

B_013

SNTP における取得時刻の精度向上に関する一考察

A study of timestamp accuracy improvement on SNTP.

牛木一成 橋本圭輔 伊藤智子 小河原成哲

Kazunari Ushiki Keisuke Hashimoto Tomoko Ito Masanori Ogawara

エヌ・ティ・ティ・コムウェア株式会社 研究開発部

NTT COMWARE CORPORATION. Research and Development Department.

1. はじめに

近年、放送と通信の連携が進んでいる。例えば IP 放送が RTP/UDP を用いたマルチキャストにより実験されており、現在はロス・ジッタ・遅延といったネットワーク品質についての評価段階にある。

我々はネットワークにおける RTP パケットを対象にした品質測定を、主に導入価格の観点から市販 PC 上でソフトウェアを動かすアプローチを選択し、安価なソリューションを提供すべく検討している。測定項目は数時間から数日といった長時間にわたるジッタやロス、そして片方向遅延であり、特に遅延に関しては個々のパケットが通過した時刻を精度高く把握することが肝要である。

本論では、検証作業の中で得た、SNTP プロトコルでの NTP サーバのオーバーヘッドの在処と、その在処により導出される時刻補正式について述べる。

2. PC クロックと時刻取得方式

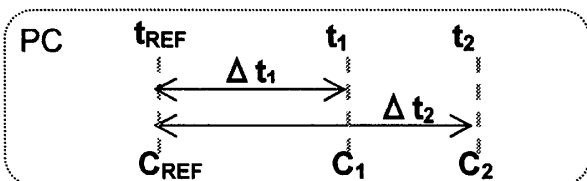
PC のクロックは基準周波数のずれが存在し、PC のタイマーは誤差が線形に増加する。また基準周波数は緩やかに変動し、PC のタイマーには緩やかな時間の揺れが生じる。すなわち PC のタイマーは短時間的には線形補正することができるが、長時間的には揺れがあり不正確という特性を持つ[1]。このため測定直前に PC の時刻校正をし、測定中は PC が自走する方法は長時間測定においては精度の観点から望ましくない。よって時刻計算の方法は、定期的に時刻問合せして基準時刻を確定し、パケット通過までのクロック値差分から得た時間を基準時刻に加算してパケット通過時刻を得ることにした。時刻計算の考え方を図 1 に示す。

まず基準時刻 (t_{REF}) と、基準となる PC クロック値 (C_{REF}) を得ておく。次にパケット通過したタイミングで PC クロック値 (C_1, C_2) を得る。基準となる PC クロック値 (C_{REF}) とパケット通過したタイミングで得た PC クロック値 (C_1, C_2) の差分から得た時間 ($\Delta t_1, \Delta t_2$) を基準時刻 (t_{REF}) に加えパケット通過時刻 (t_1, t_2) とする。

$$\Delta t_n = (C_n - C_{REF}) \div \text{PC クロック周波数}$$

$$t_n = t_{REF} + \Delta t_n$$

PC 外部の時計から時刻を得るには NTP サーバ[2]を用い、問合せプロトコルはレスポンスの速さから SNTP とした。

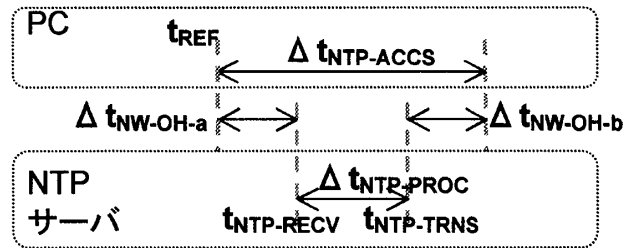


<図1 クロック差分による時刻計算>

3. オーバーヘッドの按分ロジック

NTP サーバに対し SNTP プロトコルで問合せたところ、問合せにかかる時間 ($\Delta t_{NTP-ACCS}$) は NTP サーバでの処理時間 ($\Delta t_{NTP-PROC}$) の倍以上の時間であった。具体的には NTP サーバでは $300 \mu\text{sec}$ 程度の処理時間が AP の問合せ開始から時刻取得完了では $800 \mu\text{sec}$ 以上となり、 $500 \mu\text{sec}$ 以上のオーバーヘッドが存在していた。オーバーヘッドの配分は未知である。ここでは簡単のため按分した。

