

第 12 部

NTP

第 1 章

はじめに

NTP [25] ワーキンググループでは、昨年度に引続き文部省国立天文台の研究グループとともに、インターネット上に正確な時刻情報を提供するための機構の設計・実装・運用について検討を行った。

その結果、「GPS 衛星からの電波を受信し時刻情報を出力する装置（以下、GPS 受信機）」が出力する秒信号等を時刻源とする NTP サーバを各組織に設置し、これを stratum 1 として運用すれば、実用上は問題ない精度の時刻情報を保時できることが明らかになった。さらに、各組織に設置されたこれらの NTP サーバが、国立天文台において運用されているセシウム原子時計を時刻源とする NTP サーバ (stratum 1 として運用) を参照すれば、いっそう精度の高い時刻を維持できることも明らかになった。しかし、各組織が一斉に国立天文台の stratum 1 を参照すると、組織間を結ぶバックボーンネットワークを流れる NTP パケットのトラフィックが不必要に増加することになる。この問題の解決策の一つに、組織間のコーディネーションを慎重に行い、組織間を結ぶバックボーンネットワークに流れる NTP パケットの量を最小に押えるという方法がある。組織間のコーディネーションは重要であるが、これは本質的な解決策ではない。

そこで、NTP ワーキンググループでは、コーディネーションにたよる方法ではなく、バックボーンネットワークに NTP プロトコルを流さずに組織間で時刻同期を行う方法を検討した。そのために、

- UNIX 上で各種時刻信号の比較が高精度で行えるように、カーネルに変更を加え、外部時刻信号を直接メモリ上に高速で取り込む方法を開発・実装した。
- 次に、GPS 受信機を時刻源とする NTP サーバを開発・実装した。
- さらに、セシウム原子時計を直接取り込んでいる UNIX 上で NTP stratum 2 サーバ (stratum 1 ではない) を運用し、両者の比較を行うことにより、stratum 2 サーバの精度・確度を計測した。
- 最後に、セシウム原子時計と直接比較することにより、GPS 受信機の時刻源としての精度・確度を計測した。

その結果、

1. 国立天文台のセシウム原子時計が刻む時刻 (UTC(NAOT)¹) と各社の GPS 受信機が出力する時刻の偏差を計測する。
2. これをインターネット上に定期的に通報する。
3. 各組織では、この値を取り込んで NTP サーバの時刻を修正する。

という方法をとれば、自組織の stratum 1 の精度をいっそう向上させることが可能となることを明らかになった。

本報告書では、これらの点について報告する。

なお今年度の報告書は、NTP ワーキンググループの大野浩之 (WIDE プロジェクト, 東京工業大学)、福島登志夫、久保浩一、松田浩 (以上、文部省国立天文台) の 4 名が分担して執筆し、全体の調整は大野が行った。

¹UTC(NAOT) とは、国立天文台三鷹 (NAOT) における協定世界時 (UTC) のことである。

第 2 章

外部時刻信号の UNIX カーネルへの取り込み

2.1 昨年度の NTP サーバの構成

昨年度の報告書で紹介した NTP サーバ [144] [145] は、以下のような手順でセシウム原子時計から時刻情報の供給を受けて稼働した (図 2.1)。

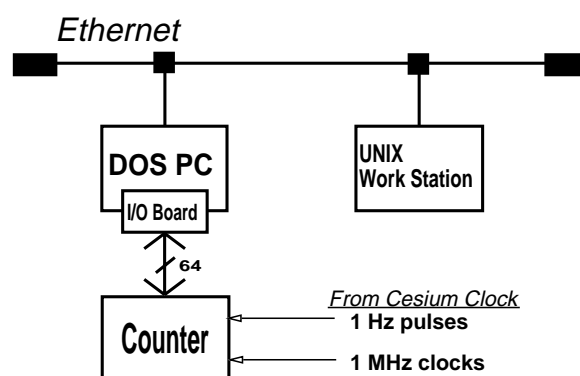


図 2.1: 昨年度試行した NTP サーバ (Mk-I) の構成

1. まず、セシウム原子時計から得られる UTC に同期した 2 つの信号 (1pps と 1MHz) を「2 台の 32bit カウンタで構成されるカウント回路」(以下カウンタ回路 [146]) に入力する。
2. 次に、カウンタ回路の出力を MS-DOS を OS とする NEC PC-9801 上で稼働するアプリケーションプログラム中の `timeval` 型の構造体¹ の変数に取り込む²。

¹

```
struct timeval {
    long tv_sec; /* 秒単位の時間を格納 */
    long tv_usec; /* 秒以下の時間をマイクロ秒単位で格納 */
}
```

²データの取り込みには 2 組の 32bit パラレル I/O ボードを利用した

3. 取り込んだ時刻情報を、NTP とは別のプロトコルを用い、同じ LAN につながっている UNIX ワークステーションに送る。
4. このワークステーション上で PC-9801 から送られて来た時刻情報を時刻源とする NTP サーバを運用する [147] [148]。

ここではこの方式を Mk-I と呼ぶ。

Mk-I は UTC を保時するセシウム原子時計から時刻を取り込みインターネットに時刻情報を供給するおそらく世界最初のシステムという点では十分意義があったがシステム構成 [149] [150] は冗長であった。すなわちもし以下のいずれかが実現できれば PC-9801 かワークステーションの一方が不要になるとともに両者を結ぶ独自のプロトコルも不要となる。

- カウンタ回路を接続した DOS マシン (PC-9801) 上に NTP サーバを移植する。
- UNIX ワークステーションにアクセスタイムが 1 μ 秒以下の 64bit パラレル入力ボードを装着し UNIX 上のプロセスからこのポートにアクセスする。

しかし昨年度は上記の 2 方法はいずれも実現しなかった。これは以下の理由による。

- DOS マシン上に NTP サーバを移植するには大幅な書き直しが必要であった。
- UNIX ワークステーションにパラレルポートを増設するにはコストがかかる上パラレルポートのデータをどのようにして取り込むかについての議論が不十分であった。

2.2 本年度の NTP サーバの構成

今年度は Mk-I に以下のような改造を施した。

- パーソナルコンピュータを NEC PC-9801 から IBM PC/AT 互換機 (以後 PC と略する) に変更する。
- OS を MS-DOS から UNIX (BSD/386) に変更する。

すなわちカウンタ回路からのデータ取り込みは従来通りパーソナルコンピュータで行うが OS は DOS ではなく UNIX に変更した。これにより、以下のような改善がなされた。

- Mk-I とは異なりワークステーションは不要となった
- PC を使うので安価なパラレルボードを利用できる。
- PC 本体のハードウェアに関する情報が公開されている。
- OS が UNIX なので既存の NTP サーバを無改造で利用できる。

- OS のソースコードを参照できるので、たとえば OS に「パラレルボードの値を読み取る」といった機能を組み込む必要が生じたら、これを容易に実現することができる。

この改造に伴い、Mk-I では DOS 上の一つのアプリケーションソフトウェアとして実現していた「カウンタ回路から時刻情報を読み取り timeval 型の変数に現在時刻として格納」する機能は UNIX のカーネル内に取り込んだ。詳細は次節で述べるが、この改造により当該計算機の `gettimeofday()` システムコールは時刻決定に際してカウンタ回路の値すなわちセシウム原子時計が刻む時刻を参照するようになった。このため `xntpd`³を含む既存の現在時刻を取り扱うアプリケーションソフトウェアは何の修正も再コンパイルもなしにセシウム原子時計精度の時刻を取得できるようになった。この方式を Mk-II と呼ぶ。Mk-II の導入により Mk-I では 2 台必要だった計算機を 1 台にまとめることができただけでなく両者を結ぶ独自の時刻同期プロトコルも不要となった。

ところで著者らは Mk-II 方式による NTP サーバが複数台必要であると考え、新たに PC (OS を BSD/386 にした Gateway2000/4DX2-66V) を 3 台準備した。これらを Mk-II 2,3,4 号機と各々呼ぶことにする。これにより NTP サーバとなり得る PC が合計 4 台になったので、各々を運用系 (1 台) バックアップ (待機) 系 (1 台) 研究実験系 (2 台) と位置付け、外部に対する NTP サービスは運用系とバックアップ系が行い研究開発といった運用中止や再起動を伴う作業は研究実験系上で自由に実施できるようにした。なお昨年度に導入した PC-9801 用パラレル入力ポートを引続き有効利用するために、Mk-II 1 号機では「IBM PC 互換機に PC-9801 のボードを接続するためのアダプタ」を介して PC-9801 用パラレル入力ポートの値を読み込んでいるが、Mk-II 2 号機以降では、PC 用の 64bit パラレルポートを PC に直接装着しカウンタ回路の出力を読み込むようにした。また NTP に関する通信がそれ以外の通信の影響を受けて遅延することを避けるため各 PC にはイーサネットボードを 2 枚ずつ装備し、NTP 用のサブネットとそれ以外の目的で利用するネットワークを分離可能にした (図 2.2)。

