

NTP (Network Time Protocol)

Protocole de synchronisation d'horloges dans Internet



Institut de Recherche en Informatique de Toulouse

Z. Mammeri
IRIT - UPS

Cours de DEA, 1998-2004



1. Principes et concepts généraux de NTP

1.1. NTP : c'est quoi ?

- **Protocole de synchronisation d'horloges de hôtes et routeurs dans Internet.**
- Temps NTP basé sur le temps **UTC** (NTP et UTC peuvent être différents à cause des secondes intercalaires).
- Le temps NTP contient le nombre de secondes écoulées depuis le 1 janvier 1900.
- Plusieurs dizaines de millions de clients NTP à travers le monde.
- Plusieurs milliers de serveurs NTP à travers le monde.
- Dans beaucoup de pays, il y a au moins un **serveur primaire**.
- NTP dans la navette spatiale et prévu pour les sondes (Mars...)
- **NTP fournit une précision de synchronisation selon le contexte (... , ms, μ s, ...)**
- Logiciel NTP disponible sur presque tous les types de machines et OS
- Windows/XP possède un client NTP (en natif)

- **Versions de NTP**

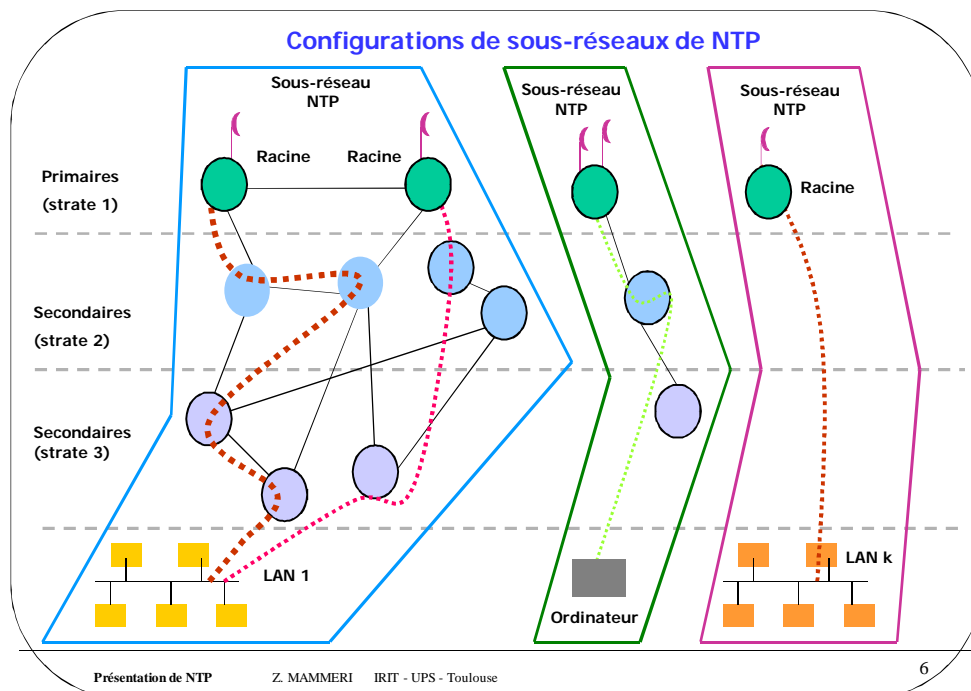
- Version 0.0 : RFC 778 (Avril 1981)
- Version 0.1 : RFC 958 (Septembre 1985)
- Version 1 : RFC 1059 (Juillet 1988)
- Version 2 : RFC 1119 (Octobre 1989)
- **Version 3 : RFC 1305 (Mars 1992)**
- Version 4 : RFC 2030 (depuis Octobre 1996) Non encore approuvée par l'IETF

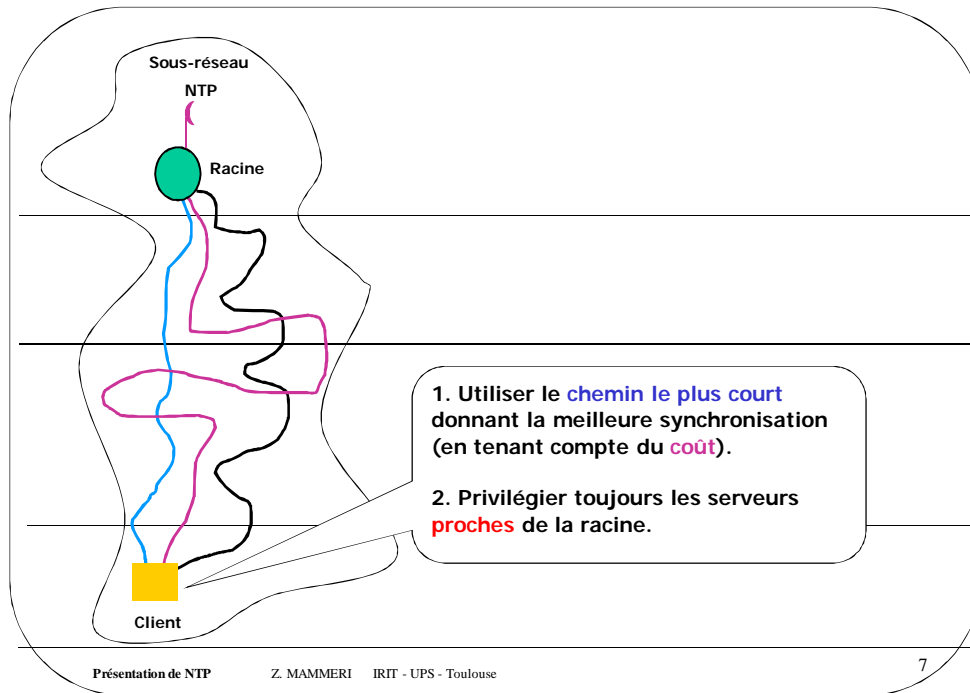
1.2 Besoins/exigences à prendre en compte par NTP

