rfc5905

Network Time Protocol v4

npv4 est largement utilisé pour synchroniser les horloges système via un jeu de serveurs de temps distribué et de clients. Ce document décris l'architecture, le protocole, les états machine, les structures de données et les algorithmes.

Le modèle de sous-réseau NTP inclus des serveurs de temps primaire largement accessibles, synchronisés par ondes radio ou filaires. Le but du protocole NTP est de transmettre les informations de temps depuis ces serveurs primaires vers des serveurs de temps secondaires et les clients via des réseaux privées et Internet. Les algorithmes d'ajustement de précision mitigent les erreurs qui peuvent résulter de problèmes réseaux, erreurs de serveurs, et d'actions hostiles possibles. Les serveurs et clients sont configurés tel que les valeurs atteignent le client depuis les serveurs primaires via le branchement de serveurs secondaires.

Le design NTPv4 surmonte les lacunes importantes de NTPv3, corrige certains bugs, et incorpore de nouvelles fonctionnalités. En particulier, Les définitions d'horodatage NTP étendus encouragent l'utilisation des types de données à virgule flottante double via l'implémentation. En résultat, la résolution de temps est meilleur qu'une nanoseconde, et la résolution de fréquence est inférieur à une nanoseconde par seconde.

Des améliorations supplémentaires incluent un nouvel algorithme de discipline qui est plus réactif aux fluctuations de fréquence hardware des horloges système. Les serveur primaire utilisant des machines modernes sont précis de l'ordre ed quelques dizaines de microsecondes. Les serveurs secondaires et clients sur les LAN rapide sont précis de l'ordre de quelques centaines de microsecondes avec des intervals d'interrogation de 1024 secondes, qui était le maximum avec NTPv3. Avec NTPv4, les serveurs et les clients sont précis de l'ordre de quelques dizaines de millisecondes avec des intervals d'interrogation jusqu'à 36 heures.

Modes d'opération

Une implémentation NTP opère comme serveur primaire, secondaire, ou client. Un serveur primaire est synchronisé sur une horloge de référence directement traçable en UTC (ex : GPS, Galileo, etc.). Tous les serveurs et clients qui sont pleinement conforme NTPv4 doivent implémenter toute la suite d'algorithmes décris dans ce document. Pour maintenir la stabilité dans de grands réseaux NTP, les serveurs secondaires devraient être pleinement conforme NTPv4. Les algorithmes alternatifs peuvent être utilisés, mais leur sortie doit être identique aux algorithmes décris dans cette spécification.

Modes de protocole

Il y a des variantes de protocole NTP : symétrique, client/serveur, et broadcast. Chacun est associé avec un mode d'association (une description de la relation entre 2 protagonistes NTP). De plus, les associations persistante sont mobilisées au démarrage et ne sont jamais démobilisées. Les associations éphémères sont mobilisées à l'arrivée d'un paquet et sont démobilisées au timeout ou sur erreur.

Association Mode	Assoc. Mode Value	Packet Mode Value
Symmetric Active	1	1 or 2
Symmetric Passive	2	1
Client	3	4
Server	4	3
Broadcast Server	5	5
Broadcast Client	6	N/A

Dans la variante client/serveur, un client persistant envoie des paquets mode 4 à un serveur qui retourne des paquets mode 3. Les serveurs fournissent la synchronisation à un ou plusieurs clients, mais n'acceptent pas la synchronisation depuis ces derniers. Un serveur peut également être une horloge de référence qui obtient son temp directement depuis une source de temps standard tel qu'un recepteur GPS ou un modem.

Dans la variante symétrique, un paire opère comme client et serveur en utilisant une association symétrique passive ou active. Une association symétrique active envoie des paquets mode 1 à un paire symétrique actif associé. Alternativement, une association passive symétrique éphémère peut être mobilisée jusqu'à l'arrivée d'un paque symétrique actif sans association. Cette association envoie des paquets mode 2 et persiste jusqu'à une erreur ou un timeout. Les paires envoient et reçoivent la synchronisation entre-eux. Pour ce document, un paire opére comme un client, donc les références à un client implique un paire également.

Dans la variante broadcast, une association serveur broadcast persistante envoie périodiquement des paquets mode 5 qui peuvent être reçus par plusieurs clients. À la réception d'un tel paquet sans association correspondante, une association client broadcast éphémère (mode 6) est mobilisée et persiste jusqu'à une erreur ou un timeout. Il est utile de fournir un initial quand le client opère en mode client et échange de nombreux paquets avec le serveur, donc pour calibrer le délai de propagation et pour lancer le protocole Autokey security, après lequel le cient revient en mode client broadcast. Un serveur broadcast pousse la synchronisation aux clients et autres serveurs.

En suivant les conventions établies par l'industrie du téléphone, le niveau de chaque serveur dans la hiérarchie est définis par un numéro de strate. Les serveurs primaires sont assignés au strate 1. Plus le niveau de strate augmente, plus sa précision se dégrade, dépendant de la qualité du réseau et de la stabilité de l'horloge. Les erreurs, mesurés par les distances de synchronisation, augmentent approximativement en proportion aux numéro de strate et délais mesurés.

Comme pratique standard, timer la topologie réseau devrait être organisé pour éviter les boucles de temps et minimiser la distance de synchronisation. Dans NTP, la topologie de sous-réseau est déterminée en utilisant une variante de l'algortihme de routage distribué Bellman-Ford, qui calcule le chemin de plus cours partant des serveurs primaires. L'algorithme réorganise automatiquement le sous-réseau, pour produire le temps de plus précis et le plus fiable, même quand il y a des erreurs dans le timing du réseaux

Découverte de serveur dynamique

Il y a 2 associations spéciales, les clients manycast, et les serveurs manycast, qui fournissent une fonction de découverte de serveur dynamique. Il y a 2 types d'association de client manycast : persistant et éphémère. Le client manycast persistant envoie des paquets client mode 3 à une adresse multicast ou broadcast IPv4 ou IPv6. Si un serveur est utilisable pour la synchronisation, il retourne un paquet serveur (mode 4) en utilisant l'adresse unicast du client. À la réception du paquet, le client mobilise une association client éphémère (mode 3). L'association persiste jusqu'à erreur ou timeout.

Un client manycast continue d'envoyer des paquets pour rechercher un nombre minimum d'associations. Il commence avec un TTL égal à 1 et incrémente jusqu'à ce qu'à la valeur minimale d'associations soient atteinte ou quand le TTL atteint la valeur max. Si la valeur max est atteinte et qu'il n'y a pas suffisamment d'associations, le client stoppe la transmission pendant un certain temps, efface toutes les associations mobilisées, et répète le cycle de recherche. Si un nombre minimum d'associations ont été mobilisés, le client commence à transmettre un paquet par période pour maitenir les associations. Le champ constraints limite la valeur minimum à 1 et max 255. Ces limites peuvent être définis individuellements.

Les associations éphémères sont en concurence entre-elles. À mesure que des associations éphémères sont mobilisées, le client lance les algorithme d'aténuation pour les meilleurs candidats, les associations éphémère restantes sont démobilisées. De cette manière, la population inclus seulement les meilleurs candidats qui ont répondus le plus récemment avec un paquet NTP.

Définitions

Certains termes technique sont définis dans cette section. Un timescale est une trame de référence où de temps est exprimé comme valeur d'un compteur binaire augmentant monotoniquement avec un nombre indéfinis de bits. Il compte en secondes et fractions de secondes, quand un point décimal est employé. Le timescale UTC est définis par ITU-R TF.460.

UTC représente le temps solaire tel que disséminé par les laboratoires standards nationaux. Le temps système est représenté par l'horloge système maintenu par le hardware et le système d'exploitation. Le but des algorithmes NTP est de minimiser la différence de temps et la fréquence entre UTC et l'horloge système. Quand ces différences sont réduites sous la tolérance nominale, l'horloge système est dite synchronisée à UTC.

La tade d'un événement est le temp UTC auquel l'événement UTC prend place. Les dates sont des valeur éphémères désignées pavec T majuscule. Le temps courant est un autre timescale qui coïncide avec la fonction de synchronisation NTP.

Un timestamp T(t) représente soit la date UTC ou un décalage de temps depuis UTC au temps t. Disons T(t) étant l'offset de temp, R(t) l'offset de fréquence, et D(t) l'âge moyen (d'abord dérivé de R(t) en respect de t). Alors, si T(t_0) est l'offset de temps UTC déterminé à t = t_0, l'offset de temps UTC au temps t est :

```
T(t) = T(t_0) + R(t_0)(t-t_0) + 1/2 * D(t_0)(t-t_0)^2 + e
```

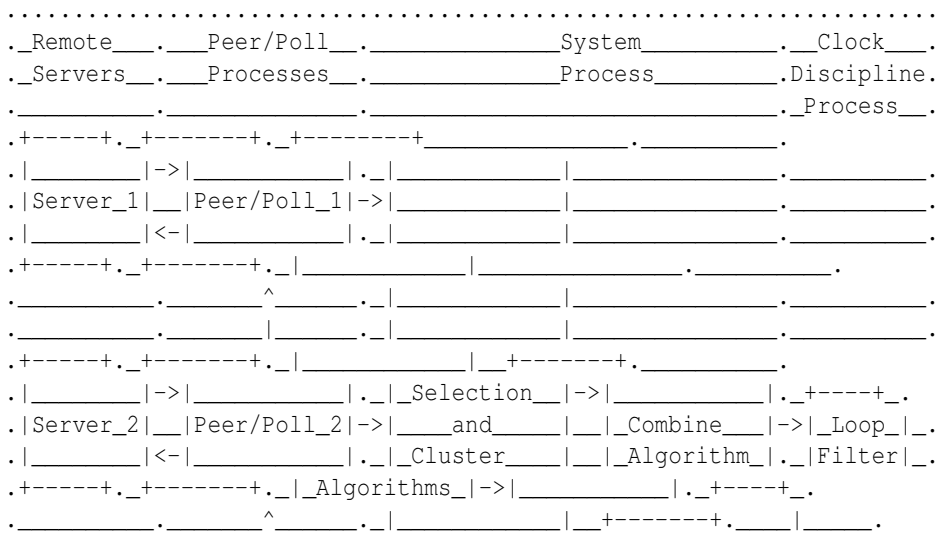
où e est une erreur stochastique discuté ultérieurement. Bien que D(t) est important en caractérisant les oscilateurs de précision, il est ordinairement négligé pour les oscilateurs des ordinateurs. Dans ce document, toutes les valeurs de temps sont en seconde (s) et toutes les fréquences sont en secondes-par-secondes (s/s). C'est souvent convenable pour exprimer les offsets de fréquence en part-par-million (ppm), où 1 ppm est égal à 10^(-6) s/s

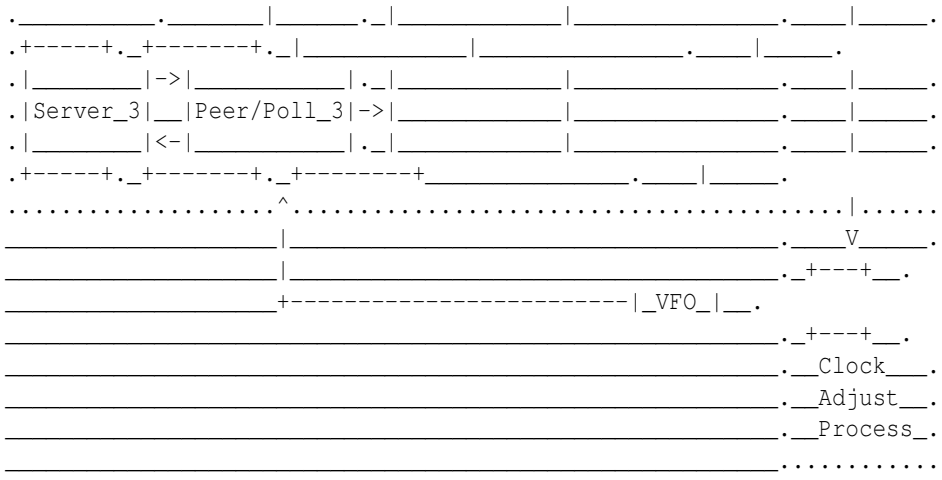
Il est important dans les application de gestion de temps d'évaluer les performance de la fonction de timekeeping. Le modèle de performance NTP inclus 4 statistiques qui sont mis à jours chaque fois qu'un cilent fait une mesure avec un serveur. L'offset (theta) représente l'offset de temp maximum de l'horloge serveur relatif à l'horloge système. Le délai (delta) représente le délai entre le client et le serveur. La dispersion (epsilon) représente l'erreur maximum inhérente à la mesure. Cela augmente à un taux égal au maximum de tolérance de fréquence de l'horloge système disciplinée (PHI), typiquement 15 ppm. Le jitter (psi) est définis comme moyenne root-mean-square (RMS) des différences d'offset les plus récents, et représente l'erreur nominale en estimant l'offset.

