

ネットワーク時空情報サーバと動的ユーザランド時刻クライアントの開発

町澤 朗彦^{†a)} 岩間 司[†] 鳥山 裕史[†]

Development of a Network Space-Time Server and a Dynamic Userland Time Client

Akihiko MACHIZAWA^{†a)}, Tsukasa IWAMA[†], and Hiroshi TORIYAMA[†]

あらまし ユビキタス社会では、時空（時刻と位置などの空間座標）情報を利用することにより、分散している多数の端末を有機的に結び付けた効果的な利用が容易となる。本論文では、GPS 電波の届きにくい屋内と GPS 機能を有しない端末を対象に、ネットワークを介して時空情報を提供するサーバ/クライアントシステムを開発したので報告する。サーバには、GPS 信号を利用して、毎正秒に高精度な時刻情報と位置・速度情報をネットワーク上にブロードキャストする構造の単純な専用ハードウェアを開発した。一方、PC 用クライアントソフトウェアには、処理コストが小さく精度の高い動的ユーザランド時刻取得関数を実装した。時刻同期方式には、毎正秒パケット到着間隔方式 (PAI) を用いている。従来方式と比べた結果、精度も処理コストも 1 けた以上の改善が確認された。しかも、提案方式は、時空情報パケットが毎秒ネットワークに供給され、同期所要時間も数秒以内であるため、省電力のために必要なときだけ稼動するユビキタスノードには最適と考えられる。また、位置情報は GPS アンテナ位置となるため、精度は建物位置程度の粗粒度精度を与える。

キーワード ユビキタスコンピューティング、時空情報、高精度時刻同期、ロケーションウェアコンピューティング

1. ま え が き

ユビキタス社会では、時刻と空間座標（位置及び速度）などの時空情報を利用することにより、分散している多数の端末を有機的に結び付けた効果的な利用が可能となる [1], [2]。必要とされる時空情報の精度は用途に依存するが、近年、時刻情報では計測・制御分野 [3] で高精度（マイクロ秒）な要求が高まっている一方で、空間情報については、緊急通話での位置情報通知をはじめとする位置情報サービスなどで、粗粒度（数十から数百メートル）精度の利用が活発化してきている。

筆者らは、これまで、ソフトウェア処理のみでサブマイクロ秒精度を可能とする毎正秒パケット到着間隔時刻同期方式 (PAI 方式) に関して、原理的な研究を

進めてきたが [4], [5]、実用化するには、GPS 受信機のジッタの影響やアプリケーションプロセスへの時刻供給方法などの検討が不足していた。一方、ネットワークを介した位置情報の提供方法として、DHCP の位置情報オプションが提案されているが [6]、送受信を必要とし、省電力機器に適したブロードキャストを用いた時刻供給方式は、まだ提案されていない。

そこで、本論文では、ネットワークに時空情報を提供するネットワーク時空情報 (Network Space-Time: NST) サーバを試作し、高精度時刻同期のために GPS を時刻源とする PAI 方式の同期特性を検証するとともに、アプリケーションプロセスへの高精度な時刻提供手法を提案する。更に、粗粒度空間情報を提供するために、時刻同期パケットの未利用フィールドに空間情報を載せることにより、新たなネットワーク負荷を与えずに、空間情報を供給するシステムを提案する。空間情報と時刻情報を統合することにより、広域ネットワークでの複数地域情報選択問題を解決することができる。

[†] 情報通信研究機構, 小金井市

National Institute of Information and Communications Technology, 4-2-1 Nukui-Kitamachi, Koganei-shi, 184-8795 Japan

a) E-mail: machi@nict.go.jp

今回、試作したサーバは小型・低コストでありながら、サブマイクロ秒の精度を有し、クライアントでは、時刻同期デーモンで更新される時刻変換パラメータを、共有メモリを用いてアプリケーションプロセスから動的に利用することにより、高精度な時刻をシステムコールより1けた以上高速に取得することが可能となり、精度76 ns (標準偏差)を実現した。更に、広域に広がるネットワーク上にサーバが複数存在する場合でも、時空情報パケットが同時に送出される特徴を利用し、最寄りの位置を特定することができる。

本論文では、まず、2. で高精度時刻同期と粗粒度精度の空間情報に関する研究をまとめ、3. で今回試作したNSTサーバの動作原理と時空情報パケットフォーマットについて述べ、4. でクライアントの実装方法と性能について述べる。

2. 関連研究

IEEE 1588 では、高い時刻精度を得るために、クライアントのネットワークインタフェースでのタイムスタンプなど、ハードウェアの支援を必要としているが[3]、筆者らはソフトウェア処理のみで、サブマイクロ秒精度を得ることのできる毎正秒パケット到着間隔 (PAI: Packet Arrival Interval) 時刻同期方式を開発してきた[5]。しかし、文献[5]では原理的な動作確認のために、時刻源には、要求精度と比べて無視し得るピコ秒精度の原子時計が用いられているが、協定世界時との同期やコストを考慮すると、実用的にはGPSの利用が望ましい。しかし、一般の民生品として用いられているGPS受信機の時刻精度仕様はマイクロ秒であり、クライアントの要求精度と同程度の精度しか有しておらず、無視することができないため、GPSを用いた場合の同期過程を含むシステム全体での同期精度と挙動を実証する必要がある。

