

Simulation réseau en SNT et NSI

Introduction

On retrouve la notion de réseau à plusieurs reprises dans le programme de SNT ou dans les programmes de NSI. Tout ce qui touche au réseau peut être relativement difficile à enseigner. En effet, il faut prendre garde à ne pas tomber dans la simple évocation du contenu des différents protocoles (TCP, IP, HTTP...).

Pour rendre ces séquences sur le réseau un peu plus “vivantes”, il est possible de faire “manipuler” les élèves. Malheureusement, pour des questions légitimes de sécurité, les réseaux des établissements sont dans la plupart des cas très verrouillés et empêchent toutes manipulations sortant un peu de l’ordinaire. L’utilisation d’un logiciel de simulation peut permettre aux élèves de tout de même mettre en pratique les notions abordées.

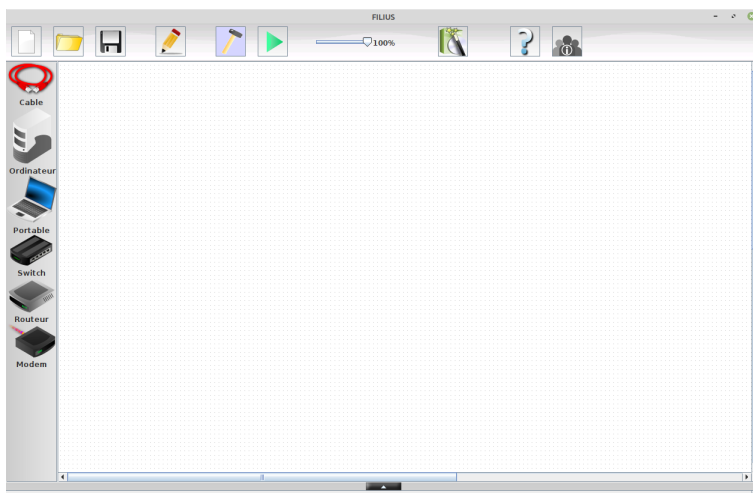
Il existe beaucoup de logiciels de simulation de réseau informatique. Voici une liste non exhaustive des logiciels que l’on peut trouver :

- CERTA
- GNS3
- packet tracer de CISCO
- Filius
- ...

Certains des logiciels évoqués ci-dessus sont extrêmement complets, mais un peu difficiles à prendre en main pour des élèves de seconde. D’autres sont faciles à prendre en main, mais trop simplistes. Un de ces logiciels offre un bon compromis “facilité de prise en main” / “réalisme” : Filius.

Présentation et prise en main de Filius

Filius est un logiciel pour Linux et Windows (il est aussi possible de le faire fonctionner sous Mac OS assez simplement), libre et gratuit, développé par Stefan Freischlad. Vous pouvez le télécharger sur le site de l’auteur (<https://www.lernsoftware-Filius.de/Herunterladen>).



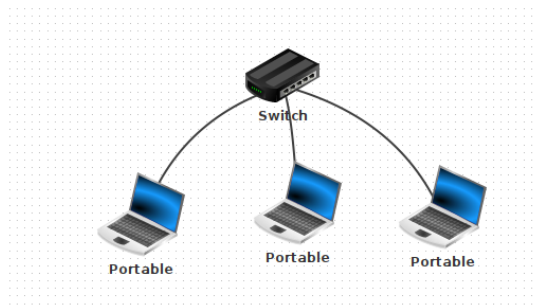
interface de Filius

L'interface de Filius se divise en deux grandes parties : une partie édition et une partie simulation.

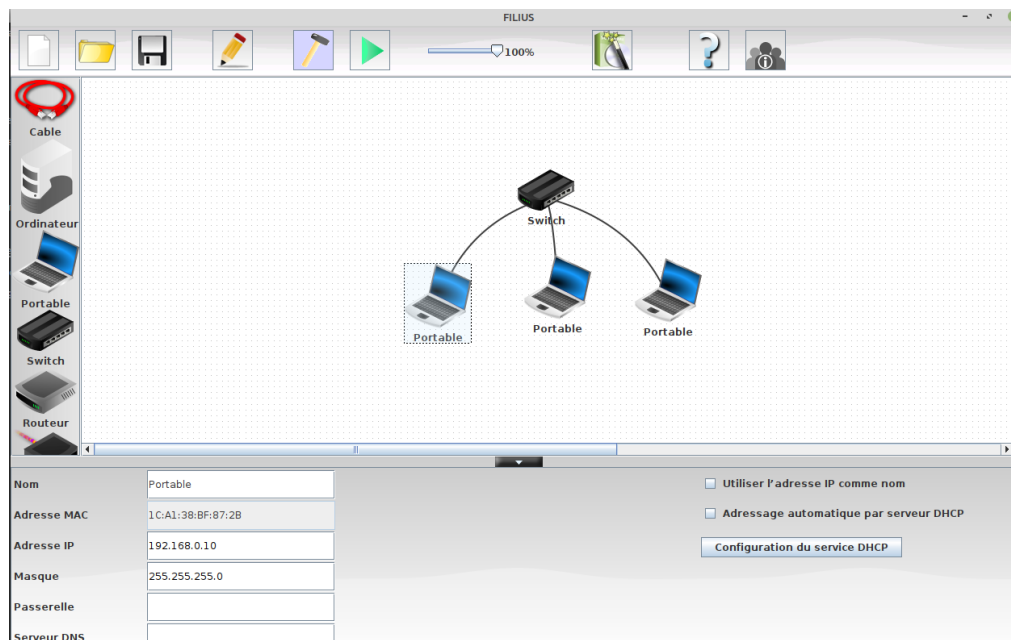
La partie édition permet de créer son propre réseau à partir des éléments suivants :