Bien que les statistiques theta, delta, epsilon, et psi représentent des mesures d'horloge système relatives à chaque serveur de temps séparémment, le protocole NTP inclus des mécanismes pour combiner les statistiques de nombreux serveurs pour une discipline plus précise et pour calibrer l'horloge système. L'offset système (THETA) représente l'offset max estimé pour la population serveur. Le jitter système (PSI) représente l'erreur nominale en estimant l'offset système. Les statistiques delta et epsilon sont accumulées à chaque niveau de strate depuis l'horloge de référence pour produire le délai racine (DELTA) et la dispersion racine (EPSILON). La distance de synchronisation (LAMBDA) égal à EPSILON + DELTA / 2 représente l'erreur maximum du à toutes les causes.

Modèle d'implémentation

La figure ci-dessous montre l'architecture d'une implémentation multi-threadée typique. Elle inclus 2 processus dédiés à chaque serveur, un processus paire pour reçevoir les messages du serveur ou de l'horloge de référence, et un processus d'intérrogation pour transmettre des messages au serveur ou à l'horloge de référence.





Ces processus opèrent sur une structure de données commune, appelée association, qui contient les statistiques décrites plus haut, ainsi que d'autres données décrites dans la section "Peer Process". Un client envoie des paquets à un ou plusiers serveurs et traite les paquets retournés à leur réception. Le serveur interchange les adresses source et de destination et les ports, change certains champs dans le paquet et le retourne immédiatement (dans le mode client/serveur) ou ultérieurement (dans les modes symétriques). Comme chaque NTP message est reçu, l'offset theta entre l'horloge du paire et l'horloge système est calculée avec les statistiques associées delta, epsilon, et psi.

Le processus système inclus la sélection, cluster, et les algorithmes de combinaison qui mitigent les différents serveurs et horloges de référence pour déterminer le candidat le plus fiable et le plus précis pour synchroniser l'horloge système. L'algorithme de sélection utiliser les principes de détection de faut Byzantine pour supprimer les candidats présumés incorrects appelés "falsetickers" de la population, laissant seulement les bons candidats appelés les "truechimers". Un truechimer est une horloge qui maintient un timekeeping précis d'un standard précédemment publié et validé, alors qu'un falseticker est une horloge qui montre un temps inconsistant. L'algorithme cluster utilise les principes statistiques pour trouver le jeu de truechimers les plus précis. L'algorithme combine calcule l'offset d'horloge final en créant une moyenne statistique des truechimers.

Le processus de discipline d'horloge est un processus système qui contrôle le temps et la fréquence de l'horloge système, ici représenté comme un oscilateur à fréquence variable (VFO). Les timestamps données par le VFO ferment la boucle qui maintient le temps système. Associé avec le processus de discipline d'horloge on trouve le processus d'ajustement d'horloge, qui se lance une fois par secondes pour injecter un offset de temps calculé et maintient la fréquence constante. La moyenne RMS de la différence d'offset de temps passé représente l'erreur nominale (system clock jitter). La moyenne RMS des différences d'offset de fréquence passées représente l'échelonnement de fréquence de l'oscilateur (frequency wander).

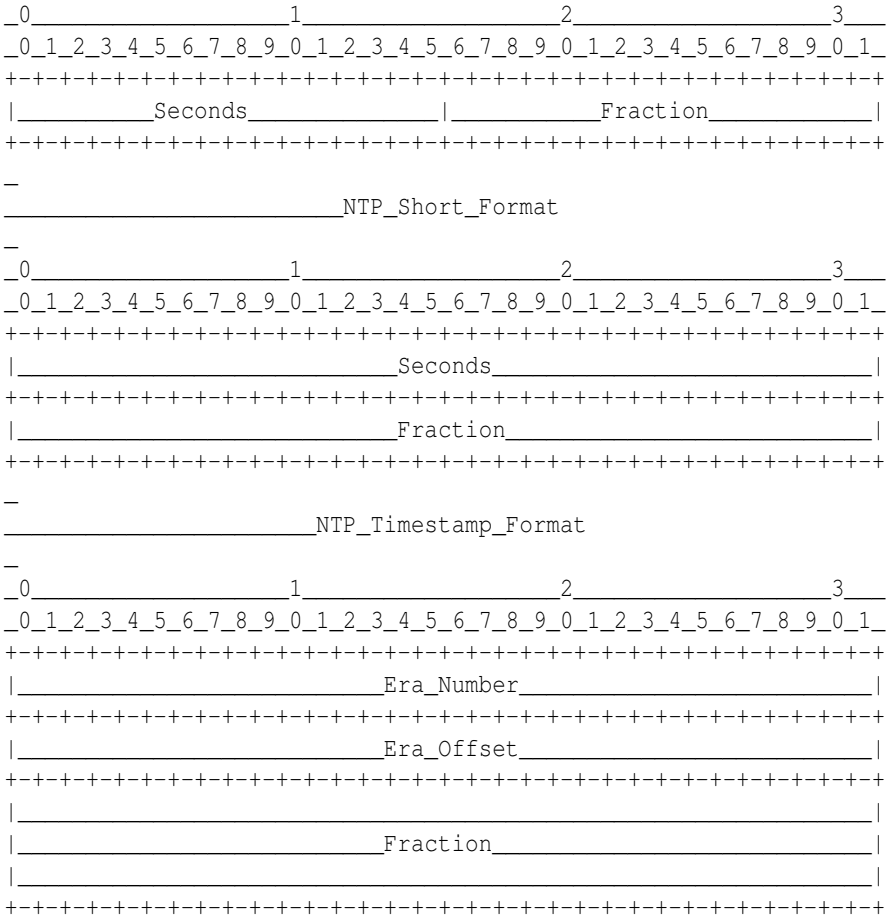
Un client envoie des messages à chaque serveur avec un interval d'interrogation de 2^tau secondes, comme déterminé par l'exposant d'interrogation pau. Dans NTPv4, les plages tau vont de 4 (16s) à 17 (36h). La valeur de tau est déterminée par l'algorithme de discipline d'horloge pour correspondre à la constante de boucle de temps T_c = 2^tau. Dans le mode client/serveur, le serveur répond immédiatement ; cependant, dans les modes symétriques, chacun des 2 paires gèrent tau comme une fonction de l'offset de temps système et de jitter système, donc ils ne peuvent pas être d'accord avec la même valeur. Il est important que le comportement dynamique de l'algorithme de discipline d'horloge soit contrôlé pour maintenir la stabilité dans le sous-réseau NTP. Cela exige que les paires soient d'accord sur un tau commun égal à l'exposant d'interrogation minimum de tous les paires.

Le modèle d'implémentation inclus des moyens pour définir et ajuster l'horloge système. Le système d'exploitation est supposé fournir 2 fonctions : une pour définir le temps directement, par exemple, la fonction Unix settimeofday(), et une autre pour ajuster le temps en petits incréments avançant ou retardant le temps par une quantité définie, par exemple, la fonction Unix adjtime(). Dans ce design le processus de discipline d'horloge utilise la foncion adjtime() si l'ajustement est inférieur au seuil définis, et la fonction settimeofday() au delà.

Types de données

Toutes les valeurs de temps NTP sont représentées au format complément de 2, avec les bits numérotés en big-endian. Il y a 3 formats de temps NTP, un format de date 128bits, un format d'horodatage 64bits, et un format cours 32bits. Le format de date 128 bits est utilisé quand

un stockage suffisant et la taille de mots sont disponibles. Il inclus le champs des secondes 64 bits signés couvrant 584 milliards d'années et un champs 64bits de fraction résolvant .05 attosecond. Par convention dans le mappage entre les formats, le champs secondes est divisé en un champ 32 bits Era Number, et nu champ 32 bits Era Offset. Les ères ne peuvent pas être produites par NTP directement, et il n'est pas nécessaire de le faire. Si nécessaire, ils peuvent être dérivés par des moyens externes, tel que le système de fichier ou un hardware dédié.



Le format 64bits est utilisé dans les en-têtes de paquet et d'autres emplacement avec une taille de mot limitée. Il inclus un champ 32 bits non-signé des secondes couvrant 126 ans et un champ 32 bits fractionnel résolvant 232 picosecondes. Le format cours 32 bits est utilisé dans les champs d'en-tête delay et dispersion où la résoluction complète n'est pas justifiée. Il inclus un champ 16 bits non-signé des secondes, et un champ 16 bits de fraction.

Dans les formats de date et d'horodatage, L'époque, ou la date de base de l'ère 0 est 0 h 1 Janvier 1900 UTC, quand tous les bits sont à zéro. Il devrait être noté que strictement parlant, UTC n'existait pas avant le 1 Janvier 1972, mais il est convenable de supposer qu'il à toujours existé. Les dates sont relatives à cette époque ; les valeurs supérieurs à 0 représentent le temps après cette date ; les valeurs inférieurs à 0 représentent les dates avant cette date.. Noter que le champ Era Offset du format date et le champ Seconds du format timestamp ont la même interprétation.

Les timestamp sont des valeurs non-signées, et les opérations sur elles produisent un résultat dans la même ère ou ère adjacente. L'ère 0 inclus les dates de l'époque premier jusqu'à une date en 2036, quand le champ timestamp entoure la date de base pour l'ère 1 est établie. Dans ces format, un valeur 0 est un cas spécial représentant un temps inconnu ou désynchronisé. La figure ci-dessous montre des dates NTP historiques avec leur Jours au calendrier Julien modifié (MJD), l'ère NTP, et l'horodatage NTP :

Date	MJD	NTP Era	NTP Timestamp Era Offset	Epoch
1 Jan -4712	-2,400,001	-49	1,795,583,104	1st day Julian
1 Jan -1	-679,306	-14	139,775,744	2 BCE

```
|_1_Jan_0_____|_-678,491___|_-14_|_171,311,744___|_1_BCE_____________|
|_1_Jan_1_____|_-678,575___|_-14_|_202,939,144___|_1_CE______________|
|_4_Oct_1582__|_-100,851___|_-3__|_2,873,647,488_|_Last_day_Julian__|
|_15_Oct_1582_|_-100,840___|_-3__|_2,874,597,888_|_First_day________|
|_____________|____________|_____|_______________|_Gregorian________|
|_31_Dec_1899_|_15019______|_-1__|_4,294,880,896_|_Last_day_NTP_Era_|
|_____________|____________|_____|_______________|_-1_______________|
|_1_Jan_1900__|_15020______|_0___|_0_____________|_First_day_NTP____|
|_____________|____________|_____|_______________|_Era_0____________|
|_1_Jan_1970__|_40,587_____|_0___|_2,208,988,800_|_First_day_UNIX___|
|_1_Jan_1972__|_41,317_____|_0___|_2,272,060,800_|_First_day_UTC____|
|_31_Dec_1999_|_51,543_____|_0___|_3,155,587,200_|_Last_day_20th____|
|_____________|____________|_____|_______________|_Century__________|
|_8_Feb_2036__|_64,731_____|_1___|_63,104________|_First_day_NTP____|
|_____________|____________|_____|_______________|_Era_1____________|
+---------+--------+---+----------+------------+
```

Disons P le nombre de bits significatifs dans une fraction de secondes. La résolution d'horloge est définis avec 2^(-p), en secondes. Pour minimiser le biais et aider à rendre les horodatages non-prédictibles par un intrus, les bits non-signifiants devraient être mis à une chaîne aléatoire non-biaisée. La précision d'horloge est définis comme temps courant pour lire l'horloge système, en secondes. Noter que la précision définis de cette manière peut être supérieure ou inférieur à la résolution. Le terme rho, représentant la précision utilisée dans le protocole, est le plus grand des 2.

La seule opération arithmétique permise sur les dates et horodatage est la soustraction en complément de 2, Donnant un résultat 127bits ou 63bits signé. Il est important de noter que l'arithmétique en complément de 2 ne distingue pas les valeurs signées et non-signées (bien que les comparaisons puissent prendre en compte le signe) ; seule les instructions de branchement conditionnels le font. Ainsi, bien que la distinction est faite entre les dates signées et non-signées, elles sont traitées de la même manière. Une perception hasardeuse avec les calculs d'horodatage 64-bits dans une ère, tels qu'il sera possible en 2036, peut résulter par un dépassement. En fait, si le client est définis dans les 68 ans du serveur avant que le protocole soit démarré, les valeurs correctes sont obtenues même si le client et le serveur sont dans des ères adjacentes.

Certaines valeurs de temps sont représentées au format exposant, incluant la précition, constante de temp, et interval d'interrogation. Il y a un entier signé 8-bits en log2 (log base 2) secondes. Les seules opérations arithmétiques permises sur eux sont l'incrément et le décrément. Dans ce document, pour simplifier la présentation, une référence à une de ces variables par nom signifie la valeur exponentielle, par exemple, l'intervalle d'interrogation est 1024 secondes, alors que la référence par nom et exposant signifie la valeur actuelle, par exemple, l'exposant d'interrogation est 10.

Pour convertir le système de temps en un autre format que NTP, il est nécessaire que le nombre de secondes s depuis l'époque première vers le temps système soit déterminé. Pour déterminer l'ère et l'horodatage s,

era = s / 2^(32) et timestamp = s - era * 2 ^(32)

qui fonctionne pour les dates positives et négatives. Pour déterminer s en donnant l'ère et le timestamp,

s = era * 2^(32) + timestamp

La conversion NTP et temps système peut être un peut désordonné, et est au-delà du périmètre de ce document. Noter que le nombre de jours dans l'ère 0 est plus que le nombre de jours dans la plupart des autres ères, et cela ne se reproduira pas jusqu'en 2400 dans l'ère 3.

