データベースシステム 125－7
（２００１．７．１７）

時刻情報を用いたアプリケーションのための高精度NTPサーバの設計

北口善明[†‡] 岡沢治夫[‡] 櫻田武嗣[‡] 杉浦一徳[‡] 木俵豊[‡] 勝本道哲[‡] 中川晋一[‡]

[†]通信・放送機構 GENESIS プロジェクト <kita@genesis.tao.go.jp>
[‡]独立行政法人 通信総合研究所

あらまし – インターネットによる電子商取引等の急速な発展に伴い，時刻情報を利用したデータ管理や認証等が行われ始めている．そのため，様々なサーバの時刻の正確性が求められおり，時刻の精度はその同期方法に起因している．インターネットにおいて，各サーバの時刻同期には NTP が用いられおり，NTP サーバでの時刻の精度が重要なものとなっている．本報告では，時刻情報の精度向上によるアプリケーションの可能性の検討と，これまでの PC の時刻揺らぎを抑えることのできる高精度 NTP サーバの設計について述べる．

The Design of the High-Accurate NTP Server for the Application using Time Information

Yoshiaki Kitaguchi[†‡] Haruo Okazawa[‡] Takeshi Sakurada[‡] Kazunori Sugiura[‡]
Yutaka Kidawara[‡] Michiaki Katsumoto[‡] Shin-ichi Nakagawa[‡]

[†]Telecommunications Advancement Organization of Japan
[‡]Communications Research Laboratory

Abstract – Thanks to rapid progress in electronic commerce and the like on the Internet, data has been managed and authenticated using time information. That makes various servers to pursue time accuracy that depends on its synchronization method. NTP has been used for time synchronization at each server on the Internet, so time accuracy at NTP servers has become crucial. In this report, we will describe possibility of applications with more accurate time information and designing high-accurate NTP servers that can restrain time fluctuation usually happened in conventional PCs.

1. はじめに

近年のインターネットの発展に伴い，電子商取引が広く利用されている．電子商取引では，取引に用いられるデジタルデータに付与する時刻情報が必要である．また，この時刻情報は，分散システム環境でのデータ同期やストリーム通信でのタイムスタンプ等に用いられており，その精度が重要となっている．

この時刻情報を正確に保つためには，時刻同期を定期的に実施する必要があり，その同期をネットワーク上で行う一般的な方法が NTP（Network Time Protocol）[1]である．NTP では後述するように負荷分散のための階層構造を持ち，上位サーバとネットワーク上での時刻同期を行う。そのため，最上位サーバ（Stratum 1）の時刻精度が NTP の階層構造全体に大きく影響すると言える．

本稿では，時刻情報の必要性についてまとめ，時刻の精度向上によるアプリケーションの可能性の検討を行う．また，ネットワーク時刻同期における精度向上に関して，これまでの PC の時刻揺らぎを抑えること

のできる高精度 NTP サーバの設計について述べる.

2. ネットワーク時刻同期プロトコル

2.1 NTP

NTPはネットワークを介して時刻同期を行うプロトコルとして広く利用されている. この NTP では、世界協定時間（UTC: Coordinated Universal Time）を得たサーバ（Stratum 1）を頂点として階層構造（図 1 参照）を形成し, 各 NTP サーバでは自分より一階層上のサーバまたは同階層のサーバから時刻情報を得て同期を取っている. 時刻同期に用いる時刻情報は, NTP タ

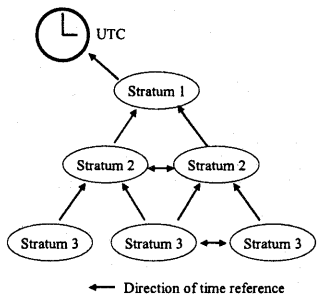


図 1 NTP の階層構造

イムスタンプ形式で持っており, 64 ビットの無符号固定小数点数により, 1900 年 1 月 1 日 0 時からの秒数で表現され, 最大値 4,294,967,295 秒を約 100 ピコ秒の精度で表現できるように設計されている. 現在はマイクロ秒以下の精度は利用されないため, 下位ビットは 0 で埋められているが, 将来, 時刻精度の向上が求められた場合でも十分対応できる設計である.