また、アプリケーションプロセスが時刻を取得する関数 (例えば `gettimeofday()/clock_gettime()`) は一般にはシステムコールとして与えられるが、システムコールはオーバーヘッドが大きいため、Obata はユーザランドで実行するユーザランド `gettimeofday()`^(注1)を開発した。しかし、プロセッササイクルカウンタ (Processor Cycle Counter: PCC) から時刻への変換パラメータに、コンピュータの起動時に取得された静的な値を使っているが、PCを駆動している水晶の動作周波数は温度変化が激しいため、精度が低い欠点を有していた。筆者らは、これまで、PCでジッタの少

ないPCCを計測する手法[4]や毎正秒パケット到着間隔時刻同期方式 (PAI) [5]などの研究を進めてきており、時刻同期プロセスでの高精度な内部時計を構成することに成功しているが、他のアプリケーションプロセスへの時刻供給手法については、まだ示されていない。

一方、空間情報では、社会の安全・安心に対する要請の高まりから、米国の緊急電話に関するExtended 911指令で、発信者の位置情報を50mから300mの精度で特定できることが求められており、我が国でも同様の指針[7]が出されている。また、子供たちの安全を確保するための「位置確認」機能が携帯電話各社から提供されてきている。更に、コンテキスト依存情報提供などの位置情報サービスでも、粗粒度精度の位置情報の利用が活発になってきている。さて、屋外ではGPSを用いることにより時空情報が得られるが、屋内にはGPSの電波は届きにくい。屋内でもGPS機器が利用できるように、GPSリピータなどが開発されているが、元来屋内の利用を想定している機器はGPS機能を有してはいない。最近、インターネットを利用した音声通話 (VoIP) の利用が急速に拡大しており、近い将来、携帯電話と同様に用いられる可能性が高くなってきているが、VoIPでは位置情報の特定が困難であることが指摘されている^(注2)。しかし、これらのネットワーク機能を有している機器では、ネットワークを介して時空情報が提供されれば、新たなハードウェアを追加することなく、容易に時空情報を利用することが可能である。

ネットワーク上で位置情報を得る手段として、DHCPの位置情報オプション[6]が提案されている。しかし、DHCPを用いた場合、送受信が必要であるが、送信は受信よりも多くの消費電力を必要とする。センサノードなど消費電力に厳しいシステムでは、送信を避ける必要がある。一方、位置情報をブロードキャストすれば、クライアントは受信のみで済むが、近年、広域イーサネットやVPNの利用が増加し、本社と支社が同一ネットワークセグメント (サブネット) で結ばれるなど、ネットワークが地域的に広がってきている。このような広域にわたるネットワークで、各地域の位置情報がブロードキャストされた場合、最寄りの位置情報を特定することは困難である。

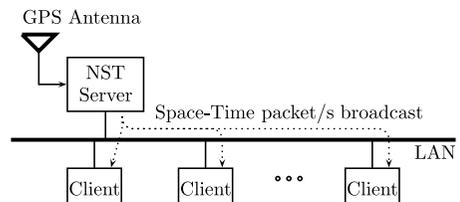
(注1): <http://www.cs.rpi.edu/~obatan/ugettime/ugettime-0.1.1.tar.gz>

(注2): <http://www.voip911.gov/>

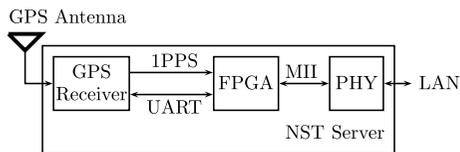
3. ネットワーク時空情報サーバ

3.1 動作原理及び仕様

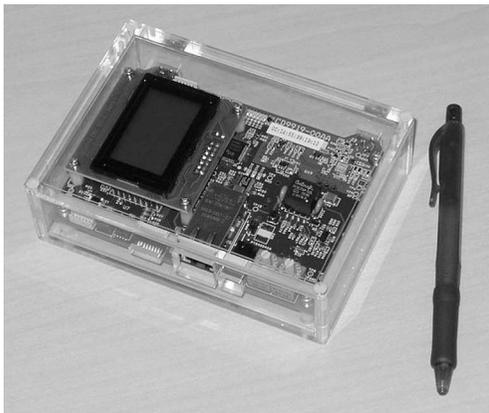
ネットワーク時空情報 (NST: Network Space-Time) サーバは、GPS 信号を利用して、毎正秒に時空情報パケットをネットワークにブロードキャストする。図 1 に (a) システム構成、(b) サーバのブロック図及び (c) 概観を示す。NST サーバは、LAN 上で、最も GPS アンテナからアクセスの良い場所（例えば建物の最上階など）に設置する。これは、一般的に、GPS に用いられる同軸ケーブルよりも LAN ケーブルの方が、敷設コストや距離の制約が小さいからである。また、NST サーバの構造は単純であり、主要な回路は、1 チップ化された GPS 受信モジュール、FPGA とネットワークの PHY だけで構成可能である。ネットワーク機能も、イーサネットブロードキャストフレームを送出す



