

RFC 868 : Time Protocol

Stéphane Bortzmeyer

<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 3 septembre 2010. Dernière mise à jour le 4 septembre 2010

Date de publication du RFC : Mai 1983

<https://www.bortzmeyer.org/868.html>

Bon exemple d'une ère révolue, où les RFC pouvaient faire deux pages (copyright et autre "boiler-plate" compris), et spécifier un protocole qui s'implémentait en quinze minutes. Celui-ci décrit le protocole "Time", qui permettait de récupérer l'heure d'une machine distante.

Le but était de pouvoir synchroniser plus ou moins les horloges, en interrogeant celle d'une machine distante. Certes, la précision d'une telle mesure est faible (car on ne connaît pas le temps de retour de l'information, celle-ci est donc déjà dépassée lorsqu'elle arrive) mais cela convenait, à cette époque où NTP était loin dans le futur (NTP est aujourd'hui normalisé dans le RFC 5905¹).

Difficile de faire plus simple que ce protocole "Time". Comme beaucoup de protocole de cette époque (finger, whois, ...), il a un numéro de port réservé, 37. Il suffit de se connecter à la machine distante, sur ce port (en TCP ou en UDP) et on récupère un entier de 32 bits (le RFC ne prend pas la peine de préciser s'il est gros-boutien ou petit-boutien... ; ni d'ailleurs si l'entier est signé ou non) qui indique le nombre de secondes écoulées depuis le 1er janvier 1900 à zéro heure (UTC, même si le RFC utilise encore le vieux terme anglo-saxon-centriste de GMT ; à l'époque, les étrangers restaient à leur place et ne réclamaient pas un temps « universel »). Au fait, pourquoi 1900, époque où les ordinateurs n'existaient pas, plutôt que 1970 qui aurait été plus logique ? Mystère. Les commentaires dans le source d'OpenBSD sont amusants :

1. Pour voir le RFC de numéro NNN, <https://www.ietf.org/rfc/rfcNNN.txt>, par exemple <https://www.ietf.org/rfc/rfc5905.txt>

```

/*
 * Return a machine readable date and time, in the form of the
 * number of seconds since midnight, Jan 1, 1900. Since gettimeofday
 * returns the number of seconds since midnight, Jan 1, 1970,
 * we must add 2208988800 seconds to this figure to make up for
 * some seventy years Bell Labs was asleep.
 */
u_int32_t
machtime(void)
{
    struct timeval tv;

    if (gettimeofday(&tv, NULL) < 0)
        return (0L);

    return (htonl((u_int32_t)tv.tv_sec + 2208988800UL));
}

```

Si vous êtes fort en calcul, vous avez déjà vu que 32 bits permettront d’aller jusqu’en 2036 (cette valeur de la date limite indique d’ailleurs que les auteurs du RFC pensaient à des entiers non signés, malgré le curieux exemple d’une valeur négative que donne le RFC à un moment). Après, il faudra se résigner à abandonner complètement ce protocole.

En attendant, il existe des mises en œuvre de ce RFC pour tous les Unix, typiquement en passant par `inetd`. Le service est en général coupé par défaut, car il n’a plus guère d’utilité pratique aujourd’hui. Sur une Debian, on trouve dans `/etc/inetd.conf` :

```

#time          stream  tcp     nowait  root    internal
#time          dgram  udp     wait    root    internal

```

et on peut activer le service en décommentant la ligne `time` et en rechargeant `inetd`.

Côté client, le protocole est binaire (son frère, “*DayTime*”, normalisé dans le RFC 867, est, lui, en mode texte lisible par un humain) donc `telnet` n’est pas très utile. Voici un petit programme client en Go qui se connecte à un serveur “*Time*” et indique l’heure qu’il y a sur cette machine distante :

```

% ./time-rfc-868 foo.bar.example
Time at foo.bar.example:37 is 3492608823 (which is Sat Sep  4 17:07:03 UTC 2010)

```

Le source est en (en ligne sur <https://www.bortzmeyer.org/files/time-rfc-868.go>). Merci à Kim-Minh Kaplan pour son débogage. On peut mettre des bogues dans un programme aussi court.

Pour revenir aux entiers signés ou non, le RFC donne un exemple de valeur négative, qui n’est pas compatible avec les autres données du document. Cela a fait l’objet d’un rapport de bogue <http://www.rfc-editor.org/errata_search.php?eid=2528>.