adresse: <https://stackoverflow.com/a/58637725>

Entry - Champs de saisie — Tkinter pour ISN, [sans date]. [en ligne]. [Consulté le 5 juillet 2022]. Disponible à l'adresse: <http://tkinter.fdex.eu/doc/entw.html>

Exemple de cahier des charges à télécharger - Word, PDF..., [sans date]. *Cahiers des charges*. [en ligne]. [Consulté le 22 juin 2022]. Disponible à l'adresse: <https://cahiersdescharges.com/exemple-cahier-des-charges-pdf/>

FERGUSSON, Kymberly, 2018. Automatically create draw.io diagrams from CSV files. *draw.io*. [en ligne]. 3 avril 2018. [Consulté le 13 juin 2022]. Disponible à l'adresse: <https://drawio-app.com/automatically-create-draw-io-diagrams-from-csv-files/>

GONZALEZ, Diego, 2022. How can i fit my tkinter app to any size screen? *Stack Overflow*. [en ligne]. 11 janvier 2022. [Consulté le 9 juin 2022]. Disponible à l'adresse: <https://stackoverflow.com/q/70666707>

GROVES, Eddie, 2021. Answer to « Draw.io - how to export all tabs to images using command line ». *Stack Overflow*. [en ligne]. 10 février 2021. [Consulté le 6 août 2022]. Disponible à l'adresse: <https://stackoverflow.com/a/66145548>

GT565K, 2014. Split string with multiple delimiters in Python. *Stack Overflow*. [en ligne]. 1 mars 2014. [Consulté le 14 juillet 2022]. Disponible à l'adresse: <https://stackoverflow.com/q/4998629>

HASSAN, Rami, 2020. Answer to « Tk treeview column sort ». *Stack Overflow*. [en ligne]. 16 août 2020. [Consulté le 18 août 2022]. Disponible à l'adresse: <https://stackoverflow.com/a/63432251>

Home · TomSchimansky/CustomTkinter Wiki, [sans date]. *GitHub*. [en ligne]. [Consulté le 28 juin 2022]. Disponible à l'adresse: <https://github.com/TomSchimansky/CustomTkinter>

How do I get an event callback when a Tkinter Entry widget is modified?, [sans date]. [en ligne]. [Consulté le 6 juillet 2022]. Disponible à l'adresse:

<https://www.tutorialspoint.com/how-do-i-get-an-event-callback-when-a-tkinter-entry-widget-is-modified>

How do you create a clickable Tkinter Label?, [sans date]. [en ligne]. [Consulté le 12 août 2022]. Disponible à l'adresse: <https://www.tutorialspoint.com/how-do-you-create-a-clickable-tkinter-label>

How to Use Tkinter Treeview To Display Tabular & Hierarchical Data, [sans date]. *Python Tutorial - Master Python Programming For Beginners from Scratch*. [en ligne]. [Consulté le 26 juin 2022]. Disponible à l'adresse: <https://www.pythontutorial.net/tkinter/tkinter-treeview/>

How to Validate IPv4 and IPv6 Prefix ? - IT Service Management - Question - ServiceNow Community, [sans date]. [en ligne]. [Consulté le 6 juillet 2022]. Disponible à l'adresse: https://community.servicenow.com/community?id=community_question&sys_id=c8469e6fdb021450fb4ae15b8a9619c6

HU, Jinku, 2020. Comment changer l'état du bouton Tkinter. *Delft Stack*. [en ligne]. 23 janvier 2020. [Consulté le 5 juillet 2022]. Disponible à l'adresse: <https://www.delftstack.com/fr/howto/python-tkinter/how-to-change-tkinter-button-state/>

Installation — N2G 0.3.0 documentation, [sans date]. [en ligne]. [Consulté le 19 juin 2022]. Disponible à l'adresse: <https://n2g.readthedocs.io/en/latest/Installation.html>

ipaddress — IPv4/IPv6 manipulation library — Python 3.10.6 documentation, [sans date]. [en ligne]. [Consulté le 7 août 2022]. Disponible à l'adresse: <https://docs.python.org/3/library/ipaddress.html#logical-operators>

JERRYMOUSE, 2012. Executing Javascript from Python. *Stack Overflow*. [en ligne]. 2 juin 2012. [Consulté le 25 juin 2022]. Disponible à l'adresse: <https://stackoverflow.com/q/10136319>

JOE, 2011. Answer to « Split string with multiple delimiters in Python ». *Stack Overflow*. [en ligne]. 14 février 2011. [Consulté le 14 juillet 2022]. Disponible à l'adresse: <https://stackoverflow.com/a/4998655>