Dans la description des variables d'état qui suivent, une référence explicite au type entier implique un entier non-signé 32 bits. Cela simplifie la vérification des limites, vu que seul la limite supérieure doit être définie. Sans référence explicite, le type par défaut est 64-bits double flottante.

Structures de données

Les machines d'état NTP sont définies dans les sections suivantes. Les variables d'état sont séparées en classes en accord avec leur fonction dans les en-têtes de paquet, processus paires et d'interrogation, le processus système, et le processus de discipline d'horloge. Les variables de paquet représentent les variables d'en-tête NTP dans les paquets transmis et reçus. Les variables des paires et d'interrogation représentent le contenu de l'association pour chaque serveur séparément. Les variables systèmes représente l'état du serveur tel que vu pas ses clients dépendants. Les variables de discipline d'horloge représentent le travail interne de l'algorithme de discipline d'horloge.

Conventions de structure

Pour distinguer les différentes variables de même nom, mais utilisé dans différents processus, la convention de nommage ci-dessous est adoptée. Une variable d'un paquet reçu v est un membre de la structure de paquet r avec le nom pleinement qualifié r.v. De manière similaire, x.v est une variable de paquet transmise, p.v est une variable paire, s.v une variable système, et c.v une variable de discipline d'horloge. Il y a un jeu de variables paire pour chaque association ; il y a seulement un jeu de variable d'horloge et système.

Name	Description
r.	receive packet header variable
x.	transmit packet header variable
p.	peer/poll variable
s.	system variable
c.	clock discipline variable

Paramètres globaux

En plus de ces classes de variable, un monbre de paramètres globaux sont définis dans ce document, incluant ceux affichés avec des valeurs ci-dessous :

Name	Value	Description
PORT	123	NTP_port_number
VERSION	4	NTP_version_number
TOLERANCE	15e-6	frequency_tolerance_PHI_(s/s)
MINPOLL	4	minimum_poll_exponent_(16_s)
MAXPOLL	17	maximum_poll_exponent_(36_h)
MAXDISP	16	maximum_dispersion_(16_s)
MINDISP	.005	minimum_dispersion_increment_(s)
MAXDIST	1	distance_threshold_(1_s)
MAXSTRAT	16	maximum_stratum_number

Bien que des paramètres globaux sont nécessaires pour l'interopérabilité, une collection plus grande est nécessaire dans une implémentation. L'appendix A.1.1 contient ceux utilisés par le squelette pour les algorithme d'atténuation, de discipline d'horloge, et des fonctions dépendante de l'implémentation. Certaines valeurs de paramètres sont gravés dans la pierre, comme le port NTP assigné par l'IANA et le numéro de version assigné NTPv4.

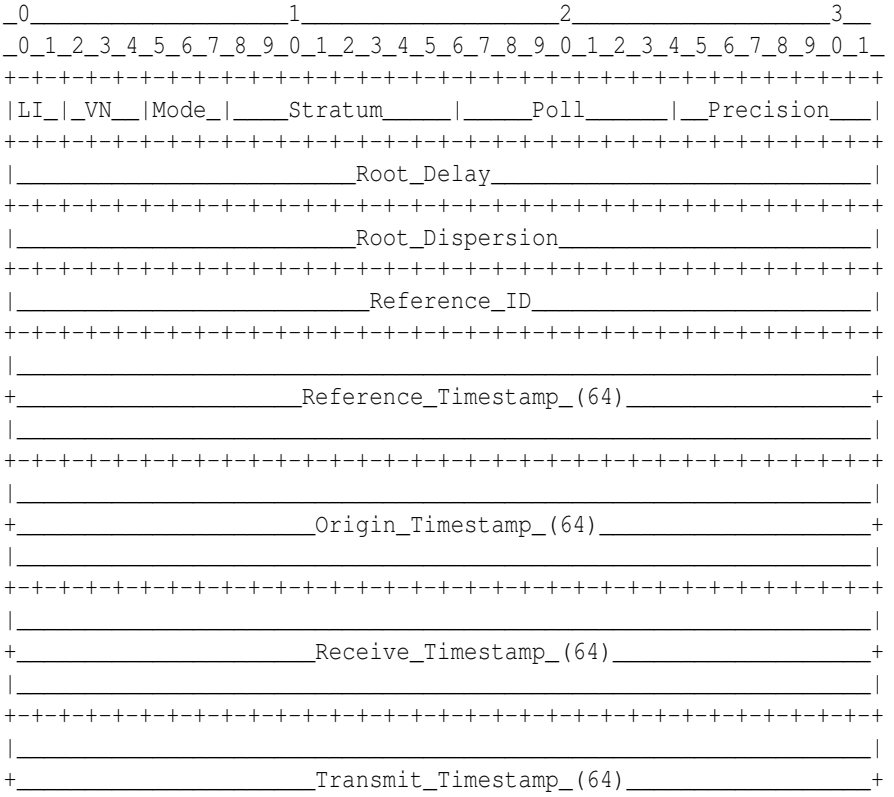
Bien qu'affiché avec des valeurs fixées dans ce document, certaines implémentations peuvent ajuster ces variables.

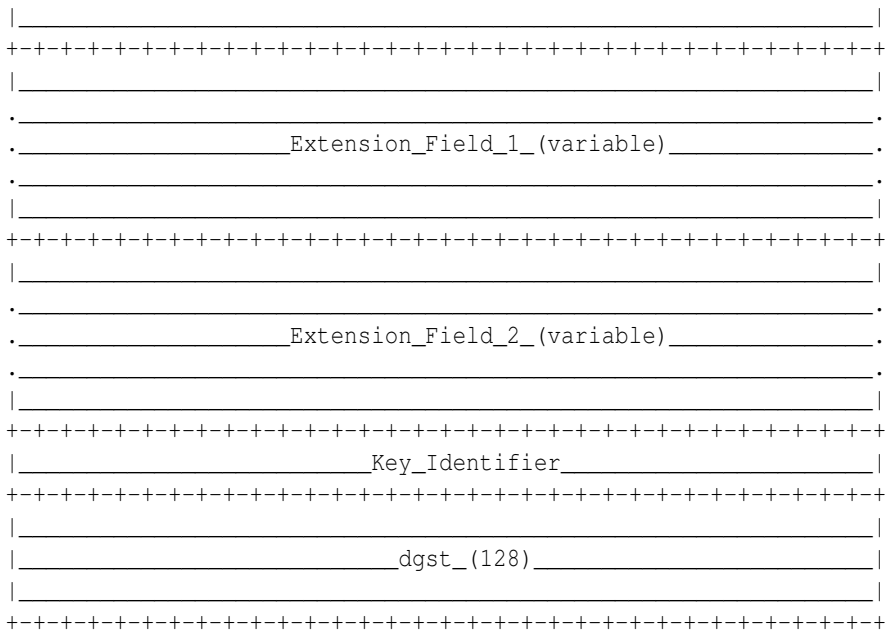
Variables de l'en-tête de paquet

Les variables d'état les plus importantes d'un point de vue externe sont les variables d'en-tête de paquet. L'en-tête NTP consiste d'un numéro entier 32bits dans l'ordre d'octet réseaux. Le format du paquet consiste de 3 composants : l'en-tête lui-même, un ou plusieurs champs d'extension, et un MAC optionnel. L'en-tête est identique à l'en-tête NTPv3. Les champs d'extensions sont utilisées par les algorithmes cryptographiques à clé publique Autokey décris dans la rfc5906. Le MAC est utilisé par Autokey et l'algorithme à clé symétrique définis dans cette rfc.

Name	Formula	Description
leap	leap	leap_indicator_(LI)
version	version	version_number_(VN)
mode	mode	mode
stratum	stratum	stratum
poll	poll	poll_exponent
precision	rho	precision_exponent
rootdelay	delta_r	root_delay
rootdisp	epsilon_r	root_dispersion
refid	refid	reference_ID
reftime	reftime	reference_timestamp
org	T1	origin_timestamp
rec	T2	receive_timestamp
xmt	T3	transmit_timestamp
dst	T4	destination_timestamp
keyid	keyid	key_ID
dgst	dgst	message_digest

Le paquet NTP est un datagramme UDP. Certains champs utilisent plusieurs mots et d'autres sont empaquetés dans des champs plus petits dans un mot. L'en-tête de paquet NTP ci-dessous a 12 mots suivi par des champs d'extension optionnels et finalement un MAC consistant du champ Key Identifier et du champ Message Digest





Les champs d'extension sont utilisé pour ajouter de capacités optionnelles, par exemple, le protocole de sécurité Autokey. Le format du champ d'extension est présenté dans l'ordre pour que le paquet soit parcouru sans connaissance des fonctions du champ d'extension.

Excepté pour une variation mineur en utilisant la famille d'adresse IPv6, ces champs sont compatibles avec NTPv3. Les champs d'en-tête de paquet s'appliquent aux paquets transmis et reçus. Les champs et variables de paquet associés (entre parenthèses) sont interprétés comme suit :

LI Leap Indicator (leap) : Entier 2-bits d'indication d'alerte d'une seconde à insérer ou supprimer dans la dernière minute du mois courant :

Valeur	Signification
0	pas d'alerte
1	la dernière minute a 61 secondes
2	la dernière minute a 59 secondes
3	inconnu (horloge désynchronisée)

VN Numéro de version sur 3bits, actuellement 4

Mode : entier 3bits représentant le mode :

Valeur	Signification
0	Réservé
1	Actif symétrique
2	Passif symétrique
3	client
4	server
5	broadcast
6	Message de contrôle NTP
7	Réservé

Stratum : Entier 8bits représentant la strate :

Valeur	Signification
0	Non-spécifié ou invalide
1	Serveur primaire, équipé avec un récepteur GPS
2-15	serveur secondaire (via NTP)

16_____Désynchronisé
17-255_Réservé

Il est courant de mapper la valeur de strate 0 dans les paquets reçus à MAXSTRAT (16) dans la variable paire p.stratum et de mapper les valeurs p.stratum de MAXSTRAT ou supérieur à 0 dans les paquets transmis. Cela permet aux horloges de référence, qui normalement apparaîssent à la strate 0, d'être convenablement mitigée en utilisant les même algorithmes de sélection d'horloge utilisé pour les sources externe.

Poll Entier signé 8-bits représentant l'interval maximum entre les messages successifs, en secondes log2. Les limites suggérées pour ces interval minimun et maximum sont 6 et 10.

Precision : Entier 8-bits représentant la précision de l'horloge système, en secondes log2. Par exemple, une valeur de -18 correspond à une précision d'environ une microseconde. La précision peut être déterminée quand le service démarre d'abord au temps minimum des nombreuses itérations pour lire l'horloge système.

Root Delay Délai total pour l'horloge de référence, au format NTP court

Root Dispersion Dispersion totale de l'horloge de référence, au format NTP court.

Reference ID Code 32-bits identifiant le serveur particulier ou l'horloge de référence. L'interprétation dépend de la valeur dans le champ stratum. Pour les paquets de strate 0, c'est une chaîne ASCII 4 caractères, apperé le 'kiss code', utilisé pour débugger ou superviser. Pour un strate 1, c'est une chaîne ASCII 4 octets assigné à l'horloge de référence. La liste des identifiants de référence est maintenue par l'IANA :

```
+----+-------------------------------------+
|_ID___|_Clock_Source_____________________________________________|
+----+-------------------------------------+
|_GOES_|_Geosynchronous_Orbit_Environment_Satellite_______________|
|_GPS__|_Global_Position_System___________________________________|
|_GAL__|_Galileo_Positioning_System_______________________________|
|_PPS__|_Generic_pulse-per-second_________________________________|
|_IRIG_|_Inter-Range_Instrumentation_Group_________________________|
|_WWVB_|_LF_Radio_WWVB_Ft._Collins,_CO_60_kHz_____________________|
|_DCF__|_LF_Radio_DCF77_Mainflingen,_DE_77.5_kHz__________________|
|_HBG__|_LF_Radio_HBG_Prangins,_HB_75_kHz_________________________|
|_MSF__|_LF_Radio_MSF_Anthorn,_UK_60_kHz__________________________|
|_JJY__|_LF_Radio_JJY_Fukushima,_JP_40_kHz,_Saga,_JP_60_kHz_______|
|_LORC_|_MF_Radio_LORAN_C_station,_100_kHz_________________________|
|_TDF__|_MF_Radio_Allouis,_FR_162_kHz_____________________________|
|_CHU__|_HF_Radio_CHU_Ottawa,_Ontario_____________________________|
|_WWV__|_HF_Radio_WWV_Ft._Collins,_CO_____________________________|
|_WWVH_|_HF_Radio_WWVH_Kauai,_HI__________________________________|
|_NIST_|_NIST_telephone_modem_____________________________________|
|_ACTS_|_NIST_telephone_modem_____________________________________|
|_USNO_|_USNO_telephone_modem_____________________________________|
|_PTB__|_European_telephone_modem_________________________________|
+----+-------------------------------------+
```

Sous une strate 1 (les serveurs secondaires et clients), c'est l'identifiant de référence du serveur et peut être utilisé pour détecter les boucles de temps. Avec IPv4, l'identifiant est une adresse IPv4. Avec IPv6, ce sont les 4 premiers octets du hash MD5 de l'adresse IPv6.

