Devoir à rendre le 17 décembre 2012
Déclaration de services
Jean.Saquet@unicaen.fr

1 Présentation du sujet

La réalisation devra utiliser `marionnet` mais l'idée est basée sur la situation suivante :
Un groupe de personnes (travaillant dans un même domaine par exemple) désirent partager certains services installés sur leurs réseaux personnels ou de petites entreprises. Ces services peuvent évoluer au cours du temps, ils peuvent être ajoutés, retirés, remplacés. Ces services n'étant pas publics, leurs possesseurs n'ont pas envie de les déclarer dans un DNS. Le but va donc être pour chacun d'informer ses partenaires sur les services qu'il leur met à disposition, ceux-ci étant enregistrés dans un serveur connu de tous les partenaires.

On suppose que tous les partenaires ont un préfixe IPv6, les services seront accessibles uniquement en IPv6 et donc sans passer par un NAT.

Tous les partenaires devront donc renseigner leur serveur avec les paramètres des services qu'il offre, et pourront interroger les serveurs des autres partenaires pour savoir quels services ils offrent, et y accéder.

2 Simulation sous `marionnet`

Le réseau d'un partenaire sera simulé sous `marionnet`, un routeur connecté avec l'extérieur simulera la passerelle d'accès vers l'Internet de ce partenaire. On utilisera les adresses déjà utilisées en TP dans l'unité réseaux, en section 3 du TP2, soit les préfixes `2001:660:7101:001X::/64` (X allant de 1 à F), et pour adresse externe de la passerelle `2001:660:7101:ffff:10::1X/80`.

Chacun de ces réseaux utilisera la machine d'adresse 2 (`2001:660:7101:001X::2`) comme serveur des paramètres de ses services. Attention, ce ne sera pas un serveur DNS ! Cependant il sera référencé dans DNSTP comme `ns.zonexx.tp.info.unicaen.fr`, avec son adresse v6 (attention xx équivalent décimal de 1X).

2.1 Routage IPv6

Le premier travail à effectuer sera donc de configurer un réseau avec quelques machines, de leur donner des configurations IPv6, manuelle pour la machine 2, auto-configurées pour les autres. L'adresse d'un DNS ne sera pas nécessaire pour le moment. Quagga devra être lancé avec RIPng pour assurer le routage entre ce réseau et les réseaux similaires. Pour tester, vous devrez donc configurer deux réseaux.

2.2 Serveur de paramètres de services : SRVS(serveur de services)

Comme écrit plus haut, la machine d'adresse `2001:660:7101:001X::2` va héberger un serveur dont il faudra écrire le code, et qui pourra :
- recevoir des demandes d'enregistrement ou de suppression de services de la part de toutes les machines du réseau dans lequel elle se trouve (`2001:660:7101:001X::/64`).
- répondre à des demandes d'obtention de paramètres de serveur, ou de liste, provenant d'une machine des autres réseaux connus.

Les paramètres d'un services seront enregistrés dans une ligne de texte de la forme :
<Nature du service> <IP de la machine hôte> <Numéro de port>

On utilisera le protocole dont les commandes sont :
- REGISTER <Nature du service> <IP de la machine hôte> <Numéro de port>
  cette commande pourra être envoyée par toute machine du même réseau que le serveur SRVS sollicité. Ce dernier devra enregistrer le service SRVS dans sa base de données. Il devra également retourner un message de succès ou d'erreur en précisant ce qui a été ajouté ou la cause de l'erreur.
- UNREGISTER <Nature du service> <IP de la machine hôte> <Numéro de port>
  Supprime l'enregistrement correspondant dans le serveur et renvoie la réponse.
- LIST [<Nature du service>]
  Interroge le SRVS de la machine serveur sollicitée, laquelle doit renvoyer la liste de ses services disponibles sous forme d'une suite de lignes ou une seule ligne selon le cas. Si <Nature du service> est précisée, elle ne renvoie que la liste des services de cette nature connus d'elle.

Les réponses sont de la forme :
- +OK <parametres>
- -ERR <message d'erreur>

les paramètres de retour en cas de succès dépendent évidemment de la commande. En particulier, ce sera une seule ligne si la réponse est constituée des paramètres d'un service, de plusieurs lignes sinon.

3 Logiciels client et serveur

Le langage n'est pas imposé, toutefois, c'est sans doute un peu complexe pour netcat associé à des scripts, il est conseillé d'utiliser un langage de programmation classique et ses primitives de programmation réseau. Le cours les présentera en `Python`.

Il faut réaliser un serveur et un client. Ce dernier devra comporter une interface utilisateur (en mode texte éventuellement) pour lui permettre de choisir ses requêtes, lancer les commandes correspondantes et afficher les réponses.

Le serveur doit bien entendu fonctionner en permanence. Par contre, il ne sera pas nécessaire de prévoir des requêtes simultanées, les réponses étant immédiates. On pourra jute prévoir dans le client un mécanisme de ré-essai pour le cas où la requête serait refusée pour cause de serveur occupé.

4 Tests

Vous devez écrire quelques exemples de services pour les déclarer, les tester. Ce peut être des services très simples (écho, heure, ...) ou la configuration de quelques logiciels libres utilisant des protocoles connus(client/serveur ftp, ...). Votre rapport devra mentionner la réalisation de ces tests. Pensez à filtrer les accès indésirables (seule une machine du même réseau peut déclarer un service, seuls les partenaires peuvent accéder aux serveurs).

5 Conditions, présentation du travail

Votre réalisation doit pouvoir fonctionner sous `marionnet` sur les machines libre-service de la salle 409. Le langage de programmation du serveur et du client n'est pas imposé.

Le travail est à réaliser en binôme (ou seul). Vous devrez rendre votre travail pour le 17 décembre 2012 en téléchargeant une archive targz sur le serveur de devoirs.

Dans le cas présent cette archive doit contenir :
- un fichier `noms.txt` contenant les noms des étudiants ou étudiantes composant le binôme (s'il y a lieu),
- Le fichiers .mar correspondant à un réseau configuré.
- les sources de vos programmes client et serveur
- un rapport réalisé de préférence avec LaTeX (résultat en pdf) précisant les services que vous avez définis, comment vous les nommez, comment le serveur stocke les services déclarés, les solutions aux problèmes rencontrés.
- tout autre fichier que vous jugerez nécessaire.