KOS, 2012. Answer to « Split string with multiple delimiters in Python ». *Stack Overflow*. [en ligne]. 1 novembre 2012. [Consulté le 14 juillet 2022]. Disponible à l'adresse: <https://stackoverflow.com/a/13184791>

langfr-420px-Cisco_logo_blue_2016.svg.png (420x222), [sans date]. [en ligne]. [Consulté le 18 août 2022]. Disponible à l'adresse: https://upload.wikimedia.org/wikipedia/commons/thumb/0/08/Cisco_logo_blue_2016.svg/langfr-420px-Cisco_logo_blue_2016.svg.png

LESZEK, 2020. Draw.io - how to export all tabs to images using command line. *Stack Overflow*. [en ligne]. 22 décembre 2020. [Consulté le 6 août 2022]. Disponible à l'adresse: <https://stackoverflow.com/q/65404843>

LEVON, 2020. Answer to « Executing Javascript from Python ». *Stack Overflow*. [en ligne]. 8 avril 2020. [Consulté le 25 juin 2022]. Disponible à l'adresse: <https://stackoverflow.com/a/61104170>

LIVNI, Jonathan, 2011. Answer to « Split string with multiple delimiters in Python ». *Stack Overflow*. [en ligne]. 14 février 2011. [Consulté le 14 juillet 2022]. Disponible à l'adresse: <https://stackoverflow.com/a/4998688>

MADONIUS, 2015. Answer to « Tk treeview column sort ». *Stack Overflow*. [en ligne]. 9 juin 2015. [Consulté le 18 août 2022]. Disponible à l'adresse: <https://stackoverflow.com/a/30724912>

MIRKO, 2018. Answer to « Executing Javascript from Python ». *Stack Overflow*. [en ligne]. 16 septembre 2018. [Consulté le 25 juin 2022]. Disponible à l'adresse: <https://stackoverflow.com/a/52359306>

MULYALIN, Denis, [sans date]. *n2g: Need To Graph*. [en ligne]. version 0.3.2. [Consulté le 19 juin 2022]. Disponible à l'adresse: <https://github.com/dmulyalin/N2G>

NORRIS, Ron, 2017. Answer to « Tkinter Toplevel always in front ». *Stack Overflow*. [en ligne]. 20 juillet 2017. [Consulté le 9 juin 2022]. Disponible à l'adresse: <https://stackoverflow.com/a/45215220>

OAKLEY, Bryan, 2022. Answer to « How can i fit my tkinter app to any size screen? » *Stack Overflow*. [en ligne]. 11 janvier 2022. [Consulté le 22 août 2022]. Disponible à l'adresse: <https://stackoverflow.com/a/70669697>

openpyxl - A Python library to read/write Excel 2010 xlsx/xlsm files — openpyxl 3.0.10 documentation, [sans date]. [en ligne]. [Consulté le 19 juin 2022]. Disponible à l'adresse: <https://openpyxl.readthedocs.io/en/stable/>

Packaging · TomSchimansky/CustomTkinter Wiki, [sans date]. *GitHub*. [en ligne]. [Consulté le 5 août 2022]. Disponible à l'adresse: <https://github.com/TomSchimansky/CustomTkinter>

PAUL, 2013. Answer to « Split string with multiple delimiters in Python ». *Stack Overflow*. [en ligne]. 9 janvier 2013. [Consulté le 14 juillet 2022]. Disponible à l'adresse: <https://stackoverflow.com/a/14233023>

Python Imaging Library (PIL) — Documentation Bibliothèques Python 1.0.0, [sans date]. [en ligne]. [Consulté le 18 août 2022]. Disponible à l'adresse: <https://he-arc.github.io/livre-python/pillow/index.html>

Python (langage), 2022. *Wikipédia*. [en ligne]. [Consulté le 14 août 2022]. Disponible à l'adresse: [https://fr.wikipedia.org/w/index.php?title=Python_\(langage\)&oldid=195831595](https://fr.wikipedia.org/w/index.php?title=Python_(langage)&oldid=195831595)Page Version ID: 195831595

PYTHON, Real, [sans date]. Reading and Writing CSV Files in Python – Real Python. [en ligne]. [Consulté le 4 août 2022]. Disponible à l'adresse: <https://realpython.com/python-csv/>

python-igraph, [sans date]. [en ligne]. [Consulté le 19 juin 2022]. Disponible à l'adresse: <https://igraph.org/python/>

RATNAKUMAR, Sridhar, 2009a. Answer to « Tk treeview column sort ». *Stack Overflow*. [en ligne]. 28 décembre 2009. [Consulté le 18 août 2022]. Disponible à l'adresse: <https://stackoverflow.com/a/1967793>

RATNAKUMAR, Sridhar, 2009b. Tk treeview column sort. *Stack Overflow*. [en ligne]. 27 décembre 2009. [Consulté le 18 août 2022]. Disponible à l'adresse: <https://stackoverflow.com/q/1966929>

RESEARCH, Daniel Ellis, 2021. Simple Documentation Generation in Python using Pdoc. *CEMAC*. [en ligne]. 25 mai 2021. [Consulté le 4 août 2022]. Disponible à l'adresse: <https://medium.com/cemac/simple-documentation-generation-in-python-using-pdoc-16fb86eb5cd5>

