
Meinberg Funkuhren

# Kurzbeschreibung Funktionsweise 'ntpd'

# Table of Contents

# Kurzbeschreibung Funktionsweise 'ntpd'

Die Referenzimplementierung des NTP-Protokolls, **`ntpd`**, ist eine freie Software, die durch das NTP-Projekt bzw. durch die Network Time Foundation gepflegt wird.

Meinberg unterstützt das NTP-Projekt durch Spenden, führt Tests durch, und steuert gelegentlich auch Patches bei. Die Software wird nur als Quellcode veröffentlicht. Bei Meinberg wird der veröffentlichen Quellcode für Windows compiliert und ein Setup-Programm erstellt, um die Installation unter Windows zu vereinfachen. Das Installationsprogramm kann kostenlos hier heruntergeladen werden:
https://www.meinberg.de/german/sw/ntp.htm#ntp_stable

Viele Informationen zu NTP gibt es in den Whitepapers auf unserem Webserver:
https://www.meinberg.de/german/info/#whitepaper

Der NTP-Service (`ntpd`) kann unter Windows gegebenenfalls den `w32time`-Service ersetzen. Allerdings ist es nicht sinnvoll, `ntpd` auf Domain Controllern in einem Windows Active Directory zu installieren. `w32time` auf einem Domain Controller trägt sich als authoritative Zeitquelle in der Domain ein, so dass Domain-Clients mit `w32time` automatisch erkennen, an welchen Rechner sie ihre NTP-Anfragen senden sollen. `ntpd` unterstützt dies jedoch nicht. Siehe auch Kapitel 12.5 in diesem Whitepaper:
https://www.meinberg.de/download/burnicki/Computer%20Time%20Synchronization%20Concepts%20-%202014-04-29.pdf

In regelmäßigen Intervallen vergleicht `ntpd` seine eigene Systemzeit mit allen konfigurierten Zeitquellen/NTP-Servern. Er klassifiziert dabei auch die Zeitquellen je nachdem, wie groß der Netzwerk-Jitter während der Abfrage ist, ob die ermittelten Zeitdifferenzen übereinstimmen, usw., und wählt auf Grund der Ergebnisse die “beste” Zeitquelle aus, den sogenannten **`system peer`**. Siehe auch:
https://www.meinbergglobal.com/download/ntp/docs/html/prefer.html
https://www.meinbergglobal.com/download/ntp/docs/html/select.html

Beim Vergleich mit der “besten” Zeitquelle wird nicht nur die momentane Zeitdifferenz berechnet, sondern auch, ob und wie stark sich die Zeitdifferenz bei mehreren aufeinanderfolgenden Zeitvergleichen ändert, d.h. wie stark die lokale Systemzeit von der Referenzzeit wegdriftet. Diese Drift wird verursacht durch eine herstellungsbedingte Ablage der tatsächlichen Frequenz des Oszillators auf dem Mainboard von der Sollfrequenz. Zusätzlich ändert sich dieser individuelle, mittlere Driftwert noch mehr oder weniger mit der Umgebungstemperatur im Rechnergehäuse, wenn die CPU mehr oder weniger stark ausgelastet ist, und mit der Temperatur im Serverraum, wenn z.B. die Klimatisierung ein- oder aussetzt.

Wenn die ermittelte Zeitabweichung unter dem sogenannten **`step threshold`** (standardmäßig 128 ms) liegt, wird die Systemzeit kontinuierlich “weich” so nachgeführt, dass sowohl die Zeitabweichung als auch die Zeitdrift minimiert werden. Siehe auch:
Time Synchronization Accuracy With NTP

Wenn die ermittelte Zeitdifferenz größer ist als der `step threshold`, aber kleiner als der sogenannte **`panic threshold`** (standardmäßig 1000 s), wird die Systemzeit hart gesetzt. bei jedem Setzen der Systemzeit werden die bisher ermittelten Werte zur Zeit- und Driftkompensation

verworfen, und die Regelung beginnt komplett von Neuem.

Wenn die ermittelte Zeitdifferenz größer ist als der `panic threshold`, wird dies nur einmalig nach dem Start des Service akzeptiert (wenn die Kommandozeilenoption '-g' verwendet wurde, was normalerweise der Fall ist), und die Systemzeit wird gesetzt.