Reference Timestamp Temps quand l'horloge système a été définie ou corrigée, au format timestamp NTP

Origin Timestamp Temps auquel le client à envoyé la requête au serveur, au format timestamp NTP.

Receive Timestamp Temp auquel le serveur a reçu la requête du client

Transmit Timestamp Temp auquel le serveur en émit la réponse au client

Destination Timestamp Temp auquel le client a reçu la réponse du serveur

Note : le champ Destination Timestamp n'est pas inclus comme champ d'en-tête, il est déterminé à l'arrivée du paquet et disponible dans la structure de données du tampon du paquet.

Si NTP a accès à la couche physique, les timestamp sont associés avec le début du symbole après le début de la trame. Sinon, les implémentations devraient tenter d'associer le timestapm au premier point accessible dans la trame.

Le MAC consiste de l'identifiant de clé suivi par le Hash. Le hash, ou cryptosum, est calculé comme dans la rfc1321 sur tous les en-têtes NTP et les extensions de champ optionnels, mais pas le MAC lui-même

Extension Field n Voir plus bas pour une description du format de ce champ

Key Identifier Entier non-signé 32-bits utilisé par le client et le serveur pour désigner une clé MD5 128-bits secrète

Message Digest Hash MD5 128-bits calculé

Il devrait être noté que le calcule MAC utilisé ici diffère de ceux définis dans les rfc1305 et rfc4330, mais est consistant avec la manière dont les implémentations existantes génèrent un MAC.

Paquet Kill-o'-Death

Si le champ stratum est 0, le champ Reference Identifier peut être utilisé pour transporter des messages utiles pour reporter le status et le contrôle d'accès. Ils sont appelés des paquets KoD (Kiss-o'-Death) et les messages ASCII qu'ils transportent sont appelés des codes Kiss. Les paquets KoD prennent leur nom de leur première utilisation qui consistait à dire à un client d'arrêter d'envoyer des paquets qui violent les contrôles d'accès au serveur. Les codes Kiss peuvent fournir des informations utiles pour un client intelligent, soit NTPv4 soit SNTPv4. Les codes Kiss sont encodés en chaînes ASCII 4 caractères. Une liste de codes définis est donné ci-dessous. Les destinataires des Kiss codes doivent les inspecter et, dans les cas suivants, effectuer ces actions :

a. Pour DENY ou RSTR, le client doit démobiliser toutes associations avec ce serveur et arrêter d'envoyer des paquets à ce serveur.

b. Pour RATE, le client doit immédiatement réduit sont interval d'intérrogation vers ce serveur et continuer à le réduire chaque fois qu'il reçoit un RATE

c. Les Kiss codes commençant avec X sont pour des expérimentation et développement et doivent être ignorés s'ils ne sont pas retenus

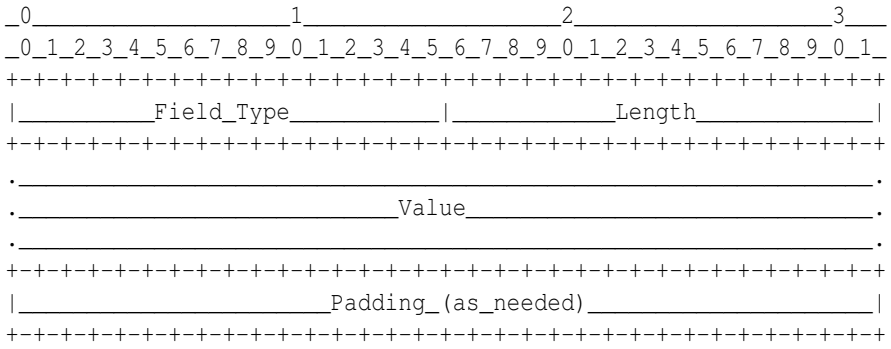
d. Autre que les conditions ci-dessus, les paquets KoD n'ont pas de signification de protocole et sont supprimés après inspection

Code	Meaning
ACST	The association belongs to a unicast server.
AUTH	Server authentication failed.
AUTO	Autokey sequence failed.
BCST	The association belongs to a broadcast server.
CRYP	Cryptographic authentication or identification failed.
DENY	Access denied by remote server.
DROP	Lost peer in symmetric mode.
RSTR	Access denied due to local policy.
INIT	The association has not yet synchronized for the first time.
MCST	The association belongs to a dynamically discovered server.
NKEY	No key found. Either the key was never installed or is not trusted.
RATE	Rate exceeded. The server has temporarily denied access because the client exceeded the rate threshold.
RMOT	Alteration of association from a remote host running ntpdc.
STEP	A step change in system time has occurred, but the association has not yet resynchronized.

Le Receive Timestamp et Transmit Timestamp ne sont pas définis dans un paquet KoD et ne doivent pas être considéré par les clients.

Format de champ d'extension NTP

Dans NTPv4, un ou plusieurs champs d'extension peuvent être insérés après l'en-tête et avant le MAC, qui est toujours présent quand un champ d'extension est présent. Ce document ne définis que le format du champ, pas son contenu. Un champ d'extension contient une requête ou un message de réponse au format :



Tous les champs d'extension sont remplis avec des 0 à des limites word (4 octets). Le champ de type de champ est spécifique à la fonction définie et n'est pas élaborée ici. Bien que la longueur de champ minimum contenant les champs requis est de 4 words, une longueur maximum reste à établir.

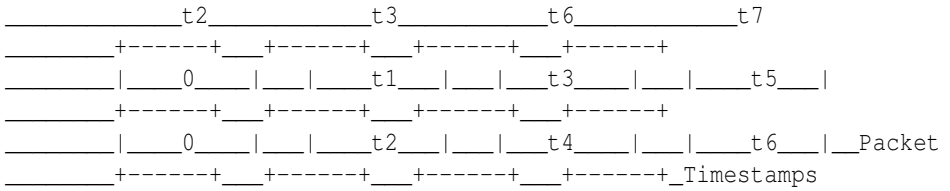
Le champ Length est un entier 16-bits non-signé indiquant la longueur du champs d'extension en octets, incluant le padding.

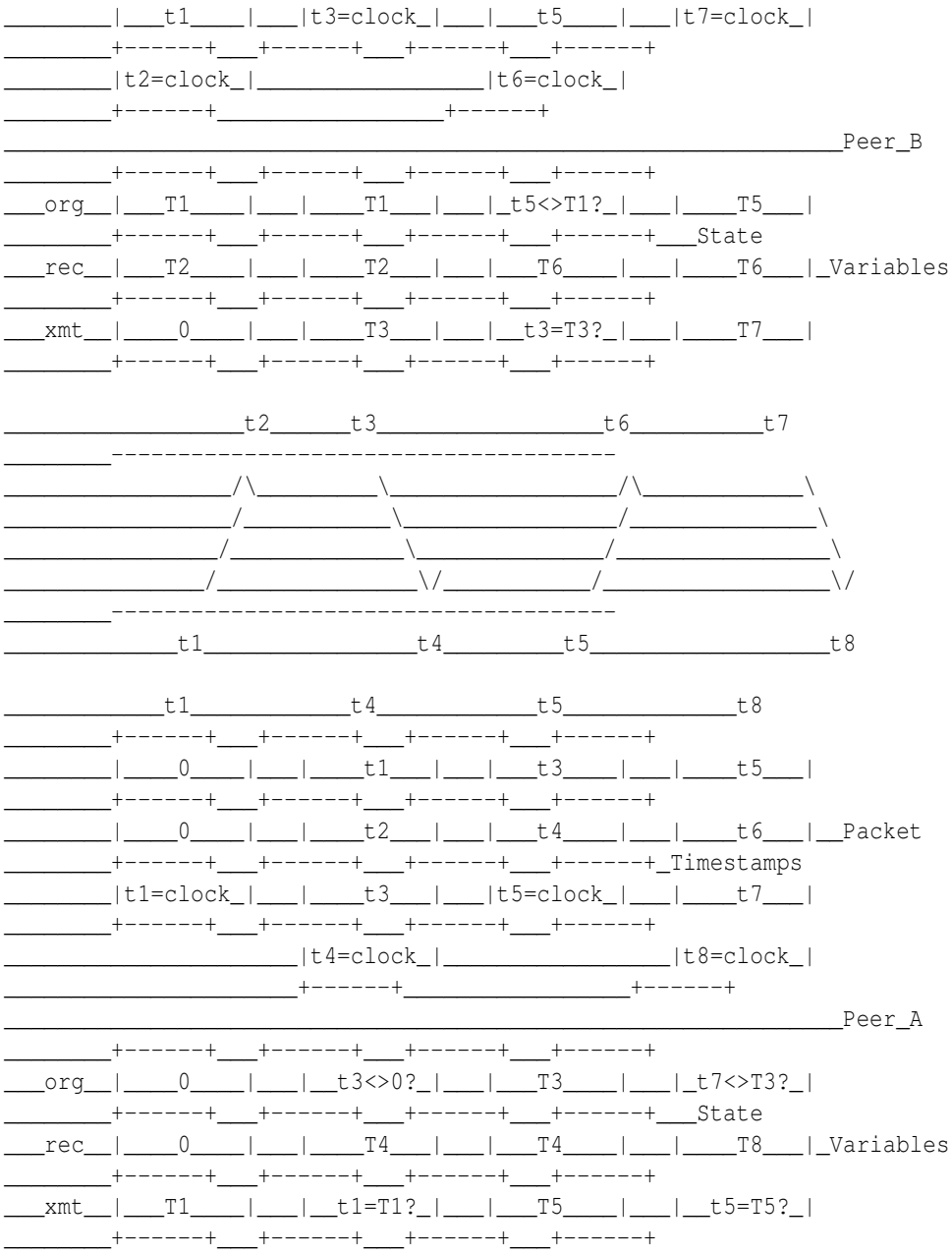
Protocol On-Wire

Le cœur du protocole NTP on-wire est le mécanisme core qui échange les valeurs de temps entre les serveurs, paires, et client. Il est résistant à la perte de paquets ou de paquets dupliqués. L'intégrité des données est fournie par les checksums IP et UDP. Aucun contrôle de flux ou de facilité de retransmition ne sont fournis ou nécessaires. Le protocole utilise les timestamp, qui sont soit extraits des en-tête de paquet ou pris depuis l'horloge système une fois reçu ou envoyés. Les timestamps sont des données de précision et devrait être frappés en acs de retransmition L2 et corriger le temps de calcule du MAC à transmettre.

Les messages NTP utilisent 2 modes de communication différents, one-to-one, et one-to-many, communément référré à unicast et broadcast. Pour ce document, le terme broadcast est interprété comme mécanisme one-to-many. Pour IPv6, c'est équivalent au broadcast IPv4 et au multicast IPv4. Pour IPv6 c'est équivalent au multicast IPv6. L'IANA a alloué l'adresse de multicast IPv4 224.0.1.1 et l'adresse multicast IPv6 se terminant par :101, avec le préfix déterminé par les règle de scoping. Tout autre adresse multicast non-allouée peut être également utilisée en plus de ces adresses multicast.

Le protocole on-wire utilise 4 horodatages numérotés t1 à t4 et 3 variables d'état org, rec, et xmt. La figure ci-dessous montre le cas le plus général où chacun des paires, A et B, mesurent indépendamment l'offset et le delais relatif à l'autre. Pour l'illustration, Les horodatages de paquet sont affichés en minuscule, et les variables d'état sont affichés en majuscule. Les variables d'état sont copiées des horodatages de paquet à l'arrivée et au départ d'un paquet.





Le premier paquet transmis par A contient seulement le temps d'origine t1, qui est copié dans T1. B reçoit le paquet au temps t2 et copie t1 dans T1 et le temps reçu t2 dans T2. À cet instant ou ultérieurement à t3, B envoie un paquet à A contenant t1 et t2 et le temps t3. Les 3 horodatages sont copiés dans les variables d'état correspondants. A reçoit le paquet à t4 contenant les 3 horodatages t1, t2 et t3 et le temps de destination t4. Ces 4 horodatages sont utilisés pour calculer l'offset et le délai de B relatif à A, comme décris plus bas.

Avant de mettre à jours les variables d'état xmt et org, 2 vérifications sont effectuées pour éviter les duplicats, bugs, ou paquets relayés. Dans l'échange ci-dessus, un paquet est dupliqueé ou rejouté si le temps t3 correspond à la variable d'état T3. Un paquet est buggé si t1 dans le paquet ne correspond pas à la variables d'état xmt T1. Pour protéger les replay du dernier paquet transmis, la variable d'état xmt est mises à 0 immédiatement après la vérification réussie.