- **Fonctionner selon l'esprit Internet**
- Hétérogénéité des noeuds (capacité de stockage et traitement)
- Nombre de noeuds (de quelques sites à des milliers/millions de sites)
- Classes de réseaux (LAN, WAN, ..., avec/sans fil, ...)
- Performances (précision de synchro : ... 1s ... 1 ms ... 1ns ...)
- Période synchronisation (heures, ..., secondes...)
- Compatibilité avec IP et UDP (pas de connexion, pas de contrôle de flux,...)
- Tolérance aux fautes (malicieuses ou non)
- Sécurité
- Coûts (pour les noeuds qui supportent la synchronisation)

1.3 Principes de conception d'un réseau de synchronisation NTP

- Un (sous-)réseau NTP = serveurs + Hôtes (clients)
- Serveurs **primaires** (de strate 1) : se synchronisent directement sur une source UTC via GPS, radio...
- Serveurs **secondaires** (de strate ≥ 2) : se synchronisent entre eux et avec les serveurs primaires via Internet
- **Clients** : synchronisent leurs horloges sur les serveurs.
- Communication entre serveurs et clients en **point à point, multicast** ou **diffusion**. Communication **non orientée connexion** (pour minimiser les latences).
- Mécanismes de **sécurité** (optionnels).
- Mécanismes de **redondance** (utiliser plusieurs serveurs et chemins de communication) pour tolérer les fautes.
- Utilisation d'algorithmes (filtrage, sélection et combinaison) fondés sur les principes du **maximum de vraisemblance** pour atténuer la variation des délais.





1.4 Modes d'opération de NTP

- Un site **hôte** exécute NTP en échangeant des messages avec des sites **pairs**.
 - L'exécution de NTP est représentée localement à l'aide d'une **association**.
- Mode d'association : **Client**, **Serveur**, **Symétrique actif**, **Symétrique passif** ou **Diffusion**.

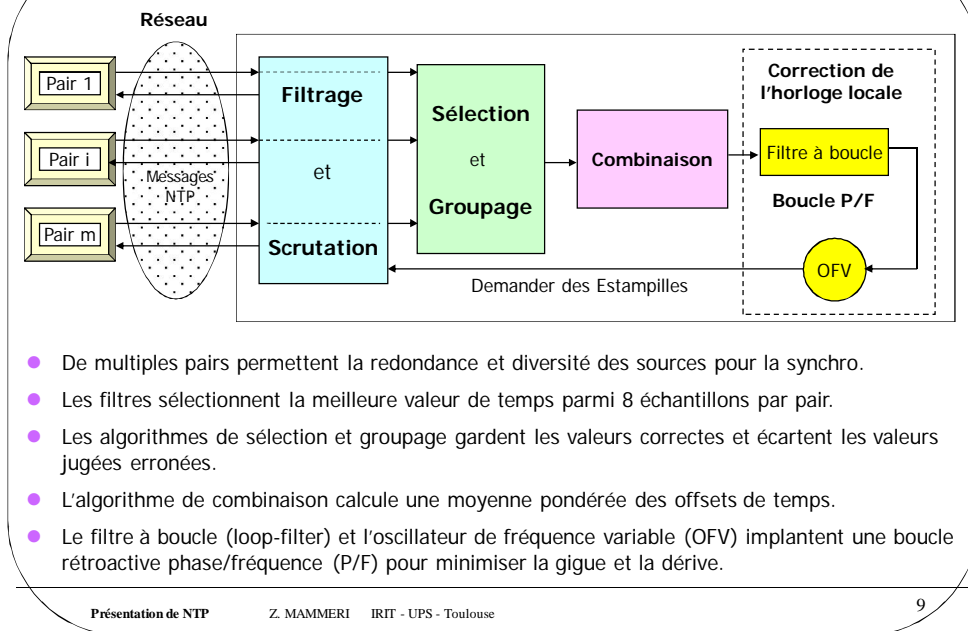
- Fonctionnement **asymétrique**

- **Client/serveur** : le **Client** envoie des messages périodiques de demande de synchronisation sur son pair, mais ne demande pas à synchroniser son pair. Un même hôte peut être client de plusieurs serveurs. Le **Serveur** accepte de synchroniser son pair (le client) mais n'accepte pas d'être synchronisé par son pair. Le serveur répond aux requêtes du client en lui envoyant les estampilles adéquates. Un même hôte peut être serveurs de plusieurs clients.
- **Diffusion** : L'hôte (ex. un serveur de temps sur un LAN) envoie des messages périodiques pour synchroniser tous les pairs à l'écoute, mais il n'accepte pas d'être synchronisé par ces pairs

