Network Time Protocol

Timothy Myers
Tennessee Tech
344 East 12th Street Apt. 4
Cookeville, TN 38501
931-206-2255
temyers240@gmail.com

ABSTRACT
This paper discusses the Network Time Protocol (NTP) as it is a foundation protocol for all networks even including the Internet. Both the current Internet standard of NTP version 3 and the newer NTP version 4 are invaluable for keeping data and communications open within the networks that we use every day of our lives. The precision provided by NTP is especially important when LAN , or other multi-player games require coordination between the player actions of anywhere from two to up to two hundred and fifty six players at a time. In many of these games, units of time measuring in fractions of a second dictate the end results of the game. Unfortunately, the implementation required for such precision as is needed for things like multi-player gaming is quite hardware intensive. For integrated systems and others, especially where such exacting times keeping is not required there is a form of NTP called SNTP or Simple Network Time Protocol which while not so robust in features is much easier on limited resources.

Keywords
NTP, SNTP, time synchronization, multi-player gaming

1. INTRODUCTION
The goal of this document is to present a general understanding of NTP, one of its uses, and one of its variations. Towards this goal I hope to convey as much of the workings of NTP as I understand them. It is my wish that with the knowledge that I am able to present, the applications and variations of this protocol will be just that much simpler to understand and realize.

Many applications use NTP, from large distributed computing services [2] to small scripts, and they all use it for one simple thing, to tell time, but not *a* time, *the* time. Knowing this, one of the most interesting places to see NTP in my opinion is in the area of multi-player gaming, which is in very many ways similar to the large distributed systems mentioned earlier, where precise times-keeping and sequence keeping is required.

There is also a highly useful variation on the standard NTP protocol called Simple Network Time Protocol (SNTP). An attempt will be made to distinguish this protocol from NTP as much as possible. I fully realize that the distinction will not be terribly great as related as the protocols are, but I do hope to bring to light the cases where the full NTP protocol are neither required nor wanted.

2. FOUNDATION
A first question to ask about NTP is what exactly does it do? The answer is quite simple. It tells you how far away from the right time you are, and it does this very quickly. The speed of the protocol is quite important, because when you are dealing with time you need to be as accurate as possible. One example that I have found is the NaradaBrokering system [2]. A system that keeps track of the order of events in a distributed working environment, the challenge being that different computers have different errors in their clocks [2]. Here especially extreme accuracy is needed for an effective system.

To ensure this accuracy the protocol is broken up into many stratums, or layers, with stratum 0 being a device that knows the exact time, such as an atomic clock. There is then one or more computers in the stratum 1 layer that talks that stratum 0 device directly to get the most accurate time possible. These stratum 1 servers then serve that time to one or more stratum 2 clients which may also be servers to sundry stratum 3 clients. This process continues until the clients on the bottom most stratum receive the time. This tree structure is implemented to minimize network delay while determining *the* time, because, although NTP does factor in network time when calculating time, the client must wait for the round trip to the sever to continue with its operation, and the nearest stratum 1 server may be prohibitively far away.

Once a client has found a suitable server, the client sends a very specific packet to the server to request the current time. This packet includes the header shown in Figure 1 which hold many bits of information vital to the successful working of the NTP algorithms. The fields I will mention here are the four 64 bit time stamps in the middle of the header, these are vital to the algorithms used to generate *the* time, directly from [4]:

Reference Timestamp: Time when the system clock was last set or corrected, in NTP timestamp format

Originate Timestamp: Time at the client when the request departed for the server, in NTP timestamp format

Receive Timestamp: Time at the server when the request arrived from the client, in NTP timestamp format
Transmit Timestamp: Time at the server when the response left for the client, in NTP timestamp format

There is also one more value that is computed rather than sent in the header and that is a value I will call Destination, again from [4]:
Destination: Time at the client when the reply arrived from the server, in NTP timestamp format