2.2 時刻同期の精度

時刻のオフセット値は上位サーバの参照時刻と自分の内部時刻との差で得られるが, ネットワークを介して時刻同期を行う場合ネットワークの通信遅延の影響を受けることになる. NTP ではこの通信遅延の影響を双方向通信により打ち消し, 時刻同期の精度を高めている. 図 2 に NTP での通信概念を示す. クライアントでの送受信時間をそれぞれ T_1,T_4, サーバでの送受信時間をそれぞれ T_2,T_3, クライアントとサーバ間の時刻オフセットを δ_t とすると, クライアントの送信パケットの通信遅延 D_1 と受信パケットの通信遅延 D_2 は次のように定義できる.

$$D_1 = T_2 - (T_1 + \delta_t) \quad (1)$$

$$D_2 = (T_4 + \delta_t) - T_3$$

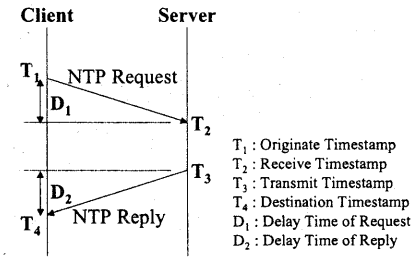


図 2 NTP の時刻同期のながれ

この式(1)と式(2)から δ_t は式(3)のように求まる.

$$\delta_1 = \frac{(T_2 - T_1) + (T_3 - T_4) + (D_1 - D_2)}{2} \quad (3)$$

NTP では往復の通信遅延時間差がない（$D_1 - D_2 = 0$）として計算するため, 往復の通信遅延の差すなわち通信遅延の“揺らぎ”が大きい場合に, 測定される時刻オフセット値に大きく影響を与えることがわかる.

2.3 現在の精度と問題点

Stratum 1 サーバでは, GPS 受信機の低廉化が進んだことからGPSが外部時刻源として近年広く利用されている. 現在, GPS を用いた Stratum 1 サーバの時刻精度は10マイクロ秒よりも大きなものであることが知られている（表 1 参照）[2]. また, Stratum 1 サーバを参照している Stratum 2 サーバやクライアントの時刻精度は Internet Multifeed 社による運用データ[3]や文献[4]による実験データより数ミリ秒から数十ミリ秒となっており, Stratum 1 サーバでの時刻精度より悪くなっていることがわかる. この原因は, 前述したように, 通信遅延の揺らぎのためであり, この揺らぎを解消できないことが NTP での問題となっている.

表 1 各 OS 毎の Trimble Palisade Receiver による時刻精度

Platform	OS	NTP Source	Accuracy
i386 (PC)	Linux	NTP Distribution	10 us
i386 (PC)	Windows 2000/NT	Trimble NTP	1 ms
SUN	Solaris, Sun OS 4	NTP Distribution	50 us
HP	HPUNIX 9,10,11	HP NTP	50 us
Various	FreeBSD	NTP Distribution	20 us
Cisco Router	Model 7200	Cisco	20 us

3. 時刻精度の必要性

次に，現在のインターネット技術において時刻情報が必要または影響を大きく受けるアプリケーションについてまとめる．

3.1 時刻認証

時刻認証技術（デジタルタイムスタンプ技術）は，デジタルデータがある特定時刻に存在していたこと，およびその後変更が行われていないことを証明する技術である．電子商取引において，「誰が，いつ，どのようなデータを生成し，交信したか」という情報の証明は重要であり，これを第三者によって証明する機能を有するものである．