さらに記憶装置については、新たに容量 1.6GB のハードディスクを導入して、これら 4 台の NTP サーバで NFS で共用することとし、今後の拡張に備えて CD-ROM 及び MO も導入した。加えて近い将来無停電電源を設置し PC の電源とする予定である。これはセシウム原子時計やカウンタ回路が最初から無停電電源装置に接続されているのに対し PC はバックアップされておらず保時室が停電するとサービスが停止してしまうという問題に対処するための措置である。

2.3 UNIX カーネルの改造

本節では、カウンタ回路の出力をカーネル内部に取り込むためにカーネルに施した改造について述べる [151]。

³NTP をサポートする daemon プログラム

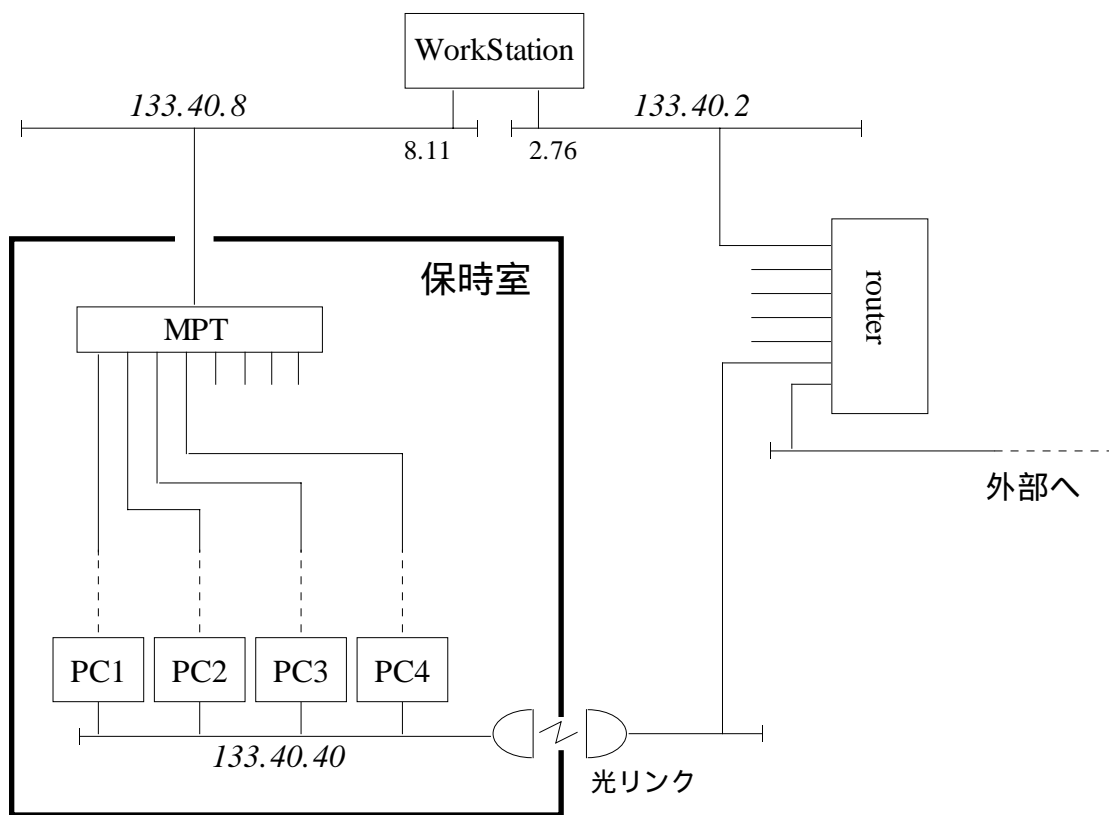


図 2.2: ネットワーク構成

すでに述べたように、著者らは PC の OS に UNIX (BSD/386) を採用し、このカーネルの改造を行ったが、カーネル改造の話は当然のことながらカーネル内部に立ち入った内容になる。そこで、今回は BSD/386 のカーネルの内容そのものについて直接言及することは避け、全く同様な改造手法を適用できる NetBSD 0.9 のカーネルを例にして述べる。

カーネルを改造し、カウンタ回路の出力 (セシウム原子時計の刻む時刻) をカーネル内部にとりこむには、`gettimeofday()` を改造するのが簡便である。そこで、カーネルのソースを調べると、`gettimeofday()` の事実上の実体は `/sys/arch/i386/i386/microtime.s` 中で定義されている `microtime()` であることがわかる。NetBSD に限らず、多くの OS では時刻を取得するとき、スピードの遅いリアルタイムクロックにいちいちアクセスすることはせず、起動時などに一度だけ参照して「現在時刻を保持する変数」に格納し、以後は定期的に発生するタイマ割込みの際にこの変数を必要なだけアップデートするのが普通である。NetBSD もこの方式を採用しており、現在時刻を保持する `struct timeval` 型の変数 `time` を 10msec ごとのタイマ割込み時に 10msec 分だけ増加させている。ただし、このままでは時計の粒度は 10msec になってしまうので、`microtime()` は単に変数 `time` の値を返すのではなく、「最後にタイマ割込みが発生してから『現在』までの経過時間」を 1.193MHz で自走するカウンタを利用して計測し、その結果を変数 `time` に加算したものを返している。これにより、`microtime()` は 1 μ 秒の粒度を確保している。なお、この自走カウンタは IBM/PC 本体に内蔵されているものである。この自走カウンタが 1MHz で駆動されていれば話は簡単であるが、現実には 1.193MHz に固定されており変更できないため、自走カウンタによって求めた 1.193MHz のパルス数を求め、これを 1.193 で割るという効率のよくない作業を `microtime()` 内部で毎回行っている。

著者らは `microtime()` 内に、カウンタユニットの内容を読み取るための I/O 命令を追加した。PC に装着したパラレル入力ボードはアクセスタイムが 500nsec と高速であるが、カウンタユニットにはラッチ機構がないため値が変化している途中の値を読み込んでしまう恐れがある。そのため、機械接点のチャタリング除去と同じ手法で 2 度ずつ読み取り、1 度目と 2 度目のビットパターンの変化が妥当なものであるかどうか判断している。

現時点では `microtime()` には、カウンタユニットの値を読み取るコードと、上記の (従来の) 方式による時刻取得コードの両方が含まれており、カーネル内に新たに設けた大域変数

```
static struct pctime_and_cesium {
    struct timeval pctime;
    struct timeval cesium;
    struct timeval timediff;
}
```

に、両方式で求めた時刻を格納している (`pctime` には従来の方式で求めた時刻が、`cesium` にはカウンタユニットの値が格納される)。さらに、各々の方式で求めた時刻の差を計算

し、上記構造体の `timediff` に代入している。

したがって、`microtime()` が従来方式の時刻を返すように設定し、この PC を NTP stratum 2 で運用すると、`gettimeofday()` を実行するたびに、上記構造体の `pctime` には、NTP によって制御されている時刻が格納され、`timediff` にはカウンタ回路の値 (すなわちセシウム原子時計が刻む UTC) との時差が入る。すなわち、NTP で制御される時計のゆらぎが観測できる。これについては後述する。

なお、`microtime()` が最終的に上記構造体の `pctime` と `cesium` のどちらを返すかを実行中に変更する場合には、カーネル内にある「どちらの値を返すかを決定する変数」を直接書き換えている。

第 3 章

GPS 受信機に基づく stratum 1 NTP サーバの設計と実装

GPS 受信機を用いると、衛星が受信可能な場所であれば、UTC に同期した時刻をすぐに利用できる。この利点を生かすことにより手軽に NTP サーバを構成できる。しかし、一般に購入可能な民生用 GPS 受信機については、その種類こそ多いが、時刻情報は取り出せても 1pps 信号が出せるものは少ない。