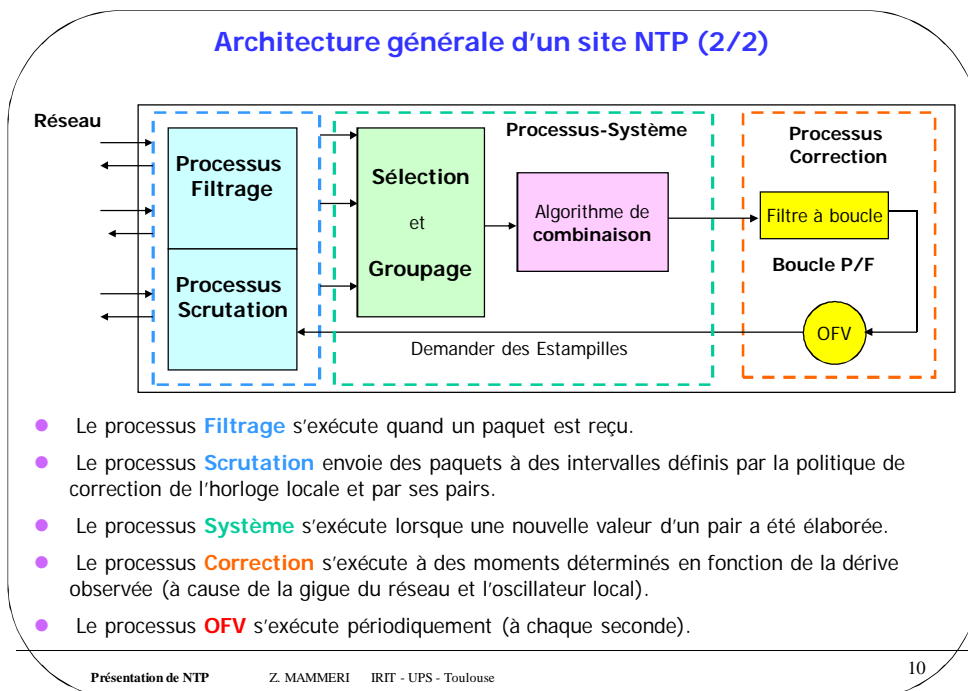
- Fonctionnement **symétrique**

- **Symétrique actif** : l'hôte envoie des messages périodiques (indépendamment de l'état de son pair) pour synchroniser son pair et se synchroniser sur son pair.
- **Symétrique passif** : une association de ce type est créée lorsque l'hôte reçoit un message d'un pair fonctionnant en mode Symétrique actif. Cette association existe tant que le pair est atteignable et fonctionne avec un niveau de strate inférieur ou égal à celui de l'hôte. L'hôte concerné accepte de se synchroniser sur son pair et de le synchroniser.

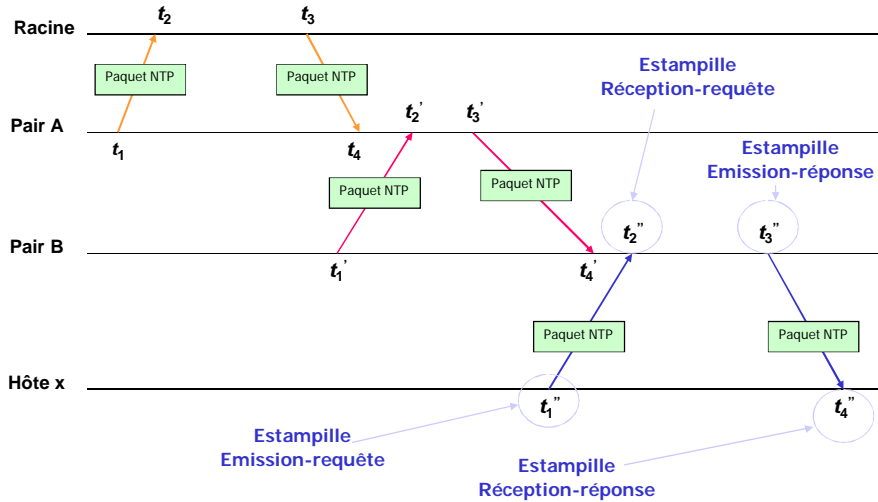
1.5 Architecture générale d'un site NTP (1/2)



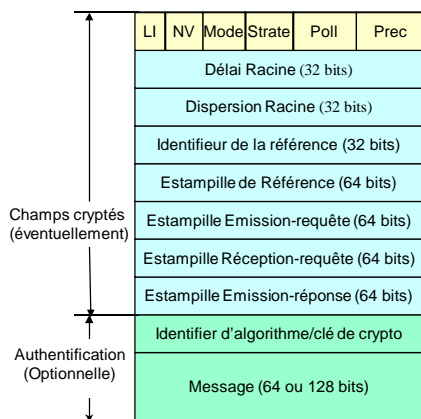
Architecture générale d'un site NTP (2/2)



2. Protocole et algorithmes



2.1. Format d'entête de paquet NTP (1/2)



LI (2 bits) : (leap indicator) indique qu'une seconde doit être insérée/supprimée de la dernière minute de la journée courante

NV (3 bits) : numéro de version (actuellement = 3)

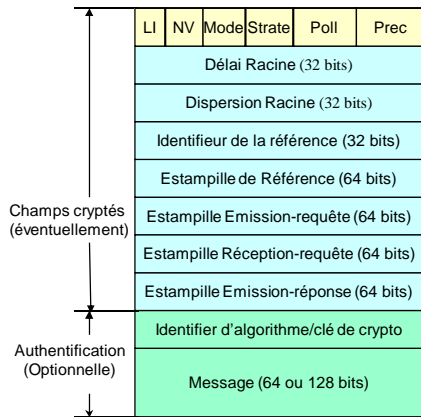
Mode (3 bits) : (client, serveur, diffusion, symétrique...)