RITZEL, Jochen, 2011. Answer to « Split string with multiple delimiters in Python ». *Stack Overflow*. [en ligne]. 14 février 2011. [Consulté le 14 juillet 2022]. Disponible à l'adresse: <https://stackoverflow.com/a/4998687>

SHISHERE, 2020. Answer to « Tk treeview column sort ». *Stack Overflow*. [en ligne]. 19 août 2020. [Consulté le 18 août 2022]. Disponible à l'adresse: <https://stackoverflow.com/a/63491566>

SKITZAFREAK, 2017. Tkinter Toplevel always in front. *Stack Overflow*. [en ligne]. 20 juillet 2017. [Consulté le 22 août 2022]. Disponible à l'adresse: <https://stackoverflow.com/q/45214662>

STYLE, Graphic, 2020. Signification des couleurs pour votre marque : comment les utiliser ? [en ligne]. 7 mai 2020. [Consulté le 18 août 2022]. Disponible à l'adresse: <https://www.graphicstyle.fr/communication/signification-des-couleurs/>

THOM, Pete, 2021. Answer to « Draw.io - how to export all tabs to images using command line ». *Stack Overflow*. [en ligne]. 15 juillet 2021. [Consulté le 6 août 2022]. Disponible à l'adresse: <https://stackoverflow.com/a/68390549>

TkDocs Tutorial - Styles and Themes, [sans date]. [en ligne]. [Consulté le 30 juin 2022]. Disponible à l'adresse: <https://tkdocs.com/tutorial/styles.html>

tkinter — Interface Python pour Tcl/Tk — Documentation Python 3.10.6, [sans date]. [en ligne]. [Consulté le 17 juin 2022]. Disponible à l'adresse: <https://docs.python.org/fr/3/library/tkinter.html>

Tkinter Grid Geometry Manager, [sans date]. *Python Tutorial - Master Python Programming For Beginners from Scratch*. [en ligne]. [Consulté le 22 juillet 2022]. Disponible à l'adresse: <https://www.pythontutorial.net/tkinter/tkinter-grid/>

tkinter.messagebox — Tkinter message prompts — Python 3.10.6 documentation, [sans date]. [en ligne]. [Consulté le 6 juillet 2022]. Disponible à l'adresse: <https://docs.python.org/3/library/tkinter.messagebox.html>

tkinter.ttk — Tk themed widgets — Python 3.10.6 documentation, [sans date]. [en ligne]. [Consulté le 5 juillet 2022]. Disponible à l'adresse: <https://docs.python.org/3/library/tkinter.ttk.html#tkinter.ttk.Treeview>

Treeview Update Records - Python Tkinter GUI Tutorial #121, 2020. [en ligne]. [Consulté le 3 juillet 2022]. Disponible à l'adresse: https://www.youtube.com/watch?v=IKiNlSS_cms

TRUONG, Kien, 2012. Answer to « Executing Javascript from Python ». *Stack Overflow*. [en ligne]. 13 avril 2012. [Consulté le 25 juin 2022]. Disponible à l'adresse: <https://stackoverflow.com/a/10136615>

Validation in entry widget in tkinter | Python, 2018. [en ligne]. [Consulté le 5 juillet 2022]. Disponible à l'adresse: <https://www.youtube.com/watch?v=oRYshQCOHOs>

VARUN, [sans date]. Python : Count elements in a list that satisfy certain conditions — thisPointer. [en ligne]. [Consulté le 13 juillet 2022]. Disponible à l'adresse: <https://thispointer.com/python-count-elements-in-a-list-that-satisfy-certain-conditions/>

Annexe 1 : Manuel d'utilisation



Créateur de schéma réseau

Préambule

Ce manuel permet à l'utilisateur·rice de comprendre le fonctionnement et les fonctionnalités de l'application et est écrit avec un langage épïcène³.

Table des matières

Importation d'un fichier csv	39
Ajout d'un lien	41
Modification d'un lien	43
Suppression d'un lien.....	45
Enregistrement du diagramme	46
Modification du diagramme sur le fichier *.drawio	48
Exportation sous un format spécifique (png, pdf, ...).	49

³ Le langage épïcène utilise différentes règles et pratiques cherchant à éviter toute discrimination sexiste.
Exemple : utilisateur·rice, auteur·e, ceux·elles

Importation d'un fichier csv

Un fichier csv peut être importé dans le format semblable au tableau ci-dessous séparé par des « ; » :

Exemple

Type de machine	Nom de la machine	Adresse réseau	Interface	Liaison
Routeur 1921	Routeur-1	10.136.8.0/26	ge0/0	PC1
PC	PC1	10.136.8.0/26	Intel	Routeur-1
...

⚠ La création du csv doit être correctement établie au risque d'obtenir des erreurs lors de l'exécution du schéma.