(a) Space-time information broadcasting network



(b) Block diagram of network space-time server



(c) An overview of a NST server

図 1 ネットワーク時空情報 (NST) サーバ
Fig. 1 Network space-time (NST) server.

るだけなので、IP スタック等は不要である。

さて、NST サーバの動作は、GPS 受信モジュールの UART (Universal Asynchronous Receiver Transmitter) 端子を介して得られる時空情報を用いて、次の正秒における時空情報パケットを作成し、1 plps/s (Pulse per Second) 信号を待つ。1 plps/s 信号が入力されたら、用意しておいた時空情報パケットをネットワークにブロードキャストする。なお、これらの機能は FPGA によるハードウェアによって実装しているため、処理ジッタは生じない。ただし、正秒のタイミングは、GPS 受信機の精度に依存し、民生品の仕様精度はマイクロ秒程度である^(注3)。

ここで、今回試作した NST サーバの仕様を表 1 に示す。また、図 2 に、本サーバで用いている GPS 受信モジュールの発生する 1 plps/s 信号のタイミング精度を示す。(a) はオフセットの時間変動を示し、(b) はオフセットの大きさをヒストグラムで表している。なお、ヒストグラムのピン幅は 5 ns である。また、オフセットは日本標準時に同期させた Cs 原子時計との差を示している。

さて、オフセットの変動は ±50 ns ほどに収まっているが (標準偏差 17 ns)、数十秒間にわたるドリフトが観測された。これは、GPS 衛星の移動に起因していると思われるが、GPS 受信モジュール GXB5005 の時刻取得アルゴリズムは公開されていないため、詳細は不明である。

3.2 時空情報パケットフォーマット

本サーバは、図 3 で示すフォーマットのペイロードを有する IPv4/UDP パケットをイーサネット上に

表 1 NST サーバの諸元
Table 1 Specification of NST server.

| Item | Spec. |
|----------------------|-----------------------------------|
| GPS receiver module | SONY GXB5005 |
| FPGA | Xilinx Spartan-3 |
| PHY | Intel LXT972A, 100Base-TX |
| Packet Tx timing | every second (jitter 17 ns r.m.s) |
| outbound latency | 320 ns (TYP) |
| Positioning accuracy | 2 m (ANT.) |
| Packet format | IPv4/UDP |
| Tracking sensitivity | -152 dBm |
| Setting | none |
| size | 13 cm × 10 cm × 4 cm |
| Weight | 333 g (incl. case weight 216 g) |
| Sound noise | none (fan-less) |
| Power supply | PoE or AC adapter |

(注3): 例えば、本サーバで使用している SONY GXB5005 など。

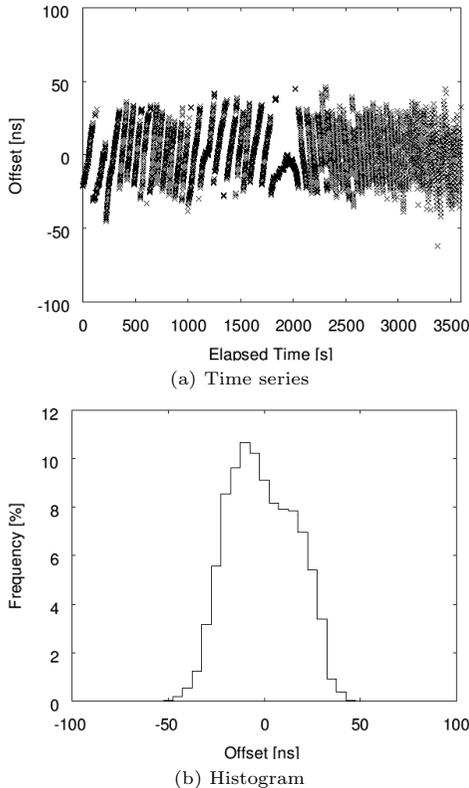


図 2 GPS 受信モジュールの 1pls/s タイミング精度
Fig.2 1 pls/s timing accuracy of GPS receiver.

| | | | | | |
|-------|----------------------|---------|-----|----------|----|
| | 0 | 8 | 16 | 24 | 31 |
| 0000h | Magic | | Ver | GPS mode | |
| 0004h | LeapSec | Reserve | | | |
| 0008h | UNIX Time (TAI): sec | | | | |
| 000Ch | outbound latency: ns | | | | |
| 0010h | Reserve | | | | |
| 0014h | Reserve | | | | |
| 0018h | Latitude: degree | | | | |
| 001Ch | Longitude: degree | | | | |
| 0020h | Altitude: m | | | | |
| 0024h | Track: degree | | | | |
| 0028h | Speed: km/h | | | | |
| 002Ch | Reserve | | | | |

図 3 時空情報パケットフォーマット
Fig. 3 The packet format of a space-time information packet.