Strate (8 bits) : numéro de strate de l'horloge locale (1: horloge primaire, 2-255 : horloges secondaires)

Poll (8 bits) : (polling) intervalle maximum entre deux messages NTP. (exprimé en \log_2 secondes)

Prec (8 bits) : entier signé indiquant la précision de l'horloge locale. La valeur réelle de la précision est arrondie à la puissance de 2 la plus proche, puis exprimée en \log_2 .
 Par ex. une horloge avec une fréq à 50Hz a une précision de 20 ms. Ce qui donne un champ précision = -5 (car 2^{-5} est la valeur la plus proche de 0,02 s)

Format d'entête de paquet NTP (2/2)



Délai Racine : nombre réel signé indiquant le délai total d'aller-retour par rapport à la source de la référence primaire (ce champ peut être positif ou négatif selon la précision et la dérive de l'horloge de l'émetteur du paquet)

Dispersion Racine : nombre réel indiquant l'erreur maximale par rapport à la source de la référence primaire (ce champ est toujours positif)

Identifieur de la référence : code de l'horloge de référence (ex. de codes aux USA : DSN, NIST, DYS, LORC, GPS, GOES...)

Estampille de référence : temps local auquel l'horloge locale a été corrigée (pour la dernière fois) ou initialisée

Estampille Emission-requête : temps local auquel la requête est émise par le site client vers le site serveur

Estampille Réception-requête : temps local auquel la requête est arrivée au site serveur

Estampille Emission-réponse : temps local auquel la réponse a été émise par le site serveur vers le site client

Authentication : utilisé si des mécanismes de sécurité sont implantés.

Format d'estampille NTP

Estampille NTP (nombre non signé, 64 bits)

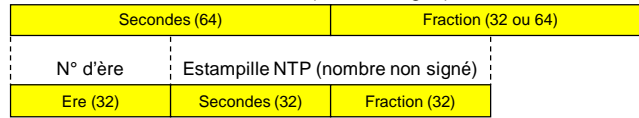
Secondes (32 bits)	Fraction (32 bits)
--------------------	--------------------

- Estampille NTP = temps écoulé depuis le 1er janvier 1900
 - Précision de représentation ≈ 200 picosecondes
 - Précision dépendante de l'interface locale de lecture et recopie d'estampille.
- Tous les 136 ans (2^{32} secondes) : retour à zéro de l'estampille NTP !!!
 - ➔ prochain retour à zéro : le 7 février 2036
 - Mécanismes de recouvrement pour la cohérence des estampilles

Format de date NTP

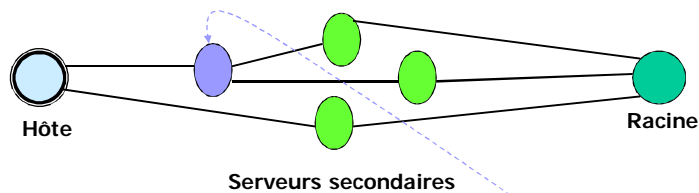
- Estampille NTP = date NTP tronquée (moins de bits)
- Date NTP : non gérée directement par NTP (les paquets NTP contiennent des estampilles).

Date NTP (nombre signé)



An	M	J	NJJ	Date NTP	Ere	Estampille	
-4712	1	1	0	-208 657 814 400	-49	1 795 583 104	Premier jour de l'ère julienne
1	1	1	1 721 426	-59 926 608 000	-14	202 934 144	Premier jour de notre ère
1582	10	15	2 299 161	-10 010 304 000	-3	2 874 597 888	Premier jour de l'ère grégorienne
1900	1	1	2 415 021	0	0	0	Premier jour de l'ère NTP 0
1970	1	1	2 440 588	2 208 988 800	0	2 208 988 800	First day Unix Era
1972	1	1	2 441 318	2 272 060 800	0	2 272 060 800	Premier jour UTC
2000	1	1	2 451 545	3 155 673 600	0	3 155 673 600	Premier jour du 21ème siècle
2036	2	7	2 464 731	4 294 944 000	0	4 294 944 000	Dernier jour de l'ère NTP 0
2036	2	8	2 464 732	4 295 030 400	1	63 104	Premier jour de l'ère NTP 1
3000	1	1	2 816 788	34 712 668 800	8	352 930 432	4294967296

2.2 Principales informations gérées par NTP



Pour chaque pair i

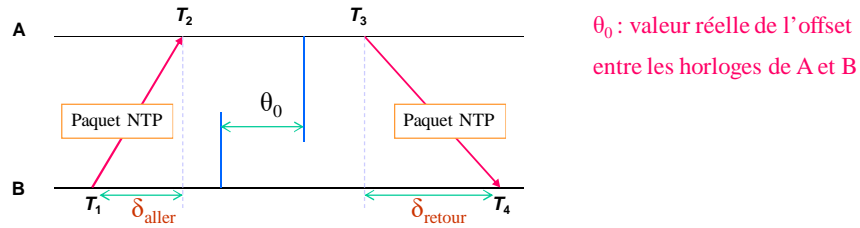
- θ_i : offset entre l'horloge locale et celle du pair i
- δ_i : délai d'aller-retour entre l'hôte et le pair i
- ϵ_i : dispersion (erreur maximale) entre l'hôte et le pair i
- λ_i : distance de synchro entre l'hôte et le pair i

Pour la racine

- Θ : offset entre l'horloge locale et celle de la racine
- Δ : délai d'aller-retour entre l'hôte et la racine
- E : dispersion (erreur maximale) entre l'hôte et la racine
- Λ : distance de synchro entre l'hôte et la racine