- Câble
- Ordinateur (qui joue souvent le rôle de serveur)
- Portable (qui joue souvent le rôle de client)
- Switch
- Routeur
- Modem

On peut très rapidement créer un réseau simple en disposant les différents éléments par “glisser-déposer” et en reliant ces éléments avec des câbles :

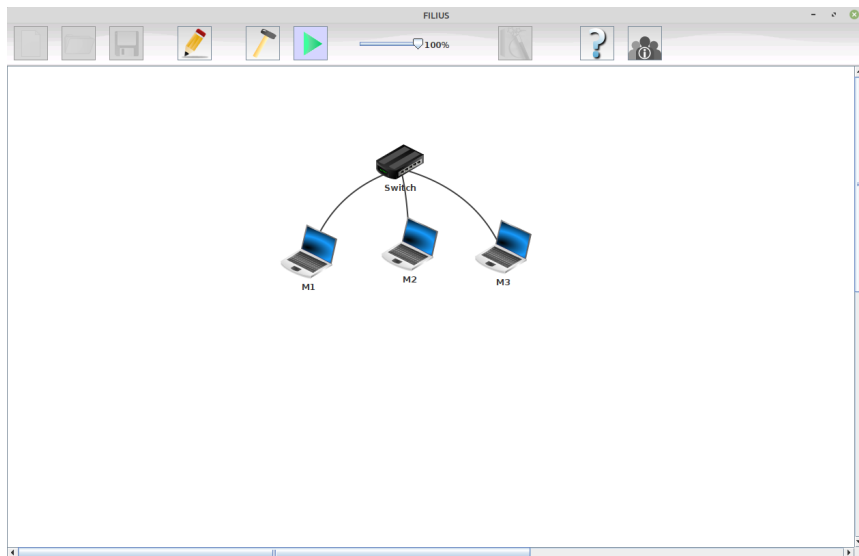


Une fois les éléments du réseau en place, il est nécessaire de configurer les noms, les adresses IP... en cliquant sur les différentes machines :



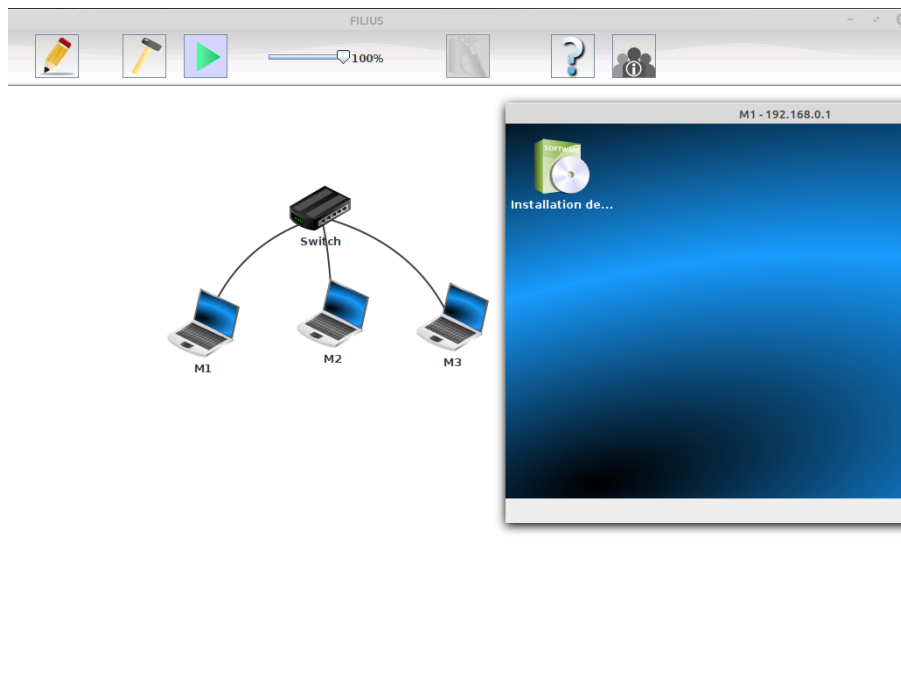
configuration (nom, adresse IP, masque....) d'une machine

Une fois la configuration terminée, il est possible de quitter le mode "conception" pour passer en mode "simulation" (Ctrl + R).

