

M1 Informatique Réseaux

Devoir comptant pour le contrôle continu

Gaétan Richard, Jean Saquet
gaetan.richard@unicaen.fr, jean.saquet@unicaen.fr

4/01/2012

1 Avant de commencer

Ce TP fait parti du contrôle continu de l'unité réseau. Il vous faudra rendre un fichier `.mar` contenant le travail réalisé ainsi qu'un rapport expliquant les choix effectués et les actions réalisées.

Le tout sera renvoyé dans une archive nommée `nom-id.tar.gz` (où `nom` est votre nom et `id` un nombre aléatoire entre 13 et 275) dans le dossier `/home/etudiants/data/m1/tp5m1` avant le lundi 15 octobre à 13h37.

2 Architecture

On s'intéresse à l'architecture présentée dans la figure 1. Le point crucial est que les deux routeurs (E1 et E2) disposent de deux liaisons entre eux appartenant à deux réseaux distincts.

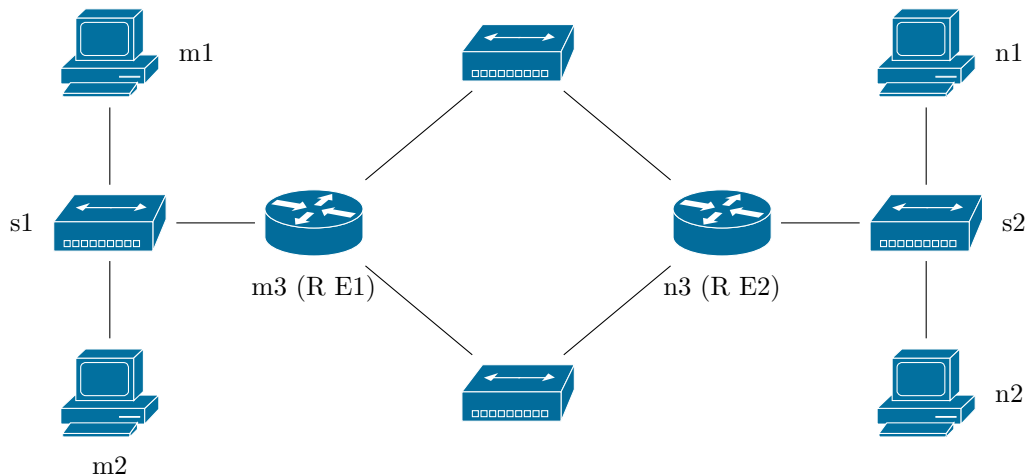


FIGURE 1 – Notre réseau

3 Adressage

On utilise les réseaux :

- 192.168.1.0/24, 2001:660:7101:1::/64 pour le réseau comprenant `s1` ;

- 192.168.2.0/24,2001:660:7101:2::/64 pour le réseau comprenant s2;
- 10.1.0.0/16,2001:660:7101:ffff:1:/80 et 10.2.0.0/16,2001:660:7101:ffff:2:/80 pour les deux réseaux d'interconnexion entre E1 et E2.

Configurez les machines en utilisant des adresses statiques.

4 Routage

Ajouter les routes nécessaires pour permettre la communication entre les machines m et n. On configurera les routes par défaut afin que les routes de n3 à m3 et de m3 à n3 ne soit pas l'inverse l'une de l'autre.

On veillera à ce que ces règles soient fonctionnelles dès le lancement suivant de la machine.

5 Redondance couche routage

Comme on dispose de deux connexions entre les deux routeurs, nous allons essayer de mettre en place une redondance.

Dans un premier temps, écrire deux scripts sur le routeur E1 : **route-main.sh** et **route-backup.sh** qui permettent de configurer le routage pour utiliser respectivement la connexion par défaut et de secours (la connexion par défaut correspond à la route que vous avez choisie de m3 vers n3 à la section 4 ci-dessus).

Faire de même pour le routeur E2.

Trouver une commande permettant de s'assurer que la connexion entre les deux routeurs est active.

Écrire un script **daemon.route.sh** qui utilise le test précédent et passe automatiquement sur le liens de secours si le lien principal tombe et revient sur le lien principal quand celui-ci est de nouveau actif.

6 Redondance de services

On lance maintenant un serveur apache sur la machine n1 à l'aide de la commande **/etc/init.d/apache2 start**.

Écrire un script **test-apache.sh** permettant de tester depuis n2 si le service apache est lancé sur n1.

Mettre en place sur n2 un script **apache-backup.sh** qui démarre le serveur apache dès que celui de n1 ne répond plus et l'éteint lorsque le serveur de n1 redevient disponible.

Sur n3, écrire un script **redir.sh** qui redirige les requêtes à destination du serveur *web* de n1 sur celui de n2 si le premier n'est pas accessible mais que le second l'est. Faire en sorte que ce script supprime la redirection dès que le serveur sur n1 redevient disponible.

A Scripts

Pour ceux qui ne sont pas familiers avec cette notion, un script est un fichier texte (dont l'extension est de préférence *.sh*) contenant une suite de commandes UNIX.

Il faut prendre soin d'ajouter la ligne **#!/bin/sh** au début et de le rendre exécutable avec la commande **chmod u+x fichier.sh**.

Il est ensuite possible de la lancer à l'aide de la ligne **./fichier.sh** à partir du répertoire le contenant.

Dans un script, il est possible d'utiliser les relations **com1 || com2** (resp. **com1 && com2**) qui exécute la commande 2 si la commande 1 a échoué (resp. réussi).