分散システム特論レポート

RFC 1305 Network Time Protocol(NTP)

情報工学専攻修士 1 年 牛島研究室 川野朋彦

1999 年 2 月 8 日

1 RFC1305

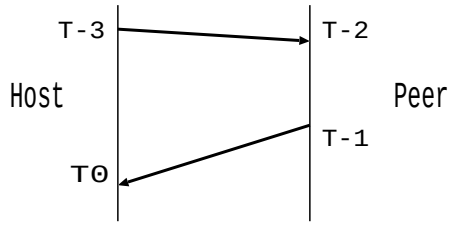
RFC1305 では、Network Time Protocol(NTP) の形式的な構造を特定し、NTP の実装に対する有益な情報をまとめる。

1.1 用語ほか

1.1.1 用語

- stability: 時計の stability は、どれだけ絶え間ない周波数を維持できるか。
- accuracy: その周波数と時刻が、どれだけ国際標準に匹敵するか。
- precision: stability と accuracy の値が、特定のシステムの中でどれだけ正確に維持 されるか。
- skew: 2 つの時計間の周波数の差。
- peer: 通信相手
- synchronization distance: dispersion と片側の絶対遅延の和。以下では「同期距離」。

1.1.2 変数



- peer との時計の差: offset$= \frac{(T_{-2}-T_{-3})+(T_{-1}-T_0)}{2}$
- peer との往復遅延: roundtrip delay$= (T_0 - T_{-3}) - (T_{-1} - T_{-2})$
- peer の時計に対するローカル時計の最大誤差: dispersion$= \rho + \phi\tau + \epsilon\sigma$
- 測定誤差: $\rho = 1 \ll sys.precesion$

- skew-error 蓄積: $\phi\tau$
- 最大 skew rate: $\phi = \frac{NTP.MAXSKEW}{NTP.MAXAGE}$
- peer の最終更新からのインターバル: $\tau = sys.clock - peer.update$

1.1.3 定数

- NTP.MAXAGE(86400 秒):参照時計が最後に更新されてから正当だろうと考えられる時間
- NTP.MAXSKEW(1 秒):NTP.MAXAGE によって決定するインターバルの下で、ローカル時計の skew による最大 offset 誤差

2 NTP

NTP は、多様なインターネット運用の中で、時刻を同期し、時刻分散を同格にするメカニズムを提供する。

NTP の目的は、

- 国際標準時計を持つサーバから、時刻合わせ情報を、他のタイムサーバにインターネットを介して運ぶこと
- 時計をクロスチェックすること
- 設備もしくは伝播の誤りによる誤差をやわらげること

である。

2.1 特徴

データの健全性は IP や UDP のチェックサムによって提供される。フロー制御や再転送機能は提供されず、必要でない。

Internet Assigned Numbers Authority によって NTP に割当てられたポート 123 を使う。

メッセージには単一のフォーマットが使われるので、プロトコルは簡単に実装される。

時計同期にはその性質から、正確な (ミリ秒のオーダで) 時刻合わせを維持するためには長い期間と複数の比較が要求される。このように、accuracy は、時刻合わせが為された時刻に直接依存する。

2.2 構造

2.2.1 クライアント /サーバ

NTP デーモンの動作のモードには、以下の 5 つのモードがある。

- Symmetric Active
- Symmetric Passive
- Client

- Server
- Broadcast

クライアントは、peer 毎にタイマを持っている。そのタイマを用いてサーバに定期的に NTP メッセージを送って、サーバから受け取った応答を処理する。

サーバは、クライアントから受け取ったパケットのアドレスとポートを置き換えて、メッセージの決まったフィールドを上書きして、素早くメッセージを返す。

2.2.2 プライマリ/セカンダリ

NTP モデルにおいて、国際標準に同期された数多くのプライマリ参照源は、バックボーンゲートウェイのような広くアクセス可能な資源に接続され、プライマリタイムサーバとして運用される。