ブロードキャストする。さて、ネットワーク遅延はパケット長に依存するが、時空情報パケットの長さは、NTP パケットに等しい長さで設計してあるため、ネットワーク遅延を補償する際に、NTP を利用することも可能である（ネットワーク遅延の補償については、3.1 参照）。したがって、時刻情報を配信するだけでも、UDP ペイロードを 48 バイトとする必要があり、空間情報を同時に載せても、新たなトラフィックは発生していないと考えることができる。なお、時空情報パケットのトラフィック量は毎秒 1 パケット（608 bit/s、IP ヘッダ含む）であり、回線に占める割合は、FastEthernet（100 Mbit/s）でわずかに 0.0007%（IFG 及び FEC 含む）である。なお、各フィールドは以下のとおりである。

- Magic* : 時空情報パケットを識別するためのビットパターン (6A88h)
- Ver* : パケットフォーマットバージョン (現在は 2)
- GPS mode* : GPS 測位モード (1: 非測位, 2: 2D 測位, 3: 3D 測位)
- LeapSec* : 累積うるう秒 (TAI-UTC)
- UNIX Time (TAI)* : 1970/01/01 を起点とする積算秒数で、うるう秒もカウントアップする
- outbound latency* : 1 pls/s の立上りから、パケット先頭がネットワークインタフェースから送出されるまでに要する時間
- Latitude* : 緯度
- Longitude* : 経度
- Altitude* : 海拔高度
- Track* : 進行方向
- Speed* : 速度

4. 時空クライアント

4.1 動作原理

時空クライアントは特殊なハードウェアを用いず、一般的なパーソナルコンピュータ上にソフトウェアで実装した。クライアントは、時空デーモン NSTd と、アプリケーションとのインタフェースである API (Application Programming Interface) からなる。クライアントの構成を図 4 に示す。

NSTd では、時空情報パケットから時刻情報 (*T*) と空間情報 (*P*) を取り込むとともに、受信時の割込み

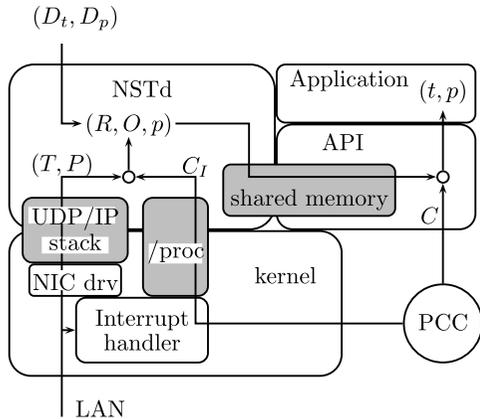


図 4 クライアント処理のブロック図
Fig. 4 Block diagram of client process.

ハンドラでの PCC の値 (C_I) を `/proc` ファイルシステムを介して取得し、PAI 同期アルゴリズムに基づいて、PCC と時刻を変換するために必要なパラメータ (R : 周波数, O : オフセット) を時空情報パケット受信ごとに算出する。なお、アンテナ位置からクライアントまでの遅延時間とクライアント内部の時間 ($s_{incoming} + s_{interaction}$) は事前知識 D_t として補償している [5]。また、サブマイクロ秒ほどの精度を求めない場合には、NTP と同様の往復遅延から推定することも将来的には可能である。今回開発した NST サーバでは、時刻同期に関して GPS 時刻源での PAI 方式の性能を実証するために、NTP サーバ機能を実装してはいないが、既に、NTP サーバ機能の FPGA による実装は済んでおり [8]、NST サーバへの実装も容易である。一方、空間情報に関しては、本論文では、粗粒度精度を対象としているため、NST サーバから提供される P をそのまま使うことが可能であるが、GPS アンテナ位置からの相対位置 D_p を事前知識として与えられれば、より正確な空間情報 p を算出することが可能となる。

API では、アプリケーションからのリクエストに応じて PCC (C) を取得し、毎秒更新されている動的時刻変換パラメータを共有メモリを介して取得し、現在時刻 t と空間情報 p を返す。なお、 t は次式に従って与えられる。

$$t = RC + O \quad (1)$$

API には、現在時刻を取得する `nst_gettime()` と空間情報を取得する `nst_getlocation()` の二つを用意して

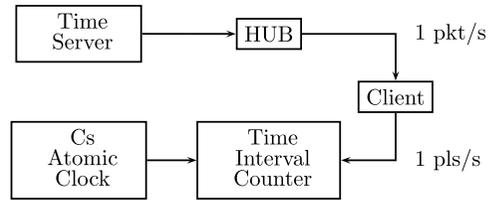


図 5 同期精度計測ブロック図
Fig. 5 A block diagram for synchronization accuracy measurement.