著者らは、1pps 信号が出せる GPS 受信機として、現在最も安価でコンパクトな古野電気の GN-72 を使うことにした。GN-72 は、全長 20cm 強のアンテナと電源 (DC 5V) を接続することにより、世界中どこにいても常時、高精度の測位情報 (経緯度など受信機の場所) 及び UTC に同期した 1pps 信号 (以後、正秒信号と呼ぶ) を出力する。図 3.1 は、右から、GN-72 本体 (受信機)¹ に電源等を組み込んだ GPS 1 秒信号発生器、GN-72 のアンテナ、1MHz 信号を取り出すための水晶発振器 (後述) の外観を示したものである。

図 3.1 でもわかるように、GN-72 の受信機本体は手帳ほどの大きさである。GN-72 の制御及び測位・時刻等のデータの取得は、RS-232C を使いパーソナルコンピュータ等で行なう。すべての情報は GN-72 の D-sub 9 ピンコネクタより RS-232C レベルのシリアルデータとして得られる。このときコネクタから +5V の電源を供給しており、同時に同じコネクタを通じて正秒信号も得ることができる。GPS を利用するには視野内に使用可能な衛星が複数個 (できれば 4 個以上) あることが必要であるので、アンテナは見通しがよく周りに反射物等のない場所、例えば高い建物の屋上等に設置する。

GN-72 を用いた NTP サーバを実現するに当たっては、セシウム原子時計に基づく NTP stratum 1 サーバの試作用に、昨年度開発したカウント回路 (図 3.2) をそのまま流用できるように、GN-72 の正秒信号出力の波形変換回路及び 1MHz 信号発生器を設計した。すなわち、著者らの GPS 受信機の利用方法は、GPS 受信機 (GN-72) からの秒信号と水晶発振機からの 1MHz 信号をカウンタ回路でカウントし、その結果をパラレルインタフェースを介して UNIX のカーネルに読み込むというものである。この方法以外にも、xntpd の配布パッケージに含まれる GPS 受信機用ドライバに見られるような方法、すなわち時刻情報を RS-232C 経由で取り込むという方法がある。この方法だと、仮に 19200bps で通信を行っても、1bit 分すなわち 50 μ 秒程度の誤差がシリアルインタフェースまわりで発生する上、割込み処理のオーバーヘッドもある。このため、この方法は最初から検討の

¹GPS 1 秒信号発生器の左下隅の "GPS" と書かれている長方形のユニットが GN-72 本体である。



図 3.1: GN-72 本体を内蔵した GPS 1 秒信号発生器 (右),GN-72 のアンテナ (中) と水晶発振器の外観

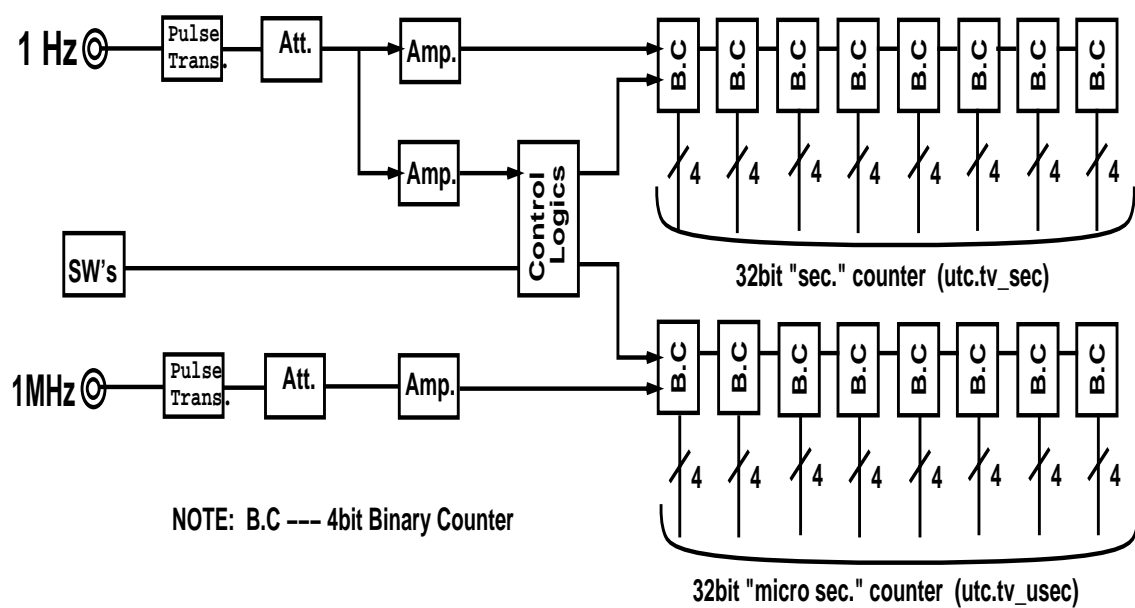
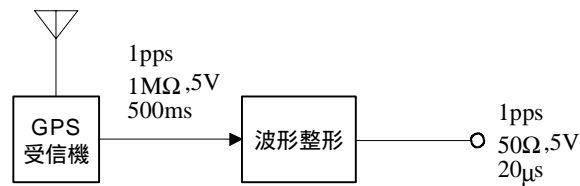


図 3.2: カウンタ回路の構成

対象外であった。

GN-72 より出力される正秒信号は、TTL レベル (5V , 1M Ω , パルス幅 500ms) であるので、まずこれをワンショットマルチバイブレ - タ (Single Shot) をへて 50 Ω バッファアンプにより、5V、50 Ω 、パルス幅 20 μ s の出力信号に変換する (図 3.3(1))。これはセ

(1)GPS 1秒信号発生器



(2) 水晶発振器

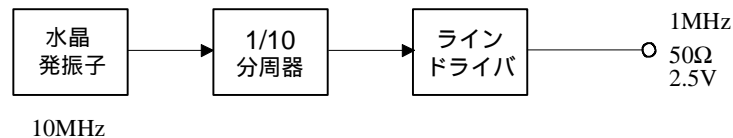


図 3.3: GPS 1 秒信号発生器 (上) と水晶発振器 (下) のブロック図

シウム原子時計の正秒信号 (10V、パルス巾 20 μ s、立ち上がり時間 50ns 以下) と同等の信号形態にするための工夫である。1MHz 信号発生器は、($10 \times 10^6 \pm 500$)Hz の水晶発振子を用いて、カウンタ IC による分周器により 10 分周し、出力に 50 Ω ラインドライバを用いて、1MHz、50 Ω 、2.5V の矩形波を得ている (図 3.3(2))。これはセシウム原子時計の 1MHz 信号 (1V-rms、50 Ω 、正弦波) に相当するものである。なお、カウント回路では 1MHz 信号をゼロクロスコンパレ - タで矩形波に変換している。

これらを用いることで GPS 受信機から得られる正秒信号と、1MHz 信号をカウント回路を通して計算機に取り込むことができる。PC から見た場合、カウンタ回路を駆動する信号源が何であるかは何ら意味をもたないので、PC のカーネルとして前節で述べたセシウム原子時計を時刻源とする NTP サーバ用のカーネルを用意すれば、GPS に基づく NTP サ - バを実現できる。

第 4 章

NTP stratum 2 のゆらぎ

すでに述べたように、今回開発した、「カウンタ回路の出力をカーネル内に取り込む機能」を持つカーネル (改造カーネル) は、カーネルの内部変数に以下の 2 つの時刻を保持している。

- カウンタ回路から読み込んだ値 (に一定値を加えたもの)。すなわち UTC に同期した時刻 (以後、セシウム時と呼ぶ)。
- PC に内蔵された水晶発振器を時刻源として求めたもの。すなわち従来の方法で求めた時刻 (以後、PC 時と呼ぶ)。

この 2 つの時計を用いれば、インターネット上の他組織で稼働中の stratum 1 を参照する stratum 2 がどの程度の精度の時刻情報を維持しているのか (言い替えれば、stratum 2 はどの程度ゆらぐのか) を調べることができる。

