

Livrable sur le déploiement et implementation de la solution Projet reseau :

Simulation de la consommation des nœuds et classification du trafic dans un reseau mailé sans fil programmable .

Nom participants

Sous la supervision de : Pr Djotio, Mlle Nouhou , Mr bessala

Administration reseau 4Gi Enspy 2023 2024

Introduction

Ce document presente la procedure d'utilisation de la classification en temps reel des trafic générer par l'outil D_itg dans une architecture SDN.

I) Predisposition

Nous utiliserons le système d'exploitation Linux ubuntu dont les caracteristiques sont les suivants

Distribution ID : ubuntu

Description : ubuntu 18.04.6 LTS

Relase : 18.04

Codename : bionic

Nous devons avoir python2 , python3 , ryu , D_itg , mininet , mininet_wifi installer sur notre ubuntu

II) Structuration du dossier

Le dossier s'appelle : PROJET_RESEAU

Il est constitué de 5 dossiers :

binary : comporte le fichier binaire du modèle de machine Learning utilisé

dataset : comporte tous les datasets utilisés pour l'entrainement du modèle

Documents : comporte tous les documents important relatif au projet (rapports, livres utiles etc.)

Notebook : comporte les différents notebook utilisés lors de la phase d'entrainement des algorithmes de machine Learning.

src : comporte les fichiers de lancement du projet (Le wifi_AP_ST_Topo.py , wifi_classifier2.py,wifi_monitor_v2.py)

III) configuration du fichier wifi_classifier2.py

Avant de lancer le projet , il faut déjà copier le dossier et le coller dans Document de votre machine ubuntu .

```
proj_location_Src = "/home/bessala/PROJET_RESEAU/src/"  
proj_location_dataset = "/home/bessala/PROJET_RESEAU/dataset/"  
proj_location_Binary = "/home/bessala/PROJET_RESEAU/binary/"
```

Ensuite aller dans /PROJET_RESEAU/src/wifi_classifier2.py, modifier : « bessala » par le nom de votre machine.

IV) lancement du projet

Ouvrez deux terminaux ubuntu avec la commande : ctrl+alt+t

Passez en mode sudo : sudo su , puis entrez votre mot de passe .

Dans chacun des terminaux , naviguez : cd Document/PROJET_RESEAU/src

Dans le terminal 1 , lancez le classifieur : python3 wifi_classifier2.py « unsupervised » ou « supervised » , selon votre model de classification choisi.

Puis Dans le terminal 2, lancez la topologie mininet : python2 wifi_AP_ST_TOPO.py

Nb : servez-vous de la tabulation pour aller plus vite .

Une fois le reseau virtuel mis en place avec mininet , on peut commencer à generer le trafic grâce à d_itg.

Enter dans le terminal 2 , taper « xterm h1 h2 » .vous pouvez changer d'hote ou de station à votre convenance en entrant la commande « dump » dans mininet pour visualiser tous les équipements créer par notre topologie . pour une bonne prise en main de mininet ,ryu , openflow, ouvrez le document:/PROJET_RESEAU/documents/TP_4GI_SD_WMN.pdf

Une fois les deux invites de commandes ouverts , on va generer un trafic de h1(10.0.0.1) vers h2(10.0.0.2) . h1 sera l'émetteur , et h2 le recepteur .

Par exemple on veut generer le trafic DNS ,

Dans le xterm de h2 entrez la commande : ITGRecv -l recv.log

Dans le xterm de h1 entrez la commande : ITGSend -t 15000 -a 10.0.0.2 -rp 10003 DNS

Dans nos travaux, nous nous sommes focalisés sur les trafic que D_itg permet de générer simplement .

Nom trafic	Code emetteur	Code recepteur	Commentaire
Ping	Ping « ip machine à ping »	Ping « ip machine à ping »	Il suffit de lancer un ping all dans la console de mininet_wifi
Dns	ITGSend -t 15000 -a 10.0.0.2 -rp 10003 DNS	ITGRecv -l recv.log	L'emeteur envoie pendant 15000 mili secondes des paquets DNS au recepteur d'adresse ip 10.0.0.2
telnet	ITGSend -t 15000 -a 10.0.0.2 -rp 10002 Telnet	ITGRecv -l recv.log	
Quake			
VoIP	ITGSend -t 15000 -a 10.0.0.2 -rp 10001 VoIP -x G.711.2 -h RTP -VAD	ITGRecv -l recv.log	
CSI			
CSA			

Pour en savoir plus sur ces trafic voir le document : D_ITG-2.8.1-manual.pdf dans PROJET_RESEAU/documents/

Une fois le trafic lancé, on observe sur le terminal où on a lancer la wifi_classifier2.py supervised une classification en temps réel .

```
root@bessala-VirtualBox: /home/bessala/Documents/PROJET_RESEAU/src
Fichier Édition Affichage Rechercher Terminal Aide
[-952630682837242029, '10.0.0.1', '10.0.0.2', array(['TELNET'], dtype=object), '
INACTIVE', 'INACTIVE']
1
['TELNET']
[-4068018769424495448, '10.0.0.2', '10.0.0.1', array(['TELNET'], dtype=object),
INACTIVE', 'INACTIVE']
2
{'PING': 0, 'DNS': 0, 'TELNET': 772, 'VOICE': 0, 'CSI': 0, 'CSA': 3, 'QUAKE3': 3
56, 'unknown traffic': 0}
{'shared services': 0, 'dedicated services': 3, 'realtime services': 1128, 'unkn
own services': 0}
['TELNET']
[-952630682837242029, '10.0.0.1', '10.0.0.2', array(['TELNET'], dtype=object), '
INACTIVE', 'INACTIVE']
1
['TELNET']
[-4068018769424495448, '10.0.0.2', '10.0.0.1', array(['TELNET'], dtype=object),
INACTIVE', 'INACTIVE']
2
{'PING': 0, 'DNS': 0, 'TELNET': 774, 'VOICE': 0, 'CSI': 0, 'CSA': 3, 'QUAKE3': 3
56, 'unknown traffic': 0}
{'shared services': 0, 'dedicated services': 3, 'realtime services': 1130, 'unkn
```

Une fois le trafic terminer, on prend on observe le tableau qui nous donne aussi la classification du nombre de service dédié rencontrer. (shared services etc) .

Une fois fini , on ferme mininet_wifi avec la commande « quit » , on ferme le classifieur avec ctrl+c

Conclusion

..... ;

Dans le but d'améliorer nos travaux , vous pouvez nous contacter à