表 2 同期精度の比較に用いたシステム
Table 2 Systems to evaluate synchronized accuracy.

| System | Time Server | Client | Time Func. |
|-----------|--------------|-------------------|-----------------------------|
| NST | NST server | NSTd | <code>nst_gettime()</code> |
| NTP | HW Sntp srv. | <code>ntpd</code> | <code>gettimeofday()</code> |
| IEEE 1588 | OnTime T200 | PTPd | <code>gettimeofday()</code> |

表 3 測定に用いた PC の諸元
Table 3 Specification of the PC.

| Item | Spec. |
|--------------------|----------------------------------|
| Mother board | Supermicro P4DPR-iG2 |
| CPU | Intel Xeon 2.4 GHz |
| Network controller | ADMtek AN983(B) drv tulip-0.9.14 |
| Digital IO Board | Interface PCI-2703 |
| OS | linux 2.4.32 |

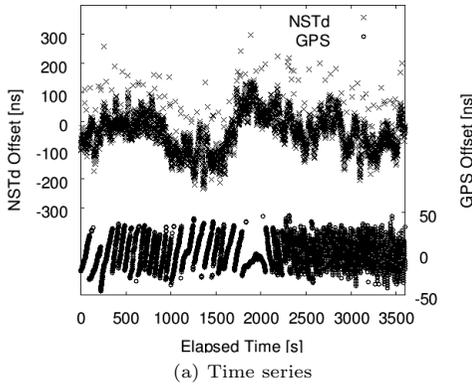
いる。

NST サーバが設置されている LAN では、時空情報が 1 秒間隔でブロードキャストされているため、センサネットのノードのように、省電力を図るために必要なときのみ電源の入るようなクライアントでも、起動から 1 秒以内に時空情報を得ることができる。しかも、能動的にリクエストパケットを送出する必要がないため、省電力なクライアントにも適している。

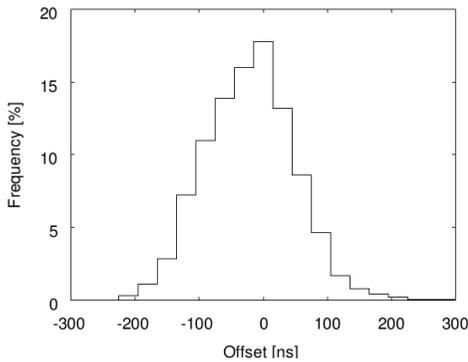
4.2 時刻精度評価

同期精度を評価するために、図 5 の構成で、クライアントは時刻サーバに同期するとともに、毎秒パルスが発生させ、Cs 原子時計の毎秒パルスとのオフセットをインターバルカウンタにより計測した。精度の評価には、表 2 に示すように、提案方式の他に、NTP と IEEE 1588 を準備した。NTP サーバには、ハードウェア Sntp サーバ [8] を使い、IEEE 1588 サーバには OnTime 社 T200^(注4) を使用した。`ntpd` のバージョンは 4.2.0、PTPd [9] のバージョンは 1b4 を用いた。また、クライアント用 PC の諸元を表 3 に示す。`ntpd` では、ポーリング間隔を最小の 16 秒とし、T200 の同期パケットの送出間隔は 1 秒としている。

(注4): <http://www.ontimenet.com/>



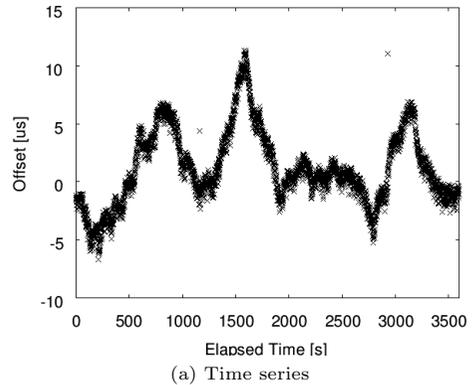
(a) Time series



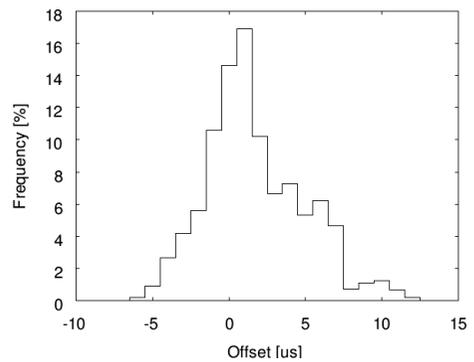
(b) Histogram

図 6 NSTd 同期精度

Fig. 6 Synchronization error of NSTd.