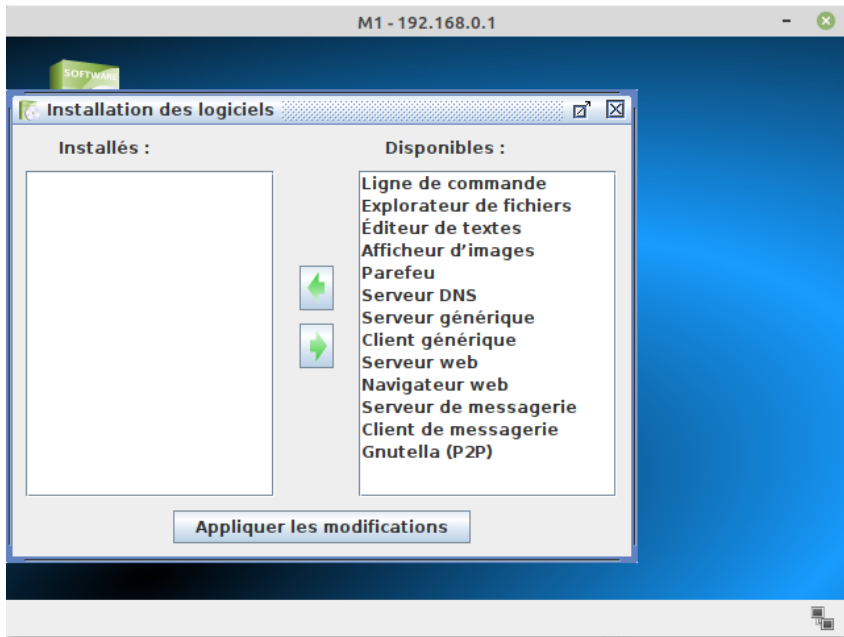


Filius en mode simulation

Dans le mode simulation, il est possible, en cliquant sur une machine, d'ouvrir une fenêtre qui va vous permettre d'installer différents logiciels :



Les logiciels "installables" sont relativement nombreux :



Par exemple, il est possible d'installer la "Ligne de commande" sur une machine et de saisir la commande "ipconfig" afin d'obtenir des informations sur la configuration réseau de la machine.



Ou bien encore il est possible d'effectuer un ping vers une autre machine :

```
Ligne de commande
move / mv      déplace/renomme un fichier
cat / type    affiche le contenu d'un fichier
del / rm      supprime une fichier ou un dossier
mkdir         crée un dossier
cd            change le dossier courant
pwd          affiche le chemin du dossier courant
dir / ls     liste le contenu du dossier courant
ipconfig     affiche les paramètres du réseau
netstat      affiche la liste des connexions en cours
arp          affiche la table (ARP) de résolution d'adresses
host         résout un nom d'hôte en adresse IP
route        affiche la table de routage
ping         teste la connexion avec un autre ordinateur
tracert      analyse les sauts nécessaires pour atteindre une destination
exit        quitte la ligne de commande

=====

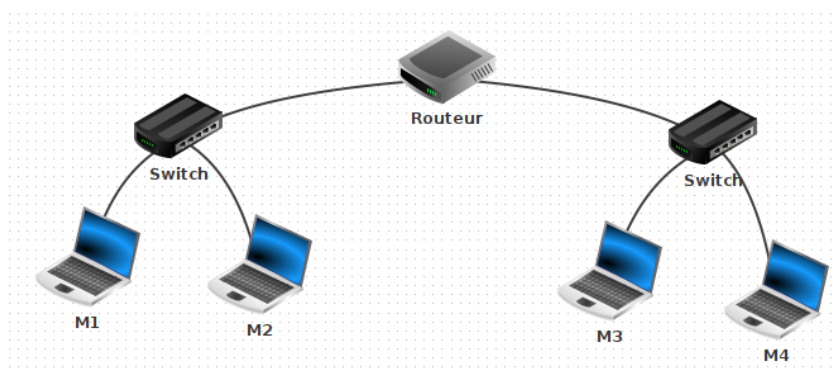
/> ping 192.168.0.1
PING 192.168.0.1 (192.168.0.1)
From 192.168.0.1 (192.168.0.1): icmp_seq=1 ttl=64 time=436ms
From 192.168.0.1 (192.168.0.1): icmp_seq=2 ttl=64 time=209ms
From 192.168.0.1 (192.168.0.1): icmp_seq=3 ttl=64 time=208ms
From 192.168.0.1 (192.168.0.1): icmp_seq=4 ttl=64 time=208ms
--- 192.168.0.1 Statistiques des paquets ---
4 paquets transmis, 4 paquets reçus, 0% paquets perdus

/>
```

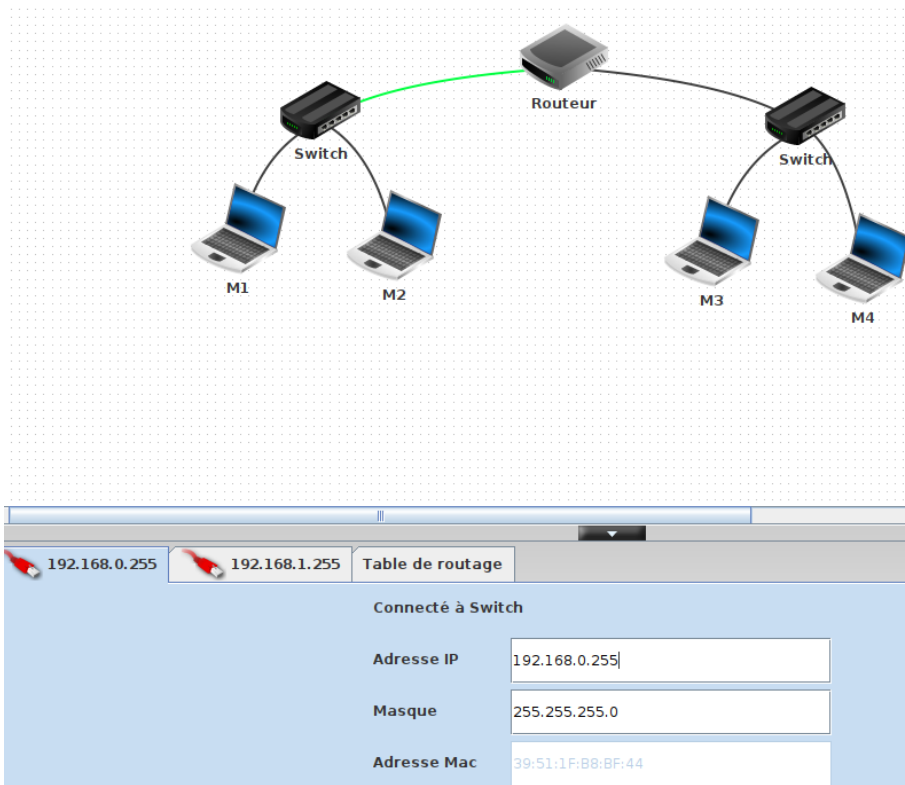
Exemples d'activités possibles en SNT et en NSI

