

TP n°5

Protocole SNTP

Introduction

Le protocole NTP Network Time Protocol permet de régler avec une grande précision l'horloge d'un hôte à partir d'une ou plusieurs sources de temps distantes. NTP est défini dans le RFC 1305, un document de 35 pages qui décrit une discussion détaillée des algorithmes de réglage que les clients NTP doivent employer.

Pour beaucoup d'applications, la complexité de NTP n'est pas nécessaire. RFC 1305 définit NTP Simple Network Time Protocol, une version simplifiée de NTP qui ne définit que le protocole et pas les algorithmes à employer. Tout client ou serveur NTP est *a fortiori* un client ou serveur NTP, mais l'inverse n'est pas vrai.

Le but de ce TP est d'implémenter un client NTP.

1 Le protocole

Lorsqu'il détermine que c'est nécessaire, le client NTP envoie au serveur une requête sous forme d'un paquet UDP qui contient la date à laquelle ce message est transmis. Le serveur répond aussitôt que possible avec une réponse qui contient quatre dates :

- la date à laquelle la requête a été transmise selon le client ;
- la date à laquelle la requête a été reçue selon le serveur ;
- la date à laquelle la réponse a été transmise selon le serveur ;
- la date à laquelle l'horloge du serveur a été réglée pour la dernière fois.

En outre, la réponse contient un certain nombre de données que nous ne considérerons pas dans ce TP, et qui permettent à NTP mais pas nécessairement NTP d'avoir une idée de la précision des résultats.

On remarquera que la réponse contient la date que le client avait utilisée dans la requête ; ceci permet au client d'envoyer une réponse comme à l'habitude par avec une requête donnée et donc de contourner les problèmes dus à la nature non-abandon du transport.

1.1 Format des données

Date NTP Une date NTP est représentée sous la forme d'un entier de 48 bits exprimant un temps en unités de 2^{-32} secondes écoulées depuis l'époque d'origine. RFC 1700 définit le format d'une telle date comme suit :

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
seconds																															
seconds Fraction -padded																															

Message NTP Les requêtes et les réponses NTP ont le format suivant :

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	
LI		N			Mode			tratum								Po								Prec s on								
Root De ay																																
Root D s pers on																																
Reference Ident er																																
Reference																mestamp																🔒
Or g nate																mestamp																🔒
Rece ve																mestamp																🔒
ransm t																mestamp																🔒

Les champs d'un tel message sont définis comme suit :

- *LI* sera ignoré dans ce P et vaudra toujours 0 ;
- *VN* est la version du protocole employée et vaudra toujours 3 ;
- *Mode* est le type de message et vaudra 3 pour une requête et 4 pour une réponse ;
- *Stratum*, *Poll* et *Precision* seront ignorés dans ce P ;
- *Reference Timestamp* sera ignoré dans ce P ;
- *Originate Timestamp* vaut 0 pour une requête et pour une réponse est la copie du *Transmit Timestamp* de la requête correspondante ;
- *Receive Timestamp* vaut 0 pour une requête et pour une réponse est la date de réception de la requête correspondante ;
- *Transmit Timestamp* est la date de transmission de ce paquet.

Le paquet peut être suivi de 4 octets de données supplémentaires que nous ne transmettrons jamais et qui seront ignorés lors de la réception.

2 Code fourni

Vous trouverez sur la page web des `NtpMessage` Java pouvant vous aider. Il permet de manipuler un message NTP et récupérer facilement les champs du message NTP. Par exemple :

- `NtpMessage msg = new NtpMessage(packet.getData());`
Renvoie un objet `NtpMessage` à partir d'un paquet `DatagramPacket`.
- `NtpMessage.encodeTimestamp(array, pointer, timestamp)` Encode le *timestamp* dans le message *array* à partir de la position en octet *pointer*.
- `msg.toString()`
Renvoie en chaîne de caractère le contenu du message.
- `msg.receiveTimestamp()`
Renvoie le temps de réception du message. Recevra *timestamp*.

3 Exercices

Exercice 1 [Premier contact NTP]

Écrivez en Java un programme qui exécute les actions suivantes :

1. crée une requête NTP ;
2. envoie cette requête dans un paquet UDP vers une machine munie d'un serveur NTP ;
3. attend une réponse sur le port UDP ;
4. affiche le contenu de la réponse ;
5. calcule le délai de propagation entre vous et le serveur ; prendre en compte le délai de transmission

Pour des raisons techniques, vous ne pouvez pas tester votre programme sur Internet, mais uniquement dans la France.

Exercice 2 [Découverte de serveur]

Certains supports physiques comme le réseau Ethernet utilisé dans les machines de la France, permettent de diffuser une trame à tous les ordinateurs qui y sont connectés. Cette possibilité est re-exportée par le protocole IPv4 à l'adresse . 88. [P. 3 t

- 4. Afficher les adresses des serveurs N-P ayant répondu à votre requête.
- 5. Déterminez le nombre de serveurs N-P de 1-4FR.