(a) Time series



(b) Histogram

図 7 NTP 同期精度

Fig. 7 Synchronization error of NTP.

クライアントでの毎秒パルスの発生は時刻取得関数、つまり、提案方式では `nst_gettai()`、NTP と IEEE 1588 は `gettimeofday()`、をループ内で繰り返して呼出し、秒の小数部が所定の値を超えたら、速やかにループから抜け出し、デジタル出力ポート（インターフェース社 PCI-2703 デジタル IO ボード^(注5)）にパルスを発生させる。したがって、毎秒パルスには、クライアントの時刻オフセットと時刻取得関数の呼出しに必要な時間を解像度とする誤差を含むことになる。

計測結果を図 6（提案方式）、図 7（NTP）、図 8（IEEE 1588）に示す。ただし、図 6(a) では、時空情報パケットの時刻精度である GPS 受信モジュールの 1 pps/s 信号の時間変動（図 2）も併せて表示している。なお、計測データは、同期開始後 6 時間以上経過し、十分収束した状態で取得した。各図の (a) はオフセットの時間変動であり、(b) はオフセットのヒストグラムである。図 7(a) 及び図 8(a) ではオフセットのドリフトが観測された。また、ドリフト以外の帯状に広がる細かなジッタは `gettimeofday()` による解像度の限

界を示していると考えられる。時刻取得の所要時間は、`gettimeofday()` が、CPU クロックで 2100 clock 必要であるのに対し、`nst_gettai()` では 140 clock しか要しない。一方、提案方式（図 6）ではドリフトもジッタも小さく、標準偏差は 76 ns である。更に、図 6(a) より、時刻源である GPS モジュールで生じる 1 pps/s 信号のドリフトを含む時間変動の影響はわずかであり、サブマイクロ秒の精度が実現されていることが確認された。

次に同期に要する初期時間を計測した結果を図 9 に示す。収束するまでに、`ntpd` は約 1 時間、`PTPd` は約 1000 秒必要とするのに対し、提案方式は同期開始後 3 秒でマイクロ秒以下に収束している。

さて、本システムは他の一般通信（クロストラヒック）も存在するイーサネットでの使用を想定している。ネットワークスイッチ（スイッチングハブを含む）での、キューイング遅延については、既に検討している

(注5): <http://www.interface.co.jp/>

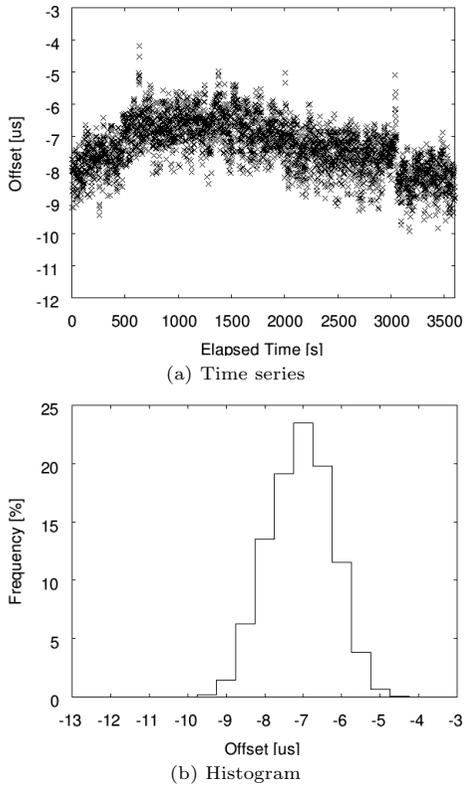


図 8 IEEE 1588 同期精度

Fig. 8 Synchronization error of IEEE 1588.

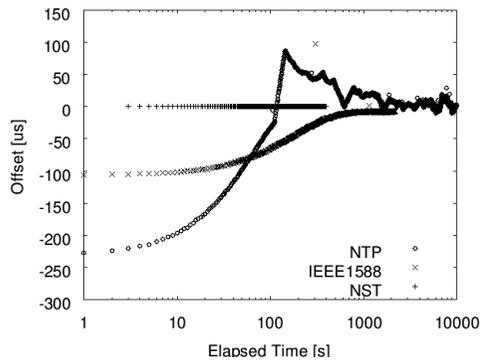


図 9 収束速度

Fig. 9 Speed of convergence.

が [5], PAI 方式は, 毎正秒に送出されるパケットの到着間隔の平均がキューイングの影響を受けにくいことが示されている. また, イーサネットではコリジョンも発生するが, コリジョンによる再送遅延は, 上限のある非負のランダムな値 (“backoff” [10]) であるため, 数学的にはキューイング遅延と同様に扱うこと

が可能である. 更に, 毎正秒に送出される時空間情報パケット中のタイムスタンプは 1 秒ずつカウントアップすることが分かっているため, コリジョンによるパケットロスも検出が容易であり, 一部のパケットが到着しなくとも, PAI 方式で用いている到着間隔のメジアンは影響を受けにくい. したがって, PAI 方式はコリジョンにも強い方式と考えられる.