Source_Synchro =
Horloge source de
synchronisation
actuellement utilisée

2.3 Modèle d'estimation du délai et offset

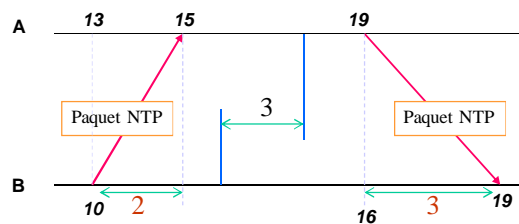


$$\text{Estimation Délai Aller Retour} = \delta = \delta_{\text{aller}} + \delta_{\text{retour}} = (T_4 - T_1) - (T_3 - T_2)$$

$$\text{Estimation Offset} = \theta = \frac{(T_2 - T_1) + (T_3 - T_4)}{2}$$

$$\theta - \frac{\delta}{2} \leq \theta_0 \leq \theta + \frac{\delta}{2}$$

Exemple (délais connus + θ_0 connue)



$$\text{Estimation Délai Aller Retour} = \delta = 2 + 3 = (20 - 10) - (19 - 15) = 10 - 5 = 5$$

$$\text{Estimation Offset} = \theta = \frac{(15 - 10) + (19 - 19)}{2} = \frac{5}{2}$$

$$3 - \frac{5}{2} \leq \theta_0 \leq 3 + \frac{5}{2}$$

Erreur d'estimation d'offset

$$\text{Estimation_Offset} = \theta = \frac{(T_2 - T_1) + (T_3 - T_4)}{2}$$

$$\theta - \frac{\delta}{2} \leq \theta_0 \leq \theta + \frac{\delta}{2}$$

$$\text{Erreur brute d'estimation d'offset} = |\theta_0 - \theta| \leq \delta/2$$

$$\text{Erreur précise d'estimation d'offset} = |\theta_0 - \theta| \leq \varepsilon + \delta/2$$

$$\theta - \varepsilon - \frac{\delta}{2} \leq \theta_0 \leq \theta + \varepsilon + \frac{\delta}{2}$$

$$\text{Dispersion} = \varepsilon = \rho + \varphi_A \times (T_4 - T_1) + \varphi_B \times (T_3 - T_2)$$

ρ : erreur maximale d'accès à l'horloge locale (granularité de l'horloge + interface),

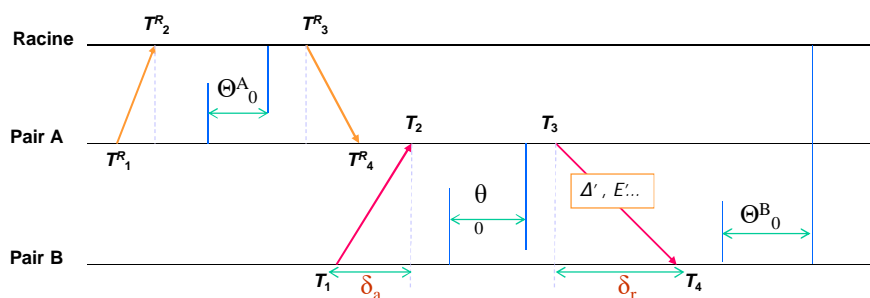
$\rho = 2^{\text{Precision}}$ (Precision : précision de l'horloge locale exprimé en log2)

φ_x : tolérance (erreur) maximale de la fréquence de l'horloge x .

(φ_x est fixée par défaut par NTP à 1/86400)

(φ_x = offset maxi causé par la dérive de l'horloge divisé par la durée maxi de validité d'une valeur d'horloge de référence)

Modèle d'estimation du délai et offset par rapport à la Racine (Primaire de référence)



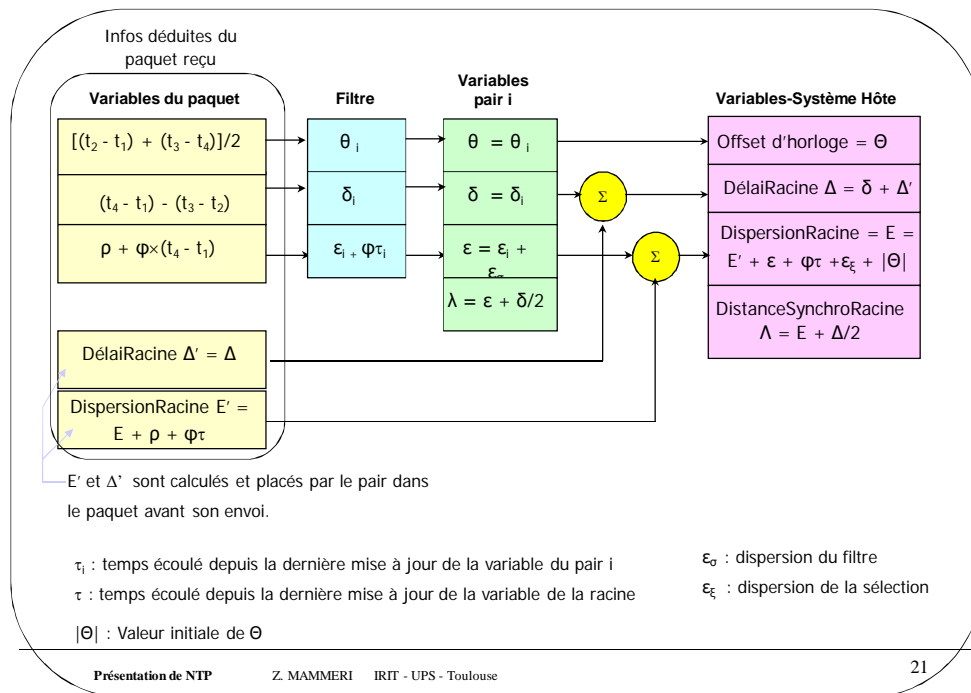
$$\text{Estimation_DélaiAller_Retour_B\%Racine} = \Delta' + \delta$$

$$\text{Dispersion_B\%Racine} = \mathbf{E} = \mathbf{E}' + \varepsilon + \varphi_A \tau + \varepsilon_\xi + |\Theta|$$

τ : intervalle de temps depuis la dernière mise à jour de la dispersion par rapport à la racine

ε_ξ : dispersion de sélection

$|\Theta|$: première valeur d'offset fournie à l'algorithme de synchronisation au moment d'initialisation de l'horloge locale.