 Créateur de schéma réseau

— □ ×



Importer un fichier

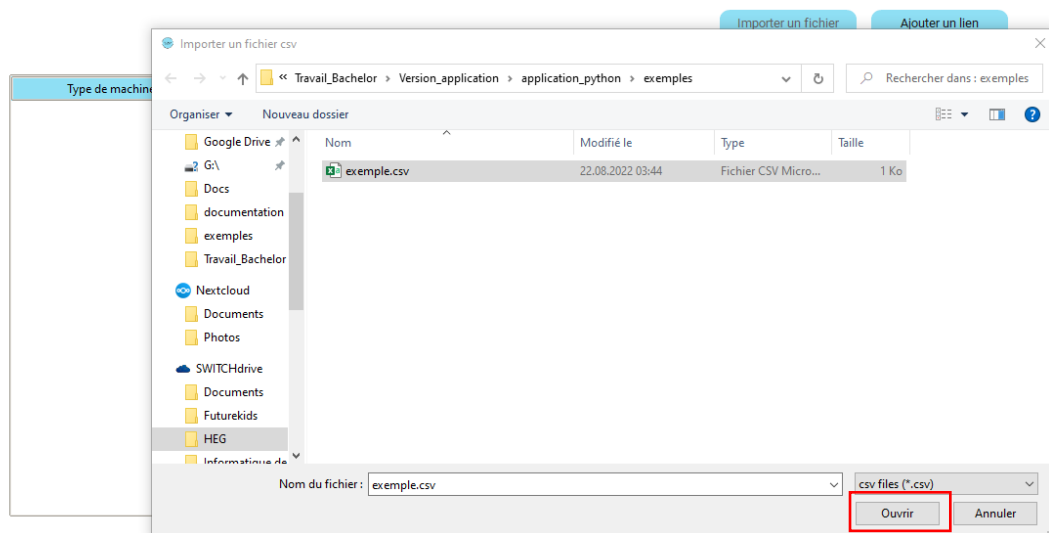
Ajouter un lien

Type de machine	Nom de la machine	Adresse réseau	Interface	Liaison
-----------------	-------------------	----------------	-----------	---------

Une seconde s'ouvre, afin de choisir le fichier à importer doté d'une extension .csv. En cliquant sur le bouton « Ouvrir », les données sont chargées et insérées dans l'application lorsque celles-ci ont été correctement faites.

Créateur de schéma réseau

— □ ×



Si les adresses sont sous le protocole EIGRP 65100, le nom de la salle apparaît. Le résultat de l'importation est la suivante :

Créateur de schéma réseau

— □ ×



Vous êtes dans la salle FR31

Ajouter un lien

Type de machine	Nom de la machine	Adresse réseau	Interface	Liaison
Routeur 1921	R1	10.136.0.0/22	ge0/0	EIGRP 65100
Routeur 1921	R1	10.136.8.0/26	ge0/1	S2
PC	PC1	10.136.8.0/26	Realtek	S2
Commutateur 3660	S2	10.136.8.0/26	fa0/1	R1
Commutateur 3660	S2	10.136.8.0/26	fa0/2	PC1
Commutateur 2960	S1	10.136.9.0/26	fa0/1	R1
PC	PC1	10.136.9.0/26	Intel	S1
Routeur 1921	R1	10.136.9.0/26	fe0/1/0	S1
Commutateur 2960	S1	10.136.9.0/26	fa0/2	PC1
Routeur 1841	R2	10.136.10.0/25	serial0/1/0	

Créer le schéma

Ajout d'un lien

L'application offre également la possibilité d'ajouter un lien qui doit être **bidirectionnel** pour son bon fonctionnement.

Il faudra tout d'abord cliquer sur le bouton « Ajouter un lien », puis une seconde fenêtre s'ouvre.

Créateur de schéma réseau

Schéma de réseau
CISCO

Vous êtes dans la salle FR31

Ajouter un lien

Type de machine	Nom de la machine	Adresse réseau	Interface	Liaison
Routeur 1921	R1	10.136.0.0/22	ge0/0	EIGRP 65100
Routeur 1921	R1	10.136.8.0/26	ge0/1	S2
PC	PC1	10.136.8.0/26	Realtek	S2
Commutateur 3660	S2	10.136.8.0/26	fa0/1	R1
Commutateur 3660	S2	10.136.8.0/26	fa0/2	PC1
Commutateur 2960	S1	10.136.9.0/26	fa0/1	R1
PC	PC1	10.136.9.0/26	Intel	S1
Routeur 1921	R1	10.136.9.0/26	fe0/1/0	S1
Commutateur 2960	S1	10.136.9.0/26	fa0/2	PC1
Routeur 1841	R2	10.136.10.0/25	serial0/1/0	

Créer le schéma

Après avoir rempli chaque champ, une vérification sera effectuée avant l'ajout dans le tableau de lien sur la fenêtre principale dès que l'utilisateur-e cliquera sur le bouton « Ajouter ».

