

RFC 7164 : RTP and Leap Seconds

Stéphane Bortzmeyer

<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 1 avril 2014

Date de publication du RFC : Mars 2014

<http://www.bortzmeyer.org/7164.html>

Les secondes intercalaires sont une source de complication et de polémique depuis longtemps. Voilà qu'elles affectent même le protocole de distribution de multimédia RTP : si on regarde un film, ou qu'on a une communication téléphonique, au moment où une seconde intercalaire est ajoutée, RTP va mal le vivre. Ce RFC propose donc un léger changement dans la façon dont RTP tient compte du temps, afin de rester heureux même pendant les secondes intercalaires.

RTP est normalisé dans le RFC 3550¹. Il est conçu pour l'envoi de flux multimédias (audio ou vidéo) entre un émetteur et un récepteur. Les caractéristiques propres des flux multimédias font que TCP n'est pas très adapté, donc RTP utilise UDP. UDP, par défaut, ne prévoit pas de rétroaction (l'émetteur ne sait pas ce que le récepteur a reçu) donc RTP ajoute des fonctions permettant d'informer son pair de ce qui se passe. Certains flux étant temps réel, ces informations ("*sender reports*" et "*receiver reports*", section 6.4 du RFC 3550) incluent des estampilles temporelles. Un des champs de ces informations, "*NTP timestamp*", fait référence à une horloge absolue (l'horloge située sur le mur, synchronisée avec l'échelle UTC). Une horloge qui a des soubresauts, les secondes intercalaires. Or, jouer proprement un flux audio ou vidéo nécessite au contraire une horloge sans à-coups.

La section 3 de notre RFC est un rappel sur les secondes intercalaires. Il y a deux échelles de temps standard. L'une est bien adaptée à l'informatique, à l'Internet et aux activités scientifiques en général, c'est TAI, une échelle fondée sur un phénomène physique (les atomes de césium dans une horloge atomique) et où le temps est strictement croissant et prévisible. Une autre échelle est bien adaptée à l'organisation des pique-niques et autres activités au grand air, c'est UTC. La référence d'UTC est la rotation de la Terre. Comme la Terre, au contraire des atomes de césium, est imparfaite, UTC ne garantit pas que le temps progresse régulièrement, et n'est pas prévisible (je ne peux pas dire aujourd'hui quelle sera l'heure UTC dans cinq cents millions de secondes, alors que c'est trivial en TAI). Depuis 1972,

1. Pour voir le RFC de numéro NNN, <https://www.ietf.org/rfc/rfcNNN.txt>, par exemple <https://www.ietf.org/rfc/rfc3550.txt>

UTC est définie comme étant TAI plus la correction des secondes intercalaires, qui ajoutées de temps en temps, permettent de compenser les variations de la rotation de la Terre. Ces secondes intercalaires sont forcément le dernier jour du mois et, en pratique, ont toujours été fin décembre ou fin juin (cf. texte UIT TF.460 <<http://www.itu.int/rec/R-REC-TF.460>>). Elles sont annoncées moins d'un an à l'avance dans le bulletin C <<http://datacenter.iers.org/web/guest/eop/-/somos/5Rgv/product/16>> (rappelez-vous que la rotation de la Terre n'est pas prévisible), par l'IERS.

S'il y a une seconde intercalaire, elle est forcément à la fin de la journée. Elle peut être positive (la dernière seconde de la journée est doublée, et s'affiche « 23 h 59 mn 60 s ») ou négative (la dernière seconde de la journée est omise). Jusqu'à présent, toutes les secondes intercalaires étaient positives, puisque la Terre ralentit (le jour devient plus long). À noter qu'il y a depuis des années une discussion sur la suppression des secondes intercalaires (ce qui serait une mauvaise idée <<http://seenthis.net/messages/32168#message32940>>), mais qu'elle n'est pas encore close. Voir mon article sur ce sujet <<http://www.bortzmeyer.org/leap-second-or-not-leap-second.html>> et le document de l'UIT <<http://www.itu.int/pub/R-QUE-SG07.236-2001>>.