現在，IETF PKIX で標準化が進められている時刻認証プロトコル（TSP: Time Stamp Protocol）[5]の基本的な仕組みは，利用者が，正確な時刻情報をなんらかの形で取得している時刻認証局（TSA: Time Stamp Authority）に対して，時刻認証したいデータとその要求を送り，TSA はそのデータに時刻証明データを付与して戻すというものである[6]. この場合 TSA の信頼性が重要であり，その向上のために複数の分散時刻証明を利用する方式も提案されている[7]. この方式を用いると，一つの TSA が不正を働いたとしてもタイムスタンプの偽造や改竄を行うことは難しくなる．

3.2 分散システム

分散データベースでは，応答時間の改善のために複数のトランザクション（データ編集処理）を同時に実行する．この時，データベースの内部状態の一貫性を保証する仕組みが同時実行制御であり，この同時実効制御では，トランザクション毎に時刻印（タイムスタンプ）を付け，この時刻印を用いて制御を行う時刻印方式が一般的な手法となっている．すなわちトランザクションの開始時刻情報を基に同一データへの操作衝突を回避している．

また，分散システム間の時刻同期を正確に行うことにより，分散したシステムログの整合性を維持することができる．これにより不正アクセスが発生した場合の追跡調査が可能となる。

3.3 リアルタイム通信

インターネット上でリアルタイム通信を行う場合，転送効率や遅延の面から UDP を用いる場合が多い．この UDP では，通信の信頼性を保証しないため転送時にデータの欠損が生じる場合がある．そこで，実時間転送を踏まえたマルチキャスト上の軽いトランスポート・プロトコルとして設計された RTP（Real-time Transport Protocol）[8]が一般的に用いられる．RTP ではヘッダにパケットシーケンス番号とタイムスタンプを含み，受信側でタイミング制御ができる仕組みを提供している．さらに RTP は上流のサイトで複数のソースから提供される音声，映像を合成して一つの RTP のパケットとして再送信するミキサ機能や，符号化形式の変換を行うトランスレータ機能を有している．

この RTP ヘッダが持つ時刻情報は，NTP のタイムスタンプとは異なり 32 ビット長で，最初にサンプリングを行った時間を基準に単調増加し、ペイロードとして運ばれるデータのフォーマットに依存する．また，RTP の制御用プロトコルである RTCP（Real-time Transport Control Protocol）では，センダとレシーバ間の通信遅延を測定するために NTP タイムスタンプ形式を利用している．

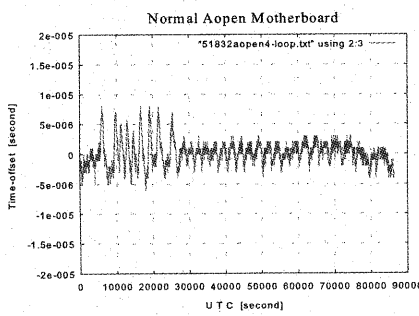
4. 高精度時刻サーバ

以上のように，インターネット上で時刻情報を必要とするアプリケーションは多く存在している．これらに影響を与える NTP による時刻同期では，2 章で述べたように，Stratum 1 サーバの時刻精度の影響が大きく，また通信遅延の影響にも左右されることがわかっている．以下に PC の時刻の揺らぎを押さえることのできる高精密 NTP サーバの設計についてまとめる．

4.1 PC クロックの精度向上

Stratum 1 自身の精度は，PC 内部のクロックの精度に大きく依存している．しかし現在の PC で用いられている内部クロックの精度は低く，この改善が必要である．PC で内部クロック 14.318MHz 基準信号の発振源に利用されている水晶発振子は，ほとんどが ±20ppm[1]の精度であり，この値は一日で約 1.8 秒の誤差が生じること（≒1.8/day）を意味している．そこで我々はその水晶発振子をより精度の良いものに置き換えることを実施した[9]. この置き換えを実施するために，PC への組み込みが容易に可能な外部信号入力デバイスを開発した．このデバイスは，外部からの 10MHz 信号を 14.318MHz に変換して入力できるもので，時刻源としては 10MHz 信号が生成できればどのようなものでも利用できる．