Créateur de schéma réseau

Schéma de réseau
CISCO

Vous êtes dans la salle FR31

Ajouter un lien

Ajouter
UN LIEN

Machine: Routeur 1841

Nom de la machine: R3

Adresse de réseau / préfixe
Préfixe compris entre 22 et 30
ex: 10.136.8.0/26

10.136.10.0/25

Liée à la machine: R2

Interface: serial0/1/0

Ajouter

La ligne a été ajoutée :

⚠ Ne pas oublier de définir le lien de manière bidirectionnel, afin que le diagramme soit cohérent.

Si l'utilisateur ajoute une machine avec l'adresse 10.136.0.0/22, ce dernier sera automatiquement lié avec l'EIGRP 65100.

Créateur de schéma réseau

— □ ×



Vous êtes dans la salle FR31

Ajouter un lien

Type de machine	Nom de la machine	Adresse réseau	Interface	Liaison
Routeur 1921	R1	10.136.8.0/26	ge0/1	S2
PC	PC1	10.136.8.0/26	Realtek	S2
Commutateur 3660	S2	10.136.8.0/26	fa0/1	R1
Commutateur 3660	S2	10.136.8.0/26	fa0/2	PC1
Commutateur 2960	S1	10.136.9.0/26	fa0/1	R1
PC	PC1	10.136.9.0/26	Intel	S1
Routeur 1921	R1	10.136.9.0/26	fe0/1/0	S1
Commutateur 2960	S1	10.136.9.0/26	fa0/2	PC1
Routeur 1841	R2	10.136.10.0/25	serial0/1/0	
Routeur 1841	R3	10.136.10.0/25	serial0/1/0	R2

Créer le schéma

Modification d'un lien

Lorsqu'un lien est sélectionné dans le tableau, deux nouveaux boutons apparaissent : « Supprimer un lien » et « Modifier un lien ».

Lorsque l'utilisateur·rice cliquera sur le bouton « Modifier un lien », une seconde fenêtre s'ouvrira, afin de permettre à celui-elle-ci de modifier le ou les champs voulus.

Créateur de schéma réseau

Schéma de réseau
CISCO

Vous êtes dans la salle FR31

Supprimer un lien **Modifier un lien** Ajouter un lien

Type de machine	Nom de la machine	Adresse réseau	Interface	Liaison
Routeur 1921	R1	10.136.8.0/26	ge0/1	S2
PC	PC1	10.136.8.0/26	Realtek	S2
Commutateur 3660	S2	10.136.8.0/26	fa0/1	R1
Commutateur 3660	S2	10.136.8.0/26	fa0/2	PC1
Commutateur 2960	S1	10.136.9.0/26	fa0/1	R1
PC	PC1	10.136.9.0/26	Intel	S1
Routeur 1921	R1	10.136.9.0/26	fe0/1/0	S1
Commutateur 2960	S1	10.136.9.0/26	fa0/2	PC1
Routeur 1841	R2	10.136.10.0/25	serial0/1/0	
Routeur 1841	R3	10.136.10.0/25	serial0/1/0	R2

Créer le schéma

La validation des champs s'effectue, lorsque l'utilisateur·rice clique sur le bouton « Modifier ».

Créateur de schéma réseau

Schéma de réseau
CISCO

Vous êtes dans la salle FR31

Modifier un lien

Modifier UN LIEN

Machine: Routeur 1841 Nom de la machine: R2

Adresse de réseau / préfixe: 10.136.10.0/25 Liée à la machine: R3

Interface: serial0/1/0

Modifier

Le résultat de la modification est comme ceci :

⚠ Ne pas oublier que lorsque l'adresse de réseau est modifiée, le lien devrait également être modifier, afin de garder la cohérence du diagramme.

Créateur de schéma réseau

— □ ×



Vous êtes dans la salle FR31

Supprimer un lien

Modifier un lien

Ajouter un lien

Type de machine	Nom de la machine	Adresse réseau	Interface	Liaison
Routeur 1921	R1	10.136.8.0/26	ge0/1	S2
PC	PC1	10.136.8.0/26	Realtek	S2
Commutateur 3660	S2	10.136.8.0/26	fa0/1	R1
Commutateur 3660	S2	10.136.8.0/26	fa0/2	PC1
Commutateur 2960	S1	10.136.9.0/26	fa0/1	R1
PC	PC1	10.136.9.0/26	Intel	S1
Routeur 1921	R1	10.136.9.0/26	fe0/1/0	S1
Commutateur 2960	S1	10.136.9.0/26	fa0/2	PC1
Routeur 1841	R2	10.136.10.0/25	serial0/1/0	R3
Routeur 1841	R3	10.136.10.0/25	serial0/1/0	R2

Créer le schéma

Suppression d'un lien

Lorsqu'une ligne du tableau et le bouton « Supprimer un lien » sont sélectionnés, la ligne disparaît, ainsi que ses possibles liaisons.

