Network Time Protocol

Ahmed Almohsin
amalmohsin21@tntech.edu

ABSTRACT
In this paper, we describe the network time protocol.

Categories and Subject Descriptors
C.2.2 [**Computer-Communication Networks**]: Network protocols

General Terms
Algorithms, Management, Measurement, Documentation, Performance, Design, Reliability, Experimentation, Security, Standardization, Languages, Verification.

Keywords
Network, synchronization, protocol.

1. Introduction
The Network Time Protocol (NTP) is a time synchronization system for computer clocks that operates through the Internet network. It was mainly developed at Delaware University and is evolved from Time Protocol and the ICMP Timestamp Message, but is specifically designed for high accuracy, stability, and reliability, even when used over typical Internet paths involving multiple gateways and unreliable networks. There are three versions of NTP that have been developed over the years. The first was developed in 1988, the second was in 1989, and the third version that we use today was developed in 1992. The current version is compatible with all the previous versions and in 1995 a second type of NTP was defined called the Simplified NTP (SNTP). The complete kind of NTP implements all the features of NTP but the simplified kind implements only a subset of NTP. SNTP types can work only as a client and can have only 1 defined server at any given time.

2. How it works
NTP works by obtaining the reference clocks from the machines in the network. The times are passed using UDP/IP packets because of the fast connection setup and response times. The times are used to compute some additional statistical values that describe the quality of the time it sees. Thus it maintains an estimate of the quality the reference clocks of the network and its own reference clock.

3. Characteristics
There are a few main characteristics that define NTP. It is fully automatic meaning it keeps synchronization continuously. It is suitable to synchronize one computer or a whole network of computers. NTP carries UTC time which is independent of time zones and day-light savings time; this is set by the individual computer or user. The accuracy of NTP's synchronization can reach up to 1 millisecond.

A NTP primary server is a computer connected to a high precision reference clock equipped with the NTP software. Other computers automatically query the primary server to synchronize their clocks. The primary server is also called stratum 1, the computers that connect to it are called stratum 2, and the ones connected to those stratum 3, and it keeps going all the way to 16 stratums. Being higher in the stratums will cause the accuracy to be less and less as you get further from the primary server.

4. In the operating systems
The current Operating Systems, ones based on Linux/Unix and Windows, use NTP to synchronize their clocks. Windows XP uses NTP but computers running Windows server 2003 use SNTP. Windows synchronizes the clock when it starts up and at intervals while running in order to insure that software activation doesn't fail due to clock times too far apart (Microsoft TechNet 2004). Unix started out using it's own synchronization method called "timed" but it was developed only to keep synchronization over a local Ethernet and it's clock discipline algorithms failed in comparison to that which was later implemented in NTP. NTP version 0 was implemented in 1985 in Unix by Louis Mamakos and Michael Petry at the University of Maryland. Fragments of the Unix code still survive in the software running today (Mills 2003).

5. Brief history
It was discovered that NTP could be used for more than just synchronizing time. Around 1985 Project Athena at MIT was developing the Kerberos security model, which provides cryptographic authentication of users and services. The fundamental part of the Kerberos design was the use of a ticket to access computer and network services. Tickets have a designated lifetime and must be securely revoked when their lifespan has ended. Thus, all Kerberos facilities had to have secure time synchronization services so NTP was chosen. NTP alone was inadequate to deflect sophisticated attacks so a method of authenticating NTP packets using symmetric key cryptography with keyed message digests and private keys was implemented (Mills 2004).

6. Conclusion
NTP started out as a way to keep our clocks synchronized across a network. It then grew to an even larger network in the Internet,

and is even being used to authenticate users for computer and network services. NTP today is major part of how, if, and when our networks operate.

7. Python NTP code sample

```
from socket import *
import struct
import sys
import time

TIME1970 = 2208988800L

client = socket( AF_INET, SOCK_DGRAM )
data = '\x1b' + 47 * '\0'
client.sendto( data, ('time.tntech.edu', 123 ))
data, address = client.recvfrom( 1024 )
if data:
    print 'Response received from:', address
    t = struct.unpack( '!12I', data )[10]
    t -= TIME1970
    print '\tTime=%s' % time.ctime(t)
```

8. Ruby NTP code sample

```
require 'optparse'
require 'socket'

SNTP_MSG = "\010" + "\0" * 47
SNTP_PORT = 123
NTP_UNIX_TIME = 2208988800

set_time = false

unless ARGV[0]
  puts opts
  exit
end

data = UDPSocket.open do |sd|
  sd.send(SNTP_MSG, 0, ARGV[0], SNTP_PORT)
  sd.recvfrom(64)[0]
end

data = data.unpack('N12')[10].to_i - NTP_UNIX_TIME
time = Time.at(data)
```

9. References

Microsoft TechNet. (2004). Windows Time Service. http://technet.microsoft.com/en-us/library/bb490605.aspx

Mills, David L. (1994). *Precision Synchronization of Computer Network Clocks*. ACM SIGCOMM Computer Communication Review. Vol 24, Issue 2. pg 28-43. http://doi.acm.org/10.1145/185595.185651

Mills, David L. (2003). *A brief history of NTP time: memoirs of an Internet timekeeper*. ACM SIGCOMM Computer Communication Review. Vol 33, Issue 2. pg 9-21.

http://doi.acm.org/10.1145/956981.956983