

Yet Another Network Simulator
Développement d'un outil de simulation de
réseaux

Sylvain Daste
Avec l'aide de Mr. Chaput Enseignant Chercheur à l'Irit
ENSEEIH
IRIT

Toulouse 2018-18-09

Table des matières

1	Remerciement	3
2	Introduction	4
2.1	Objectifs de Yane	4
2.2	Objectifs du stage	4
3	Yane - Utilisation	5
3.1	Simuler un réseau	5
3.2	Exemples d'utilisation	6
4	Yane - Fonctionnement et architecture	7
4.1	Choix des langages	7
4.1.1	Développeur	7
4.1.2	Utilisateur	7
4.2	Modularité	7
4.3	YAML	7
4.4	Tmux	7
5	Docker	8
5.1	Docker Bases	8
5.2	Docker et Yane	8
5.3	8
6	L'interaction avec Yane	9
6.1	tmux	9
7	Conclusion	10

1 Remerciement

Je tiens à remercier mon maitre de stage, Mr. Emmanuel Chaput, pour son accueil, le temps passé ensemble et toute l'aide qu'il m'a apporté. Grâce aussi à sa confiance j'ai pu m'accomplir totalement dans mes missions. Il fut d'une aide précieuse dans les moments les plus délicats. Je remercie également l'Irit de m'avoir chaleureusement accueilli au sein de leur équipe.

2 Introduction

2.1 Objectifs de Yane

Yet Another Network Simulator a pour objectif d'émuler des réseaux complets. Les réseaux simulés par Yane sont des réseaux internet. Par conséquent Yane se base uniquement sur les protocoles TCP/IP (IPv4 seulement). Il doit être en mesure d'émuler ou de virtualiser jusqu'à 20 stations. Les stations peuvent être de nature différentes. Par exemple émuler un serveur DNS, des routeurs, des bases de données. Ces différents besoins imposent à Yane de pouvoir émuler la plusieurs OS ou de virtualiser différentes applications. Yane permettra ainsi de simuler des réseaux de taille moyenne sur une seule machine et ainsi être utilisé en TP à l'ENSEEIHHT afin d'apprendre à manipuler et comprendre les réseaux.

Le choix fait par Mr. Chaput était de rendre Yane modulaire afin de pouvoir intégrer différentes techniques d'émulation au fur et à mesure de son développement.

Une fois le réseau émulé, l'utilisateur doit pouvoir interagir avec celui-ci. Les interactions provoqueront l'envoi de paquets entre les différentes stations. L'utilisateur aura aussi la possibilité d'écouter le réseau.

Mr. Chaput avait déjà développé une base simple de Yane. Il avait fournit un ensemble d'exemples d'utilisation de Yane. Nous passerons en revue les principaux exemples car ils permettent de comprendre le fonctionnement de Yane.

2.2 Objectifs du stage

L'objectif de ce stage était d'améliorer Yane, un logiciel simple capable de simuler/émuler des réseaux internet. Mr Chaput avais déjà fournit un socle avec les fonctionnalités les plus basiques. Mes missions étaient de :

- fournir une documentation des fonctionnalités à la fois d'un point de vue utilisateur que développeur.
- développer de nouvelles techniques d'émulation, Docker notamment.
- développer Yane de façon à faciliter l'intégration de nouvelles fonctionnalités.
- essayer plusieurs interfaces d'interaction avec Yane (tmux, terminator,...)

Le développement de Yane est accessible sur Github :
<https://github.com/Manu-31/yane.git> ce qui permet un suivit agile.

Annonce du plan

3 Yane - Utilisation

Dans cette partie nous allons rapidement voir comment Yane peut être utilisé et les fonctionnalités qu’il propose. Puis nous verrons quelques exemples d’utilisation.

3.1 Simuler un réseau

Une fois que l’utilisateur a spécifié le réseau à simuler, il doit créer un fichier `yane.yml`. Ce fichier sera la représentation du réseau. Mais l’utilisateur pourra aussi, via ce fichier, demander plusieurs chose à Yane, grâce aux différentes fonctionnalités qu’il possède. Dont par exemple, spécifier quelles méthodes de virtualisation ou d’émulation doivent être utilisés.

Le code suivant montre un exemple simple de ce que j’ai dû dans un premier temps réaliser :

Listing 1 – Réseau à 2 stations

```
network:
name: basic
version: 1.0
hosts:
  - host-a
  - host-b

links:
  - host-a:v0:192.168.1.1/24!host-b:v0:192.168.1.2/24

consoles:
  - all

dumpif:
  - all
```

Yaml permet d’être assez explicite. On peut clairement lire que ce fichier générera un réseau avec 2 stations reliées par un lien. On peut spécifier la configuration des interfaces des stations grâce à la balise ‘link’. Si l’on souhaite relier plus de 2 machines entre elles Yane est capable d’émuler un pont. Il faut alors employer la méthode suivante :

Listing 2 – Les ponts

```
...
bridges :
  - __BRIDGE_NAME__
    interfaces: host-a:v0:192.168.1.1/24!\
```

```
host-b:v0:192.168.1.2/24!\  
host-c:v0:192.168.1.3/24
```

3.2 Exemples d'utilisation

La première mission de mon stage était de fournir une documentation de Yane mais également de comprendre Yane en fournissant de nouveaux exemples d'utilisation de Yane.

4 Yane - Fonctionnement et architecture

4.1 Choix des langages

4.1.1 Développeur

Yane a été développé en BASH. Celui-ci permet de contrôler les "namespaces" et Docker facilement. Par conséquent cela à permis un développement plus rapide de Yane. Cependant BASH a montré ses limites notamment lorsque Yane à besoin de faire les choses par soi-même. Par exemple de simples calculs en BASH peuvent être délicats à mettre en oeuvre.

4.1.2 Utilisateur

L'utilisateur devra fournir une description des réseaux en Yaml. Yet Another Markup Language est un langage de représentation de données. Pour plus d'informations sur Yaml : <http://yaml.org>

4.2 Modularité

4.3 YAML

4.4 Tmux

5 Docker

5.1 Docker | Bases

5.2 Docker et Yane

5.3

6 L'interaction avec Yane

6.1 `tmux`

7 Conclusion