Wenn die Systemzeit stabil läuft, sollte ein Setzen der Systemzeit nur einmalig nach Start des Dienstes erforderlich sein. Anschließend sollten die Abweichungen dauerhaft so klein sein, dass die kontinuierliche “weiche” Nachführung ausreicht, größeren Zeitdifferenzen erst gar nicht mehr entstehen zu lassen. Die Driftkompensation ist bei `ntpd` auf 500 PPM (500 Microsekunden pro Sekunde) begrenzt, das sollte jedoch für reale Systeme in jedem Fall ausreichen.

Wenn während des Betriebes die Zeitabweichung schon gering war, dann aber plötzlich eine Zeitabweichung auftritt, die größer ist als der `step threshold`, wird *nicht* sofort die Systemzeit hart gesetzt, denn auch eine Änderung der Netzwerk-Route im WAN könnte der Grund dafür sein. Stattdessen wird zunächst das **`stepout threshold`**-Intervall (standardmäßig 300 s) abgewartet, ob die große Zeitabweichung bestehen bleibt.

Wenn nach Ablauf des `stepout threshold`-Intervalls die Zeitabweichung immer noch groß ist, wird die Systemzeit hart gesetzt, wenn die Abweichung unter dem `panic threshold` liegt. Wenn sie jedoch über dem `panic threshold` liegt, beendet sich der NTP-Service mit einer Meldung im Syslog/Eventlog, die sinngemnäß sagt: *“Jemand hat die Zeit verstellt. Das kann nur der Administrator gewesen sein, und der weiß, was er tut. Ich kann da nichts mehr machen und beende mich.”*.

Das ist auch der Grund dafür, dass es *keine* gute Idee ist, die Zeitsynchronisierung zu testen, indem man die Systemzeit im laufenden Betrieb verstellt und dann von `ntpd` erwartet, dass die Zeit danach schnell wieder korrigiert wird. Siehe auch Kapitel 6.1.18.1 im Whitepaper “Time Synchronization Concepts”.

Nur wenn die unkorrigierte Systemzeit zwar driftet, aber ansonsten stabil ist und nicht hin- und herspringt, kann `ntpd` exakt den Zeitoffset und die Zeitdrift bestimmen und diese kompensieren. Wenn z.B. durch eine unsauber programmierte Treibersoftware die Systemzeit zusätzlich falsch geht (z.B. durch verlorene Timer Ticks), oder wenn das System in einer virtuellen Maschine (VMware, Hyper-V, ...) läuft und die Systemzeit durch die Virtualisierung mit einem mehr oder weniger starken Jitter behaftet ist, kann ein Programm wie `ntpd` dies nur sehr begrenzt kompensieren.

VMware hat es bereits seit längerer Zeit geschafft, diesen Jitter so gering wie möglich zu halten. Bei Microsofts Hyper-V funktioniert das erst in aktuellen Versionen von Windows Server 2016 zufriedenstellend, aber auf einer physikalischen Maschine ist das Zeitverhalten immer noch am besten.

## w32time

Bei `w32time` hängt das Verhalten im Detail sehr stark von der Programmversion ab, d.h. auch von der genauen Windows-Version und dem Service Pack Level.

Auch die Installation ist jedoch entscheidend. Wenn eine Maschine als “Standalone”-Rechner installiert wird, wird `w32time` oft automatisch so konfiguriert, dass er nur einmal pro *Woche*

den Server `time.microsoft.com` kontaktiert und dann die Systemzeit stellt, die danach eine Woche lang frei läuft. Selbst wenn ein anderer Zeitserver eingetragen wird, wird nicht unbedingt automatisch das Abfrage-Intervall verkürzt, sondern man muss das manuell ändern.

Wird ein Rechner dagegen gleich als Mitglied einer Windows-Domäne eingerichtet, wird er normalerweise so konfiguriert, dass er sich zyklisch auf den Domain Controller synchronisiert, der sich als authoritative Zeitquelle im Directory eingetragen hat.

## Stratum-Werte und Synchron-Status

Während in der Telekommunikation der Stratum-Wert eine gewisse Genauigkeitsklasse angibt, wird der Stratum-Wert bei NTP nur für die verschiedenen Hierarchie-Ebenen verwendet. Ein NTP-Server mit einer Funkuhr, einem GPS-Empfänger oder Ähnlichem bildet die Spitze der Hierarchie und ist im Netzwerk als “Stratum 1”-Server sichtbar. Ein `ntpd` als Client dieses “Stratum 1”-Servers wird selbst zum “Stratum 2”-Server, dessen Clients werden wiederum zum “Stratum 3”, usw.