Activités sur le routage (SNT et NSI terminale)

Au-delà des exemples de réseaux relativement simples (switch + quelques machines) déjà vu ci-dessus, il est possible avec Filius de mettre en place des routeurs pour les communications inter-réseaux locaux (activités possibles en SNT et en terminale NSI)

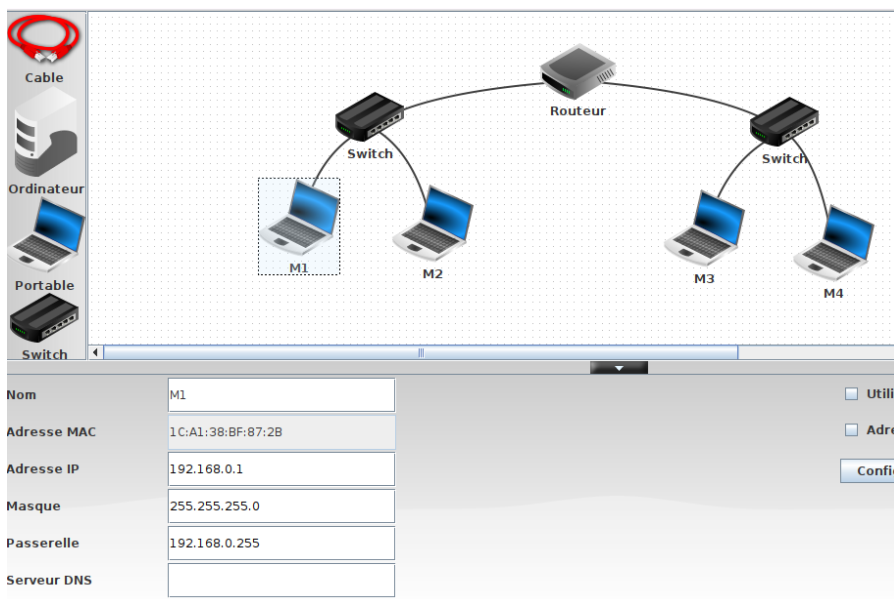


Après avoir configuré les interfaces réseau du routeur :



configuration du routeur

il est nécessaire de renseigner l'adresse de la passerelle (adresse du routeur) pour chaque machine :



configuration des clients (adresse de la passerelle)

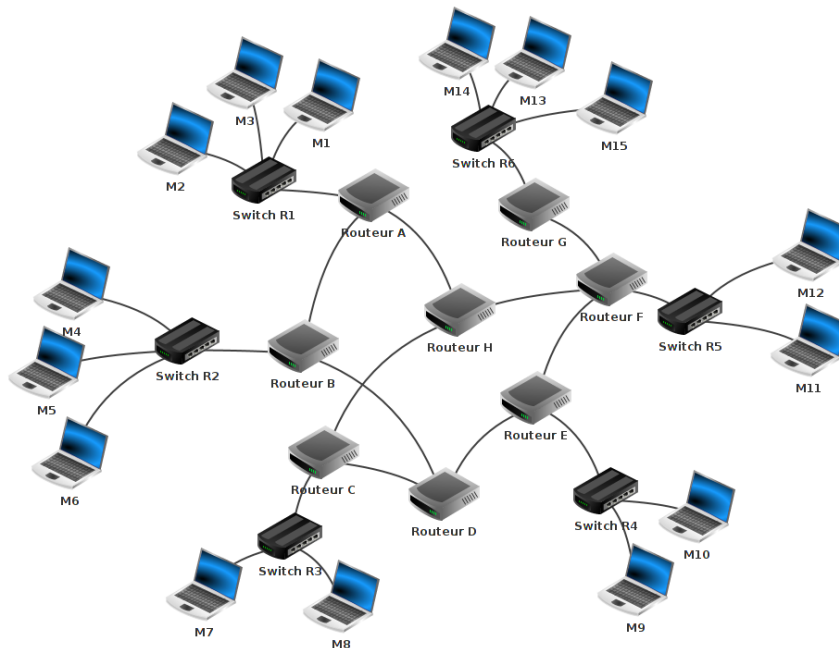
Une fois le routeur et les clients bien configurés, nous pouvons faire communiquer 2 machines situées dans 2 réseaux locaux différents (par exemple M1 et M3) à l'aide d'un *ping*. Nous pouvons ensuite utiliser la commande *tracroute* pour vérifier que les paquets de données passent bien par le routeur pour aller, par exemple de M1 à M3

```
> /> tracroute 192.168.1.1
Établissement de la connexion avec 192.168.1.1 (en 20 sauts max.).
 1  192.168.0.255
 2  192.168.1.1

192.168.1.1 a été atteint en 2 sauts.
/>
```

tracroute de M1 (192.168.0.1) à M3 (192.168.1.1) en passant par le routeur

Nous pouvons aussi faire travailler les élèves avec des réseaux plus complexes (composés de nombreux réseaux locaux et de plusieurs routeurs) :



Ce genre de réseau peut simuler un “mini-internet” (l’enseignant peut fournir aux élèves le réseau déjà configuré).