[1] parts per million：100 万分の 1 の量を表す．20ppm なら 1 秒間で 20 マイクロ秒ずれることを意味する．



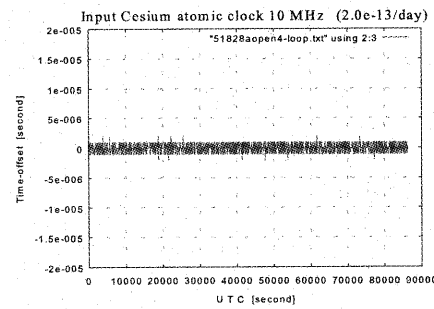


図 3 Stratum 1 サーバの時間精度 [左：内部水晶発振子，右：外部信号セシウム時計] (GPS との比較)

このデバイスを用いて，セシウム原子時計[2]と高精度水晶発振子のそれぞれの信号を利用した場合の時刻安定度を測定した．この結果両者の精度は大きく変わらず，PC の時刻精度を 1 マイクロ秒オーダに向上することができた（図 3 参照）[9]．また実験結果を表 2 に示す．外部信号の利用により，時刻精度の向上が可能となったが，1 マイクロ秒以下の精度を得ることは不可能であった．これは，今回 OS として用いた Linux の時間粒度が 1 マイクロ秒であることに原因があると思われる．セシウム原子時計のような高精度な外部入力信号を用いる場合，現状の Linux OS ではその性能を十分に利用できないということが示された．

表 2 時刻源の時刻精度と Stratum 1 の時刻精度

時刻源	時刻源の精度	Stratum 1 精度
PC 内部水晶発振子	1.8/day	10 マイクロ秒
高精度水晶発振子	5.0×10^{-10}/day	0.57 マイクロ秒
セシウム原子時計	2.0×10^{-12}/day	0.55 マイクロ秒

4.2 通信の揺らぎへの対応

現在のインターネット上には，時刻同期を行うための NTP サーバが充分に存在しておらず，これにより通信遅延の大きなサーバと同期をせざるを得ない状況が発生している．通信遅延が大きいということはその経路に存在するルータの数が多いことを意味しており，ルータの処理時間による通信の揺らぎが大きくなる．そのため，同一セグメントや自サイトに NTP サーバを設置することで NTP サーバへの通信遅延を小さくすることが可能で，これにより揺らぎの影響を小さくできると思われる．そのためには低コストで Stratum 1 サーバが構築できることが重要であると言え、今回我々が開発した PC デバイスを用いることで，容易に構築が可能であると思われる．

4.3 高精度時刻サーバのメリット

このような NTP サーバのメリットを以下にまとめる．

(1) Stratum 1 の時刻精度を 1 マイクロ秒にすることで，各下位層のサーバの時刻精度向上につながる
(2) 汎用的なハードウェアや OS を用いて実現できることから，ネットワーク上に容易に高精度な NTP サーバを構築でき，各サイト・各セグメントに設置することで，通信遅延による揺らぎの影響を小さくできる

これらのことを踏まえて，インターネットにおける各アプリケーションが，ミリ秒以下の時刻精度を NTP サーバにより利用できる場合の可能性について以下に検討を行う．

5. 高精度化による可能性の検討

5.1 リアルタイム時刻認証

電子商取引でのトランザクションが発生した時間をリアルタイムで認証する仕組みを考える．図 4 にその概念図を示す．まず利用者は HTTP 等を利用してオンライントレードサイトにアクセスし欲しい情報を得る．商品の購入を決定すると，まず利用者は自分自身の秘密鍵にてその購入トランザクションのデータに署名する．そして近隣の TSA に“オンライントレードサイトへ送信するパケット”を“時刻証明を要求するデータ”として送信する．TSA では受信時の時刻をデータに付加し，TSA の秘密鍵で署名する．最後に TSA は送られて来たデータからオンライントレードサイトへのア