Les 4 horodatages les plus récents, T1 à T4, sont utilisés pour calculer l'offset de B relatif à A.
theta = T(B) - T(A) = 1/2 * [(T2-T1) + (T3-T4)]
et le délai
delta = T(ABA) = (T4-T1) - (T3-T2).
Noter que les quantités dans les parenthèses sont calculées depuis des horodatages non-signés 64-bits et résulte en valeurs signées avec 63 bits significatifs plus le signe. Ces valeurs peuvent représenter les dates de 68 ans dans le passé à 68 ans dans le future. Cependant, l'offset et le délai sont calculés comme somme et différences de ces valeurs, qui contiennent 62 bits signifiants et 2 bits de signe, donc il peuvent

représenter des valeurs non ambigües de 34 ans dans le passé à 34 ans dans le future. En d'autre termes, le temps du client doit être définis dans les 34 ans du serveur avant que le service soit démarré. C'est une limitation fondammentale avec l'arithmétique entier 64-bits.

Dans les implémentations où l'arithmétique double flottante est disponible, les différence de premier ordre peuvent être convertis en double flottant et les sommes de second ordre peuvent être calculés dans cet arithmétique. Vu que les termes de second ordre sont typiquement très petits relatifs aux magnitudes d'horodatage, il n'y a pas de perte de signification, ni de plage ambigüe restaurée depuis 34 ans à 68 ans.

Dans certains scénarios où l'offset de fréquence initial du client est relativement grand et que la propagation de temps et petite, il est possible que le délai de calcul devienne négatif. Par exemple, si la différence de fréquence est de 100 ppm et que l'interval T4-T1 est de 64 s, le délai apparent est -6,4ms. Vu que les valeurs négatives sont mal gérés dans les calculs suivants, la valeur du delta devrait être ramené à au moins s.rho, où s.rho est la précision système, exprimée en secondes.

La discution ci-dessus assume le cas le plus général où 2 paires symétriques mesurent indépendamment les offsets et délais entre-eux. Dans le cas d'un serveur sans état, le protocole peut être simplifié. Un serveur sans état copie T3 et T4 du paquet client dans T1 et T2 du paquet serveur et construit à T3 avant de l'envoyer au client.

Noter que le protocole on-wire tel que décris résiste aux replay d'un paquet réponse serveur. Cependant, il ne résiste pas aux replay d'un paquet request client, qui peut résulter en une réponse serveur avec de nouvelles valeurs de T2 et T3 et résulter dans des offsets et délais incorrect. Cette vulnérabilité peut être évitée en définissant la variable d'état xmt à 0 après avoir calculé l'offset et le délai.

Processus paire

Les déscriptions de processus à suivre incluent un listing des variables d'état importantes suivies par une vue générales des opérations implémentées comme routines. Il est fait fréquemment référence au squelette dans l'appendice. Ce squelette inclus des fragments C qui décrivent les fonctions plus en détail. Il inclus les paramètres, variables, et déclarations nécessaires pour une implémentation NTPv4 conforme. Cependant, de nombreuses variables additionnelles et routines peuvent être nécessaire dans une implémentation fonctionnelle.

Le processus paire est appelé à l'arrivée d'un paquet serveur ou paire. Il lance le protocole on-wire pour déterminer le décalage l'horloge et le délai et calcule des statistiques utilisées par les processus systèmes et d'intérrogation. Les variables paire sont instanciées dans la structure de données d'association quand la structure est initialisée et mis à jours dans les paquets arrivants. Il y a un processus paire, un processus d'interrogation, et un processus d'association pour chaque serveur.

Variables de processus paire

Les figures ci-dessous sommarisent les noms commun, noms de formule, et description courte des variables paires. Ces noms commun et noms de formule sont interchangeable ; les noms de formules sont prévus pour augmenter la lisibilité des équations dans cette spécification. Sauf mention, toutes les variables paire ont le préfix p.

Variables de configuration de processus paire

Name	Formula	Description
srcaddr	srcaddr	source_address
srcport	srcport	source_port
dstaddr	dstaddr	destination_address
dstport	destport	destination_port
keyid	keyid	key_identifier_key_ID

Variables de paquet de processus paire

Name	Formula	Description
leap	leap	leap_indicator
version	version	version_number
mode	mode	mode
stratum	stratum	stratum
ppoll	ppoll	peer_poll_exponent
rootdelay	delta_r	root_delay
rootdisp	epsilon_r	root_dispersion
refid	refid	reference_ID
reftime	reftime	reference_timestamp

Variables d'horodatage de processus paire

Name	Formula	Description
org	T1	origin_timestamp
rec	T2	receive_timestamp
xmt	T3	transmit_timestamp
t	t	packet_time

Variables de statistique de processus paire

Name	Formula	Description
offset	theta	clock_offset
delay	delta	round-trip_delay
disp	epsilon	dispersion
jitter	psi	jitter
filter	filter	clock_filter
tp	t_p	filter_time

Les variables de configuration suivantes sont normalement initialisées quand l'association est mobilisée, soit depuis un fichier de configuration ou à l'arrivée du premier paquet pour une association inconnue.

srcaddr adresse IP du serveur distant ou de l'horloge de référence. Devient l'adresse IP de destination dans les paquets envoyés depuis cette association.

srcport Numéro de port UDP du serveur ou de l'horloge de référence. Devient de port de destination dans les paquets envoyés depuis cette association.

dstaddr Adresse IP du client. Devient l'IP source dans les paquets envoyés depuis cette association

dstport Port client, généralement 123. Devient le port source dans les paquets émis depuis cette association.

keyid ID de clé symétrique pour la clé MD5 128-bits utilisés pour générer et vérifier le MAC. Le client et le serveur ou le paire peuvent utiliser différentes valeurs, mais elles doivent mapper la même clé

Les variables de paquet de processus paire sont mis à jours depuis les en-tête de paquet quand ils arrivent. Elles sont interprétés de la même manière que les variables de paquet de même nom. Il est préférable pour le traitement ultérieur de convertir les valeurs au format NTP court r.rootdelay et r.rootdisp en double flottantes comme variables paires.

Les variables d'horodatage de processus paire incluent les horodatages échangés par le protocole on-wire. La variable t est le compteur de secondes c.t associé avec ces valeurs. La variable c.t est maintenu par le processus d'ajustement d'horloge décrit plus bas. Il compte les secondes depuis que le service a démarré. Les variables de statistiques incluent les statistiques calculés par la routine clock_filter() décris dans la section suivante. La variable tp est le compteur de seconde associé avec ces valeurs.

Opération de processus paire

La routine de réception définis le flux de processus jusqu'à l'arrivée d'un paquet. Il n'y a pas de méthode requise pour le contrôle d'accès, bien qu'il est recommandé que les implémentations incluent un tel schéma, qui est similaire à de nombreux autres largement utilisés. La routine access() dans l'appendice décris une méthode d'implémentation des restrictions d'accès en utilisant une ACL. La vérification de format nécessite des longueur de champs et d'alignement corrects, un numéro de version acceptable, et une syntaxe de champs d'extension correct, si présent.

Il n'y a pas de pré-requis spécifique pour l'authentification ; cependant, si l'authentification est implémentée, l'algorithme de hashage MD5 doit être supporté.

Ensuite, la table d'association est parcouru à la recherche d'une adresses et port source, par exemple en utilisant la routine find_assoc() dans l'appendice. Le tableau ci-dessous est une table dispatch où les colonnes correspondent au mode de paquet et les lignes correspondent au mode d'association. L'intersection de l'association et des modes de paquets dispatche le traitement à une des étapes suivantes.

	Packet_Mode				
Association_Mode	1	2	3	4	5
No_Association_0	NEWPS	DSCRD	FXMIT	MANY	NEWBC
Symm._Active 1	PROC	PROC	DSCRD	DSCRD	DSCRD
Symm._Passive 2	PROC	ERR	DSCRD	DSCRD	DSCRD
Client 3	DSCRD	DSCRD	DSCRD	PROC	DSCRD
Server 4	DSCRD	DSCRD	DSCRD	DSCRD	DSCRD
Broadcast 5	DSCRD	DSCRD	DSCRD	DSCRD	DSCRD
Bcast_Client 6	DSCRD	DSCRD	DSCRD	DSCRD	PROC

DSCRD Indique une violation non-fatale du protocole résultant d'une erreur de programmation, d'un paquet très retardé, ou paquet rejoué. Le processus paire supprime le paquet s'il existe.

ERR Indique une violation fatale du protocole. Le paire supprime le paquet, démobilise l'association passive symétrique, et quitte.

FXMIT Indique un paquet client (mode 3) ne matchant aucune association (mode 0). Si l'adresse de destination n'est pas une adresse de broadcast, le serveur construit un paquet serveur (mode 4) et le retourne au client sans états retenu. L'en-tête du paquet serveur est construit. Un exemple est décris par la routine fast_xmit() en appendice. L'en-tête est assemblé depuis le paquet reçu et les variables système sont affichés ci-dessous. Si les variables système s.rootdelay et s.rootdisp sont stockés en double flottante, elles doivent être converties au format NTP court avant.

Packet_Variable	Variable
r.leap	p.leap
r.mode	p.mode
r.stratum	p.stratum
r.poll	p.ppoll
r.rootdelay	p.rootdelay
r.rootdisp	p.rootdisp
r.refid	p.refid
r.reftime	p.reftime
r.keyid	p.keyid

Noter que si l'authentification échoue, le serveur retourne un message spécial appelé un crypto-NAK. Ce message inclus l'en-tête NTP normal, mais avec un MAC consistant de 4 octets à 0. Le client peut accepter ou rejeter les données dans le message. À la fin de ces actions, le processus paire se termine.

Si l'adresse de destination est une adresse multicast, l'émetteur opère en mode client manycast. Si le paquet est valide et la strate serveur est inférieur à la strate client, le serveur envoie un paquet serveur ordinaire (mode 4), mais avec sont adresses de destination unicast. Un crypto-NAK n'est pas présent si l'authentification échoue. À la fin de ces actions, le processus paire se termine.

MANY Indique un paquet serveur (mode 4) qui ne matche pas une association. Ordinairement, cela ne se produit qu'en résultat d'un serveur manycast répondant à un paquet client multicast. Si le paquet est valide, une association client ordinaire (mode 3) est mobilisée et l'opération continue comme si l'association était mobilisée par le fichier de configuration.

NEWBC Indique un paquet broadcast (mode 5) ne correspondant pas à une association. Le client mobilise soit une association client (mode 3) ou client broadcast (mode 6). Le paquet est ensuite validée et les variables paires initialisées.

Si l'implémentation ne supporte pas de sécurité supplémentaire ou de fonctions de calibration, le mode d'association est mis à client broadcast (mode 6), et le processus paire quitte. Les implémentations supportant l'authentification à clé publique peuvent lancer Autokey ou un protocole de sécurité équivalent. Les implémentations devraient définir le mode d'association à 3 et lancer un court échange client/serveur pour déterminer le délai de propagation. En suivant l'échange, le mode d'association est mis à 6 et le processus paire continue en mode écoute uniquement. Noter la distinction entre un paquet mode 6, qui est réservé pour la supervision NTP et aux fonctions de contrôle, et une association mode 6.

NEWPS Indique un paquet actif symétrique (mode 1) ne matchant aucune association. Le client mobilise une association passive symétrique (mode 2). Le traitement continue dans la section PROC

PROC Indique un paquet matchant une association existante. Les horodatages du paquet sont vérifiés pour éviter les paquets invalides, dupliqués ou buggés. Noter que tous les paquets, incluant un crypto-NAK, sont considérés valides seulement s'ils survivent à ces tests.

Vérification d'erreurs de paquet

Packet_Type	Description
1_duplicate_packet	The_packet_is_at_best_an_old_duplicate_or_at_worst_a_replay_by_a_hacker._This_can_happen_in_symmetric_modes_if_the_poll_intervals_are_uneven.
2_bogus_packet	
3_invalid	One_or_more_timestamp_fields_are_invalid._This_normally_happens_in_symmetric_modes_when_one_peer_sends_the_first_packet_to_the_other_and_before_the_other_has_received_its_first_reply.
4_access_denied	The_access_controls_have_blacklisted_the_source.
5_authentication_failure	The_cryptographic_message_digest_does_not_match_the_MAC.
6_unsynchronized	The_server_is_not_synchronized_to_a_valid_source.
7_bad_header_data	One_or_more_header_fields_are_invalid._

Le traitement continue en copiant la variables de paquet dans les variables paire. Le protocole on-wire calcul le décalage d'horloge theta et le délay delta depuis les 4 horodatages les plus récents tels que décris précédemment. Bien qu'il est en principe possible de faire tous les calculs excepté les différences d'horodatage de premier ordre dans une arithmétique à point fixe, il est plus facile de convertir les différences de premier ordre en double flottante et de faire le reste des calculs avec cette arithmétique.

Ensuite, le registre p.reach (8-bits) dans le processus d'interrogation est utilisé pour déterminer si le serveur est accessible et que les données sont fraîches. Le registre est décalé à gauche de 1 bit quand un paquet est envoyé et le bit de plus à droite est mis à 0. Quand un paquet valide arrive, le bit de plus à droite est mis à 1. Si le registre contient des bits non-zéro, le serveur est considéré comme accessible ; sinon, il est inaccessible. Vu que l'intervalle d'interrogation paire peut avoir changé depuis le dernier paquet, l'intervalle d'interrogation hôte est revu.