À partir du réseau ci-dessus, il est possible d’utiliser la commande *tracroute* afin d’analyser le parcours des paquets pour aller d’une machine à une autre. On peut aussi supprimer un ou plusieurs fils de connexion dans le but de simuler la panne d’un routeur (malheureusement Filius ne permet pas de simuler ce genre de pannes). Les élèves peuvent alors, grâce à un nouveau *tracroute*, constater que les paquets empruntent un autre chemin (si c’est possible) afin d’éviter le routeur en “panne”.

Toujours à propos du routage, on peut travailler directement sur les tables de routage (routage statique).

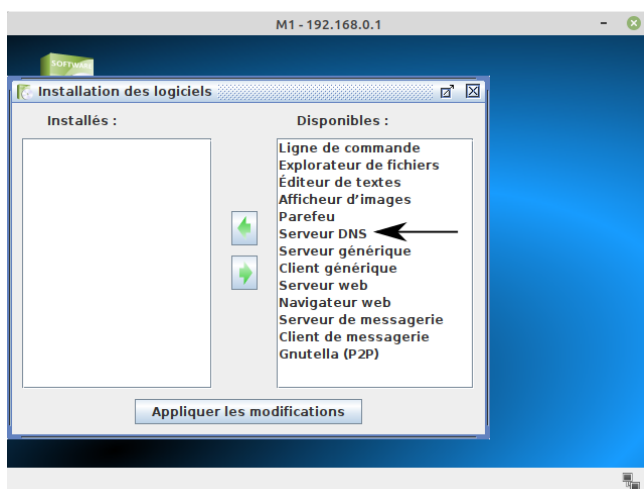
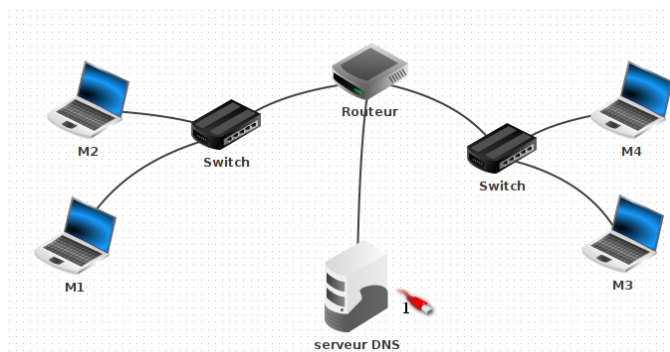
Général 192.168.0.255 192.168.1.255 Table de routage				
<input checked="" type="checkbox"/> Afficher toutes les lignes				
Nouvelle ligne		Supprimer la ligne sélectionnée		
Ouvrir dans une nouvelle fenêtre				
IP de destination	Masque	Passerelle suivante	Via l'interface	
192.168.1.255	255.255.255.255	127.0.0.1	127.0.0.1	
192.168.0.255	255.255.255.255	127.0.0.1	127.0.0.1	
192.168.1.0	255.255.255.0	192.168.1.255	192.168.1.255	
192.168.0.0	255.255.255.0	192.168.0.255	192.168.0.255	
127.0.0.0	255.0.0.0	127.0.0.1	127.0.0.1	

table de routage

Malheureusement, même si Filius permet d'établir automatiquement les tables de routage en cochant la case Routage automatique (routage dynamique) dans l'onglet Général de la page de configuration du routeur, il sera difficile de faire travailler les élèves de terminale sur les différents protocoles de routage à l'aide de Filius. En effet, le logiciel ne propose aucun paramétrage à ce niveau.

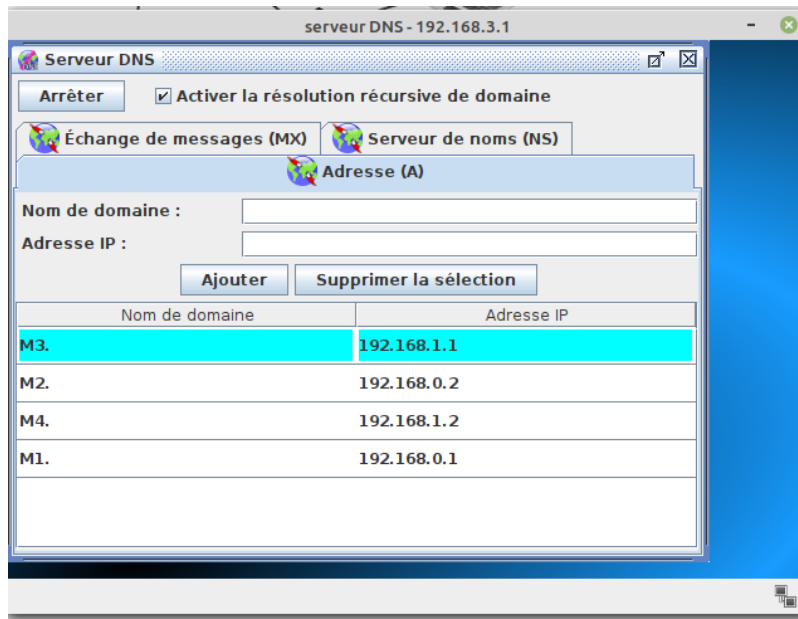
Mise en place d'un serveur DNS (SNT)

La notion de DNS est au programme de l'enseignement de SNT (module "Internet"). Filius propose la mise en place d'un serveur DNS simplifié.