[2] セシウム原子の性質を利用した時計．国際原子時計（TAI）を定義している．

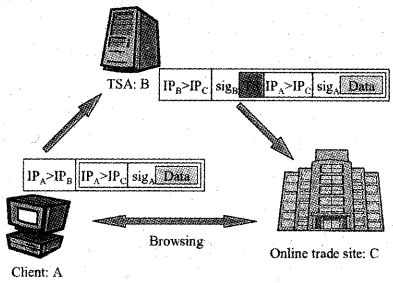


図 4 リアルタイム時刻認証のながれ

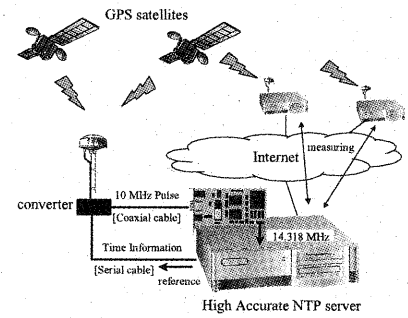


図 5 GPS を用いた NTP サーバによるネットワーク計測

ドレスを取得し，署名したデータを送信する．ここで，TSA が最終的にデータを送信するのは，利用者がオーダを出した後，価格の変動等の不利益な状況を TSA での処理が終了する前に得た場合，利用者がそのオーダをキャンセルする可能性が存在するからである．

ここで重要になるのは TSA の保持する時刻の精度証明の方法であり，以下のことが必要である．

(1) TSA の時刻精度を，TSA に時刻情報を提供している NTP サーバ，もしくはその TSA の精度を証明している上位の TSA が常時監視していること
(2) TSA の時刻監視で得られる情報を公開する手段を有すること
(3) TSA の精度の低下や時刻情報の改変があった場合に証明書の無効操作を行わなければならないこと

これらの要件を満たすシステムが利用されれば，電子商取引の利用分野がさらに拡大するだろう．

5.2 精密な通信の揺らぎ計測

ネットワークの計測を行うためには，計測対象のネットワークの両端に同期の取られた機器が必要となる．これまでの時刻同期技術では，前述してきたように通信の揺らぎの影響により数ミリ秒オーダでの同期が限界であった．そのため，近年のブロードバンドネットワークにおいては，その通信の揺らぎの大きさが同期時刻の精度より小さく，観測が難しくなっている．たとえば OC-12（転送速度は 622Mbps）では，500 バイトの典型的な大きさを持つパケットの転送時間は約 6 マイクロ秒となっている．

そこで，我々は GPS の外部入力信号を利用する高精度 NTP サーバをネットワークの各所に配置し，その NTP サーバ間の経路の通信揺らぎを充分な精度で測定する手法を提案する．図 5 に示すように，GSP 信号から 10MHz の信号を作り PC の内部クロックに利用し，同時に時刻の校正源にも GPS の時刻情報を利用する．GPS の時刻は月差±40 ナノ秒以下[10]という長期安定度を持っており，十分な測定精度を有している．これにより，GPS の時刻精度とその安定度を有する NTP サーバとなり，数マイクロ秒オーダでの通信揺らぎの測定により，ネットワークの特性を求めることが可能であると思われる．

5.3 IPv6 での適応

次世代のインターネットプロトコルとして現在実験運用から実運用への移行が進んでいる技術に IPv6（Internet Protocol version 6）[11]がある．この IPv6 の特徴は現行のプロトコル IPv4 と比べてアドレスが多いことが挙げられる．IPv6 では 128 ビットあるアドレス空間を二つに分け，上位 64 ビット（プレフィックス）は経路制御のために用い，下位 64 ビット（インターフェイス ID）は同一セグメント内でのインターフェイスの識別に使用される．このインターフェイス ID 部分の利用に関しては文献[11]では「インターフェイス ID 部分は同一リンク内で一意に決まる必要がある」と定義されており，現在の IPv6 の仕様では自動アドレス設定時にインターフェイスの MAC（Media Access control）アドレスから生成される ID を使用するか，プライバシーを保護するためにランダムに ID を生成できると定義されている[12]．