4.3 粗粒度空間情報

NST サーバから提供される空間情報は, 接続している GPS アンテナの位置であり, 建物を識別する精度と考えることができる. さて, このような数十メートル以上の精度は「粗粒度 (Coarse-grained)」と呼ばれ, 位置情報サービスで活発に利用されている. 粗粒度空間情報をネットワークで毎正秒にブロードキャストすることにより, 以下のような利点がある.

- 高精度
- 省電力
- 広域ネットワーク対応

まず, 精度に関して, 市街地で, 個々の携帯端末が直接 GPS を利用した場合, GPS 衛星からの電波が建物によって反射され (マルチパス), 数百メートルの誤差が生じる場合があるが, 屋上などの上空の見通しの良い場所に設置されたアンテナを利用し, ネットワークで配信すれば, 位置精度が向上する.

次に, GPS 信号から位置を検出するには, 受信開始後, 数分程度の時間を有するため, センサードなどの, 短時間だけ起動する省電力型機器では, 電力消費量や, 空間情報を得るまでに時間がかかるなどの問題が生じていた. また, ネットワーク上で粗粒度空間情報を得る手段として, これまで, DHCP の位置情報サーバオプションを用いる方式が提案されているが [6], 送受信が必要であり, 送信は受信よりも多くの消費電力を必要とするため, 消費電力に厳しいシステムでは, 送信を避ける必要がある. 空間情報が毎正秒にブロードキャストされていれば, クライアントは 1 秒間のみの受信で位置情報を得ることが可能である. このように配信頻度の高い時空間情報配信は, 特に, ほとんどの時間スリープ状態にあり, 短時間だけ電源がオンとなる省電力型の機器に適している.

一方, 近年利用が拡大している広域イーサネットや VPN によって, ネットワークが地域的に広がってきており, このような広域にわたるネットワークで, 各地域の位置情報がブロードキャストされた場合, 最寄りの位置情報を特定することは困難である. 例えば,

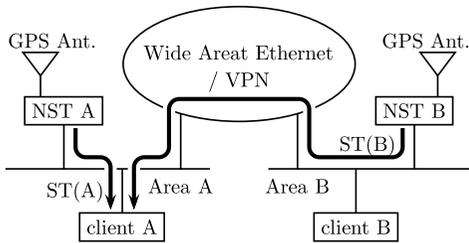


図 10 広域ネットワークでの時空情報ブロードキャスト
Fig. 10 Space-time information broadcasting in WAN.

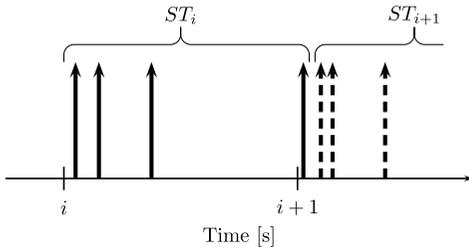


図 11 先頭パケットによる最寄りサーバの選択
Fig. 11 Nearest server selection with the lead.

図 10 に示すように、本社（地域 A）と支社（地域 B）を同一ネットワークセグメント（サブネット）で結び、両地域に設置された NST サーバから時空情報パケット（ST(A) 及び ST(B)）がブロードキャストされた場合、クライアントは両地域の時空情報を受信する。単独の時空情報パケットだけでは、最寄りの時空情報を特定することはできない。

さて、時空情報パケットは、毎正秒に全 NST サーバから同時に送出され、それぞれ、ネットワーク遅延を受けて、クライアントに到着する。ネットワーク遅延はケーブルの長さに比例する伝搬遅延成分を含むため、近いほどネットワーク遅延が小さい場合が多い。したがって、最寄りのサーバから送出されたパケットが最も早く到着する可能性が最も高い。そこで、図 11 に示すように、複数の時空情報パケットが到着した場合、最も早く到着したパケットの空間情報を、最寄りの NST サーバの位置とする。なお、図 11 で、上向き矢印は時空情報パケットの到着時刻を表し、実線 (ST_i) は時刻 i に送出されたパケット群、破線 (ST_{i+1}) は時刻 $i+1$ に送出されたパケット群を表す。また、遠方から遅れて到達したパケットが、次の正秒のパケット群を先行する形になることも考えられるが、パケット内の時刻情報を見れば、前の正秒に送出されたことが分かるため、誤ることはない。

ところで、ネットワーク遅延には、ケーブルによる伝搬遅延だけではなく、ネットワークスイッチなどの機器通過遅延も含まれる。多くのネットワーク接続機器が採用しているストアアンドフォワード方式では通過遅延はパケットサイズに依存するが、時空情報パケットのような短パケットでも 1.5 マイクロ秒ほどの通過遅延を生じ、1.5 マイクロ秒の遅延時間は、およそ 300 メートル長のケーブル（メタル及び光ファイバ）の伝搬遅延に相当する [5]。したがって、建物外からの時空情報パケットとの先着順序逆転を防ぐために、建物の内と外のネットワークを接続する装置と NST サーバの間のネットワークスイッチの段数は少ないことが望ましい。