⚠ Ne pas oublier que lorsqu'un lien est supprimé, les machines liées devront également être supprimées ou modifiées, afin de garder la cohérence du diagramme.

Créateur de schéma réseau

— □ ×

Schéma de réseau

CISCO

Vous êtes dans la salle FR31

Supprimer un lien

Modifier un lien

Ajouter un lien

Type de machine	Nom de la machine	Adresse réseau	Interface	Liaison
Routeur 1921	R1	10.136.8.0/26	ge0/1	S2
PC	PC1	10.136.8.0/26	Realtek	S2
Commutateur 3660	S2	10.136.8.0/26	fa0/1	R1
Commutateur 3660	S2	10.136.8.0/26	fa0/2	PC1
Commutateur 2960	S1	10.136.9.0/26	fa0/1	R1
PC	PC1	10.136.9.0/26	Intel	S1
Routeur 1921	R1	10.136.9.0/26	fe0/1/0	S1
Commutateur 2960	S1	10.136.9.0/26	fa0/2	PC1
Routeur 1841	R2	10.136.10.0/25	serial0/1/0	R3
Routeur 1841	R3	10.136.10.0/25	serial0/1/0	R2

Créer le schéma

En supprimant les éléments, le tableau ressemble à ceci :

Créateur de schéma réseau

— □ ×

Schéma de réseau

CISCO

Vous êtes dans la salle FR31

Ajouter un lien

Type de machine	Nom de la machine	Adresse réseau	Interface	Liaison
Routeur 1921	R1	10.136.0.0/22	ge0/0	EIGRP 65100
Routeur 1921	R1	10.136.8.0/26	ge0/1	S2
PC	PC1	10.136.8.0/26	Realtek	S2
Commutateur 3660	S2	10.136.8.0/26	fa0/1	R1
Commutateur 3660	S2	10.136.8.0/26	fa0/2	PC1
Commutateur 2960	S1	10.136.9.0/26	fa0/1	R1
PC	PC1	10.136.9.0/26	Intel	S1
Routeur 1921	R1	10.136.9.0/26	fe0/1/0	S1
Commutateur 2960	S1	10.136.9.0/26	fa0/2	PC1
Routeur 1841	R3	10.136.10.0/25	serial0/1/0	

Créer le schéma

Enregistrement du diagramme

Lorsque l'utilisateur-e est satisfait des liaisons établies, celui-elle-ci pourra créer son schéma en cliquant sur le bouton « Créer le schéma ».

Créateur de schéma réseau

Schéma de réseau

CISCO

Vous êtes dans la salle FR31

Ajouter un lien

Type de machine	Nom de la machine	Adresse réseau	Interface	Liaison
Routeur 1921	R1	10.136.0.0/22	ge0/0	EIGRP 65100
Routeur 1841	R3	10.136.0.0/22	fe0/0/1	EIGRP 65100
Routeur 1841	R2	10.136.10.0/25	serial0/1/0	R3
Routeur 1841	R3	10.136.10.0/25	serial0/1/0	R2
Routeur 1921	R1	10.136.8.0/26	ge0/1	S2
PC	PC1	10.136.8.0/26	Realtek	S2
Commutateur 3660	S2	10.136.8.0/26	fa0/1	R1
Commutateur 3660	S2	10.136.8.0/26	fa0/2	PC1
Commutateur 2960	S1	10.136.9.0/26	fa0/1	R1
PC	PC1	10.136.9.0/26	Intel	S1

Créer le schéma

II-Elle choisira le lieu et le nom de l'enregistrement du diagramme avec l'extension .drawio, ainsi que le tableau des données insérées sous un fichier avec l'extension .csv, puis l'enregistrera en cliquant sur « Enregistrer »

Créateur de schéma réseau

Schéma de réseau

CISCO

Vous êtes dans la salle FR31

Ajouter un lien

Choisissez le répertoire voulu

« Travail_Bachelor » > Version_application > application_python > exemples

Rechercher dans : exemples

Aucun élément ne correspond à votre recherche.

Nom du fichier : exemple

Type : drawio files (*.drawio)

Enregistrer Annuler

Un message avertira l'utilisateur·e du chemin où se trouve les différents fichiers :

Créateur de schéma réseau

— □ ×



Vous êtes dans la salle FR31

Ajouter un lien

Type de machine	Nom de la machine	Adresse réseau	Interface	Liaison
Routeur 1921	R1	10.136.0.0/22	ge0/0	EIGRP 65100
Routeur 1841	R3	10.136.0.0/22	fe0/0/1	EIGRP 65100
Routeur 1841	R2	10.136.10.0/25	serial0/1/0	R3
Routeur 1841	R3	10.136.10.0/25	serial0/1/0	R2
Routeur 1921	R1	10.136.8.0/26	ge0/1	S2
PC	PC1	10.136.8.0/26	Realtek	S2
Commutateur 3660	S2	10.136.8.0/26	fa0/1	R1
Commutateur 3660	S2	10.136.8.0/26	fa0/2	PC1
Commutateur 2960	S1	10.136.9.0/26	fa0/1	R1
PC	PC1	10.136.9.0/26	Intel	S1

Créer le schéma

À l'emplacement, les deux fichiers sont bien enregistrés.