Valeurs de paramètres préconisés par l'IETF (pour la version NTP 3)

● Paramètres généraux

- Numéro de port NTP = 123 (NTP.Port)
- Nombre maximum de strates = 15 (NTP.MaxStratum)

● Estampilles, dérives, dispersion et distance

- Durée maximale de validité d'une valeur d'horloge de référence : 1 jour (NTP.MaxAge)
- Intervalle min de scrutation d'un pair = 64 s (NTP.MinPoll)
- Intervalle max de scrutation d'un pair = 1024 s (NTP.MaxPoll)
- Distance maximale pour le rejet d'horloge = 1 sec (NTP.MaxDistance)
- Dérive maximale de l'horloge locale pendant NTP.MaxAge (1 jour) = 1 sec (NTP.MaxSkiew)
- Dispersion minimale utilisée par les filtres = 0.01 sec (NTP.MinDispersion)
- Dispersion maximale utilisée par les filtres = 16 sec (NTP.MaxDispersion)

● Nombres

- Nombre minimum d'horloges à sélectionner comme source à l'ajustement = 1 (NTP.MinClock)
- Nombre maximum d'horloges à sélectionner comme sources à l'ajustement = 10 (NTP.MaxClock)
- Nombre d'étages du registre de filtre = 8 (NTP.Shift)
- Poids (appliqué aux différences) utilisé par le filtre = $\frac{1}{2}$ (NTP.Filter)
- Poids (appliqué aux différences) utilisé par la sélection = $\frac{3}{4}$ (NTP.Select)

2.4 Filtrage

Principe du filtrage

1. Pour chaque site pair i , l'hôte garde les (8) échantillons les plus récents des variables θ_i , δ_i et ε_i

2. A l'arrivée d'un paquet,

i. faire les calculs suivants :

$$\theta_i = \frac{1}{2}[(t_2 - t_1) + (t_3 - t_4)] \quad /* \text{calcul d'Offset} */$$

$$\delta_i = (t_4 - t_1) - (t_3 - t_2) \quad /* \text{calcul du délai aller-retour} */$$

$$\varepsilon_i = \rho + \phi \times (t_4 - t_1) \quad /* \text{calcul de la dispersion} */$$

ii. Rajouter à toutes les variables de dispersion du pair i une quantité $\phi\tau_i$ correspondant à la dérive de l'horloge pendant l'intervalle de temps τ_i (intervalle depuis la dernière mise à jour des variables de dispersion du pair i).

3. Choix d'un couple $\langle \theta, \delta \rangle$ pour représenter le site i et calcul de dispersion

$$\varepsilon = \varepsilon_i + \varepsilon_\sigma \quad (\varepsilon_\sigma : \text{dispersion de filtrage})$$

Algorithme de filtrage

Objectif : Déterminer le meilleur échantillon pour représenter un pair donné.

Principe

1. Les échantillons $(\theta_{i,k}, \rho_{i,k}, \varepsilon_{i,k}, k=0\dots n-1)$ relatifs à un pair i , sont gérés à l'aide d'un registre à décalage à n étages (en général $n=8$). L'échantillon le plus récent se trouve dans l'étage le plus à gauche (indice 0) et le plus ancien dans l'étage le plus à droite (indice $n-1$). Quand un nouvel échantillon est calculé, le registre est décalé d'une position vers la droite et le nouvel échantillon est placé dans l'étage d'indice 0.

Un échantillon ayant une valeur par défaut ($\theta=0, \rho=0, \varepsilon=\text{MaxDispersion}$) est introduit dans le registre, s'il n'y a pas de nouvel échantillon valide au moment où celui-ci devrait être élaboré.

2. La valeur de dispersion, $\varepsilon_{i,k}$, de chaque échantillon k , se trouvant dans le registre à décalage, est mise à jour à chaque fois qu'un nouvel échantillon est introduit dans le registre : $\varepsilon_{i,k} = \varepsilon_{i,k} + \phi\tau$ (τ intervalle de temps depuis la dernière opération de décalage).

3. A chaque échantillon k , on associe une **distance de synchronisation** $\lambda_{i,k} = \varepsilon_{i,k} + |\delta_{i,k}|/2$

4. Construction d'une liste ayant des entrées de la forme $\langle \lambda_{i,k}, k \rangle$ triée selon l'ordre croissant des valeurs de la distance de synchronisation. L'entrée de tête (entrée 0) est celle dont la distance est la plus petite.