さらに，上述したインターフェイス ID の他の利用方法として，現在まで，GSE（Global Site ESD）[13]や LIN6（Location Independent Networking for IPv6）[14]が検討されており，64 ビットのインターフェイス ID 部分を体系化することで効果的な利用が実現する．そこで，この 64 ビット部分に時刻情報を利用す

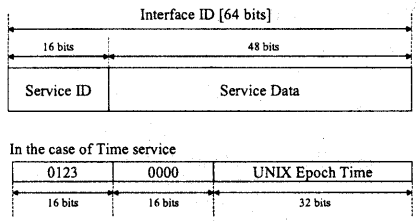


図 6 IPv6 における Service ID のアイデア

ることを考える．

図 6 に示すように，IPv6 アドレスのインターフェイス ID 部分を定義する．上位 16 ビットでそのインターフェイス ID の種類を識別し（サービス ID），下位 48 ビットはその種類によって利用方法が決定される（サービスデータ）．今回考える時刻情報の識別番号を 0123 とし，下位 48 ビットの内使用しない 16 ビットは 0 で埋めて，残りの 32 ビットにそのインターフェイスがネットワークに接続した時の時刻情報を UNIX Epoch 時間[3]で入れる事とする．このアドレスを用いる際のアドレス取得の手順を以下にまとめる．

(1) 利用者にワンタイムパスワードを発行する
(2) 利用者はクライアント端末をネットワークに接続しワンタイムパスワードを入力する
(3) 接続されたクライアントはルータに対してアドレス取得のためルータ要請送信する　このときの送信元アドレスには MAC アドレスから生成されるリンクローカルアドレスが使われる
(4) ルータはクライアントからの要請に対し，ワンタイムパスワードの確認を終えた後、このアドレス利用に対して拡張したルータ通知を発行する　このパケットの中にはルータがルータ要請を受け取った時刻情報が付加されている
(5) ルータはルータ通知のパケットの送信元アドレスと通知した時刻情報の組を保持する　これは，クライアントがリブート等でアドレス情報を失った場合に復旧させることを可能にするためである．
(6) クライアントはルータ通知を受け取ると，パケット内の時刻情報からインターフェイス ID を生成し，プレフィックスと合成して IPv6 のアドレスを構築する

この仕組みを用いると，ある時間以前にネットワークに接続された端末，つまりある時間以上連続してネットワークに接続している端末からのアクセス拒否や通信の優先度変更が可能となる．これはインターネットへのアクセスラインの時間貸しのようなサービスに適応できる．

[3] 1970 年 1 月 1 日 0 時を起源とした秒数で表される時間．

5.4 多地点ストリームデータの合成

最近，撮影位置の異なる多数のカメラを用いて，360 度全方位の動画像を合成する技術が，映画などの特殊効果として利用され始めている．異なるカメラ画像の各フレームを違和感なく合成するためには，画像フレームが撮影された時間を厳密に合わせる必要がある．例えば，通常のテレビや映画フィルムなどにおいては，秒間約 30 フレームの映像となる．この場合，フレーム間隔は約 30 ミリ秒であり，言い換えればフレームの誤差は 30 ミリ秒の誤差が発生する可能性がある．例えば，時速 60km で動く物体は，30 ミリ秒の間に 50cm 移動するため高速な移動体を多数のカメラで撮影し，それぞれの画像をフレーム毎に合成するためには，細かい精度が要求される．前述の時速 60km での動きの誤差を 1cm 以内に抑えるためには，0.5 ミリ秒程度の精度は必要となる．

このような高精度の同期を行うことは，非常に困難であり，特に大規模な環境で通信回線を使って行うことは不可能であった．しかし，我々が提案する時刻サーバを用いることで，マイクロ秒単位の精度を確保することが可能となる．そしてネットワークを経由してサーバに何台かのカメラの制御と映像転送を行わせることで，スタジオなどの限られた狭い領域での撮影しか出来なかった多地点映像合成画像を，大規模な環境で撮影・制作出来るようになる．