La statistique de dispersion epsilon(t) représente l'erreur maximum dûe à la tolérance de fréquence et le temps depuis que le dernier paquet a été envoyé. Il est initialisé :

```
epsilon(t_0) = r.rho + s.rho + PHI * (T4-T1)
```

Quand la mesure est faite à t_0 en accord avec le compteur de secondes. Ici, r.rho est la précision de paquet et s.rho est la précision système, exprimés en secondes. Ces termes sont nécessaire pour compter les incertitudes en lisant l'horloge système dans le serveur et le client.

La dispersion grandit ainsi au taux constant PHI ; en d'autres termes, au temp t, epsilon(t) = epsilon(t_0) + PHI addentry articles autoadd autofind autoprod createalpha createbeta createdb createprod findentry fullpowa generator.php genhtml genman genmd gentex html insert man md pdf regen setfor setfor2 sql temp temp-new-pc tex threads ToDo (t-t_0). Avec la valeur par défaut PHI = 15 ppm, cette quantité est d'environ 1,3 secondes par jour. L'argument t sera supprimé et la dispersion représentée simplement avec epsilon. Les statistiques restantes sont calculées par l'algorithme de filtre d'horloge décris dans la section suivante.

Algorithme de filtre d'horloge

L'algorithme de filtre d'horloge fait partie du processus paire. Il gère le flux de données on-wire pour sélectioner les échantillons le mieux possible pour représenter le temps précis. L'algorithme produit les variables affiché dans le tableau des variables de statistiques de processus paire, incluant offset (theta), délai (delta), dispersion (epsilon), jitter (psi), et le temps d'arrivée (t). Ces données sont utilisées par les algorithmes de mitigation pour déterminer le meilleur offset final utilisé pour la discipline d'horloge. Elles sont également utilisées pour déterminer la santé du serveur et s'il est acceptable pour la synchronisation.

L'algorithme de filtre d'horloge sauve l'échantillon le plus récent (theta, delta, epislon, t) dans la structure de filtre, qui fonctionne comme un registre 8 étapes. Ces échantillons sont sauvé dans l'ordre d'arrivée des paquets. Ici, t est le temps auquel le paquet est arrivé en accord avec le compteur de seconde et ne devrait pas être confondu avec la variable paire tp.

Le schéma suivant est utilisé pour s'assurer que les échantillons suffisants sont dans le filtre et que les anciennes données sont enlevées. Initialement, ces échantillons sont par défaut (0, MAXDISP, MAXDISP, 0). Quand un paquet valide arrive, les données dons décalés dans le filtre. Si les 3 bits LSB du registre sont à 0, indiquant 3 intervals d'interrogation, sont expirés sans paquet valide reçus, le processus paire appel l'algorithme de filtre d'horloge avec un faux échantillon. Se cela persiste pour 8 intervals d'interrogation, le registre retourne à la condition initiale.

Dans l'étape suivante, les étapes son copiés dans une liste temporaire et la liste triée en augmentant le delta. Disons i index des étapes commençant avec le delta le plus bas. Si le premier échantillon epoch t_0 n'est pas ultérieur au denier sampe valide epoch tp, la routine quitte sant affecter les variables de paire courant. Sinon, donnons epsilon_i la dispersion de la i-ème entrée, donc :

```
                 i=n-1
                 --     epsilon_i
epsilon_=        \      -------
                 /       (i+1)
                 --     2
                 i=0
```

Est la dispersion paire p.disp.

noter que si toutes les étapes contiennent l'échantillon par défaut avec la dispersion MAXDISP, la dispersion calculée est un peu inférieur à 16 secondes. Chaque fois qu'un échantillon valide est décalé dans le registre, la dispersion se réduit par un peut moins que la moitié, en fonction de la dispersion valide. Après 4 paquets valides la dispersion est généralement inférieur à 1 seconde.

Il est important de noter que, à la différence de NTPv3, les associations NTPv4 n'affichent pas de condition de timeout en définissant la strate à 16 et l'indicateur leap à 3. Les variables d'association retiennent les valeurs déterminées à l'arrivée du dernier paquet.

Processus système

Vu que chaque échantillon (theta, delta, epsilon, jitter, t) est produit par l'algorithme de filtre d'horloge, tous les processus paire sont scannés par les algorithmes de mitigation consistant des algorithme de sélection, cluster, combinaison, et de discipline d'horloge dans le processus système. L'algorithme de sélection scanne toutes les associations et suppriment les sources de temps incorrect. Dans une série de cycles, l'algorithme cluster supprime l'association statistiquement la plus éloignée du barycentre jusqu'au nombre minimum de survivant. L'algorithme combine produit les meilleurs statistiques finales. L'offset final est passé à l'algorithme de discipline d'horloge pour diriger l'horloge système au temps correct.

L'algorithme cluster sélectionne un des survivants comme paire système. Les statistiques associées (theta, delta, epsilon, jitter, t) sont utilisées pour construire les variables système héritées pas les serveurs et clients dépendants et sont disponibles aux autres applications tournant sur la même machine.

Variables de processus système

Les noms communs, noms de formules et descriptions courte de chaque variables système sont :

Name	Formula	Description
t	t	update time
p	p	system peer identifier
leap	leap	leap indicator
stratum	stratum	stratum
precision	rho	precision
offset	THETA	combined offset
jitter	PSI	combined jitter
rootdelay	DELTA	root delay
rootdisp	EPSILON	root dispersion
v	v	survivor list
refid	refid	reference ID
reftime	reftime	reference time
NMIN	3	minimum survivors
CMIN	1	minimum candidates

Excepté pour les variables t, p, offset, et jitter et pour les constantes NMIN et CMIN, les variables ont le même format et interprétation que les variables paires de même nom. Les paramètres NMIN et CMIN sont utilisées par les algorithme de sélection et de cluster décris dans la section suivante.

La variable t est le compteur de secondes au temps de la dernière mises à jours. La variable p est l'identifiant du paire système déterminé par la routine cluster() plus bas dans ce document. La variable precision est définie aussi large que la résolution et de temps pour lire l'horloge, en unité log2. Par exemple, la précision de l'horloge à fréquence courante s'incrémentant à 60Hz est 16ms, même quand le hardware est en nanosecondes.

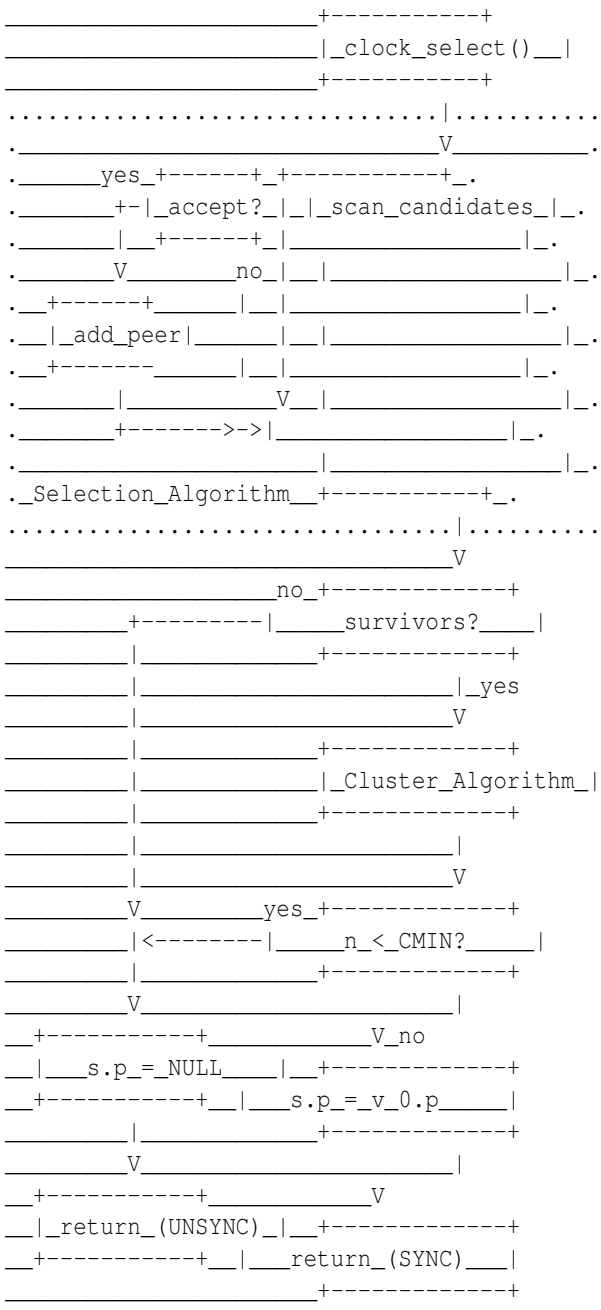
Les variables offset et jitter sont déterminée par l'algorithme combine. Ces valeurs représentent le meilleur offset finals et le jitter utilisé pour la discipline d'horloge système. Initialement, toutes les variables sont à 0, puis le leap est mis à 3 (désynchronisé), et la strate est à MAXSTRAT (16). Se rappeler que MAXSTRAT est mappé à 0 dans le paquet transmit.

Opérations de processus système

La figure ci-dessous décris les opérations du processus système effectué par la routine de sélection d'horloge. L'algorithme de sélection

produit une clique de majorité des candidats présumés correcte (truechimers) basés sur les principes d'agréments. L'algorithme cluster supprime les aberrants pour produire les survivants les plus précis. L'algorithme combine fournis le meilleur offset final pour l'algorithme de discipline d'horloge. Si l'algorithme de sélection ne peut pas produire une clique de majorité, ou s'il ne peut pas produire au moins CMIN survivants, le processus système s'arrête sans discipliner l'horloge. En cas de succès, l'algorithme cluster sélectionne le meilleur candidat comme paire système et ses variables sont hérités comme variables système.

Routine de sélection d'horloge :



Algorithme de sélection

Noter que les algorithmes de sélection et de cluster sont décris séparemment, mais combinés dans un code squelette. L'algorithme de sélection opère pour trouver un interval d'intersection contenant une clique de majorité de truechimers en utilisant les principes d'accord Bisantine originellement proposé par Marzullo, mais modifié pour améliorer la précision.

D'abord, les serveurs non utilisable en accord avec les règles du protocole sont détectés et supprimés. Ensuite, un jeu de tuples (p, type, edge) est généré pour les candidats restants. Ici, p est l'identifiant d'association et le type identifie les point supérieur (+1), moyen (0) et inférieur (-1) d'un interval de correction centré sur theta pour ce candidat. Ces 3 tuples lowpoint (p, -1, theta - lambda), midpoint (p, 0, theta) et highpoint (p, +1, theta + lambda), où lambda est la distance de synchronisation root. Les étapes de l'algorithme sont :

1. Pour chaque associations de m, place 3 tuples comme définis ci-dessus dans la liste des candidats
2. Trie les tuples dans la liste par le composant edge. Définis le nombre de falsetickers f = 0
3. Définis le nombre de midpoints d = 0, définis c = 0. Scanne du endpoint le plus bas vers le plus haut. Ajoute 1 à c pour chaque lowpoint, soustrait 1 pour chaque highpoint, ajoute 1 à d pour chaque midpoint. Is c >= m - f, stoppe, définis l = lowpoint courant.
4. Définis c = 0. Scanne du endppoint le plus haut vers le plus bbas. Ajoute 1 à c pour chaque endpoint, soustrait 1 pour chaque lowpoint, ajoute 1 à d pour chaque midpoint. Si c >= m - f, stop, définis u = highpoint courant.
5. Est-ce que d = f et l < u ? Si oui, passe à l'étable 5A, sinon 5B

5A L'interval d'intersection est [l, u]

5B Ajoute 1 à f. f < (m / 2) ? si oui retourne à l'étape 3, sinon va à l'étape 6

6 Erreur, une clique de majorité ne peut être trouvé.

Noter que l'algorithme démarre avec la supposition qu'il n'y a pas de falsetickers (f = 0) et tente de trouver un interval d'intersection non-vide contenant les midpoint de tous les serveurs correctes, par ex. truechimers. Si un interval non-vide ne peut être trouvé, il augmente le nombre de falsetickers de 1 et retente. Si un interval non-vide est trouvé et que le nombre de falsetickers est inférieurs au nombre de truechimers, une clique de majorité a été trouvé et le midpoint de chaque truechimers (theta) représente les candidats disponibles à l'algorithme cluster.

Si une clique de majorité n'est pas trouvée, ou si le nombre de truechimers est inférieur à CMIN, il y a une insuffisance de candidats. CMIN définis le nombre mininum de serveurs consistants avec les pré-reques de correction. Les opérateurs suspicieux peuvent définir CMIN pour s'assurer que plusieurs serveurs redondants soient disponibles pour les algorithmes de mitigation. Cependant, pour des raisons historiques, la valeur par défaut de CMIN est 1.