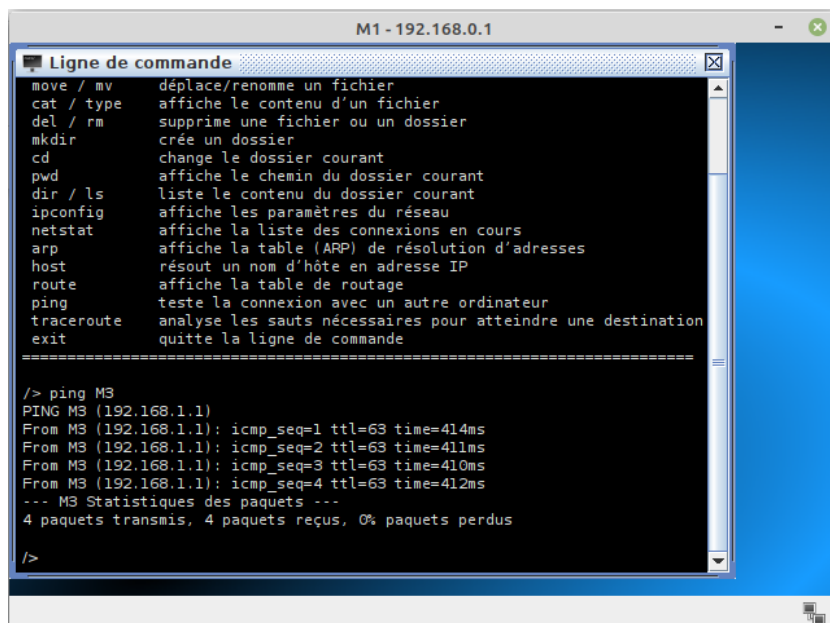


installation d'un serveur DNS

Une fois le serveur DNS correctement configuré,

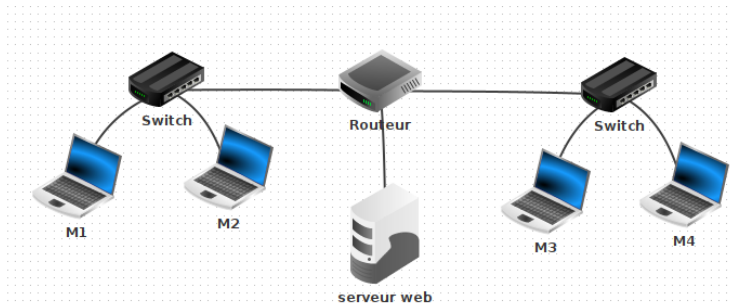


on peut demander aux élèves de vérifier qu'il est désormais possible d'utiliser les adresses symboliques à la place des adresses IP, par exemple pour l'utilisation d'un "ping" :

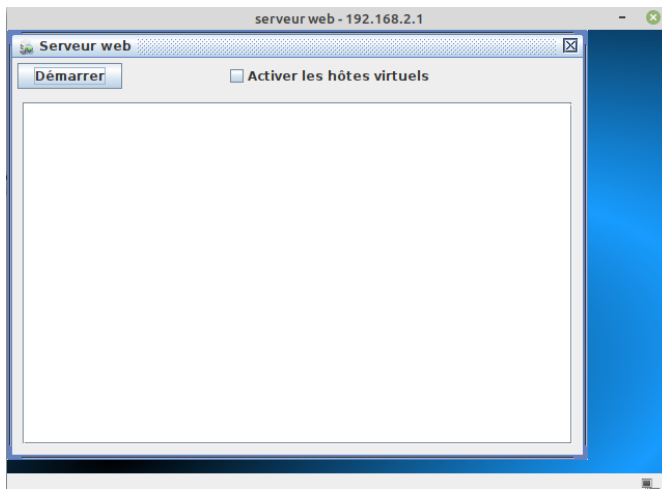


Mise en place d'un serveur HTTP (SNT et première NSI)

Comme pour le serveur DNS, Filius permet de mettre en place un serveur HTTP (serveur web). Ici aussi Filius va nous permettre de faire travailler les élèves sur beaucoup de notions étudiées en SNT et en NSI : HTML, CSS, client-serveur, serveur web, protocole HTTP...



On peut installer un “navigateur web” sur un client et un “serveur web” sur le serveur. Il suffit ensuite de démarrer le serveur web :



et d'utiliser le navigateur web du client en renseignant l'adresse IP du serveur dans la barre d'adresse du navigateur :