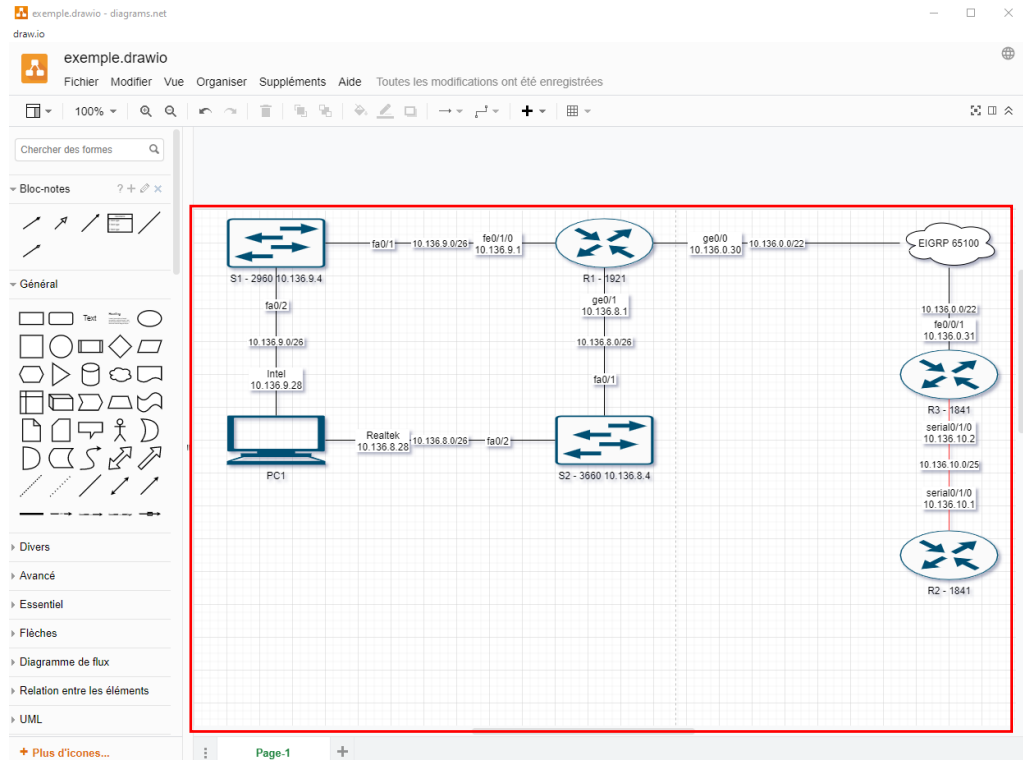
Nom

exemple.csv
exemple.drawio

```
Type de machine;Nom de la machine;Adresse réseau;Interface;Liaison
Routeur 1921;R1;10.136.0.0/22;ge0/0;EIGRP 65100
Routeur 1841;R3;10.136.0.0/22;fe0/0/1;EIGRP 65100
Routeur 1841;R2;10.136.10.0/25;serial0/1/0;R3
Routeur 1841;R3;10.136.10.0/25;serial0/1/0;R2
Routeur 1921;R1;10.136.8.0/26;ge0/1;S2
PC;PC1;10.136.8.0/26;Realtek;S2
Commutateur 3660;S2;10.136.8.0/26;fa0/1;R1
Commutateur 3660;S2;10.136.8.0/26;fa0/2;PC1
Commutateur 2960;S1;10.136.9.0/26;fa0/1;R1
PC;PC1;10.136.9.0/26;Intel;S1
Routeur 1921;R1;10.136.9.0/26;fe0/1/0;S1
Commutateur 2960;S1;10.136.9.0/26;fa0/2;PC1
```