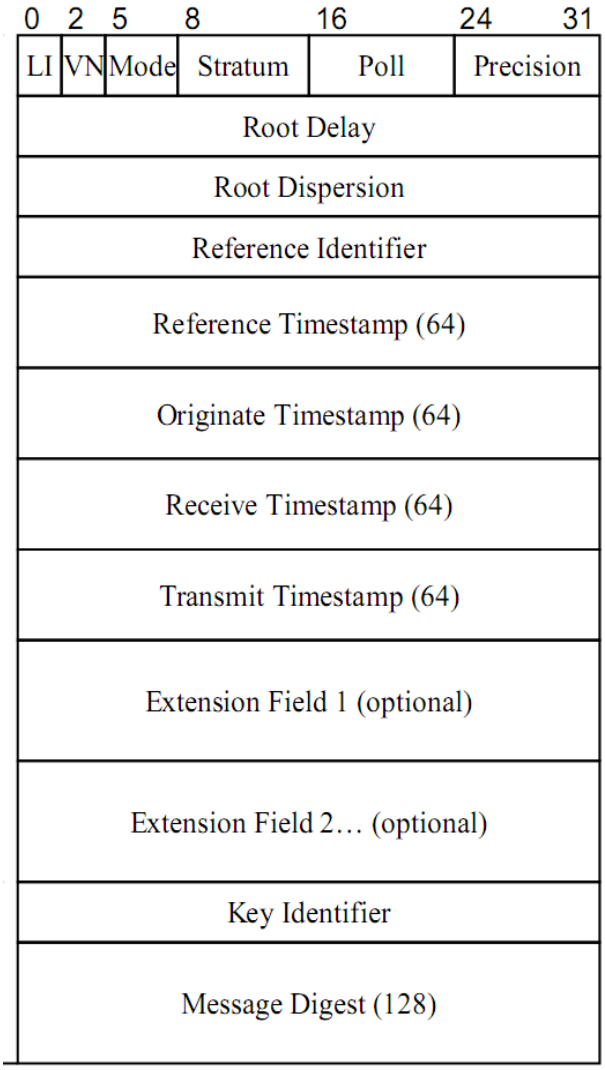

Figure 1: (S)NTP Header Format [4]

With these values, as well as the algorithms that the servers use to make sure that they pull the most accurate time possible from their servers makes finding *the* time quite attainable.

3. GAMING

Playing video games has become at least a hobby for many people and in some cases and obsession. For most people who play video games, the challenge in the game is a real goal. To that end many, many people choose to play games against other humans instead of against computers. Whether these people play over a simple local area network, or across continents through the internet there is a required constant of each player experiencing the events in the game both in the correct order and at the correct time. Some games such as MiMaze do use NTP to control almost all synchronization within the game by taking user input within the last valid time interval and using that to update the game state[3].

Unfortunately, NTP often has too much over head, and SNTP often is not accurate enough for (S)NTP to be used alone to control synchronization[6], so most games use another system to control moment to moment user actions. However, if the individual system are not synchronized then the timing data coming from the various gaming clients could lead to unfair gameplay.

This area is where NTP comes in to play, even if NTP is not used for the moment to moment gameplay, different systems not only have different times but their clocks are also operating at different speeds [2]. Even if this difference in rate is tiny, it could eventually compromise the gameplay between two, or more players. Many games, therefore may use NTP at the least at the start of a game, and hopefully at intervals to re synchronize a common time among all game clients[5].

4. SIMPLE TIME

The Simple Network Time Protocol is really a sub-protocol of NTP. In that while it is a fully realized and implemented protocol, it is fully compatible with NTP but only offers a subset of its functionality. For the following examples it is important to know that any NTP client that has only one server that it pulls time from and they have no clients that pull from it, can be thought of as an SNTP client[4].

As an example, NTP servers can, and are, clients to multiple higher stratum servers. This is, of course, so they can serve time to their clients. However pulling from several servers, they have complex algorithms to determine which of their servers is the most accurate and use it when pulling time[4]. Because an SNTP client has only one upstream server, it has absolutely no need for the algorithms to determine the best source, they are wasting resources. Because SNTP clients are only pulling from one source, they are slightly less accurate[1], especially so as in the simplest form the network travel time is not accounted for[4], but since they have no clients themselves, this is not as large a concern.

Beyond that, most NTP clients also have algorithms for gradually shifting the clock on the clients machine to the current time. However in SNTP these algorithms are missing, forcing the clock to move in set increments to match the current time. Because of the removal of these algorithms SNTP clients are better suited to systems where resources are scarce at the extremities of a network, however the reduced accuracy of SNTP must be taken into account when considering this alternative.

The scripts written in conjunction with with documents are examples of just such SNTP clients, although those clients do not even modify the system clock, just display the time to standard out.

5. CONCLSION

The Network Time Protocol in all of its variations and versions is a very important layer under any network. As people, time is very important to us, we need to have things done by a certain time, or be at an appointment at a certain time. The same is true of computers, which need correct time to determine the exact order of happenings, NTP allows for this. Whether you have a large distributed work environment, a medium sized multiplayer game going with your friends, or just want your computer to have accurate time for your own reasons, NTP and its versions allow you to choose how much importance you place on *the* time, or how much you place on your resources. Either way you have the time.

6. REFERENCES

[1] Evans, Dave 2007 *Differences between NTP and SNTP* (Sept 2010) *http://searchwarp.com/swa209278.htm*

[2] Hasan Bulut , Shrideep Pallickara , Geoffrey Fox, Implementing a NTP-based time service within a distributed middleware system, Proceedings of the 3rd international symposium on Principles and practice of programming in Java, June 16-18, 2004, Las Vegas, Nevada

[3] Lety, Emmanuel; Gautier, Laurent and Diot, Christophe, *MiMaze, a 3D Multi-Player Game on the Internet,* INRIA Sophia Antipolis

[4] *Mills, David L. 2006 Network Time Protocol Version 4 Reference and Implementation Guide* University of Delaware

[5] Pantel, L. and Wolf, L. C. 2002. On the impact of delay on real-time multiplayer games. In Proceedings of the 12th international Workshop on Network and Operating Systems Support For Digital Audio and Video (Miami, Florida, USA, May 12 - 14, 2002). NOSSDAV '02. ACM, New York, NY, 23-29. DOI= http://doi.acm.org/10.1145/507670.507674

[6] Simpson, Zachary 2000 *A Stream-based Time Synchronization Technique For Networked Computer Games* (Spet 2010) http://www.mine-control.com/zack/timesync/timesync.html