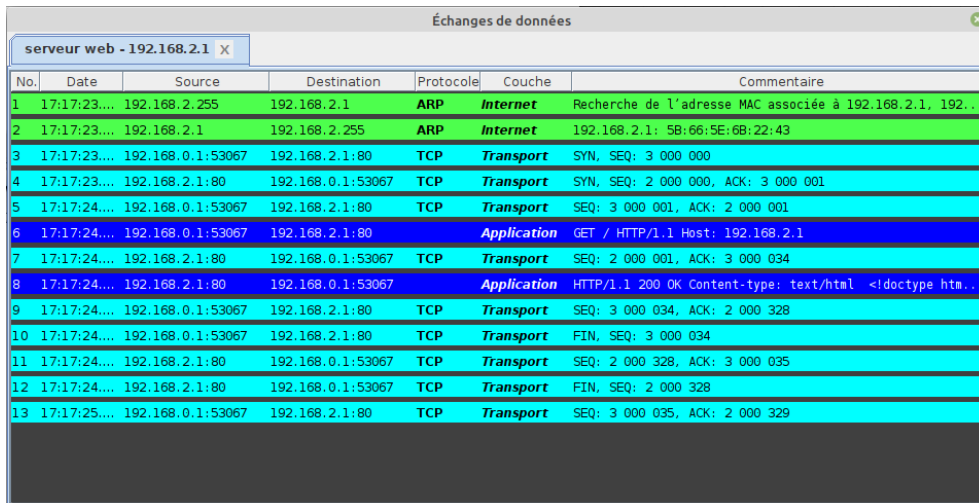


On peut aussi modifier le fichier *index.html* sur le serveur à l'aide de l'éditeur de texte et ainsi d'obtenir une page Web différente de la page proposée par défaut (travail sur HTML) :



Évidemment, il est possible d'ajouter un serveur DNS à notre réseau afin d'utiliser des adresses symboliques à la place des adresses IP dans la barre d'adresse du navigateur Web.

Filius propose aussi la possibilité d'analyser les requêtes et les réponses HTTP. Il suffit, depuis le mode simulation, de faire un clic droit sur le serveur et de choisir "Afficher les échanges de données", puis de faire une requête HTTP en utilisant le navigateur Web d'un client :



The screenshot shows a network traffic analysis window titled "Échanges de données". The window has a tab labeled "serveur web - 192.168.2.1". The main area contains a table with the following columns: No., Date, Source, Destination, Protocole, Couche, and Commentaire. The table lists 13 packets, with two highlighted in blue (lines 7 and 8).

No.	Date	Source	Destination	Protocole	Couche	Commentaire
1	17:17:23...	192.168.2.255	192.168.2.1	ARP	Internet	Recherche de l'adresse MAC associée à 192.168.2.1, 192...
2	17:17:23...	192.168.2.1	192.168.2.255	ARP	Internet	192.168.2.1: 58:66:5E:6B:22:43
3	17:17:23...	192.168.0.1:53067	192.168.2.1:80	TCP	Transport	SYN, SEQ: 3 000 000
4	17:17:23...	192.168.2.1:80	192.168.0.1:53067	TCP	Transport	SYN, SEQ: 2 000 000, ACK: 3 000 001
5	17:17:24...	192.168.0.1:53067	192.168.2.1:80	TCP	Transport	SEQ: 3 000 001, ACK: 2 000 001
6	17:17:24...	192.168.0.1:53067	192.168.2.1:80	Application	Application	GET / HTTP/1.1 Host: 192.168.2.1
7	17:17:24...	192.168.2.1:80	192.168.0.1:53067	TCP	Transport	SEQ: 2 000 001, ACK: 3 000 034
8	17:17:24...	192.168.2.1:80	192.168.0.1:53067	Application	Application	HTTP/1.1 200 OK Content-type: text/html <!doctype htm...
9	17:17:24...	192.168.0.1:53067	192.168.2.1:80	TCP	Transport	SEQ: 3 000 034, ACK: 2 000 328
10	17:17:24...	192.168.0.1:53067	192.168.2.1:80	TCP	Transport	FIN, SEQ: 3 000 034
11	17:17:24...	192.168.2.1:80	192.168.0.1:53067	TCP	Transport	SEQ: 2 000 328, ACK: 3 000 035
12	17:17:24...	192.168.2.1:80	192.168.0.1:53067	TCP	Transport	FIN, SEQ: 2 000 328
13	17:17:25...	192.168.0.1:53067	192.168.2.1:80	TCP	Transport	SEQ: 3 000 035, ACK: 2 000 329

Nous avons ci-dessus tous les échanges réseaux qui ont eu lieu entre le client et le serveur. On s'intéressera particulièrement aux 2 lignes bleu foncé qui représentent la requête HTTP (première ligne) et la réponse HTTP (deuxième ligne).

En cliquant sur la première ligne bleu foncé, on obtient la requête HTTP :

```
No. : 6 / Date: 17:17:24.300
Réseau
  Source: D3:50:8F:F3:5C:D2
  Destination: 5B:66:5E:6B:22:43
  Commentaire: 0x800
Internet
  Source: 192.168.0.1
  Destination: 192.168.2.1
  Protocole: IP
  Commentaire: Protocole :6, TTL: 63
Transport
  Source: 53067
  Destination: 80
  Protocole: TCP
  Commentaire: SEQ: 3 000 001
Application
  Commentaire: } requête HTTP
    GET / HTTP/1.1
    Host: 192.168.2.1
```

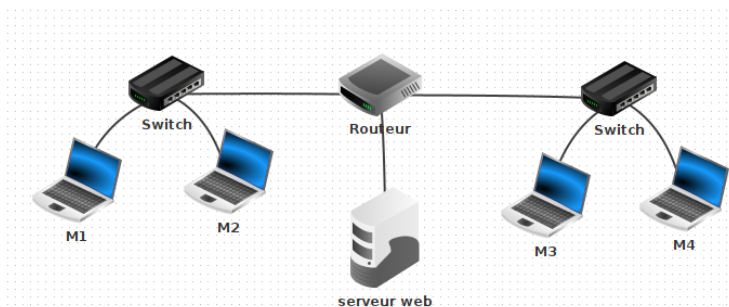
En cliquant sur la deuxième ligne bleu foncé, on obtient la réponse HTTP :

```
Application
  Commentaire: } Réponse HTTP
    HTTP/1.1 200 OK
    Content-type: text/html
    <!doctype html>
    <html lang="fr">
      <head>
        <meta charset="utf-8">
        <title>Voici mon site</title>
        <link rel="stylesheet" href="style.css">
      </head>
      <body>
        <h1>Vive la NSI !</h1>
        <form action="http://192.168.2.1">
          <p>Nom : <input value="nom"></p>
        </body>
    </html>
```

Sensibilisation sur la nécessaire sécurisation des communications

La cybersécurité est un domaine fondamental dans notre société hyperconnectée. Filius est un outil trop simple pour pouvoir étudier sérieusement ce domaine. Cependant, il est tout même possible d'utiliser Filius afin de sensibiliser les élèves sur les attaques du type "man in the middle".