Eine Sonderrolle spielt der “Stratum 16”, der in den NTP-Paketen als numerischer Wert 0 verschickt wird. Dieser Wert gibt (neben den entsprechenden sogenannten Leap Bits) an, dass der sendende NTP-Server noch nicht synchron ist.

Der Stratum-Wert dient auch dazu, einen Timing Loop zu vermeiden, wo der Server an der Spitze der Hierarchie versehentlich falsch konfiguriert wird, um sich die Zeit von einem Server am Ende der Hierarchie zu holen, der effektiv sein eigener Client ist.

Bei jeder guten NTP-Client-Implementierung akzeptiert ein Client die Zeit von seiner Zeitquelle nur, wenn diese Zeitquelle sich als “synchron zu einer anderen Zeitquelle” meldet. Die Art der Zeitquelle spielt dabei keine Rolle. Bei einem “Stratum 1”-NTP-Server mit GPS-Empfänger akzeptiert `ntpd` die Zeit vom GPS-Empfänger nur, wenn die Antenne angeschlossen und ungestörter GPS-Empfang möglich ist. Nur in diesem Fall meldet sich der “Stratum 1”-Server im Netzwerk als “synchron”.

Dabei muss man jedoch unterscheiden, ob der NTP-Server seit dem Start noch niemals synchron war, oder ob er bereits einmal synchron war, aber momentan seine Zeitquellen nicht mehr erreichbar sind.

Wenn ein `ntpd`-Server noch nie synchron zu seiner Zeitquelle war, antwortet er auf Client-Anfragen mit Paketen, in denen die beiden sogenannten **`leap bits`** auf `11` und der Stratum auf `16` (numerisch 0) gesetzt sind. Clients erkennen daran, dass der Server zwar antwortet, aber keine verläßliche Zeit liefern kann.

Erst nachdem der Server sich auf seine eigene(n) Zeitquelle(n) synchronisiert hat, werden die `leap bits` als `00` und der Stratum entsprechend der Hierarchie-Ebene (1, 2, 3, ...) gesendet.

Falls irgendwann aus irgendeinem Grund keine der auf dem Server konfigurierten Zeitquellen mehr erreichbar ist, kann die Zeit des Servers nicht mehr weiter nachgeführt werden.

In früheren NTP-Version (NTP v3) begann dann der `ntpd`-Server wieder, die Leap Bits “00” und Stratum “16” an seine Clients zu senden. Die Folge war, dass keiner den Clients mehr diesen Server als Zeitquelle akzeptiert hat, und die Zeit auf den einzelnen Clients begann, auseinander zu driften. Um dies zu verhindern, wurde damals die **`Local Clock`** als Fallback eingeführt. Darauf konnte sich der Server immer noch synchronisieren und sich “synchron” dazu melden, obwohl seine Zeit nicht

mehr wirklich synchronisiert wurde, sondern langsam wegdriftete. Immerhin konnten sich damit die Clients weiterhin auf den Server synchronisieren, und die Zeit auf jedem einzelnen Client driftete genauso wie die des Servers, das heißt, die Zeitdifferenz zwischen den einzelnen Clients blieb gering.

Nun wird ja die Zeit des Servers vor dem Ausfall der Zeitquellen gut eingeregelt und ist nicht sofort “schlecht”, wenn sie dann plötzlich nicht mehr weiter nachgeführt werden kann, weil keine Zeitquelle mehr zur Verfügung steht. Aus diesem Grund hat man sich bei NTP v4 entschieden, den Server-Status nicht mehr auf “asynchron” (Leap Bits “11”, Stratum “16” bzw. “0”) zu setzen, sondern das Feld **`Root Dispersion`** im NTP-Paket zu verwenden, um anzuzeigen, dass die Zeit des Servers driftet.

