

SAE 2.04 : Projet intégratif

par [Thibaut Karcher](#)

Introduction

Cette SAE ayant eu lieu à la toute fin de la première année du BUT réseaux-télécommunications avait pour but de réunir l'ensemble des compétences et connaissances acquises au cours de la totalité de l'année.

Cette SAE regroupe donc de nombreuses notions et fait références à l'ensemble des ressources et compétences des deux premiers semestres de BUT.

Elle était composé de 4 grandes parties toutes mises en relations :

- Partie réseau
- Partie téléphonie
- Partie collecte
- Partie Web / Base de donnée

Partie réseau

La première partie est la fondation de ce projet et est la base du projet en lui-même, en effet l'ensemble du projet a été fait sur un réseau privé que nous avons entièrement configuré de A à Z, il était donc forcément très important d'effectuer cette partie en tout premier afin de pouvoir passer à la suite sans problèmes, nous avons donc entièrement configuré notre routeur et switch afin de mettre en place tout le réseau et le segmenter.

Nous avons donc segmenté le réseau en 4 VLANS :

- VLAN voix
- VLAN users
- VLAN server

Outils utilisés

- Routeur
Cisco 1900 series
- Switch Cisco
2960 24 ports PoE
- Emulateur de terminal :
PuTTY
- Téléphone logiciel :
Linphone
- Téléphone matériel :
SIP

Ressources associés

- R2.01 | Technologies de l'Internet
- R2.02 | Administration système et fondamentaux de la virtualisation

- VLAN admin

L'adresse IP qui nous a été fournie était la suivante :
10.252.18.0 /24, nous avons donc coupé ce réseau en 4
pour chaque VLAN, on obtient donc le découpage
suivant :

- VLAN voix : 10.252.18.0 /26
- VLAN users : 10.252.18.64 /26
- VLAN server : 10.252.18.128 /26
- VLAN admin : 10.252.18.192 /26

Avec pour chaque VLAN, les 10 premières adresses en
statique, le reste en dynamique à l'aide d'un DHCP
configuré sur le routeur.

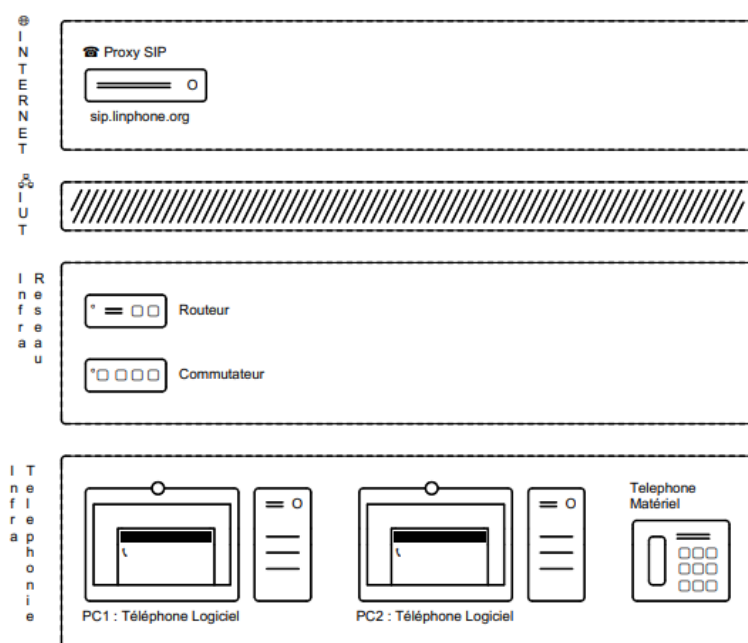
On a ensuite effectué plusieurs configurations sur le
routeur afin :

- d'ajouter des ACL afin de restreindre l'accès au

Partie Téléphonie

Dans cette partie nous avons donc effectué la mise en
place et la configuration des différents téléphones
matériels et logiciels du réseau.

Le schéma étant le suivant :



- R2.03 | Bases des services réseaux
- R2.04 | Initiation à la téléphonie d'entreprise
- R2.05 | Signaux et Systèmes pour les transmissions
- R2.06 |

Apprentissages critiques

- AC11.01 | Maîtriser les lois fondamentales de l'électricité afin d'intervenir sur des équipements de réseaux et télécommunications
- AC11.02 | Comprendre l'architecture et les fondements des systèmes

Les 2 téléphones logiciels étant sous distribution debian étaient brassés dans le VLAN users, le téléphone matériel quant à lui était brassé sur le VLAN voix.

Nous avons ensuite configuré l'ensemble des téléphones :

Configuration téléphone matériel :

1. Remise mode usine
2. On prend l'adresse IP du téléphone et on le copie / colle dans un navigateur de recherche, on tombe ensuite sur la page par défaut de connexion de Yealink
3. On configure nos compte comme suivant :

Key	Type	Value	Label	Line	Extension
Line Key1	BLF	tl.romain.bobenrieter	tl	Line 1	tl.romain.bobenr
Line Key2	BLF	tml.romain.bobenrieter	tml	Line 1	tml.romain.bobe

4. Notre téléphone matériel est configuré

Configurations des 2 téléphones logiciels :

1. Installation de Linphone
2. Connexion à Linphone avec nos comptes déjà existants
3. Test des communications
4. Les téléphones logiciels sont configurés


numériques, les principes du codage de l'information , des

communications et de l'Internet

- AC11.03 | Configurer les fonctions de base du réseau local
- AC11.04 | Maîtriser les rôles et les principes fondamentaux des systèmes d'exploitation afin d'interagir avec ceux-ci pour la configuration et l'administration des

Annexe

 [confrouteur.txt](#)

 [configswitch.txt](#)

 [main.py](#)

Partie collecte de données

Dans cette partie, le but était de récupérer via 2 capteurs donnés dans 2 topics d'un MQTT des données qui seront ensuite directement envoyés dans une base de donnée extérieures en vue de l'affichage et du tri de ces données dans un serveur web.

Pour effectuer cette collecte de donnée nous avons donc programmer un scripts python qui permet :

1. la connexion au broker ainsi qu'aux topics qui représentent mes 2 capteurs

Partie base de donnée / Web

Dans un dernier temps nous avons donc ensuite du configurer la base de donnée sur laquelle nos données récupérer via le script MQTT sont enregistrés puis nous avons mis en place un site web à l'aide du framework Django permettant l'affichage de ces données sous forme de tableau et de graphique.



Annexe

Lien vers
l'ensemble des
fichiers du site :