NTP Implementation and Assumptions about the Network

Stanislav Shalunov

June 19, 2000

DRAFT

Abstract

Under certain assumptions about statistical properties of a network the way NTP servers estimate time offsets can be improved. (These assumptions may better reflect true network properties than those made by NTP.)

1 Introduction

We consider NTP servers of stratum higher than one. Such servers use exchange of network packets with NTP servers of lower stratum to synchronize their time.

Consider computers A and B, connected via a network. Computer A wishes to synchronize its time to that of computer B. From time to time, A sends B packets that contain timestamps according to system clock on A. When B receives such packets, it sends back this packet with time according to its own clock. Having received this returned packet, A makes an estimation of time offset between its clock and clock of B.

Further, A takes action to correct its clock (speeding it up or slowing it down). We do not concern ourselves with intricate details of doing this. We're interested only in how A estimates the time difference.

Let's assume that A sends n packets at times (according to its own clock) t_1^1, t_1^2, … t_1^n, computer B receives these packets at t_2^1, t_2^2, … t_2^n (according to its clock), and computer A receives responses at t_3^1, t_3^2, … t_3^n (according to its own clock again).

Let's further assume that clocks drift only negligible amount of time during the whole measurement. This can be done because the process NTP synchronizes clock is iterative. We assume that time difference between A and B is c.

We also assume that the time it takes for a packet to get from any given point to another point is a random quantity with certain distribution, and measurements taken with significant delays (hundreds of milliseconds) are independent.

We denote the random quantity describing time it takes a packet to get from A to B by ξ_1, and from B to A by ξ_2.

2 Acknowledgments

The idea of using a similar approach belongs to Guy Almes. Anatoly Karp has spent a few hours discussing this with me, and deserves credit for most math here. So, I've simply recorded a few things (and, of course, I'm responsible for all errors).

3 Symmetric Network Paths Model

A naïve way to estimate c would be to compute

$$c^i = t_2^i - \frac{t_1^i + t_3^i}{2}$$

and further proceed to estimate

$$\hat{c} = \frac{\sum_{i=1}^n c^i}{n}. \qquad (1)$$

We have

$$c^i = c - \frac{\xi_2^i - \xi_1^i}{2}.$$

If the quantities ξ_1 and ξ_2 are distributed identically (i.e., network paths are absolutely symmetric) $\xi_2 - \xi_1$ has distribution that's symmetrical w.r.t. 0, therefore the estimate $\hat{c}$ for c is consistent.

It's my understanding that this is essentially the model that current implementaions of NTP use.

4 Assumptions about the Network

Assumption about topological symmetry of network path is reasonable. If packets take different paths from A to B than from B to A, all bets are off because there's no way to eliminate difference of times to get from A to B and from B to A from $\hat{c}$.

However, assumption about the same distribution of delays does not appear to be as reasonable: due to different network loads in different directions this assumption will frequently be false.

We present a model that requires first, but not the second assumtion to get consistent c estimations.

First, we want to describe ξ_i distributions. We might be able to go further with some generic assumptions, but in order to simplify our task we arbitrarily choose a specific distribution that *looks* consistent with measurement data.

Namely, we assume that the distribution function for ξ_i is $f_i(x)$ such that $f_i(x) = 0$ if $x < \theta_i$ and $f_i(x) = \lambda_i(1 - e^{-\lambda_i(x-\theta_i)})$ for $x \geq \theta_i$ (shifted exponential distribution).

Here, θ describes properties of the network path itself, while λ describes the degree it is loaded. We further assume that $\theta_1 = \theta_2 = \theta$ (i.e., the paths are topologically the same).

5 Time Difference Estimation

Put

$$\Delta t_1^i = t_2^i - t_1^i$$

and

$$\Delta t_2^i = t_3^i - t_2^i.$$

The method of maximum likehood produces the following estimate:

$$\hat{c} = \frac{1}{2}(\min_{i=1}^{n}(\Delta t_1^i) - \min_{i=1}^{n}(\Delta t_2^i)). \tag{2}$$

This estimate is asymptotically consistent.

We're also interested in knowing how good this estimate is. For this purpose, let's find an asumptotically consistent estimate of dispersion of $\hat{c} - c$.

Put

$$\frac{1}{\hat{\lambda}_i} = \frac{\sum_{j=1}^{n} \Delta t_j^i}{n} - \min_{j=1}^{n} \Delta t_j^i.$$

This is an asymptotically consistent estimation of the λ parameter of ξ_i, which is the same for Δt^i (which is equal to $\xi_i \pm c$).

We have

$$\hat{\sigma}^2 = \frac{1}{2} \sum_{i=1}^{2} \frac{1}{\hat{\lambda}_i^2}. \tag{3}$$

6 Critique of Assumptions

The choice of shifted exponential distribution for one-way packet delays is arbitrary. This isn't fatal, since formula 2 is asymptotically consistent under a much weaker assumption: Namely, if f_i is the distribution function of ξ_i, then if there exist positive θ, δ, and γ such that for all $x < \theta$ we have $f_i(x) = 0$, and for all $\theta < x < \theta + \delta$ we have $f_i(x) > \gamma$, then estimate 2 is asymptotically consistent.

This assumes, however, common θ for both directions. If θ is different, building an (asymptotically) consistent estimate of c appears to be very problematic. However, this should not be a problem, since different values of θ mean different network paths and will look like a systematic bias in $\hat{c}$. Such situations are already handled well in case several servers exist by voting-like procedures.

7 Conclusion

NTP servers that use the formula 1 for time offset estimation now get consistent bias if links are congested differently when going in different directions.

Using proposed formula 2 one can achieve better (asymptotically consistent) results under certain assumptions about the distribution of delays even if links are loaded differently.

If estimation of dispersion is required for statistical or other purposes, formula 3 can provide better results than naïve calculation that assumes bell-shaped distribution.