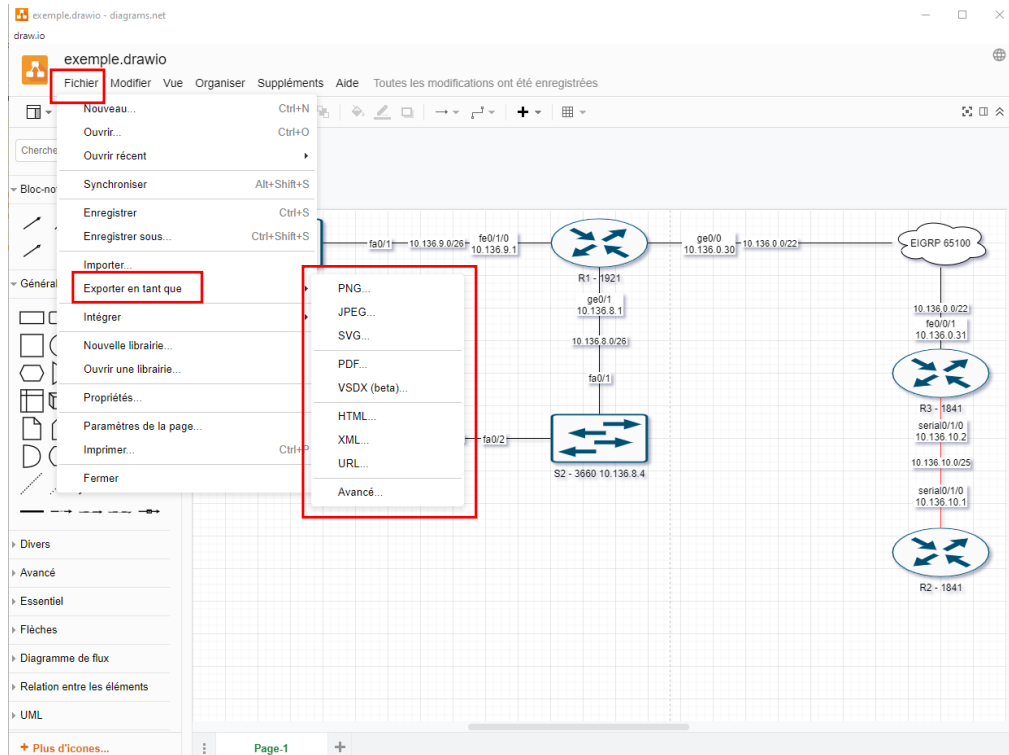
Modification du diagramme sur le fichier *.drawio

Il est possible si le diagramme est illisible de modifier l'emplacement de chaque machine.

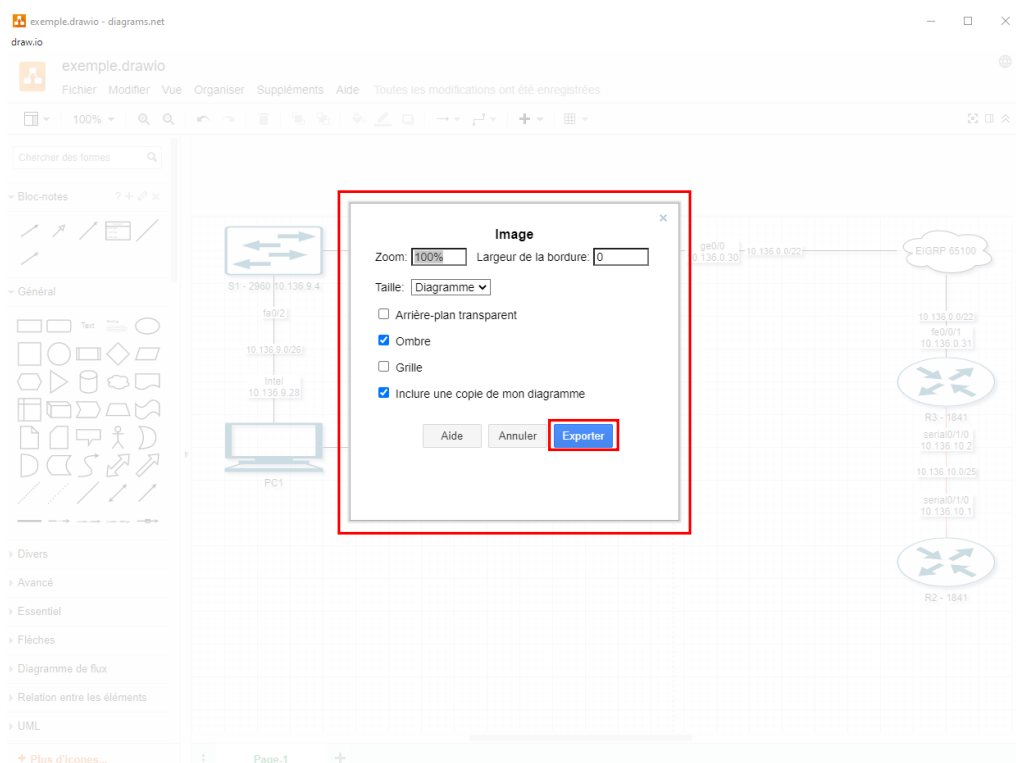


Exportation sous un format spécifique (png, pdf, ...)

Il est possible d'exporter le diagramme sous un autre format, afin de l'insérer dans un fichier, par exemple. Pour ce faire, l'utilisateur-e cliquera dans l'onglet « Fichier », puis « Exporter en tant que » et finalement sous le format voulu. Dans cet exemple, le format png sera démontré.



Il est possible de choisir des options avant l'exportation.



Il sera demandé à l'utilisateur.e le lieu de l'enregistrement et le nom du fichier.