Algorithme cluster

Les candidats du clique de majorité sont placés dans la liste de survivant v sous la forme de tuples (p, theta_p, psi_p, lambda_p), où p est un identifiant d'association, theta_p, psi_p, et stratum_p l'offset, jitter et strate courant de l'association p, respectivement, et lambda_p est un facteur de mérite égal à stratum_p addentry articles autoadd autofind autoprod createalpha createbeta createdb createprod findentry fullpowa generator.php genhtml genman genmd gentex html insert man md pdf regen setfor setfor2 sql temp temp-new-pc tex threads ToDo MAXDIST + lambda, où lambda est la distance de synchronisation root pour l'association p. La liste est traitée par l'algorithme cluster :

1. donnons (p, theta_p, psi_p, lambda_p) représentant un candidat survivant
2. Trie les candidats en augmentant lambda_p. n est le nombre de candidats et NMIN le nombre minimum requis de survivants
3. Pour chaque candidat, calculer le jitter de sélection psi_s :

```
_________+---_____________________---+^1/2
_________|_________n-1_____________________|
_________|_________--_____________________|
_________|___1____\_____________________2__|
psi_s_=_|_---_*_/__(theta_s_-_theta_j)___|
_________|__n-1___--_____________________|
_________|_________j=1_____________________|
_________+---_____________________---+
```

4. Sélectionne psi_max comme candidat avec psi_s maximum
5. Sélection psi_min comme candidat avec psi_p minimum
6. psi_max < psi_min ou n <= NMIN ? Si oui, passe à 6A, sinon 6B

6A. Fait. Les candidats suivants dans la liste survivor sont placés dans l'ordre de préférence. La première entrée représente le paire système ; ses variables sont utilisée pour mettre à jours les variables système.

6B. Supprime les candidats avec psi_max ; réduit n de 1 et retourne à l'étape 3.

Algorithme Combine

L'algorithme traite les survivants restants pour produire la meilleur donnée finale pour l'algorithme de discipline d'horloge. La routine traite l'offset paire et les statistiques jitter pour produire l'offset système THETA et le jitter paire système PSI_p, où chaque statistique serveur a un poids par le réciproque de la distance de synchronisation root et le résultat est normalisé.

Le THETA combiné est passé à la routine de mise à jours d'horloge. Le premier candidat dans la liste de survivants est nomminé comme paire système avec l'identifiant p. Le jitter paire système PSI_p est un composant du jitter système PSI. Il est utilisé avec le jitter de sélection PSI_s pour produire le jitter système : **PSI = [(PSI_s)^2 + (PSI_p)^2]^1/2**

Chaque fois qu'une mise à jours est reçue du paire système, la routine de mise à jours de l'horloge est appelée. Par règle, une mise à jours est supprimée si son temps d'arrivée p.t n'est pas strictement ultérieur à la dernière mise à jours s.t. Les labels IGNOR, PANIC, ADJ et STEP réfèrent aux codes de retour de la routine d'horloge locale décrite dans la section suivante.

IGNORE signifie que la mise à jours a été ignorée. PANIC signifie que l'offset est supérieur à seuil PANICT (1000 s) et devrait forcer le programme à quitter avec un message de diagnostique. STEP signifie que l'offset est inférieur au seuil de panique, mais supérieur au seuil d'étape (125 ms). Dans ce cas, l'horloge est décalé à l'offset correcte, mais vu que cela signifie que toute les données paire ont été invalidées, toutes les associations doivent être réinitialisées et le client recommande à l'état initial.

ADJ signifie que l'offset est inférieur au seuil d'étape et donc une mise à jours valide. Dans ce cas, les variables système sont mises à jours depuis les variables paire :

```
+-----------------------------+
|_System_Variable_<-_System_Peer_Variable__|
+-----------------------------+
|_s.leap______<-_p.leap_____________________|
|_s.stratum___<-_p.stratum_+_1______________|
|_s.offset____<-_THETA______________________|
|_s.jitter____<-_PSI________________________|
|_s.rootdelay_<-_p.delta_r_+_delta__________|
|_s.rootdisp__<-_p.epsilon_r_+_p.epsilon_+_|
|____________________p.psi_+_PHI_*_(s.t_-_p.t)_|
|____________________+_|THETA|_____________________|
|_s.refid_____<-_p.refid____________________|
|_s.reftime___<-_p.reftime__________________|
|_s.t_________<-_p.t________________________|
+-----------------------------+
```

Il y a un détail important non affiché. L'incrément de dispersion (p.epsilon + p.psi + PHI * (s.t - p.t) + |THETA|) est délimité par le bas par MINDISP. Dans les sous-réseaux avec des processeurs très rapide et des réseaux avec des délais et une dispersion très petits cela force un augmentation monotonique dans s.rootdisp (EPSILON), qui évite les boucle entre les paires opérants dans la même strate.

Les variables système sont disponible aux programmes d'application comme statistiques de performance nominale. L'offset système THETA est l'offset d'horloge relative aux sources de synchronisation disponibles. Le jitter système PSI est une estimation de l'erreur en déterminant cette valeur, appelé ailleurs l'erreur attendue. Le délais root DELTA est le délais round-trip total relatif au serveur primaire. La dispersion root EPSILON est la dispersion accumulée sur le réseau depuis le serveur primaire. Finalement, la distance de synchronisation root est définie

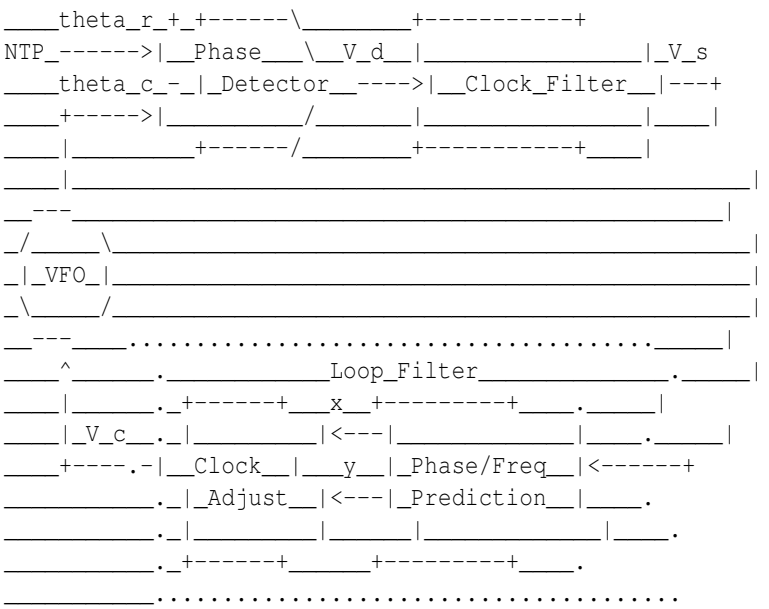
LAMBDA = EPSILON + DELTA / 2

qui représente l'erreur maximum du à toutes les causes et est désignée comme étant la distance de synchronisation root.

Algorithme de discipline d'horloge

L'algorithme de discipline d'horloge, racourcis à discipline dans la suite, fonctionne comme une combinaison de 2 sytèmes de contrôle de feedback philosophiquement différents. Dans un boucle phase-locked (PLL), la mise à jours de phase périodique aux intervals de mises à jours mu secondes sont utilisés directement pour minimiser l'erreur de temps et indirectement l'erreur de fréquence. Dans une boucle frequency-locked (FLL), les mises à jours de fréquence périodiques aux intervals mu sont utilisé directement pour minimiser l'erreur de fréquence et indirectement l'erreur de temps. PLL fonctionne généralement mieux quand le jitter réseau domine, alors que FLL dessine de fonctionnement de NTPv4.

La discipline est implémentée comme système de contrôle feedback. La variable theta_r représente l'offset de l'algorithme combine (phase référence) et theta_c l'offset VFO (phase contrôle). Chaque mise à jours produit un signal V_d représentant la différence de phase instantannée theta_r -theta_c. Le filtre d'horloge pour chaque serveur fonctionne comme une ligne de retard, avec la sortie prise au retard sélectionné par l'algorithme de filtre d'horloge. Les algorithmes de sélection, cluster, et combine combinent les données depuis plusieurs filtres pour produire le signal V_s. Le filtre loop, avec réponse impulsionnelle F(t), produit le signal V_c, qui contrôle la fréquence VFO omega_c et donc l'intégral de la phase theta_c qui ferme la boucle. Le signal V_c est généré par le processus d'ajustement d'horloge.



Ordinairement, la boucle feedback pseudo-linéaire décrite ci-dessus opère pour discipliner l'horloge système. Cependant, il y a des cas où un algorithme non-linéaire offre une amélioration considérable. Un cas est quand la discipline commence sans connaissance de la fréquence d'horloge intrinsèque. La boucle pseudo-linéaire prend des heures pour développer une mesure précise et durant ce temps l'interval d'intérrogation ne peut être être augmenté. La boucle non-linéaire décrite ci-dessous le fait en 15 minutes. Un autre cas est quand les salves occasionnelles de grand jitters sont présent à cause de liens réseaux congestionnés. L'état machine décrite ci-dessous résiste à ces erreus restant inférieur à 15 minutes.

Ce tableau contient un sommaire des variables et paramètres incluant le nom de variable (minuscule) ou paramètre (majuscule), nom de formule, et description courte. Sauf mention, toutes variables ont le préfixe assumé c. Les variable t, tc, state, hyster, et count sont des entier ; le reste des double flottants. La fonction de chacun est expliqué dans les descriptions d'algorithme dessous.

Name	Formula	Description
t	timer	seconds_counter
offset	theta	combined_offset

```
|_resid__|_theta_r____|_residual_offset__________|
|_freq___|_phi________|_clock_frequency__________|
|_jitter_|_psi________|_clock_offset_jitter______|
|_wander_|_omega______|_clock_frequency_wander___|
|_tc_____|_tau________|_time_constant_(log2)_____|
|_state__|_state______|_state____________________|
|_adj____|_adj________|_frequency_adjustment_____|
|_hyster_|_hyster_____|_hysteresis_counter_______|
|_STEPT__|_125________|_step_threshold_(.125_s)__|
|_WATCH__|_900________|_stepout_thresh(s)________|
|_PANICT_|_1000_______|_panic_threshold_(1000_s)_|
|_LIMIT__|_30_________|_hysteresis_limit_________|
|_PGATE__|_4__________|_hysteresis_gate__________|
|_TC_____|_16_________|_time_constant_scale______|
|_AVG____|_8__________|_averaging_constant_______|
+-----+--------+-----------------+
```

Le processus se termine immédiatement si l'offset est supérieur au seuil panic PANICT (1000 s). Le tableau suivant montre la fonction de transition d'état utilisé par la routine rstclock() décrite en annexe. Elle a 4 colonnes montrant respectivement le nom d'état, prédicat et action si l'offset theta est inférieur au seuil d'étape, le prédicat et les actions sinon, et finalement des commentaires.

```
+-----+--------------+-------------+---------+
|_State_|_theta_<_STEP________|_theta_>_STEP______|_Comments_____|
+-----+--------------+-------------+---------+
|_NSET__|_->FREQ______________|_->FREQ____________|_no_frequency_|
|_______|_adjust_time_________|_step_time_________|_file_________|
+-----+--------------+-------------+---------+
|_FSET__|_->SYNC______________|_->SYNC____________|_frequency____|
|_______|_adjust_time_________|_step_time_________|_file_________|
+-----+--------------+-------------+---------+
|_SPIK__|_->SYNC______________|_if_<_900_s_->SPIK_|_outlier______|
|_______|_adjust_freq_________|_else_->SYNC_______|_detected_____|
|_______|_adjust_time_________|_step_freq_________|______________|
|_______|_____________________|_step_time_________|______________|
+-----+--------------+-------------+---------+
|_FREQ__|_if_<_900_s_->FREQ___|_if_<_900_s_->FREQ_|_initial______|
|_______|_else_->SYNC_________|_else_->SYNC_______|_frequency____|
|_______|_step_freq___________|_step_freq_________|______________|
|_______|_adjust_time_________|_adjust_time_______|______________|
+-----+--------------+-------------+---------+
|_SYNC__|_->SYNC______________|_if_<_900_s_->SPIK_|_normal_______|
|_______|_adjust_freq_________|_else_->SYNC_______|_operation____|
|_______|_adjust_time_________|_step_freq_________|______________|
|_______|_____________________|_step_time_________|______________|
+-----+--------------+-------------+---------+
```

Dans les entrées de la table, l'état suivant est identifié par la flèche -> avec les action listées en dessous. Les actions tel que ajuster le temps et ajuster la fréquence sont implémentée par la boucle feedback PLL/FLL dans la routine local_clock(). Une action étape d'horloge est implémentée en définissant l'horloge directement, mais c'est fait seulement après que le seuil stepout WATCH (900s) quand l'offset est supérieur au seui d'étape STEPT (.125s). Cela résiste aux étapes d'horloge sous conditions de congestion réseau extrême.

Les statistiques jitter (psi) et wander (omega) sont calculées en utilisant une moyenne exponentielle avec un facteur de poit AVG. L'exposant de constante de temps (tau) est déterminée en comparant psi avec la magnitude de l'offset courant theta. Si l'offset est supérieur à PGATE (4) fois le jitter d'horloge, le compteur hysteresis est réduit par 2 ; sinon, il est incrémenté de 1. Si hyster augmente à la limite supérieur LIMIT (30), tau est incrémenté de 1 ; s'il est diminué à la limite basse -LIMIT (-30), tau est décrémenté de 1. Normalement, tau plane autour de MAXPOLL, mais diminue rapidement si un pique de température provoque un saut de fréquence.