また、ローカルネットワーク上のホストは、セカンダリタイムサーバとして動作し、1 つかそれ以上のプライマリサーバと NTP を実行することもできる。

2.2.3 階層構造

クライアントはまた、下位の計算機に対してサーバの役割を果たすことができ、階層構造を為す。

通常の環境のサブネットでは、ルートのプライマリサーバ、終端に向かって精度が減少するセカンダリサーバの階層的な設定が為される。

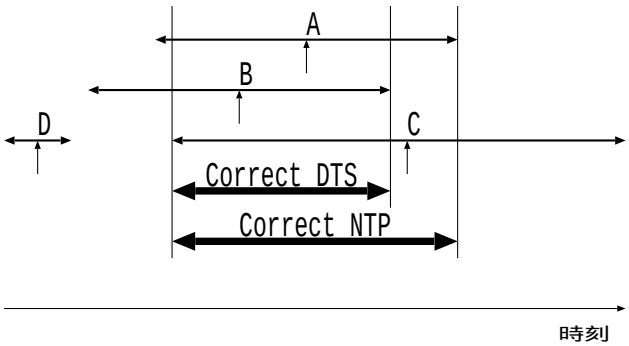
それぞれのサーバの精度は、stratum と呼ばれる数によって定義される。

2.3 時計更新

1. 1 つまたは複数のサーバから時刻情報を受け取る
   クライアントはタイマを用いて peer に定期的にメッセージを送り、応答を受け取り、フィルタリングする。
   フィルタリングでは、参照時計のデータ [offset,delay,dispersion] をいくつか (8) 取っておき、今回のデータも含めて (同期距離的に) 劣ったデータを捨てる。

```
localhost% ntpq -c 'pstatus 25220'
filtdelay=    0.53    0.53    0.53    0.53    0.53    0.53    0.53    0.53,
filtoffset=  -0.05   -0.08   -0.09   -0.11   -0.02    0.00    0.05    0.06,
filterror=    0.99    1.97    2.94    3.92    4.90    5.87    6.85    7.83
```

2. 時計を選択する
   不正確と推定される時計を除く。(intersection algorithm)



(この場合、A,B,C,D の時計のうち、D を不正確と推定。各時計の幅は遅延。)
同期距離の遠い時計を除く。(clustering algorithm)

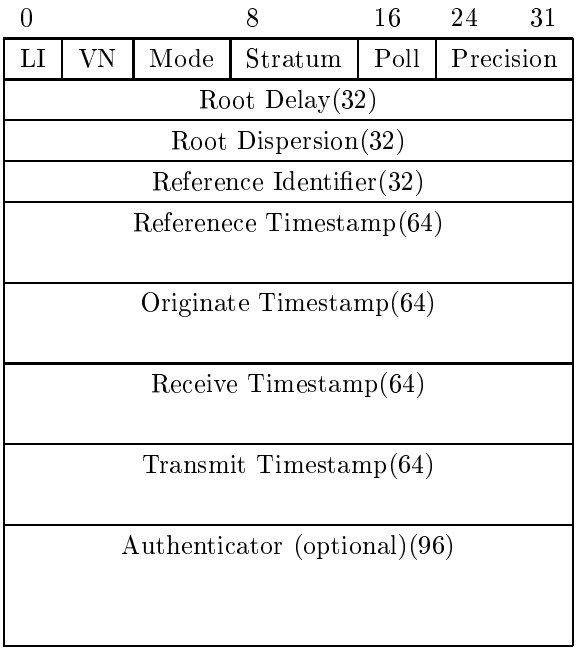
3. 時刻情報を結合する
   時計候補リストについて、offset と同期距離から時刻を結合する。
4. 時計を更新する
   それを元に更新する。

2.4 メッセージ

メッセージに含まれる項目は以下の通りである。

- Leap Indicator(LI): うるう秒の有無 (2bit code)
- Version Number(VN): バージョンナンバー (3bit integer)
- Mode: 動作モード (3bit integer)
- Stratum: 階級 (8bit integer)
- Poll Interval: メッセージ間の最大インターバル (8bit signed integer)
- Precision: 時計の精度 (8bit signed integer)
- Root Delay: プライマリ参照源からの総往復遅延 (32bit signed fixed-point number)
- Root Dispersion: プライマリ参照源に対する最大誤差 (32bit signed fixed-point number)
- Reference Clock Identifier: 参照時計の ID(32bit code)。stratum が 0 または 1 の場合は RFC で提案されている ASCII 文字列、それ以外の場合は 4 オクテットのインターネットアドレス。
- Reference Timestamp: ローカル時計を最後に修正した時刻のタイムスタンプ。
- Originate Timestamp: 要求がクライアントを出発した時刻のタイムスタンプ。
- Receive Timestamp: 要求が到着した時刻のタイムスタンプ。
- Transmit Timestamp: 応答がサービスホストを出発した時刻のタイムスタンプ。
- Authenticator: 認証情報。

NTP タイムスタンプは 64bit の符号なし固定小数点数として表され、1900 年 1 月 1 日からの秒数である。最初の 32bit は整数部で、最後の 32bit は小数部である。この表現の精度は約 200 ピコ秒となる。



3 実装例 (xntpd)

デーモンの動作のモードとして、前述の 5 つの他に、

- broadcast client
- multicast client

がある。
multicast client モードでは、デフォルトで IP アドレス 224.0.1.1 を使用する。