La référence de temps la plus commune sur l'Internet, NTP (RFC 5905) est en UTC (ce qui est une mauvaise idée, selon moi, les ordinateurs n'ayant pas besoin de savoir s'il fait jour ou pas). Au moment de la seconde intercalaire positive, l'horloge NTP ralentit (ce qui permet au moins de s'assurer que deux mesures successives donneront un temps strictement croissant) ou s'arrête, ce qui fait que la dernière seconde du jour dure deux secondes. Pour tout processus temps-réel, comme RTP, c'est un gros problème.

Les applications obtiennent en général l'heure via une interface normalisée POSIX. L'échelle de temps de POSIX est UTC, mais sans vraie spécification de la gestion des secondes intercalaires. En pratique, les différentes mises en œuvre de POSIX ont des comportements différents. Par exemple, certaines répètent l'heure, pendant la seconde intercalaire et deux lectures successives du temps à une seconde d'intervalle, peuvent donc donner une heure identique. Pire, l'horloge peut même aller en arrière (ce qu'UTC ne fait pas). D'autres ralentissent l'horloge, permettant une adaptation en douceur (et pas de répétition) mais cela n'est pas idéal non plus pour des applications temps-réel! (Un bon article sur les problèmes d'utiliser une échelle de temps à secondes intercalaires est donné dans un excellent article de Google <<http://googleblog.blogspot.com/2011/09/time-technology-and-leaping-seconds.html>>.)

Conséquences de ces comportements? La section 4 note que l'envoyeur et le récepteur RTP peuvent donc avoir une seconde entière de décalage, ce qui est beaucoup pour la plupart des applications RTP. Cela peut mener à l'abandon de paquets RTP bien reçus mais considérés comme trop vieux, ou bien au contraire à la conservation de paquets dans les tampons d'entrée/sortie même lorsqu'ils ne sont plus utiles. Cela peut se traduire par une coupure du son ou de l'image. Certains récepteurs reconnaîtront un problème et se resynchroniseront avec l'émetteur. Les autres devront attendre.

La section 5 liste les recommandations pour éviter ce problème. D'abord, elle note que les récepteurs et émetteurs qui ne sont pas synchronisés avec l'horloge au mur, mais qui se contentent de mesurer le temps écoulé depuis leur démarrage n'auront jamais le problème. Ensuite, la section 5 note aussi que les émetteurs et récepteurs RTP qui utilisent une échelle de temps sans secondes intercalaires (évidemment TAI mais aussi IEEE 1588 ou GPS) n'ont également pas de problèmes. Mais pourquoi diable RTP n'utilise pas systématiquement TAI? Cette échelle de temps a toutes les propriétés souhaitables pour la communication au sein d'un réseau informatique. Le RFC ne le dit pas, mais c'est probablement parce que TAI est, hélas, trop difficile d'accès sur la majorité des plate-formes. NTP et l'horloge du système fournissent un temps qui convient à beaucoup d'activités humaines, mais ne convient pas aux ordinateurs. Et il n'y a souvent pas de moyen simple d'accéder à TAI ou une autre échelle prédictible. (C'est le cas des fonctions POSIX `gettimeofday()` et `clock_gettime()`.)

Pour les autres, les mises en œuvre de RTP qui doivent utiliser une échelle de temps qui a des secondes intercalaires, elles sont dans l'obligation de gérer ces secondes intercalaires. Cela implique d'abord un moyen de les connaître à l'avance (par exemple via le champ "*Leap Indicator*" de NTP). Ce n'est pas forcément réaliste (bien des machines reçoivent l'heure par des mécanismes qui n'indiquent pas l'approche d'une seconde intercalaire). Une solution un peu crade mais qui marche est alors de prendre les mesures citées plus loin tous les mois, lors de la fin du dernier jour du mois.

Les secondes intercalaires négatives ne posent aucun problème. Mais, on l'a vu, toutes les secondes intercalaires jusqu'à présent ont été positives. Dans ce cas, le RFC recommande que les logiciels RTP n'envoient **pas** de "*sender reports*" pendant les deux secondes où le temps est ambigu, car une seconde intercalaire a été ajoutée. Et les récepteurs doivent ignorer de tels messages. Ce changement aux règles du RFC 3550 est la principale recommandation concrète de ce RFC. Elle ne semble pas encore mise en œuvre dans un seul logiciel RTP. Est-ce que le problème, quoique joli techniquement, est assez grave en pratique pour justifier ce changement ?