Protocoles réseaux
TD n° 5 : gestion de la pénurie IPv4 par un NAT et configuration d'interface par DHCP

Exercice 1 : NAT

Une UFR d'informatique possédait en l'an 2000 une salle de TP avec 30 postes de travail en état de marche. On attribue aux 31 machines de la salle le préfixe 194.254.199.0/24.
Un utilisateur connecté à la machine `frite-grasse` tape `ip route show` et obtient

```
default via 194.254.199.254 dev eth0
194.254.199.0/24 dev eth0 scope link src 194.254.199.42
```

1. S'il y a 30 postes de travail, quelle est la 31ème machine ?

2. Décrire l'entête et la route suivie par un paquet à destination de la machine 194.254.199.51

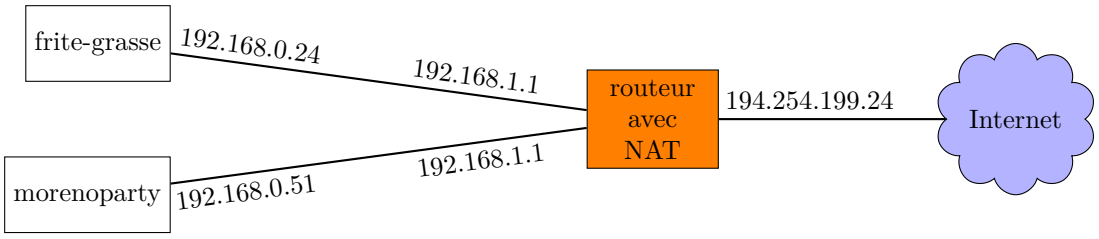
3. Même question pour un paquet à destination de 81.194.27.171.

En 2022, la pénurie [1] touchant cette UFR, toute cette salle ne bénéficie plus que du préfixe 194.254.199.24/31

4. Combien de machines accessibles depuis l'Internet global peut-on mettre dans la salle ?

En fait la salle possède maintenant 18 postes de travail en état de marche [2]. On décide d'installer un **NAT** (*Network Address Translator*) sur la passerelle. Il s'agit d'un mode optionnel d'un routeur IPv4 permettant fait de la *traduction d'adresses*, il remplace des adresses IP par d'autres adresses IP. Or ce routeur passerelle est une baie avec une trentaine de prises Ethernet :

- celle reliée à Internet correspond à l'interface `publique` du routeur, d'IP 194.254.199.25,
- les autres sont toutes reliées à l'interface `privée` du routeur, ayant l'IP 192.168.1.1



5. Décrire l'entête et la route suivie par un paquet de `frite-grasse` à destination de `morenoparty`

6. Donnez les tables de routage de `frite-grasse` et `morenoparty`.

7. Pourquoi une machine d'IP 192.168.0.24 ne peut pas communiquer *directement* (sans NAT) avec 81.194.27.171 ?

8. Décrivez les entêtes d'un paquet envoyé par `frite-grasse` à 81.194.27.171, avant et après être passé par le routeur NAT.

1. d'adresses IPv4, mais pas que...
2. en effet, la taille des groupes de TP est passée à 40 personnes

9. Que se passe-t-il lors de la transmission de la réponse ? Décrivez comment le routeur NAT viole deux des principes vus en cours. Quelles conséquences ?

On s'interesse maintenant à ce qu'il y a sous le capot du NAT. Il maintient une table de traductions dont les entrées sont de la forme :

$$\text{IP privée} : \text{port privé} \to \text{IP publique} : \text{port public}$$

Initialement, la table est vide. Lorsque le routeur NAT reçoit un paquet ayant pour source l'IP X, et le port p, et pour destination l'IP Y et le port q, si le paquet arrive par l'interface privée,

1. chercher une entrée de la forme $X : p \to Y, r$ dans la table (r peut être quelconque),
2. s'il n'y en a pas, choisir s et ajouter l'entrée $X : p \to Y : s$
3. envoyer le paquet sur l'interface publique en remplaçant l'IP source par l'IP publique du routeur NAT et le port source par r.

Si par contre le paquet arrive par l'interface publique,

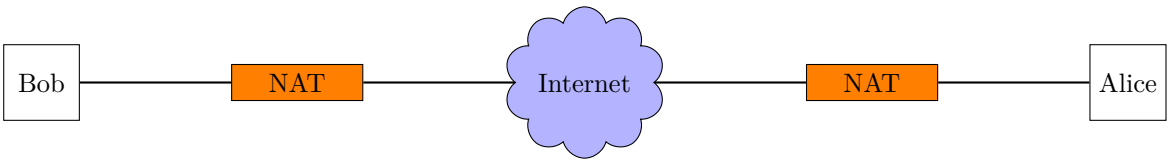
1. chercher une entrée de la forme $Z, r \to X, q$ dans la table (Z et r peuvent être quelconques),
2. s'il n'y en a pas, jeter le paquet,
3. sinon envoyer le paquet sur l'interface privée en remplaçant l'IP de destination par Z et le port de destination par r.

10. La table de traductions est mise à jour lorsque le NAT reçoit un paquet de données. Commentaire ?

11. On aurait envie de poser $s = p$ dans l'algorithme ci-dessus, c'est-à-dire utiliser un port public égal au port privé. Pourquoi n'est-ce pas possible en général ? Suggérez un algorithme pour choisir s.

Exercice 2 : Serveur derrière un NAT

On imagine la situation suivante, où Alice et Bob sont sur deux réseaux locaux distincts connectés à l'Internet chacun par un routeur NAT. Ils veulent jouer à un jeu vidéo en ligne, ce qui demande à l'un deux de lancer un serveur et à l'autre un client.