Processus l'ajustement d'horloge

Le processus d'ajustement d'horloge tourne à interval d'une seconde pour ajouter une correction de fréquence et un pourcentage fixé d'offset theta_r résiduel. theta_r est la décroissance exponentielle de la valeur theta produite par le filtre loop à chaque mise à jours. Le paramètre TC échelonne la constante de temps pour correspondre à l'interval d'interrogation. Noter que la dispersion EPSILON augmente de PHI à chaque seconde.

Le processus d'ajustement d'horloge inclus une facilité d'interruption de timer pilotant le compteur de secondes c.t. Il commence à 0 quand le service démarre et augmente à chaque seconde. À chaque interruption, la routine chock_adjust() est appelée pour incorporer les ajustement de temps et de fréquence de discipline d'horloge, puis les associations sont scannées pour déterminer si le compteur de secondes égal ou excède la variable d'état p.next définis dans la section suivante. Si c'est le cas, le processus d'interrogation est appelé pour envoyer un paquet et calculer la prochaine valeur p.next.

Processus d'interrogation

Chaque association supporte un processus d'interrogation qui est lancé à interval régulier pour construire et envoyer des paquets dans les associations symétrique, client, et serveur broadcast. Il tourne en continue, que le serveur soit accessible ou non pour pouvoir gérer le filtre d'horloge.

Variables de processus d'interrogation

Le tableau ci-dessous décris les noms communs, noms de formule, et une courte description des variables de processus paire et paramètres. Sauf mention, toutes les variables sont assumé avec le préfixe p.

Name	Formula	Description
hpoll	hpoll	host_poll_exponent
last	last	last_poll_time
next	next	next_poll_time
reach	reach	reach_register
unreach	unreach	unreach_counter
UNREACH	24	unreach_limit
BCOUNT	8	burst_count
BURST	flag	burst_enable
IBURST	flag	iburst_enable

Les variables sont allouées dans la structure de données d'association avec les variables de processus paire. La suite est une description détaillée de ces variables.

hpoll entier signé représentant l'exposant d'interrogation, en secondes log2

last Entier représentant le compteur de secondes quand le paquet le plus récent a été envoyé

next Entier représentant le compteur de secondes quand le prochain paquet doit être envoyé

reach registre entier 8-bits partagé par les processus paire et d'interrogation.

unreach Entier représentant le nombre de secondes que le serveur a été inaccessible

Opérations du processus d'interrogation

Comme décris précédemment, une fois par seconde le processus d'ajustement d'horloge est apperé. Cette routine appel la routine d'interrogation pour chaque association en retour. Si le temps du message poll suivant est supérieur au compteur de secondes, la routine retourne immédiatement. Les associations symétriques (modes 1,2), client (mode 3) et serveur broadcast (mode 5) envoient des paquets de façon routinière. Une association client broadcast (mode 6) lance la routine pour mettre à jours les variables reach et unreach, mais n'envoient pas de paquets. Le processus d'interrogation appel le processus de transmission pour envoyer un paquet. Si c'est dans un burst (burst > 0), rien d'autre n'est fait excepté l'appel à la routine de mise à jours d'interrogation pour définis le prochain interval d'interrogation.

Si ce n'est pas dans un burst, la variable reach est décalée à gauche de 1 bit avec un 0 remplaçant le bit de plus à droite. Si le serveur n'a pas été entendu les 3 derniers intervals d'interrogation, la routine de filtre d'horloge est appelée pour augmenter la dispersion.

Si le flag BURST est mis et le serveur est accessible et une source valide de synchronisation est disponible, le client envoie un burst de BCOUNT (8) paquets à chaque interval d'interrogation. L'interval entre les paquets dans le burst est de 2 secondes. C'est utile pour mesurer le jitter précisément avec les intervals d'interrogation long. Si le flag IBURST est mis et que c'est le premier paquet envoyé quand le serveur a été inaccessible, le client envoie un burst. C'est utile pour rapidement réduire la distance de synchronisation sous le seuil de distance et synchroniser l'horloge.

Si le flag P_MANY est mis dans le mot p.flags de l'association, c'est une association client manycast. Les associations client manycast envoyent des paquets mode client pour désigner des adresses de groupe multicast à l'interval MINPOLL. L'association commence avec un TTL de 1. Si au moment de l'interrogation suivante il y a moins de MINCLOCK serveurs mobilisés, le ttl est augmenté de 1. Si le ttl atteint la limite TTLMAX, sans trouver MINCLOCK serveurs, l'interval d'interrogation est augmenté jusqu'à atteindre BEACON.

La routine poll() inclus une fonctionnalité qui revient à l'interval de poll si le serveur devient indisponible. Si reach est non-zéro, le serveur est accessible et unreach est mis à 0 ; sinon, unreach est incrémenté de 1 pour chaque interrogation jusqu'au max UNREACH. Ensuite, pour chaque interrogation hpoll est augmenté de 1, qui double l'interval de poll jusqu'au maximum MAXPOLL déterminé par la routine poll_update(). Quand le serveur redevient accessible, unreach est mis à 0, hpoll est réinitialisé à la variable système tc, et l'opération reprend normalement.

Un paquet est envoyé par de processus de transmission. Certaines valeurs d'en-tête sont copiées depuis la variable paire laissée par le paquet précédent et d'autres depuis les variables système. Le tableau ci-dessous montre quelles valeurs sont copiées pour chaque champ d'en-tête. Dans ces implémentations, en utilisant les types de données double flottante pour le délai root et la dispersion root, elles doivent être converties au format court NTP. Tous les autres champs sont soit copiés tels quels depuis le paire et les variables système ou marqués comme horodatage depuis l'horloge système.

```
+----------------------+
|_Packet_Variable_<-___Variable____|
+----------------------+
|_x.leap_____<-_____s.leap_____|
|_x.version__<-_____s.version__|
|_x.mode_____<-_____s.mode_____|
|_x.stratum__<-_____s.stratum__|
|_x.poll_____<-_____s.poll_____|
|_x.precision__<-_____s.precision_|
|_x.rootdelay__<-_____s.rootdelay_|
|_x.rootdisp___<-_____s.rootdisp__|
|_x.refid____<-_____s.refid____|
|_x.reftime__<-_____s.reftime__|
|_x.org______<-_____p.xmt______|
|_x.rec______<-_____p.dst______|
|_x.xmt______<-_____clock______|
|_x.keyid____<-_____p.keyid____|
|_x.digest___<-_____md5_digest__|
+----------------------+
```

La routine de mise à jours d'interrogation est appelée quand un paquet valide est reçus et immédiatement après qu'un message poll ait été envoyé. Si dans un burst, l'intervalle d'interrogation est fixé à 2 secondes ; sinon, l'exposant de poll de l'hôte hpoll est mis au minimum de

ppoll depuis le dernier paquet reçu et hpoll depuis la routine d'interrogation, mais entre MINPOLL et MAYPOLL. Donc, la discipline d'horloge peut être suréchantillonée mais pas sous-échantillonée. C'est nécessaire pour préserver le fonctionnement dynamique de sous-réseaux et pour protéger contre les erreurs de protocole.

L'exposant poll est convertis à un interval, qui, quand ajouté à la variable de dernier temps d'interrogation, détermine la valeur de la variable de temps de prochain interrogation. Finalement, la variable de temps de dernière interrogation est mis au compteur de secondes courante.

Simple NTP

Les serveur primaires et les clients conforme avec un sous-jeu de NTP, appelé SNTPv4 (rfc4330), n'ont pas besoin d'implémenter les algorithmes de mitigation. SNTP est prévu pour les serveurs primaires équipés avec une seule référence d'horloge, et les clients avec un seul serveur et sans clients dépendants. L'implémentation NTPv4 est prévue pour les serveurs secondaires avec plusieurs serveurs montants et plusieurs clients descendants ou clients. Un serveur primaire SNTP implémentant le protocole on-wire n'a pas de serveurs upstream excepté une seule horloge de référence. En principe, c'est indistinguable d'un serveur primaire NTP qui a les algorithmes de mitigation et donc capable de mitiger entre plusieurs horloges de référence.

À la réception d'une requête cliente, un serveur primaire SNTP construit et envoie le paquet de réponse décris ci-dessous. Noter que le champs dispersion dans l'en-tête de paquets doit être mis à jours :

```
+----------------------+
|_Packet_Variable_<-___Variable____|
+----------------------+
|_x.leap________<-_____s.leap______|
|_x.version_____<-_____r.version___|
|_x.mode________<-_____4___________|
|_x.stratum_____<-_____s.stratum___|
|_x.poll________<-_____r.poll______|
|_x.precision___<-_____s.precision_|
|_x.rootdelay___<-_____s.rootdelay_|
|_x.rootdisp____<-_____s.rootdisp__|
|_x.refid_______<-_____s.refid_____|
|_x.reftime_____<-_____s.reftime___|
|_x.org_________<-_____r.xmt_______|
|_x.rec_________<-_____r.dst_______|
|_x.xmt_________<-_____clock_______|
|_x.keyid_______<-_____r.keyid_____|
|_x.digest______<-_____md5_digest__|
+----------------------+
```

Un client SNTP implémentant le protocole on-wire a un simple serveur et aucun client dépendants. Il peut opérer avec un sous-jeu du protocole NTP on-wire, l'approche la plus simple en utilisant seulement la date de transmission du paquet serveur et en ignorant tous les autres champs. Cependant, la complexité additionnelle pour implémenter le protocole on-wire est minimale donc l'implémentation complète en encouragée.

Considérations de sécurité

Les exigences de sécurité NTP sont plus stricts que d'autres services distribués. D'abord, l'opération du mécanisme d'authentification et le mécanisme de synchronisation de temps sont inextricablement liés. La synchronisation de temps fiable exige des clé cryptographiques qui sont valides seulement dans un interval de temps définis ; mais, les intervals de temps peuvent être forcés seulement quand les serveurs et client participants sont synchronisés à UTC.

Un client NTP peut prétendre avoir un temps authentique pour les applications dépendantes seulement si tous les serveurs dans le chemin vers les serveurs primaires sont authentifiés. Dans NTP chaque serveur NTP authentifie les serveurs de strate inférieur. Il est important de noter que l'authentification dans le contexte de NTP n'implique pas nécessairement que le temps est correct. Un client NTP mobilise un nombre d'associations concurrentes avec des serveurs différents et utilise un algorithme d'accord pour supprimer les serveurs non viables.

La spécification NTP assume que le but d'une intrusion est d'injecter des valeurs de temps fausses, perturber le protocole, ou obstruer le réseau, serveurs, ou clients. Il y a des mécanismes de défense déjà présents dans l'architecture NTP, protocole, et algorithmes. L'échange on-wire est résistant au spoofing, perte de paquet, et attaques replay. Les algorithmes de filtre d'horloge, sélection, et cluster sont conçus pour défendre contre les mauvais cliques de traitres byzantins. Cependant, ces mécanismes n'identifient pas ni n'authentifie de manière sûr les serveurs auprès des clients. Sans d'autres protections, un attaquant peut injecter une de ces attaques :

1. Intercepter et archiver des paquets, et toutes les valeurs publique générées et transmises sur le réseau.
2. Génerer des paquets plus vite que le serveur, réseau, ou client en peut les traiter
3. Dans une attaque wiretap, l'attaquant peut intercepter, modifier, et rejouer un paquet. Cependant il ne peut pas empêcher la transmission du paquet original.
4. Dans une attaque MITM ou masquerade, l'attaquant est positionné entre le serveur et le client, donc il peut intercepter, modifier, et rejouer un paquet et empêcher la transmission du paquet original. Cependant, il ne possède pas le clés privées.

Le modèle de sécurité NTP assume les limitation possibles suivantes :

1. Le temps de fonctionnement pour les algorithmes à clé publique sont relativement longs et variables. En général, les performances de synchronisation de temps sont dégradés si ces algorithmes doivent être utilisé pour chaque paquet NTP.
2. Dans certains modes d'opération, il n'est pas possible pour un serveur de conserver les variables d'état pour chaque client. Il est cependant possible de les régénérer pour un client à l'arrivée d'un paquet de ce client.
3. La durée de vie des valeurs cryptographiques doivent être renforcés, qui nécessite un source d'horloge fiable. Cependant, les sources qui synchronisent l'horloge système doivent être trustés. Cette interdépendance circulaire nécessite une gestion particulière
4. Les fonctions de sécurité client doivent impliquer seulement des valeurs publiques transmis sur le réseaux. Les valeurs privées ne doivent pas être révélés en dehors de la machine qui les a créés, excepté dans le cas d'un agent de confiance spécial.

À la différence du modèle de sécurité SSH, où le client doit être authentifié au serveur, dans NTP le serveur doit être authentifié auprès du client. Dans SSH, chaque adresse d'interface différente peut être liée à un nom différent, tel que retourné par une requête DNS inversée. Dans ce concept, des paires de clé publique/privée peuvent être requis. Un avantage de ce design est que la sécurité peuvent être différente pour chaque interface.

Dans le cas de NTP, les client broadcast sont vulnérables à la perturbation par mauvaise conduite ou des serveur broadcast hostiles. Le filtrage peut être employés pour limiter l'accès aux clients NTP aux serveurs de confiance. L'authentification cryptographique au niveau de client permet d'accepter seulement les messages signés.