本章では、この「stratum 2 のゆらぎ」を計測した結果について述べる。

4.1 実験方法

2 章で述べたように、改造カーネル内部に保持されている 2 つの時刻のうち、どちらの値を現在の時刻とみなすかは外部からの指示に基づいて随時変更できる。セシウム原子時計ベースの stratum 1 として運用する場合にはもちろんセシウム時を現在時刻とするように設定するが、PC 時を返すように (すなわち通常の PC と同じ挙動を取るように) し、この状態で stratum 2 として運用すれば、この PC の現在時刻 (すなわち `gettimeofday()` が返す値) は、参照関係にある stratum 1 や stratum 2 の影響をうけて変動する。このような設定のもとで、stratum 2 として時事刻々変動する PC 時と UTC に同期しているセシウム時の時刻差を記録すれば、stratum 2 として NTP サーバを運用するとどの程度の精度の時刻を確保できるかを調べることができる。なお、改造版カーネルは `gettimeofday()` システムコール実行時に 2 つの時計の時刻差をカーネル内のある変数に記録するので、時刻差の計測は以下の 3 ステップで完了する。

1. あらかじめ `/dev/kmem` を open しておく。

2. `gettimeofday()` を実行する。
3. カーネル内にある時刻差を保持している変数の値を `/dev/kmem` から読み出す。

4.2 実験結果と評価

以下に実験結果の一例を図 4.1 を示す。これは 1993 年 11 月におよそ 12 時間にわたって計測した際のデータをプロットしたものである。

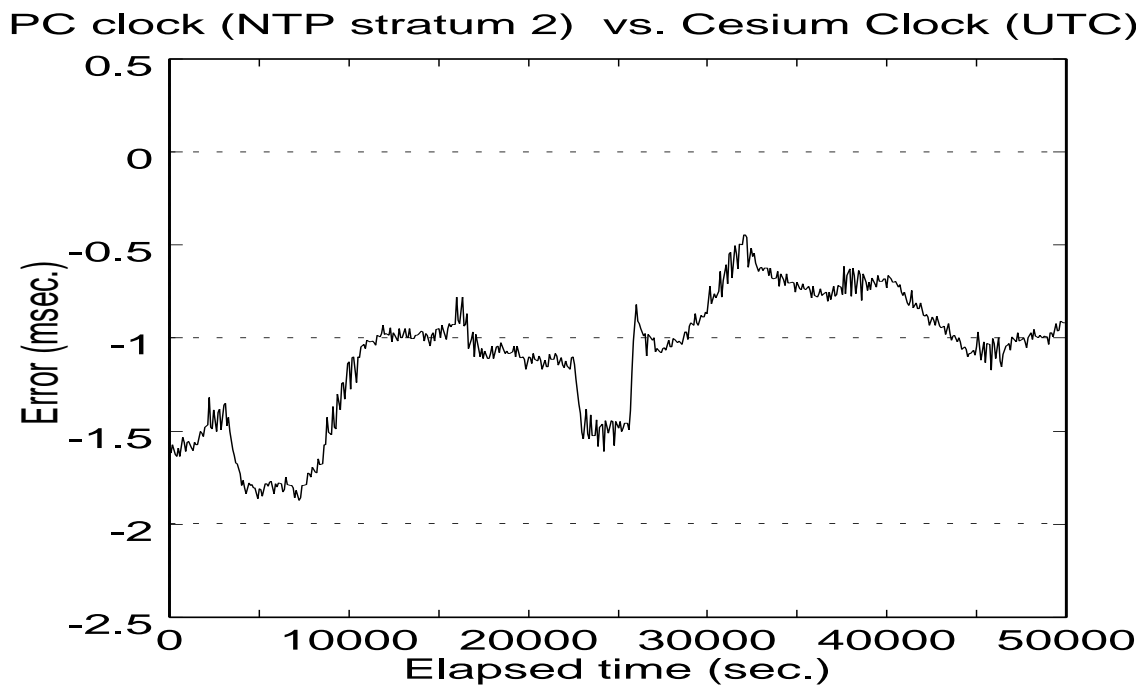


図 4.1: stratum 2 のゆらぎ

このデータは、stratum 2 は、UTC と比べておよそ 1msec 遅れており、 $\pm 0.5\text{msec}$ 程度のゆらぎがあることを示している。最近の主なワークステーションの OS のほとんどは、内部時計の粒度が $1\ \mu\text{秒}$ であることを考えると、このゆらぎと遅延はかなり大きいと言える。

第 5 章

GPS 受信機に基づく stratum 1 NTP サーバの精度と確度

ここでは、GPS 受信機として前述した GN-72 を用いて stratum 1 NTP サーバを構築した場合、サーバとして予想される性能を各種の実験を通じて評価する。

5.1 水晶発振器の精度

採用した水晶発振子の規格からすれば、試作した水晶発振器から得られる 1MHz 正弦波は、セシウム原子時計の出力する 1MHz 正弦波と同等の安定度を持つはずである。このことを確認するために、次の 3 種類の時刻供給方式を準備した。

1. 最初の時刻供給方式は、昨年度の報告書において紹介した方式と同じく、1pps 信号と 1MHz 信号の両方ともセシウム (Cs) 原子時計からカウント回路を介して計算機へ供給する方式である。以後、この方式を CS モードと呼ぶ。
2. 次の時刻供給方式では、1pps 信号はセシウム (Cs) 原子時計から、1MHz 信号は水晶発振器から各々カウント回路を介して計算機へ供給される。この方式を CX モードと呼ぶ。
3. 最後の時刻供給方式では、1pps 信号は GPS 受信機から、1MHz 信号は水晶発振器から各々カウント回路を介して計算機へ供給される。この方式を GX モードと呼ぶ。

これらの時刻供給方式間の差を調べるために、3 種類の実験系 (図 5.1 ~ 5.3) を準備した。最初の実験系 (図 5.1) は CS モードによって時刻を供給される計算機が保持する時刻同士をネットワークを介して比較するものであり、ここでは、参考のために再掲した。次の実験系 (図 5.2) は、CS モードと CX モードによって各々時刻を供給される計算機間で、ネットワークを介して時刻比較するものであり、水晶発振器とセシウム原子時計の 1MHz 信号同士を直接比較するために構成した。最後の実験系 (図 5.3) は、CS モードと GX モードによって各々時刻を供給される計算機間で、ネットワークを介して時刻比較するものであり、GPS 受信機と水晶発振器から構成される時刻信号発生システムの全体性能をみるためのものである。