1. Bob propose de lancer le serveur sur sa machine. Il dit à Alice de se connecter à 192.16.0.10 sur le port 3000. Alice lui dit que cela ne va pas marcher, pourquoi ?

2. Bob comprend son erreur, il trouve son IP publique et la donne à Alice. Celle-ci lance son client mais il ne parvient pas à se connecter, pouruqoi ?

3. Le serveur a justement une option pour inviter un client en initiant la connexion depuis le serveur [3] lorsque ce dernier est derrière un NAT. Bob entre l'IP publique et le port d'Alice mais il ne parvient pas à joindre Alice, pourquoi ?

4. Clara, une de leur amies, leur propose de les aider en leur fournissant temporairement, uniquement pour initier la connexion, un hôte sur l'Internet qui n'est pas derrière un NAT. Voyez-vous en quoi cela peut être utile ? (Plusieurs techniques sont possibles.)

3. Qui est donc client à la couche transport, mais serveur à la couche application.

Exercice 3 : DHCP

Le protocole DHCP sert à configurer automatiquement les paramètres de couche réseau d'un hôte : l'adresse IPv4 [4] de son interface réseau, le préfixe du lien attaché, adresse du serveur DNS, adresse d'une passerelle par défaut, etc. Il y a 76 options DHCP pour configurer plein de choses. Le protocole se déroule en quatre étapes :

1. le client DHCP envoie en *broadcast* un message `DHCPDISCOVER` ;
2. tous les serveurs DHCP répondent en unicast par un message `DHCPOFFER` contenant chacun une adresse IP proposée au client ;
3. le client choisit un serveur, puis envoie en *broadcast* un message `DHCPREQUEST` contenant l'adresse IP qu'il désire ;
4. le serveur répond en unicast par un message `DHCPACK` avec tous les paramètres réseau affectés au client, la durée `lease_time` du bail (voir plus loin) et deux temps T1 < T2 < `lease_time` ; ou alors par un message `DHCPNAK` s'il décide de ne pas affecter les paramètres demandés.

1. Aux étapes 2 et 4, le serveur envoie un unicast alors que le client n'a pas encore d'adresse. Commentaire ?

2. Le serveur DHCP inclus dans shncpd [5] évite ce problème en utilisant des *broadcasts* aux étapes 2 et 4. Qu'en pensez-vous ?

3. Le serveur DHCP maintient la liste des adresses affectées aux clients pour éviter d'affecter la même adresse à deux clients distincts. Que se passe-t-il s'il y a plusieurs serveurs sur le même lien ?

4. Que doit faire le serveur lorsqu'il envoie un `DHCPOFFER` ? Pourquoi le `DHCPREQUEST` de l'étape 3 est-il envoyé en broadcast ?

5. On suppose qu'un client accepte les paramètres proposés par le premier serveur qui répond. Dessinez l'automate qui décrit le comportement de ce client.

6. On suppose qu'un client attend 2 secondes après la première réception d'un `DHCPOFFER` pour collecter toutes les offres de serveur avant d'en choisir une. Dessinez l'automate de ce client.

Lorsqu'il quitte le réseau, le client envoie un message `DHCPRELEASE` pour libérer l'adresse qui lui a été attribuée.

7. Modifiez l'automate dessiné à la question 6 pour prendre en compte l'envoi de `DHCPRELEASE`.

Le message `DHCPRELEASE` peut être perdu, ou le client peut quitter le réseau sans envoyer de `DHCPRELEASE` [6]. Pour éviter de perdre des adresses IP indéfiniment, les adresses sont louées (*leased*) au client pour un temps fini (de l'ordre de quelques heures ou quelques jours). Un client maintient un *timer* qui mesure le temps depuis lequel il a acquis un bail (*lease*). Lorsque le bail arrive à expiration, le client abandonne l'adresse et recommence à l'étape 1.

8. Modifiez l'automate de la question 6 pour prendre en compte la perte d'un bail.

Le protocole décrit à la question précédente fait que le client perd temporairement son adresse, et ne garantit pas la stabilité des adresses. Pour éviter ce problème, le comportement du client est le suivant :

4. En IPv6, la situation est plus confuse : il existe deux protocoles de configuration, RA, qui fait de l'autoconfiguration sans étet, et DHCPv6, qui est analogue à DHCP(v4), mais beaucoup plus complexe. Lorsqu'il démarre, un hôte commence par essayer de s'autoconfigurer à l'aide de RA, mais un routeur peut lui demander d'utiliser DHCPv6 à la place. Android n'implémente pas DHCPv6 (et je sais pourquoi), donc en pratique c'est RA qui est utilisé.
5. `https://github.com/jech/shncpd`
6. En fait, la plupart des clients DHCP n'envoient pas de `DHCPRELEASE`.

5. au temps T1 (typiquement T1 = 50% lease_time), le client envoie un `DHCPREQUEST` unicast au serveur sélectionné. Si le serveur répond par un `DHCPACK`, le client repart pour une pleine durée de bail ; s'il répond par `DHCPNAK`, il recommence à l'étape 1 ;
6. en cas de non réponse à l'étape 5, alors au temps T2 (typiquement T2 = 87,5% lease_time), le client envoie un `DHCPDISCOVER` (broadcast) tout en conservant sa vieille adresse. Si un serveur répond par `DHCPOFFER`, le client peut envoyer un `DHCPREQUEST` tout en conservant sa vieille adresse. Si un serveur répond alors par `DHCPACK`, il abandonne l'ancienne adresse et s'affecte la nouvelle.
7. au temps lease_time, il abandonne l'adresse et recommence à l'étape 1.

9. Pourquoi les étapes 6 et 7 existent-elles ?

10. (S'il reste du temps) Modifiez l'automate de la question 6 pour qu'il implémente le protocole complet.