Il est possible de repartir de la situation précédente avec le serveur web, le routeur et les clients :



En effectuant un clic droit sur le routeur depuis le mode simulation, vous pouvez obtenir un menu qui vous permettra “d’écouter” une des interfaces réseau du routeur. Il est ensuite possible d’effectuer une requête HTTP d’un client vers le serveur web, afin de constater qu’il est possible de capturer cet échange client-serveur depuis le routeur :

The image shows a network diagram on the left and a packet capture window on the right. The diagram illustrates a central router connected to two switches. The left switch is connected to two laptops, M1 and M2. The right switch is connected to two laptops, M3 and M4. A server labeled 'serveur web' is also connected to the router. The packet capture window, titled 'Échanges de données', shows a list of captured packets. The table below represents the data shown in the window:

No.	Date	Source	Destination	Protocole	Couche	Commentaire
1	12:28:54...	192.168.2.255	192.168.2.1	ARP	Internet	Recherche de l'adresse MAC associée à 192.168.2.1. 192...
2	12:28:54...	192.168.2.1	192.168.2.255	ARP	Internet	192.168.2.1: 58:66:5E:68:22:43
3	12:28:54...	192.168.0.1:38935	192.168.2.1:80	TCP	Transport	SYN, SEQ: 5 000 000
4	12:28:54...	192.168.2.1:80	192.168.0.1:38935	TCP	Transport	SYN, SEQ: 4 000 000, ACK: 5 000 001
5	12:28:54...	192.168.0.1:38935	192.168.2.1:80	TCP	Transport	SEQ: 5 000 001, ACK: 4 000 001
6	12:28:54...	192.168.0.1:38935	192.168.2.1:80	Application	GET / HTTP/1.1	Host: 192.168.2.1
7	12:28:54...	192.168.2.1:80	192.168.0.1:38935	TCP	Transport	SEQ: 4 000 001, ACK: 5 000 034
8	12:28:54...	192.168.2.1:80	192.168.0.1:38935	Application	HTTP/1.1 200 OK	Content-type: text/html <doctype hta...
9	12:28:54...	192.168.0.1:38935	192.168.2.1:80	TCP	Transport	SEQ: 5 000 034, ACK: 4 000 328
10	12:28:55...	192.168.0.1:38935	192.168.2.1:80	TCP	Transport	FIN, SEQ: 5 000 034
11	12:28:55...	192.168.2.1:80	192.168.0.1:38935	TCP	Transport	SEQ: 4 000 328, ACK: 5 000 035
12	12:28:55...	192.168.2.1:80	192.168.0.1:38935	TCP	Transport	FIN, SEQ: 4 000 328
13	12:28:55...	192.168.0.1:38935	192.168.2.1:80	TCP	Transport	SEQ: 5 000 035, ACK: 4 000 329

Below the table, the details for packet 8 are expanded:

```

No. : 8 / Date: 12:28:54.737
  Réseau
    Source: 58:66:5E:68:22:43
    Destination: 03:50:0F:F3:5C:D2
    Commentaire: 0x000
  Internet
    Source: 192.168.2.1
    Destination: 192.168.0.1
    Protocole: IP
    Commentaire: Protocole :6, TTL: 64
  Transport
    Source: 80
    Destination: 38935
    Protocole: TCP
    Commentaire: SEQ: 4 000 001
  Application
    Commentaire:
      HTTP/1.1 200 OK
      Content-type: text/html
      <doctype html>
      <html lang="fr">
      <head>
      <meta charset="utf-8">
  
```

Cet exemple simple permet de montrer aux élèves qu’un tiers malveillant peut potentiellement intercepter nos échanges s’il possède les connaissances techniques nécessaires. Après cette constatation, il est bien évidemment possible d’évoquer le protocole HTTPS qui permet de chiffrer les communications en précisant bien que le tiers malveillant sera toujours en mesure d’intercepter les communications, mais qu’il ne pourra pas tirer la moindre information de cette interception. Il aurait été intéressant de pouvoir effectuer la même requête en utilisant le protocole HTTPS afin de montrer aux élèves que les communications sont chiffrées, mais malheureusement, Filius ne propose pas cette possibilité.

Comme nous venons de le voir au travers des différents exemples que nous venons d’évoquer, Filius est un outil bien adapté à la mise en activité des élèves de SNT ou de NSI. Évidemment, comme tous les outils, il n’est pas parfait et on peut relever quelques manques, par exemple, au niveau de l’analyse poussée des paquets ou des requêtes (impossible d’effectuer une requête HTTP de type POST, impossible d’effectuer une requête HTTPS). Pour ce genre d’études, on pourra se tourner vers des outils un peu plus spécialisés, par exemple Wireshark.