図 2 に時刻・時間の関係を示す。



<図2 按分ロジック模式図>

関係式は以下となる。

$$\Delta t_{NTP-ACCS} - \Delta t_{NTP-PROC} = \Delta t_{NW-OH-a} + \Delta t_{NW-OH-b}$$

$$\Delta t_{NW-OH-a} = \Delta t_{NW-OH-b} = (\Delta t_{NTP-ACCS} - \Delta t_{NTP-PROC}) \div 2$$

問合せ発出時刻 (t_{REF}) は式 1 で得る。

$$t_{REF} = t_{NTP-RECV} - (\Delta t_{NTP-ACCS} - \Delta t_{NTP-PROC}) \div 2 \quad \dots \text{式 1}$$

このロジックの課題は、按分という不明瞭な分配をしたことにより時刻精度の信憑性が低いことである。より正確なオーバーヘッドの配分を決定し時刻精度を高めるべく検証を進めることにした。

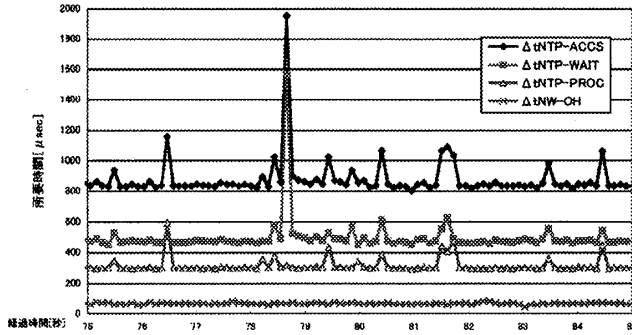
4. 実験

これまで未知であった、問合せパケットと返却パケットのやり取りのオーバーヘッドと、NTP サーバ内部でのオーバーヘッドを算出し、時刻補正式を得る。

SNTP プロトコルで問合せしているパケットの状態 (問合せパケットと返却パケットの間隔) を、ethereal で測定した。測定時間は約 160 秒間、NTP サーバへは約 0.1 秒間隔で問い合わせた。計算した時間内訳をグラフ 1 に示す。

グラフ 1 によれば問合せパケットと返却パケットのやり取りのオーバーヘッド (Δt_{NW-OH}) は $60 \pm 15 \mu\text{sec}$ で比較的安定であることから固定値とみなしてよいと思われる。

逆に NTP サーバ内で $\Delta t_{NTP-PROC}$ 以外の時間が非常に大きな値であり、オーバーヘッドのほぼ全てであることがわかる。また変動も大きく不定値であった。



<グラフ1 NTPアクセスの処理内訳>

5. 考察

(1) 補正式

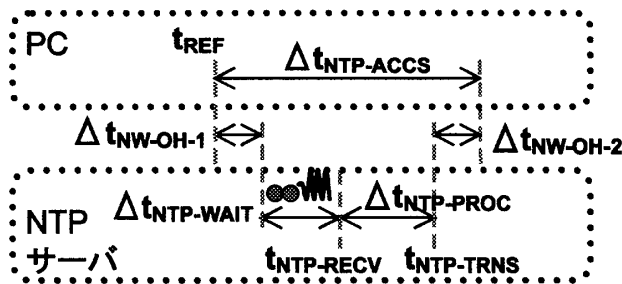
前項で示した通り、NTP サーバ内で $\Delta t_{NTP-PROC}$ 以外の時間がオーバーヘッドを支配している。なおオーバーヘッドの位置は NTP サーバが問合せパケットを受け取ってから時刻確定までの待ち行列だと捉え、NTP サーバの待ち時間 ($\Delta t_{NTP-WAIT}$) とした。図3に時刻・時間の関係を示す。

問合せ発出時刻 (t_{REF}) は補正式 (式2) で得られる。ただし Δt_{NW-OH} 内訳は測定できず不可分である。ローカル LAN 上での測定であることから通信時間が双方向で対称と仮定し、按分 ($60 \mu sec \div 2$) した。

$$t_{REF} = t_{NTP-RECV} - (\Delta t_{NTP-ACCS} - \Delta t_{NTP-PROC} - 30 \mu sec)$$