5. La dispersion du filtre, $\varepsilon_{\sigma i}$, est calculée ainsi :

Les indices sont ceux de la liste triée.

w est un facteur de poids (généralement $w=1/2$).

$$\varepsilon_{\sigma i} = \sum_{k=0}^{n-1} |\theta_{i,k} - \theta_{i,0}| w^{k+1}$$

6. Les valeurs de l'échantillon qui correspondent à l'entrée de tête de la liste (indice 0) sont sélectionnées pour représenter le pair i . Si la tête de liste contient $\langle d, j \rangle$, alors:

$$\theta_i \equiv i.\text{Offset} \leftarrow \theta_{i,j} \quad \delta_i \equiv i.\text{Délai} \leftarrow \delta_{i,j} \quad \varepsilon_i \equiv i.\text{Dispersion} \leftarrow \min[(\varepsilon_{i,j} + \varepsilon_{\sigma i}), \text{DispersionMax}]$$

2.5 Sélection de la référence

Objectif : Déterminer l'horloge qui devient la référence (source) pour l'ajustement de l'horloge locale.

Principe général : Application d'un algorithme de tolérance aux fautes pour ne garder que les pairs ayant des valeurs de dispersion jugées correctes (ex. trier et écarter les pairs avec une distance élevée, prendre la médiane...) ensuite choisir un pair parmi les pairs retenus pour servir de référence.

Algorithme de sélection de la référence préconisé par l'IETF

Utilise deux opérations (**Intersection** et **Groupage**) pour mieux sélectionner la référence en tenant compte de la distribution statistique des dispersions et en tolérant les fautes de pairs.

Algorithme d'intersection (1/2)

1. /* Sélection des pairs candidats corrects */

m := 0; /* m : nombre de pairs corrects */
ListeDispersion := Vide;

for (each Pair p in ListeCandidats)
if (p.Dispersion < MaxDispersion)

$\Lambda_p = p.DispersionRacine + p.Dispersion + \phi\tau + |p.DélaiRacine + |p.Délai||/2$ /* distance (p)*/
add(ListeDispersion, <p.Offset - Λ_p , -1>); /* Rajouter trois éléments à la liste */

add(ListeDispersion, <p.Offset, 0>);

add(ListeDispersion, <p.Offset - Λ_p , 1>);

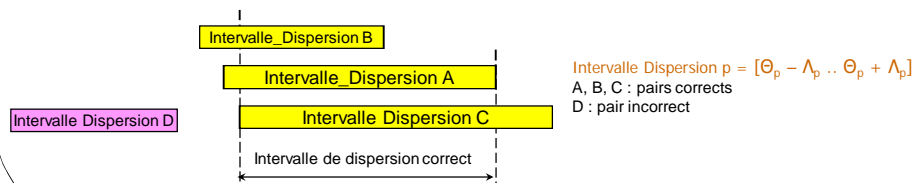
/* Les 3 éléments rajoutés à la liste définissent l'intervalle de dispersion associé au candidat p */

m:=m+1;

sort(ListeDispersion); /* tri de la liste en fonction du champ différence d'offset*/

if (m = 0) exit; /* aucun pair ne peut être utilisé comme référence – échec de la procédure de sélection */

2. /* Adaptation de l'algorithme d'intersection utilisé dans DTS (Digital Time Service) à NTP pour déterminer l'intervalle d'intersection des dispersions le plus large ne contenant que des pairs corrects*/



Algorithme d'intersection (2/2)

```

/* f : Nombre de paires incorrectes */
/* low : borne inférieure de l'intervalle d'intersection s'il existe */
/* high : borne supérieure de l'intervalle d'intersection s'il existe */
for (f from 0 to f ≥ m/2) begin
    c := 0; i := 0;
    /* Les éléments de ListeDispersion sont parcourus de la plus petite valeur de dispersion à la plus grande */
    for (each <dispersion, type> from lowest) begin /* recherche de la borne inférieure */
        i := i - type;
        low := dispersion;
        if (i ≥ m - f) break;
        if (type = 0) c := c + 1;
    endfor
    i := 0;
    /* Les éléments de ListeDispersion sont parcourus de la plus grande valeur de dispersion à la plus petite */
    for (each <dispersion, type> from highest) begin /* recherche de la borne supérieure */
        i := i + type;
        high := dispersion;
        if (i ≥ m - f) break;
        if (type = 0) c := c + 1;
    endfor;
    if (c ≤ f) break;
endfor;

/* Si à la fin de l'algorithme on a low > high, alors l'intervalle d'intersection n'existe pas */
if (low > high) exit; /* échec de la procédure de sélection */

```

Condition : au moins la moitié
des paires doivent être corrects

Algorithme de groupage (1/2)

- Phase 1 : Traitement des paires sélectionnés en tenant compte de leur strate et dispersion.
En introduisant le numéro de strate comme poids dans le calcul de dispersion, on favorise les paires ayant un numéro de strate faible (plus "proche" de l'horloge du primaire de référence) */


```

m := 0; /* m : nombre de paires corrects */
ListeDistances := Vide;
for (each Pair p in ListeCandidats) begin
    if (low ≤ p.Offset ≤ high) begin
        Λp = p.DispersionRacine + p.Dispersion + φτ + |p.DélaiRacine + |p.Délai||/2 /* distance(p) */
        dist := Λp + p.strate*MaxDispersion;
        add(ListeDistances, <dist, p>);
        m := m + 1;
    endfor;
    sort(ListeDistances); /* Tri de ListeDistances selon ordre croissant du champ dist */

```
- Phase 2 : calcul des dispersions de sélection, ε_{sp}, pour chaque paire p survivant à la phase 1.
On procède de la même manière que pour le calcul de la dispersion du filtre : */


```

for (each Pair p in ListeDistances)

```

$$\varepsilon_{sp} = \sum_{k=0}^{m-1} |\theta_{p,k} - \theta_{p,0}| v^{k+1}$$

/* v est un facteur de poids (généralement v = ¾) */

```

endfor;

```

Algorithme de groupage (2/2)