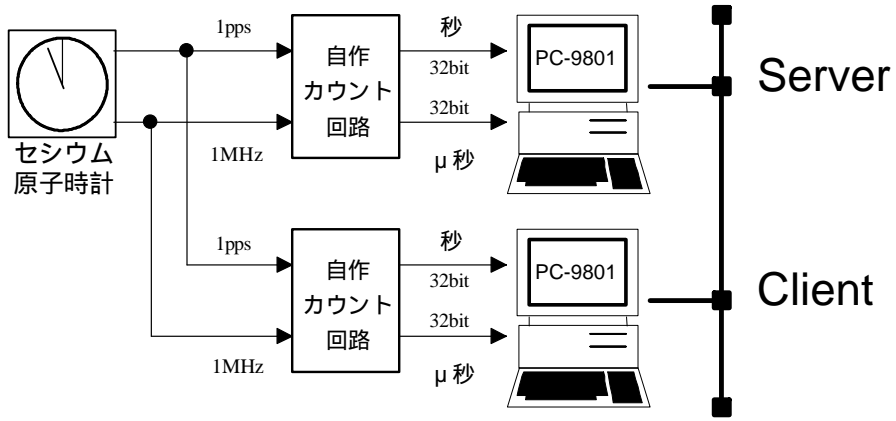


図 5.1: CS モード間の時刻比較実験ブロック図

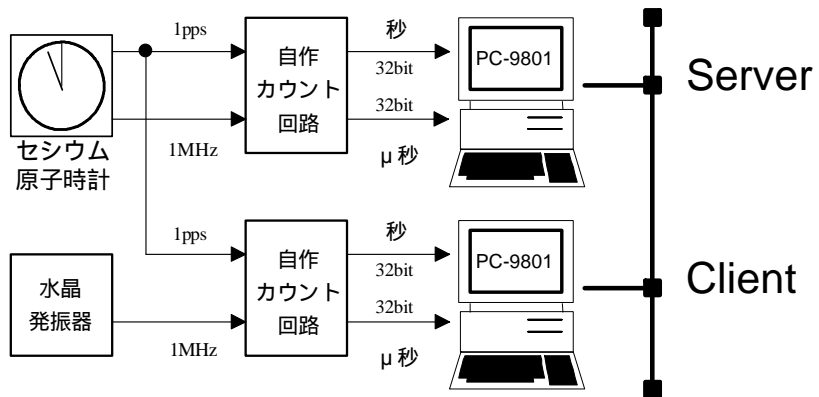


図 5.2: CS モードと CX モードの時刻比較実験ブロック図

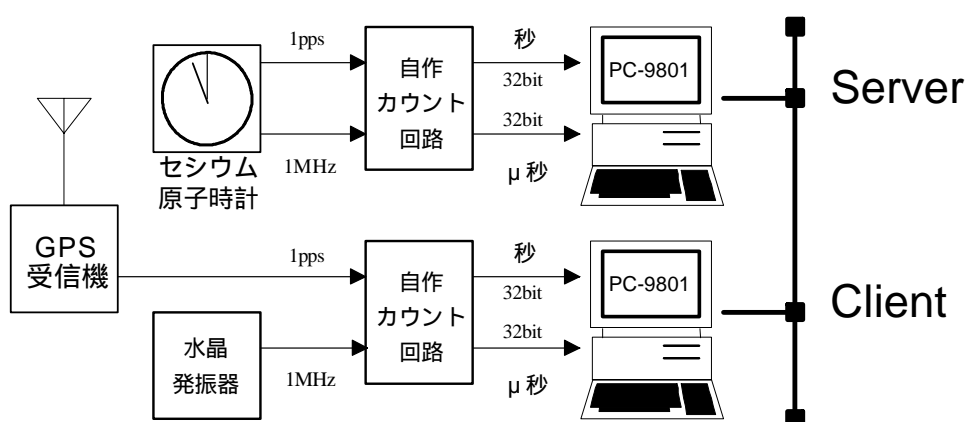


図 5.3: CS モードと GX モード間の時刻比較実験ブロック図

昨年度の報告書でも述べたが、CS モード間の時刻比較では時刻差の平均値はほぼ 0、標準偏差は約 $100\mu\text{s}$ であった。今回新たに行った CS モードと GX モード間の時刻比較でも、時刻差の平均値はほぼ 0、標準偏差はほとんど変わらず約 $100\mu\text{s}$ であった。すなわち、この結果をみる限り、ネットワークを介した時刻比較の用途には水晶発振器とセシウム原子時計の 1MHz 信号同士の差はないといってよい。一方、CS モードと GX モード間の時刻比較では、時刻差の平均値には、後に述べるような数 μs 程度のごく微小な（しかし系統的な）差が現れたが、標準偏差は変わらず約 $100\mu\text{s}$ 前後であった。従って、試作した水晶発振器は、セシウム原子時計と同等程度の 1MHz 正弦波を出力することが確認された。

5.2 GN-72 から得られる時刻信号の精度と確度

次に、GN-72 から得られる正秒信号の精度と確度を調査した。

そのためには、複数の GN-72 からの正秒信号を計測し、正秒信号のするタイミングの個々の受信機による系統的な差（すなわち機差）及び正しい協定世界時の正秒信号との差（すなわち確度）を得なければならない。また、GPS 自体が、時々刻々その配置が変る人工衛星によって運用されているため、計測された機差・確度ともに計測時刻に依存する可能性がある。したがって、この種の計測を長期間連続して行う必要がある。そこで、GN-72 を 2 台用意し、同一地点（国立天文台三鷹）において独立に作動させ、出力される正秒信号と、国立天文台三鷹においてセシウム原子時計群により維持されている協定世界時（UTC）に完全に同期した正秒信号との差を、それぞれ HP 社製の時刻間隔計測機器 HP5370A により 1994 年の 2 月から 3 月にかけて連続計測した。

この実験を実験 A と呼ぶ（図 5.4）。やむを得ない停電時などを除けば、データの欠測はほとんどなかった。

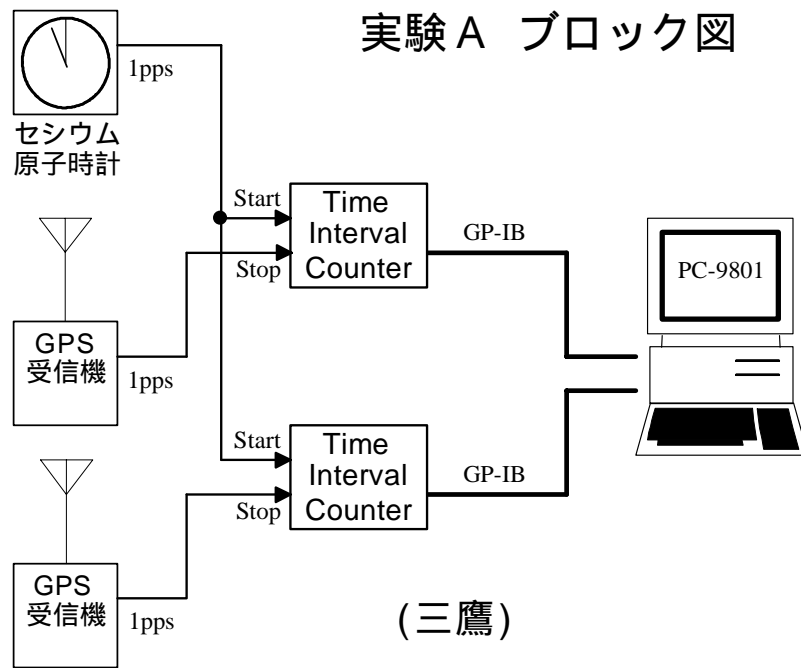


図 5.4: GPS 受信機時刻信号比較実験 (同一地点)

解析には、計測から直接得られる GPS 受信機別の差 (GPS1 - UTC、GPS2 - UTC と呼ぶ) と受信機同士の差 (GPS1 - GPS2) の各々について、毎秒の粗計測値を 60 回平均した値 (以後、毎分値と呼ぶ) を用いた。こうして得られた毎分値データの例として、GPS1 - UTC を図 5.5 に示す。

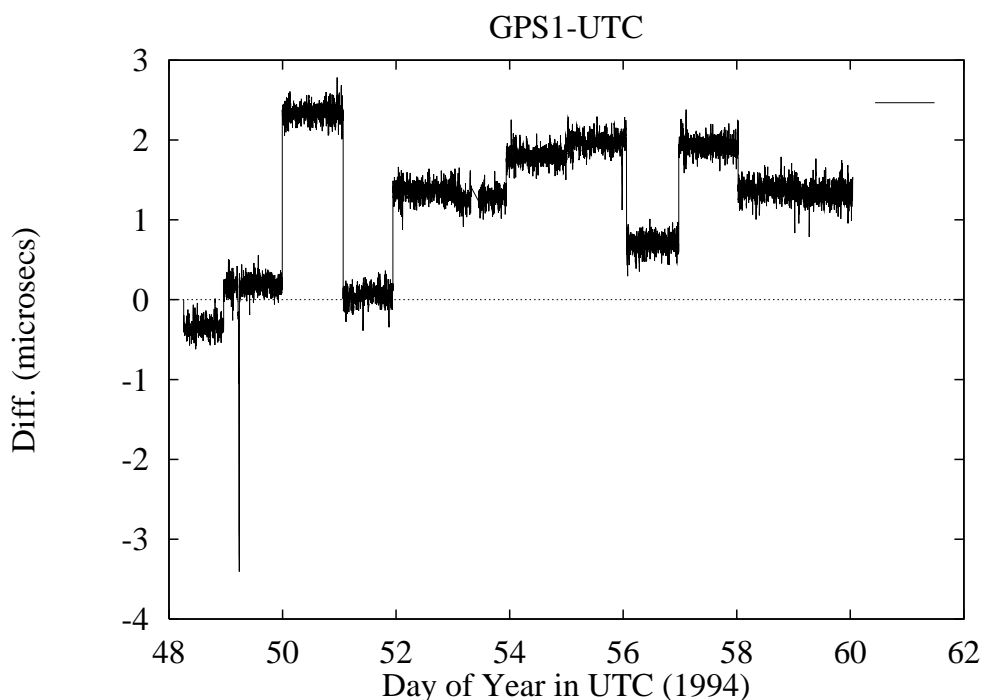


図 5.5: GN-72 の時刻信号と協定世界時との差 (実測値)

縦軸は GN-72 - UTC を表し、その単位は μs である。横軸は 1994 年の通日 (年初から数えた日数で元旦を 1 とする) である。通日の変わり目は UTC で 0 時である。

図からもわかるように、GN-72 から得られる正秒信号は、ほぼ 1 日間 (おおむね UTC の 0 時から 24 時まで) は一定であるが、日毎の値は数 μs 変化する。われわれの知る限り、正秒信号が取り出すことのできる他の GPS 受信機 (例えば、STI 社製、SONY 社製 GTT-3000、Trimble 社製 5000A) には、このような振舞は見られない。これは、GN-72 に特有な現象であろう。