Im Normalbetrieb, wenn die Zeit des Servers kontinuierlich nachgeregelt wird, bleibt der `Root Dispersion`-Wert gering, je nach Zeitquelle kleiner als eine Millisekunde oder einige Millisekunden. Wenn die Zeit des NTP-Servers anfängt zu driften, steigt der `Root Dispersion`-Wert langsam an, obwohl immer noch die Leap Bits “00” und der “gute” Stratum-Wert an die Clients gesendet wird. Jeder Client kann dann selbst entscheiden, ob er dann eine andere Zeitquelle bevorzugt, die nicht driftet, und den driftenden Server ignoriert, oder ob er den driftenden Server weiterhin verwendet, weil keine andere, bessere Zeitquelle zur Verfügung steht.

Durch Verwendung der `Root dispersion` in NTP v4 ist eigentlich die `Local Clock` als Fallback-Zeitquelle überflüssig geworden. Es gibt jedoch wenige Ausnahmefälle, z.B. wenn eine Client-Software wirklich nur dann auf eine 2. Zeitquelle umschaltet, wenn die erste nicht mehr erreichbar oder nicht synchron ist. Man kann dann auf dem Server die `Local Clock` mit Stratum “15” konfigurieren. Wenn dann keine andere Zeitquelle mehr erreichbar ist, synchronisiert sich der Server auf die `Local Clock` und wird damit selbst zum “Stratum 16”-Server, so dass seine Clients ihn als “nicht synchron” ansehen und ignorieren.

Es gibt auch durchaus NTP-Clients (darunter auch Versionen von w32time), die einen Server generell nicht mehr akzeptieren, wenn er sich z.B. mit Stratum “12” oder schlechter meldet.

Auf der anderen Seite kann es sein, dass ein `w32time`-Server immer mit einer `Root Dispersion` größer als 10 Sekunden antwortet. Dann würde ein `ntpd` als Client diesen Server nicht unbedingt als Zeitquelle akzeptieren.

## Testprogramm "ntptest"

Es gibt ein kleines Kommandozeilenprogramm “ntptest”, das einzelne Anfragen an einen NTP-Server senden kann und sowohl den Inhalt des Anfrage- als auch des Antwortpaketes genau anzeigt. Damit kann man schön testen, was ein bestimmter Server an seine Clients sendet. Weitere Infos und Download-Links:

- Utility Program 'ntptest'

## Ausfallsicherheit und Redundanz

Durch die Konfiguration von mehreren NTP-Servern bei einem `ntpd`-Client erhält man sehr einfach eine Redundanz und Ausfallsicherheit in der Zeitsynchronisation, indem man einfach mehrere NTP-Server als Zeitquellen konfiguriert. Wenn einer der Server im Vergleich zu anderen eine falsche Zeit liefert (`falseticker`), driftet, oder gar nicht mehr antwortet, wird dies erkannt und automatisch die

nächstbeste Zeitquelle ausgewählt, soweit verfügbar.

Die schlechteste Konfiguration für ntpd sind genau 2 Zeitquellen, denn wenn diese aus irgendeinem Grund unterschiedliche Zeiten liefern, aber trotzdem beide behaupten, synchron zu sein, kann der Client nicht entscheiden, welche der Zeitquellen wirklich die korrekte Zeit liefert, und es ist möglich, dass der ntpd-Client *beide* Zeitquellen ignoriert. Siehe auch: https://support.ntp.org/bin/view/Support/SelectingOffsiteNTPServers#Section_5.3.3.

Das Verhalten von `w32time` kann je nach Version anders sein als hier für `ntpd` beschrieben, z.B. dass der zweite spezifizierte Server nur abgefragt wird, wenn der erste nicht brauchbar ist oder nicht antwortet.

## Generelle Tipps

In einer gemischten Windows/Linux-Umgebung sollte ein oder mehrere Linux-Rechner als NTP-Server eingerichtet werden, wenn möglich, nicht in einer VM, sondern auf echter Hardware laufend. Wenn dafür gesorgt ist, dass diese NTP-Server eine verlässliche Zeitquelle haben und somit eine stabile Zeit mit gutem Stratum bereitstellen können, sollten Clients jeglicher Art keine Probleme mit der Zeitsynchronisierung haben.

Natürlich können auch Meinberg LANTIME-NTP-Server ins Netzwerk integriert werden und diese Rolle übernehmen. Eine Übersicht über die verschiedenen Modelle gibt es auf der Meinberg-Webseite: https://www.meinberg.de/german/products/ntp-zeitserver.htm

---

*— Martin Burnicki martin.burnicki@meinberg.de, last updated 2023-01-25*