6. 今後の課題

NTP サーバを安価に構築するために汎用的な点から Linux OS を用いたが，現在のカーネルでは 1 マイクロ秒の精度が限界であった．そのためセシウム原子時計のような精度の時刻源を用いた場合，その性能が十分発揮できないことがわかった．そのため，今後はこの OS レベルでの時間分解能の向上が必要である．その方法としてナノ秒台での時間分解能を実現する nanokernel[12]の利用やリアルタイム実効を実現する RT-Linux[13]等の利用が考えられる．また，コストを低く抑えるためには，1 マイクロ秒精度を得るために必要十分な外部入力信号の精度の見積もりも考える必要がある．

さらに，この実装を用いて，高精度なネットワーク

計測の実施，リアルタイム時刻認証やIPv6での応用等を進め，その利用価値の実証実験を行うことが必要である．

7. おわりに

本報告では，ネットワークにおける時刻同期の必要性と利用アプリケーションについて述べた．また，PC内部の基準周波数 14.318MHz を外部の高安定信号を用いて高精度化するデバイスの開発を報告し，評価した．セシウム原子時計の信号と高安定水晶発振子の信号を用いた場合での結果は，双方とも向上が図れたが，現在のLinuxの時間分解能では差異が見られないことが判明した．さらに，時効精度が向上した場合のアプリケーションの可能性について述べ，リアルタイム時刻認証やIPv6での拡張利用について提案した．

謝辞 本研究の一部は，通信・放送機構「次世代広帯域ネットワーク利用技術の研究開発プロジェクト」によっている．また，本論文執筆にあたり，有益な助言をいただいた小巻有子氏に感謝します．

参考文献

[1] D.L. Mills, "Network Time Protocol (Version 3)", RFC 1305, March. 1992.
[2] Trimble Navigation Ltd., Trimble Palisade Receiver, "http://www.trimble.com/oem/ntp/driver29.htm"
[3] Internet Multifeed Co., Experimental NTP Servers (Stratum 2), "http://www.jst.mfeed.ad.jp/"
[4] T. Nakashima and S. Ihara, "An Experimental Evaluation of the Total Cost of NTP Topology", ICOIN15, Jan. 2001.
[5] C. Adams, P. Cain, D. Pinkas, R. Zuccherato, "Internet X.509 Public Key Infrastructure Time Stamp Protocol (TSP)", "draft-ietf-pkix-time-stamp-15.txt", May. 2001.
[6] 宇根正志，松浦寛太，田倉昭，"デジタルタイムスタンプ技術の現状と課題"，"金融研究" 第19巻別冊第1号，日本銀行研究所，2000年4月
[7] A. Takura, S. Ono and S. Naito, "A secure and Trusted Time Stamping Authority", IWS'99, Feb. 1999.
[8] H. Schulzrinne, S. Casner, R. Frederick and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", RFC 1889, Jan. 1996.
[9] H. Okazawa, A. Machizawa, S. Nakagawa, Y. Kitaguchi, T. Asami and A. Ito, "Advanced NTP Synchronization Device for Internet Monitoring Tools", INET2001 Posters, Jun. 2001
[10] United States Naval Observatory, "Graph of UTC(USNO MC) - GPS values", "http://tycho.usno.navy.mil/gif/utcgps.gif"
[11] R. Hinden, S. Deering, "IP Version 6 Addressing Architecture", RFC 2373, Jul. 1998.
[12] T. Narten and R. Draves, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", RFC 3041, Jan. 2001.
[13] M. Sola, M. Ohta, Y. Muraoka and T. Maeno, "The 8+8 IPv6 Addressing Architecture", INET2000 Posters, Jul. 2000.
[14] F.Teraoka, M. Ishiyama, K. Uehara, M. Kunishi and H. Esaki, "LIN6: A Solution to Mobility and Multi-Homing in IPv6", "draft-teraoka-ipng-lin6-00.txt", Feb. 2001.