さらに解析を進めるために、計測データが区分的に階段関数状に変化するという性質を念頭において、

1. UTC で同じ日付毎に毎分値データを分け、
2. 個々のデータについて、 3σ あてはめ¹ により誤データを除外したのち、

¹ 標準偏差の 3 倍以上の偏差を持つデータを除外してからもう一度標準偏差を計算しなおし、この過程が収束するまで繰り返す方法

表 5.1: 三鷹における GN-72 の時刻精度

月日	通日	GPS1-GPS2	GPS1-UTC	GPS2-UTC
2/17	48	0.05 ± 0.06	-0.36 ± 0.08	-0.42 ± 0.10
/18	49	0.05 ± 0.06	0.17 ± 0.09	0.13 ± 0.09
/19	50	0.04 ± 0.05	2.33 ± 0.09	2.29 ± 0.10
/20	51	0.04 ± 0.06	0.03 ± 0.09	-0.01 ± 0.10
/21	52	0.04 ± 0.05	1.37 ± 0.09	1.32 ± 0.10
/22	53	0.04 ± 0.05	1.29 ± 0.09	1.24 ± 0.10
/23	54	0.04 ± 0.05	1.80 ± 0.09	1.75 ± 0.10
/24	55	0.06 ± 0.06	1.97 ± 0.09	1.92 ± 0.09
/25	56	0.04 ± 0.06	0.70 ± 0.09	0.67 ± 0.10
/26	57	0.04 ± 0.06	1.94 ± 0.09	1.90 ± 0.09
/27	58	0.04 ± 0.06	1.37 ± 0.09	1.33 ± 0.10
/28	59	0.04 ± 0.06	1.35 ± 0.10	1.31 ± 0.11
3/01	60	0.04 ± 0.04	1.28 ± 0.12	1.23 ± 0.13

(注) 表中の単位は μs である。

3. 平均値とそのまわりのバラツキを求める

ことを行った。その結果を表 5.1 に示す。ただし、表中の通日とは年初から数えた日数(元旦の通日は 1) のことであり、表中の単位は μs である。また、ここでおこなった区分的階段関数あてはめ後の残差の一例として、GPS1 - UTC の場合を図 5.6 に示す。あきらかに、残差中に周期成分など系統的な変動は見られない。他の場合も同様であった。

表 5.1 より、以下のような結論を得る。

1. GN-72 から出る正秒信号の機差はほぼ 0 である。具体的には

$$\text{GPS1} - \text{GPS2} = +40 \pm 50\text{ns} \quad (5.1)$$

2. 受信機間の差をとることによってバラツキが約 100ns から約 50ns へと半減することから、同じ地点における GN-72 の時刻信号の変動の間には高い相関があることがわかる。
3. GN-72 の正秒信号は、peak-to-peak で $2.5\mu\text{s}$ ほどランダムに日変化する。
4. 一方の GN-72 の正秒信号と協定世界時との差をモニタすることにより他方の GN-72 の正秒信号の日変化を $0.1\mu\text{s}$ 程度の精度で (1 日遅れながら) 補正できる可能性がある。

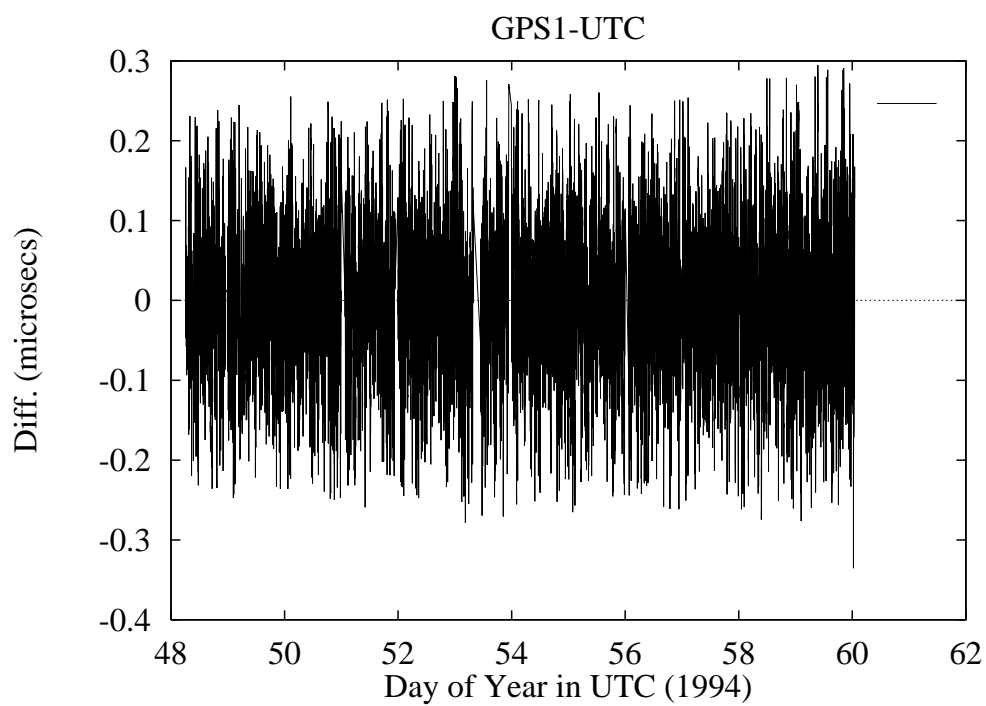


図 5.6: GN-72 の時刻信号と協定世界時との差 (区分的階段関数あてはめ後)

5.3 GN-72 の時刻信号の受信地点依存性

前節で得られた結果から、同一地点における GN-72 間の差はほぼ無視できることがわかったが、実際に運用する場合においては個々の GN-72 が置かれている場所は異なるため、これでは十分な性能評価とはいえない。

そこで、GN-72 が出力する時刻信号の、受信地点による差及び確度の時間依存性を調べることを目的として、離れた二地点 (国立天文台三鷹と国立天文台水沢) で 2 台の GN-72 を独立に作動させ、出力される正秒信号と三鷹及び水沢での協定世界時と同期されている正秒信号との差を、前節と同様の方法で 1994 年の 3 月末以降計測してきた (4 月末現在、計測を継続中である)。この実験を実験 B と呼ぶ (図 5.7)。

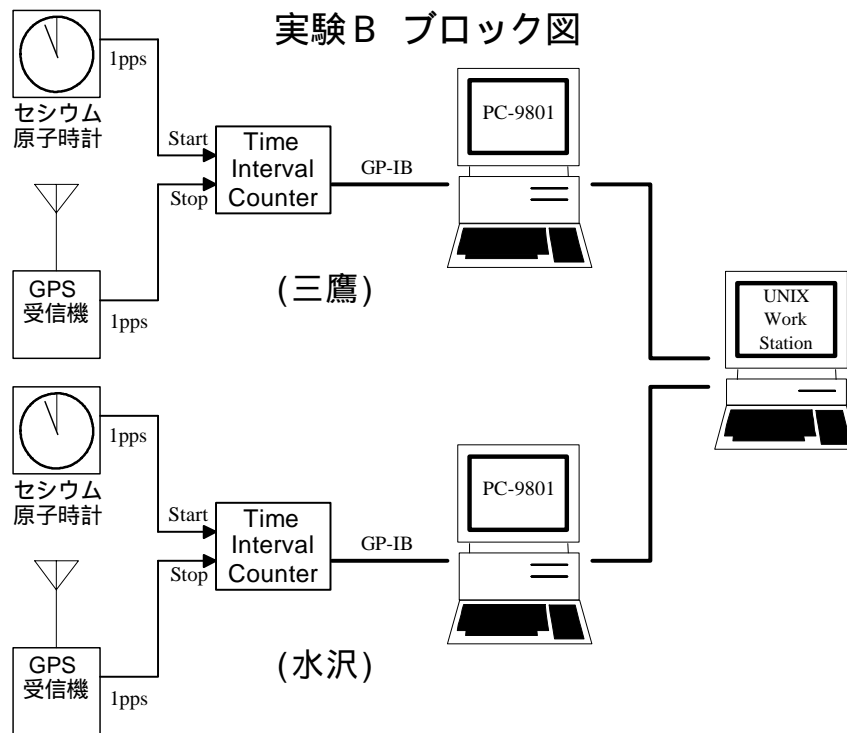


図 5.7: GN-72 時刻信号比較実験 (三鷹・水沢)

