Network Classless Time Protocol Based on Clock Offset Optimization

Omer Gurewitz, Israel Cidon and Moshe Sidi
Electrical Engineering Department
Technion, Haifa 32000
Israel

Abstract

Time synchronization is critical in distributed environments. A variety of network protocols, middleware and business applications rely on proper time synchronization across the computational infrastructure and depend on the clock accuracy. The "Network Time Protocol" (NTP) is the current widely accepted standard for synchronizing clocks over the internet. NTP uses a hierarchical scheme in order to synchronize the clocks in the network. In this paper we present a novel non-hierarchical peer-to-peer approach for time synchronization termed *CTP - Classless Time Protocol*. This approach exploits convex optimization theory in order to evaluate the impact of each clock offset on the overall objective function. We define the clock offset problem as an optimization problem and derive its optimal solution. Based on the solution we develop a distributed protocol that can be implemented over a communication network, prove its convergence to the optimal clock offsets and show its properties. For compatibility, the CTP may use the exact format and number of messages used by NTP. We also present methodology and numerical results for evaluating and comparing the accuracy of time synchronization schemes. We show that the CTP substantially outperforms hierarchical schemes such as NTP in the sense of clock accuracy with respect to a universal clock, without increasing complexity.

I. Introduction

Common distributed computation systems consist of a collection of autonomous entities linked via an underlying network and do not share a common memory or a common clock. They are equipped with distributed system software that enables the collection to operate as an integrated facility, and allow the sharing of information and resources over a wide geographic spread. Clock synchronization is a critical piece of the infrastructure for any such distributed system.

The notion "clock synchronization" relates to at least two different aspects of coordinating distant clocks. The first aspect is "frequency synchronization" which relates to the task of adjusting the clocks in the network to run with the same frequency. The second is "time synchronization" which relates to the task of setting the clocks in the network so that they all agree upon a particular epoch with respect to a Universal Time-Coordinated (UTC).

The basic difficulty in clock synchronization is that timing information tends to deteriorate over time and distance. Particularly when the frequencies of two clocks are not identical and are not known in advance. Even if the two clocks were initially time synchronized, over time they are drifting apart, hence they need to be time-synchronized from time to time. Moreover, when two remote computers are exchanging timing information, there is cumulative loss of accuracy along the path traversed by the messages exchanged, unless message transmission time is known precisely.

The application of time synchronizing in distributed systems is diverse. Server log files are used in firewall, VPN security-related activity, bandwidth usage and various logging, management, authentication, authorization and accounting functions. Since they are a collection of information from different hosts, it is essential that the time stamps be correct in order to coordinate the time of network events, which helps in understanding and tracking the time sequence of network events. For example, Cisco routers use clock synchronization in order to compare time logs from different networks for tracking security incidents, analyzing faults and troubleshooting [1].

Wireless ad-hoc networks make particularly extensive use of synchronized time. In addition to the basic requirements of traditional distributed systems, ad-hoc networks also use time synchronization for mobility prediction [2] or in sensor networks for velocity estimations [3], source localization, or to suppress redundant messages by recognizing that they describe duplicate detections of the same event by different sensors.