5. む す び

ネットワーク時空情報サーバと動的ユーザランド時刻クライアントを開発した。NST サーバは GPS を利用した簡易な構成でサブマイクロ秒精度の時刻情報と粗粒度精度の空間情報を提供する。また、動的ユーザランド時刻取得クライアントは、変動する周波数源のパラメータを動的に更新するとともに、システムコールよりも 1 けた以上高速かつ高解像度である。提案方式は、時空情報パケットが毎秒ネットワークに供給され、同期所要時間も数秒以内であるため、省電力のために必要なときだけ稼動するユビキタスノードには最適と考えられる。しかも、空間情報と時間情報が統合することにより、広域ネットワークでも最寄り位置情報を特定することが可能である。

今後は、屋内の測位システムと組み合わせることにより、高精度な空間情報を提供するシステムを開発する予定である。例えば、コストが小さく精度の高い方式として知られている超音波測位システム [11] では、センサ間の同期が必要であるが、本システムを用いれば高精度な同期が可能であるため、Active Bat [12] で使われているような電波によるトリガが不要となり、システム構成を簡略化できると考えられる。

文 献

- [1] M. Hazas, J. Scott, and J. Krumm, "Location-aware computing comes of age," IEEE Computer Magazine, vol.37, no.2, pp.95-97, 2004.
- [2] J. Weatherall and A. Jones, "Ubiquitous networks and their applications," IEEE Wireless Communications, vol.9, no.1, pp.18-29, 2002.
- [3] "IEEE1588: IEEE standard for a precision clock synchronization protocol for networked measurement

and control systems,” 2002.

- [4] 町澤朗彦, 北口善明, “割込みハンドラと高精度 PC によるソフトウェアタイムスタンプの精度改善,” 信学論 (B), vol.J87-B, no.10, pp.1678-1685, Oct. 2004.
- [5] 町澤朗彦, 岩間 司, 鳥山裕史, “毎正秒パケット到着間隔 (PAI) に基づいた時刻同期方式,” 信学論 (B), vol.J89-B, no.10, pp.1855-1866, Oct. 2006.
- [6] J. Polk, J. Schnizlein, and M. Linsner, “Dynamic host configuration protocol option for coordinate-based location configuration information,” RFC 3825, IETF, 2004.
- [7] 総務省情報通信審議会 情報通信技術分科会, “IP ネットワークにおける緊急通報等重要通信の確保方策,” 2005.
- [8] 鳥山裕史, 町澤朗彦, 岩間 司, “ハードウェア SNTP サーバの開発,” 信学論 (B), vol.J89-B, no.10, pp.1867-1873, Oct. 2006.
- [9] K. Correll, N. Barendt, and M. Branicky, “Design considerations for software-only implementations of the IEEE 1588 precision time protocol,” Proc. 2005 IEEE 1588 Conference, Zurich, 2005.
- [10] “IEEE Std 802.3-2005 part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications,” 2005.
- [11] J. Hightower and G. Borriello, “Location systems for ubiquitous computing,” IEEE Computer Magazine, vol.34, no.8, pp.57-66, 2001.
- [12] A. Harter, A. Hopper, P. Steggle, A. Ward, and P. Webster, “The anatomy of a context-aware application,” Proc. Mobicom '99, pp.59-68, 1999.

(平成 18 年 9 月 26 日受付, 12 月 22 日再受付)

鳥山 裕史 (正員)



昭 56 名工大・情報卒・昭 58 名大大学院情報工学専攻博士課程前期課程了。同年郵政省電波研究所 (現情報通信研究機構) 入所。平 2~5 ATR 通信システム研究所。平 5~6 ドイツテレコム研究所客員研究員。画像符号化, 情報通信などの研究に従事。



町澤 朗彦 (正員)

昭 59 上智大・理工・電気電子卒。同年郵政省電波研究所 (現情報通信研究機構) 入所。平 6 科学技術庁に出向し, IMnet 立上げに参与。平 8~11 Univ. Canterbury 客員研究員。平 15 JGN2 立上げに参与。

画像の高効率符号化, 視覚情報処理, 計算機ネットワーク, 時刻同期の研究に従事。



岩間 司 (正員)

昭 58 山梨大・工・電子卒。昭 60 東工大大学院修士課程了。同年郵政省電波研究所 (現情報通信研究機構) 入所。以来, 電波伝搬特性解析, 移動通信のセル構成, 標準時, 時刻認証基盤技術の研究に従事。現在, 電磁波計測部門タイムアプリケーショングループ主任研究員。平 2 本会篠原記念学術奨励賞受賞。IEEE 会員