写真 5.8、5.9 は、水沢における計測風景及びアンテナの設置状況を示している。前節において特に述べなかったが、三鷹における実験もほぼ同じ状況で実施している。

GN-72 と協定世界時との差を

$$\Delta T \equiv \text{GPS} - \text{UTC}$$

と書くことにする。三鷹と水沢における ΔT 及び両者の差 $\Delta T_{\text{水沢}} - \Delta T_{\text{三鷹}}$ の各々について、前節と同様のデータ処理を施した結果を表 5.2 に示す。データの取得状況、区分的階



図 5.8: GN-72 時刻信号計測風景 (水沢)



図 5.9: GN-72 アンテナ設置状況 (水沢)

表 5.2: 三鷹と水沢における GN-72 の時刻精度

月日	通日	$\Delta T_{\text{水沢}}$	$\Delta T_{\text{三鷹}}$	$\Delta T_{\text{水沢}} - \Delta T_{\text{三鷹}}$
3/23	82	1.017 ± 0.109	3.053 ± 0.101	-2.028 ± 0.089
/24	83	-0.113 ± 0.455	1.110 ± 0.396	-1.601 ± 0.512
/25	84	-0.733 ± 0.105	1.404 ± 0.104	-2.144 ± 0.105
/26	85	-0.741 ± 0.118	1.426 ± 0.106	-2.168 ± 0.101
/27	86	-0.716 ± 0.102	1.440 ± 0.106	-2.156 ± 0.092
/28	87	-0.697 ± 0.106	1.483 ± 0.119	-2.173 ± 0.094
/29	88	-0.678 ± 0.106	1.496 ± 0.103	-2.171 ± 0.088
/30	89	-0.628 ± 0.112	1.404 ± 0.104	-2.131 ± 0.100
/31	90	-0.689 ± 0.106	1.426 ± 0.106	-2.249 ± 0.105
4/01	91	-0.697 ± 0.107	1.440 ± 0.106	-2.272 ± 0.103
/02	92	-0.679 ± 0.105	1.483 ± 0.119	-2.270 ± 0.108
/03	93	-0.627 ± 0.115	1.496 ± 0.103	-2.223 ± 0.111
/04	94	-0.759 ± 0.104	1.506 ± 0.108	-2.352 ± 0.099
/05	95	-0.621 ± 0.097	1.559 ± 0.129	-2.293 ± 0.098
/06	96	-0.685 ± 0.105	1.567 ± 0.114	-2.358 ± 0.112
/07	97	-0.710 ± 0.105	1.585 ± 0.124	-2.397 ± 0.106
/08	98	-0.796 ± 0.101	1.597 ± 0.120	-2.480 ± 0.096
/09	99	-0.684 ± 0.096	1.598 ± 0.110	-2.381 ± 0.085
/10	100	-0.683 ± 0.101	1.694 ± 0.121	-2.398 ± 0.099
/11	101	-0.677 ± 0.111	1.679 ± 0.123	-2.439 ± 0.107
/12	102	-0.739 ± 0.105	1.699 ± 0.107	-2.513 ± 0.101
/13	103	-0.693 ± 0.100	1.690 ± 0.104	-2.530 ± 0.120
標準偏差の平均		0.106	0.111	0.101

(注1) 推定誤差の平均の計算には、異常値とおもわれる通日 83 のデータを除外した。

(注2) 表中の単位は μs である。

段関数あてはめ後の残差の様子などは、同一地点の実験結果とほぼ同じであった。

表 5.2 より、以下のような結論を得る。

1. 通日 83 のデータは階段関数あてはめ後の標準偏差(表中で±の右側の数字)が、場所によらず大きい。生データを詳細に検討した結果、これは値の階段関数的な変化が UTC での日の変わり目ではなく、中途半端な時刻で起きているためであることがわかった。
2. 今回の実験期間中では、図 5.5 にみられるような GN-72 の正秒信号の大きなジャンプは 1 回しか起きていない(通日 82 と通日 84 以降)。
3. データのばらつきはどれも 100ns 程度である。これは、三鷹同時観測のときの ΔT のバラツキと同程度である。地点間の差をとることによってバラツキが増えもしなければ減りもしないことから、同時観測のときとは異なり、異地点における GN-72 の時刻信号のゆらぎは無相関というほどではないが、その相関はかなり弱いことがわかる。
4. 1 回起きたジャンプの前後で、 $\Delta T_{\text{水沢}} - \Delta T_{\text{三鷹}}$ は有意に変化していない。このことは、一方の GN-72 と協定世界時との差をモニタすることにより、他方の GN-72 の正秒信号の日変化を 0.1 マイクロ秒程度の精度で、ほぼリアルタイムで補正できることを示唆している。

この実験により得られた $\Delta T_{\text{水沢}} - \Delta T_{\text{三鷹}}$ の値は、現実の三鷹・水沢間の協定世界時の差、すなわち実際に UTC を決定するために採用されている方法(GPS 受信機で同じ衛星を同時に受信する方法で同視野観測法 Common View Method と呼ばれる)による両者の差と一定値(数 μs) だけ異なっていた。このオフセットが生じた原因は現在も調査中であるが、従来の方法が異なる GPS 受信機を用いていること、受信機同士の機差についてはメーカーの公称値をそのまま用いていることなどの点から考えて、機差を事前に測定済の同型の受信機で得られている今回の結果のほうがより信頼度が高いと考えている。

第 6 章

広域ネットワーク上における新しい時刻同期機構の提案

NTP を利用した時刻同期システムでは、各 stratum のサーバの数、ネットワーク上での設置場所、および参照関係が不適當であると、ネットワーク上に NTP パケットが大量に流れたり特定のサーバに問い合わせが集中するといった事態に陥ることがあり得る。NTP パケットがネットワーク上の特定のホストに集中すると、時刻情報のやりとりにかかる時間とそのゆらぎが増大することになり、決定される時刻の精度を下げてしまうことになる。たとえば、国立天文台のセシウム原子時計ベースの stratum 1 を日本中の組織の stratum 2 が一斉に参照したとすると、同 stratum 1 を運用している計算機と同機周辺のネットワークに大きな負荷がかかると予想される（図 6.1）。

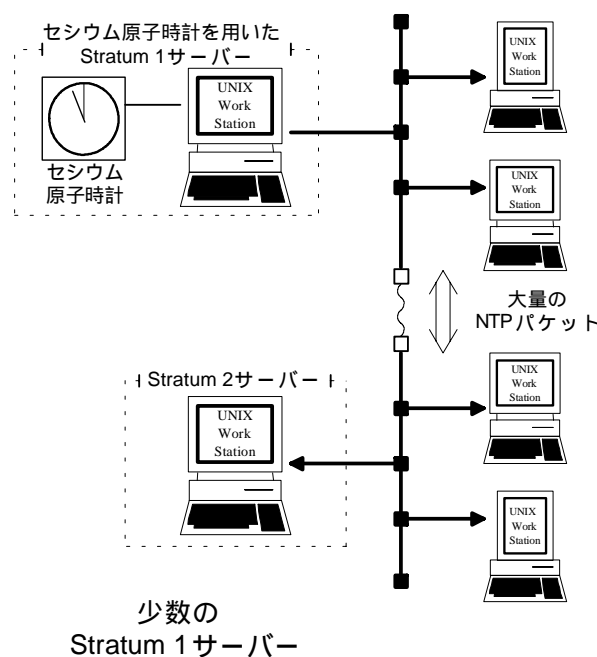


図 6.1: 従来の NTP システム

このような従来型の運用形態、すなわち各組織では stratum 2 を運用し、これらの stratum 2 はインターネット上の何箇所かにある stratum 1 を参照するという運用形態のままでは、伝送遅延時間のゆらぎ等の影響が深刻で、時刻同期精度をミリ秒のオーダーより良くすることは難しい。

そこで、手軽に実現可能な GPS 受信機を使った NTP サーバを各組織で stratum 1 として運用し、

- 組織内の計算機は、この stratum 1 を参照して時刻合わせを行う。
- 組織内の stratum 1 は、必要な場合にのみ、セシウム原子時計を使った NTP サーバにはアクセスする。

という方法が考えられる。この方法を採用すれば、インターネット上を流れる NTP パケットの量は従来の「インターネット上の少数の NTP サーバが、多数の組織からの時刻同期要求を処理する」という方法に比べて大幅に減少し、セシウム原子時計を使った NTP サーバの負担も減少する(図 6.2)。