または

$$t_{REF} = t_{NTP-TRNS} - (\Delta t_{NTP-ACCS} - 30 \mu sec) \dots (式2)$$

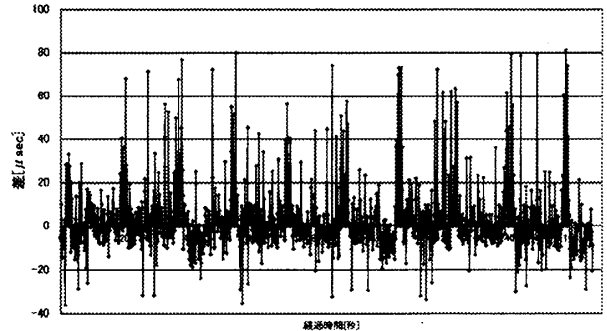


<図3 NTP待ち時間を加味したロジック模式図>

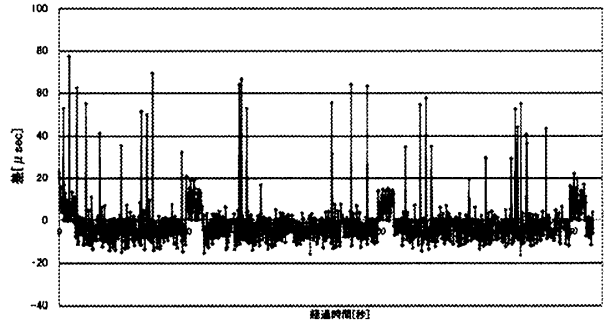
(2) 精度比較

リファレンス時刻は PC のクロックから計算して得た時刻とした。先の按分 (式1) で得た時刻と、補正式 (式2) で得た時刻に対し、リファレンスとの差を計算して比較する。ただし $\Delta t_{NTP-ACCS}$ が長い問合せはオーバーヘッドが大きくなり誤差を多く含むと考えられる。そのため $\Delta t_{NTP-ACCS}$ が 1msec 未満のデータを対象とした。按分にて計算した結果をグラフ2に、補正式にて計算した結果をグラフ3に示す。

グラフ2によれば按分でも $\Delta t_{NTP-ACCS}$ が 1msec 未満のデータとすれば、 $120 \mu sec$ の範囲に収まっていることが確認できる。一方グラフ3によれば $\pm 20 \mu sec$ に多くが収まっており、スパイク状に $80 \mu sec$ のデータがまばらにあるに留まる。グラフ2,3を比較するとグラフ2はややばらつきが多く、グラフ3はばらつきが少なく安定しているといえる。このように補正式を使用した際には安定したデータとなることを確認した。

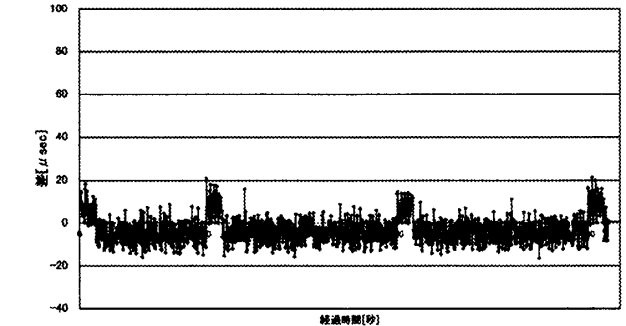


<グラフ2 1msec未満データ、按分による>



<グラフ3 1msec未満データ、補正式による>

なおグラフ3のスパイク状ノイズは簡単なフィルタ処理で除去できる。補正式のデータからメジアンフィルタとの差が $15 \mu sec$ 未満の値を抽出した結果をグラフ4に示す。グラフ4によれば、スパイク状ノイズは除去され、値は +23 から -17 の範囲に収まっている。



<グラフ4 スパイク状ノイズ除去>

6. まとめ

SNTP プロトコルを用いた NTP サーバへの時刻問合せにおけるオーバーヘッドの在処を確認し、時刻補正式を導出した。また簡単なフィルタ処理を行うことでより誤差の少ないデータを抽出が可能であることを示した。

今後は本検証結果を踏まえた片方向遅延測定装置の実装を進める。

参考

[1] 電子情報通信学会 NW システム研究会 No.160 「PCにおける時刻精度の精密計測とその評価」北口ら
 [2] EndRun Technologies 社 Tempus LX CDMA-J (<http://www.shoshin.co.jp/computer/endrun/cdma.html>)