3. /* Phase 3 : Elimination des candidats ayant des dispersions jugées importantes. A la fin de cette phase, on devrait garder un des candidats qui devient la source d'ajustement. On peut aussi garder plusieurs candidats comme sources, dans ce cas, il faut soit choisir un des candidats retenus pour devenir la source, soit appliquer la procédure de combinaison. */

```

while (true) begin
  for (each <dist, index> in ListeDistances) begin
    /* déterminer l'index i du pair ayant la valeur de dispersion de sélection,  $\epsilon_{\text{si}}$ , la plus grande */
    i:= find_index_max_DispersionSelection;
    /* déterminer l'index j du pair ayant la valeur de dispersion par rapport à la racine,  $E_j$ , la plus petite */
    j:= find_index_min_DispersionSelection;

  endfor;
  /* MinClock: nombre minimum d'horloges pour faire la synchronisation. La valeur est fixée par défaut à 1. */
  if ( $\epsilon_{\text{si}} \leq E_j$  or  $m \leq \text{MinClock}$ ) break;
  delete(ListeDistances, i); /* supprimer l'élément d'indice i de la liste */
  m:= m-1;
endwhile;

/* On essaye de garder la même référence de synchro à chaque fois où cela c'est possible. On change de référence si la
référence courante n'a pas été sélectionnée ou bien si parmi les candidats retenus, il existe un pair L dont le numéro de strate
est plus faible que celui de la référence actuelle */
if (not(Index_Référence_courante) in ListeDistances) or (strate(L) < strate(Index_Référence_courante))
  Index_Référence_courante := L;
/* fin de la sélection de la référence */

```

2.6 Combinaison d'horloges

- **Cas simple** : choisir une source parmi l'ensemble fourni par la Sélection
- Pour tenir compte des problèmes aléatoires dus au réseau de communication, sélection de plusieurs sources puis les combiner, comme le font les labos nationaux qui fournissent le temps UTC en combinant plusieurs horloges atomiques.
- **Principe** : à partir des offsets des pairs sélectionnés ($\theta_i, i=1\dots$), calculer un offset Θ pour ajuster l'horloge locale.

$$\Theta = \frac{\sum_{p \in \text{Sélectionnés}} \frac{\theta_p}{p \cdot \text{strate} \times \text{NTP.MaxDispersion} + \text{Distance}(p)}}{\sum_{p \in \text{Sélectionnés}} \frac{1}{p \cdot \text{strate} \times \text{NTP.MaxDispersion} + \text{Distance}(p)}}$$

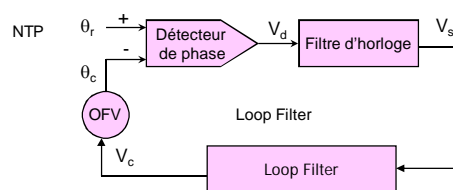
Sélectionnés : ensemble des indices des pairs sélectionnés par la fonction de groupage.

$$\text{Distance}(p) = E_p + \frac{|A_p|}{2}$$

2.7 Correction de l'horloge locale

- **Ajustement par matériel** : Modifier (accélérer/ralentir) la fréquence de génération des ticks d'horloges
- **Ajustement par logiciel** : Ajouter ou soustraire une quantité (la donnée d'ajustement) de la variable mémoire qui représente l'horloge du site.
- **Ajustement** continu
discret

Modèle d'ajustement



- θ_r : offset (phase) fourni par l'étape de sélection
- θ_0 : offset (phase) du OFV (oscillateur à fréquence variable)
- V_d : différence de phase entre NTP et OFV.
- V_s dépend de l'étage choisi dans le registre à décalage de l'horloge..
- Le processus *Loop filter* s'exécute à chaque seconde pour calculer V_c , qui contrôle la fréquence de l'oscillateur local.

Procédures du protocole NTP (3/3)

● Procédure Ajustement

- Appel de la procédure **Sélection** pour choisir une ou plusieurs sources d'ajustement.
- Appel de la procédure **Combinaison** pour combiner les sources sélectionnées.
- Appel de la procédure **Scrutation** pour renouveler (périodiquement) les échantillons.

● Procédure Initialisation

- Procédure appelée sur « reboot » ou réinitialisation du démon NTP.
- Initialisation des variables de l'hôte.
- Etablissement d'association pour matérialiser l'exécution de NTP.
- Initialisation des variables liées à la racine de synchronisation

2.9 Programmes/utilitaires NTP

- **ntpd** : démon de l'OS qui assure la synchronisation de l'horloge locale via des serveurs de temps Internet choisis à la configuration.
- **ntpdate** : programme qui met à jour (à la demande) l'horloge locale en communiquant avec un ou plusieurs serveurs sélectionnés.
- **ntpq** : utilitaire pour superviser le démon ntpd et déterminer ses performances.
- **ntpdcc** : utilitaire pour connaître l'état courant du démon ntpd et changer son état.
- **ntptrace** : utilitaire pour connaître la trace des paquets NTP depuis le l'hôte jusqu'à la racine.
- **ntp-keygen** : programme de génération de clés publiques et privées pour NTP.
- **ntpsim** : simulateur du démon ntpd.

- **Les serveurs de temps NTP français (strate 1 et 2)**

- http://www.cru.fr/NTP/serveurs_francais.html
(certains serveurs sont ntp V4)

- LAAS – Toulouse (serveur de strate 2) : ntp.laas.fr

- CERT – Toulouse (serveur de strate 2) : ntp1.onecert.fr

- **Pour plus d'infos sur NTP :**

- www.ntp.org

- <http://www.eecis.udel.edu/~mills/ntp/html/index.html>