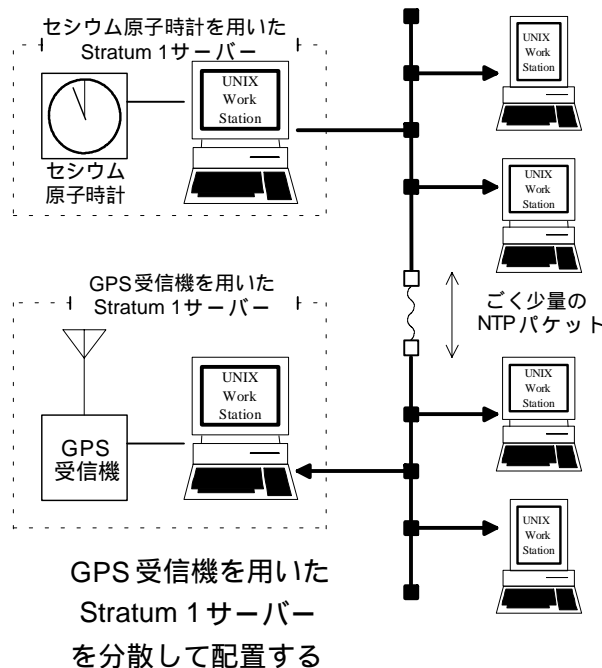


図 6.2: 分散型 NTP システム

このように、セシウム原子時計ベースの NTP サーバと GPS 受信機ベースの NTP サーバを使い分ける方式をここでは「分散型 NTP システム」と呼ぶ。分散型 NTP システムでは、セシウム原子時計を使った NTP サーバを stratum 0 として運用し、問い合わせを制限するといったことも考えられる。なお、この方式を用いた場合、各組織で運用され

る stratum 1 を参照して時刻同期を行う stratum 2 の期待される時刻同期精度はマイクロ秒程度となる。

さらに高精度の時刻同期が必要とされる場合には、GPS 受信機が出力する時刻信号と UTC の時刻差を定点（例えば国立天文台）で常時モニタし、その結果及び過去の傾向を加味した予測値を anonymous ftp, gopher, net news などの方法で常に公表し、各組織の GPS 受信機を使った NTP サーバに対し、正しい UTC に変換するための情報を提供するというしくみを用意すればよい（図 6.3）。この方式を、ここでは「時刻差予報型 NTP システム」と呼ぶ。

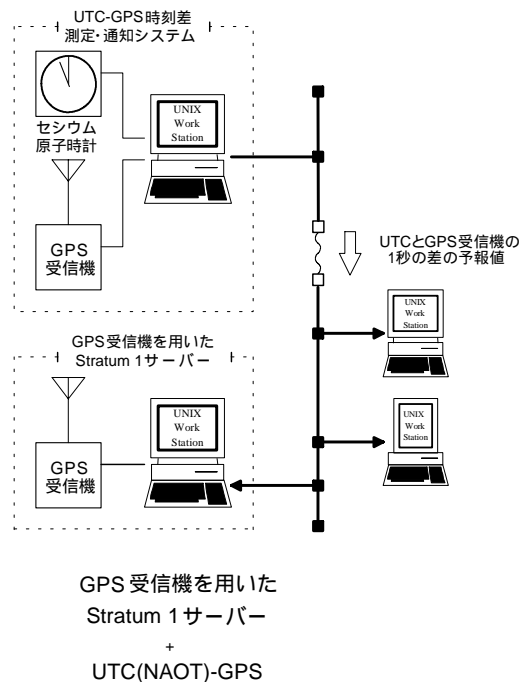


図 6.3: 時刻差予報型 NTP システム

時刻差予報型 NTP システムでは、UTC(NAOT) と GPS 受信機の時刻差の予報値を GPS 受信機の機種ごとに通知する。利用者は、この情報に基づいて時刻の補正を行うことにより、GPS 受信機を使った NTP サーバの精度をさらに約一桁向上させることができる。この方式を用いると stratum 2 で期待される時刻同期精度を数 100 ナノ秒程度とすることができる（図 6.4）。この時刻差予報型 NTP システムは、ネットワークに無用な負荷をかけずにマイクロ秒以下の精度の時刻情報を提供できるという大きな利点があるため、今後積極的に利用されるようになると思われる。

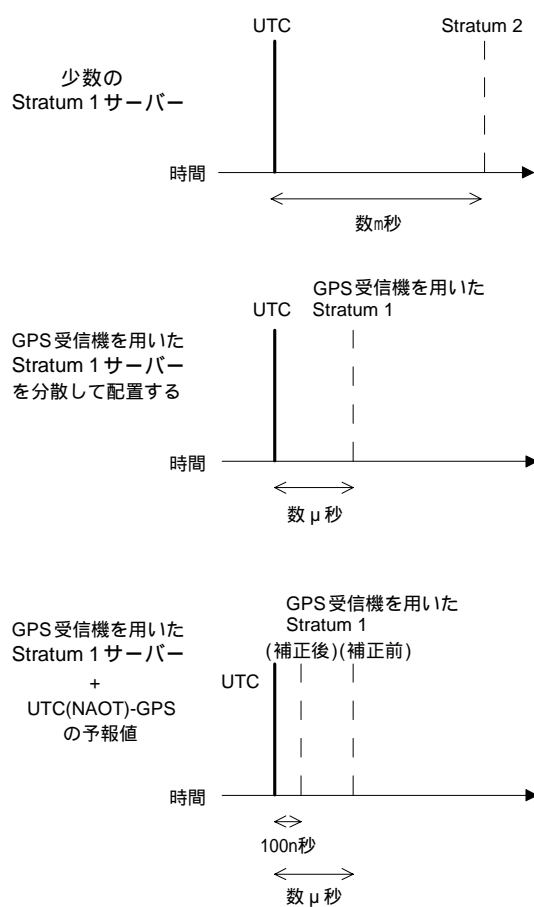


図 6.4: 各 NTP システムによる時刻同期精度

第 7 章

おわりに

本報告では、まずセシウム原子時計や GPS 受信機を用いた時刻同期機構の研究開発状況について報告し、次に広域ネットワーク上における時刻同期機構の今後のありかたについての検討結果を述べた。その概要は以下のとおりである。

- セシウム原子時計を用いた時刻同期機構を構築するため、PC 上で稼働する UNIX のカーネルを改造して、時刻情報をカーネル内に直接取り込む方法を採用した。この結果 `gettimeofday()` システムコールは、セシウム原子時計精度の時刻を返すようになった。
- GPS 受信機を利用した時刻同期機構の開発も進んだ。GPS 受信機から時刻を情報取得し RS-232C 経由でワークステーションに送る方式は、以前から `xntpd` 等で利用されているが、この方法では、時刻情報の取り込みがユーザプロセスで行われるため、十分な精度が得られないという問題があった。そこで、著者らは、GPS 受信機からの信号をカウンタ回路で数えておき、この値を UNIX のカーネル内にパラレル入力インタフェースを介して直接取り込む方法を採用した。

さらに、

- GPS 衛星からの時刻情報を時刻源として稼働する `stratum 1` を運用すれば、他組織との間で NTP による時刻同期を行わなくても実用上は十分な精度が得られる。
- このようにして運用されている `stratum 1` の時刻と UTC とのずれは、NTP によらない方法で通知でき、各組織ではこの値を利用して時刻を修正すれば各組織の `stratum 1` の精度を数 100 ナノ秒程度の精度に保つことができる。
- この方法を用いると、組織間に流れる NTP パケットを無くすることができる。

の 3 点を指摘した。

今後は、GPS 受信機を利用した `stratum 1` の普及を促進するため、さまざまな GPS 受信機用のドライバソフトウェアの開発や、GPS ベースの `stratum 1` と UTC との時刻差等の情報を提供する機構を整備する予定である。

ところで、インターネット上のほとんどの組織が GPS 受信機を利用した NTP `stratum 1` を運用するようになるまでには、しばらく時間がかかると思われる。したがって、イン

ターネット上でどのような NTP の参照関係を樹立するかについては、今後も引き続いて重要な検討事項となろう。この問題に関しては、JEPG/IP がコーディネーションを行うとのことなので、NTP ワーキンググループは、このコーディネーションにしたがうだけでなく、積極的に支援したいと考えている。