

Zeitabgleich in Internets

Johannes Franken
<jfranken@jfranken.de>

Wer das Datum von Dateien oder den Inhalt von Logfiles vergleicht, die auf verschiedenen Rechnern erstellt wurden, wird es begrüßen, wenn die Systemzeit dieser Rechner synchron läuft. Noch schöner ist es, wenn sie mit der gesetzlichen Zeit übereinstimmt.

Auf dieser Seite gebe ich einen Überblick in die hierzu häufig verwendeten Protokolle und Tools.

Inhalt

1. [Das daytime-Protokoll](#)
2. [Das time-Protokoll](#)
 - a) [rdate](#)
 - b) [netdate](#)
 - c) [localtimed](#)
3. [Das Network-time Protokoll \(NTP\)](#)
 - a) [ntpd](#)
 - b) [\(x\)ntpd](#)
 - c) [ntpq](#)
 - d) [ntpdc](#)
 - e) [Weiterführendes](#)
4. [Das Simple Network Time Protocol \(SNTP\)](#)
 - a) [net time, w32time](#)
5. [Das SMB-Protokoll](#)
 - a) [Unter LanManager \(net time\)](#)
 - b) [Unter Linux \(nettime\)](#)
6. [Weitere Möglichkeiten](#)
 - a) [Das Transmission Control Protocol \(TCP\)](#)
 - b) [Das Internet Control Message Protocol \(ICMP\)](#)
 - c) [Das Internet Protocol \(IP\)](#)

Das daytime-Protokoll

Das daytime Protokoll wurde 1983 als [RFC867](#) spezifiziert. Der daytime-server ist meist im inetd eingebaut und liefert auf Port 13 die lokale Zeit im Klartext:

```
$ netcat hamster 13  
Tue Sep 3 19:04:14 2002
```

Leider ist das Datumsformat nicht vorgeschrieben, weshalb eine automatische Verarbeitung erst nach vorheriger Vereinbarung des Formats möglich ist. Bei einigen Implementierungen fehlt sogar die Angabe zur Zeitzone. Probleme, etwa zu Beginn der Sommerzeit, sind damit vorprogrammiert. Die kleinste Einheit ist die Sekunde, damit weicht die angezeigte Zeit bis zu 1 Sekunde plus Übertragungsdauer von der tatsächlichen Zeit ab.

Das time-Protokoll

Das time Protokoll wurde kurz darauf als [RFC868](#) veröffentlicht, um die automatische Verarbeitung des Datums zu vereinfachen. Der Time-Server ist heute ebenfalls im inetd eingebaut und teilt anfragenden Clients auf Port 37 die Zahl der Sekunden seit 1900 (Greenwich Mean Time) mit. Dabei entstehen die selben Abweichungen wie beim daytime-Protokoll.

So funktioniert die Abfrage in der Praxis:

```
$ netcat hamster 37 | od -t u1
0000000 193  31 128 156
0000004
```

Der time-Server liefert uns ein 4 Byte Wort. Umwandlung in eine Zahl:

```
$ echo '156 + 128*256 + 31*256*256 + 193*256*256*256' | bc
3240067228
```

70 Jahre später (wegen des beschränkten Eingabebereichs von gnu date)

```
$ echo 3240067228 -2208988800 | bc
1031078428
```

Welches Datum ist also heute, 1031078428 Sekunden nach 1970?

```
$ date -d "1970-01-01 0:00:1031078428"
Tue Sep  3 19:40:28 CEST 2002
```

Wenn auf Ihrem System eine neuere Implementierung (als 2003?) von GNU date installiert ist, wenden Sie stattdessen bitte folgende Syntax an:

```
$ date -d "19700101 + 1031078428 seconds"
Tue Sep  3 19:40:28 CEST 2002
```

rdate

Das Standard-Tool zum Kopieren der Uhrzeit über das time-Protokoll heißt **rdate**, Beispielsweise

```
$ rdate -p hamster
Wed Sep  4 19:46:13 2002
```

zeigt die Uhrzeit des Rechners **hamster** an. Wenn man es als root und ohne den Parameter **-p** aufruft, wird diese Zeit auf dem eigenen Rechner übernommen. Natürlich kann man es auch regelmäßig per cron aufrufen.

Mehr zum Thema:
siehe [rdate\(8\) manpage](#)

Problematisch bei rdate ist, dass falsche Systemzeiten sich leicht verbreiten.

netdate

Abhilfe schafft das Tool **netdate**, indem es die zunächst die Zeiten **mehrerer** Zeitserver vergleicht und dann die Zeit ersten Servers der größten Übereinstimmungsgruppe übernimmt.

```
$ netdate tcp hamster gate hera
hamster -0.738      Wed Sep  4 20:35:16.000
gate -0.742       Wed Sep  4 20:35:16.000
hera -1.479       Wed Sep  4 20:35:16.000
hamster -0.481     Wed Sep  4 20:35:17.000
```

Mehr zum Thema:

siehe [netdate\(8\) manpage](#)

localtimed

Da time-Server die GMT-Zeit senden, müssen die Clients die Stunden für ihre Zeitzone und ggf. die Sommerzeit selbst hinzuaddieren, um die lokale Zeit zu ermitteln.

Leider gibt es immer noch time-Clients (z.B. in Kartenlesegeräten), die die übertragene Zeit als lokale Zeit interpretieren. Um dieses Problem zu lösen, habe ich einen *localtime-Dämon* geschrieben, der die lokale Zeit des Servers (inkl. Zeitzone/Sommerzeit) sendet.

Download: [localtimed \[1 kB\]](#)

Das Network-time Protokoll (NTP)

Das NTP-Protokoll wurde 1985 von Professor Dr. David L. Mills entwickelt und wird heute in Version 3 ([RFC1305](#)) eingesetzt. Inzwischen gibt es sogar Version 4, die zusätzlich IPv6-fähig ist.

Ein NTP-Client fragt die Zeit (inkl. Millisekunden!) von mehreren Zeitservern ab. Nach einiger Beobachtung und dem Einsatz ausgekügelter, mathematischer Verfahren kann er den besten Server auswählen und die Netzlaufzeit herausrechnen. Die so ermittelte Zeit weicht in der Praxis weniger als 50ms von der des Zeitservers ab.

ntpdate

Das Programm `ntpdate` übernimmt die Zeit von dem übergebenen NTP-Server:

```
$ /etc/init.d/ntp stop
Stopping NTP server: ntpd.
$ ntpdate -b gate
8 Sep 18:48:58 ntpdate[26171]: step time server 192.168.42.1 offset -0.000031 sec
```

Wenn man es ohne den `-b` Parameter aufruft, wird die lokale Uhr zur Vermeidung von Zeitsprüngen so lange beschleunigt oder verzögert, bis sie stimmt. Auf diese Weise kann es vorkommen, dass `ntpdate` für mehrere Stunden läuft. Wer `ntpdate` beim Systemstart (z.B. als `/etc/init.d/ntpdate`) aufruft, sollte daher den `-b` Parameter unbedingt setzen.

`ntpdate` kann übrigens die Uhrzeit nicht ändern, solange ein `ntpd` auf dem selben Rechner aktiv ist. Wenn `ntpdate` die Fehlermeldung

```
8 Sep 18:50:42 ntpdate[4671]: the NTP socket is in use, exiting
```

ausgibt, hatte man vergessen, den `ntpd` zu stoppen.

Mehr zum Thema:

siehe [ntpdate\(1\) manpage](#)

(x)ntpd

Der (x)ntpd [(Experimental) Network Time Protocol Demon] dient der dauerhaften Synchronisation von Systemzeiten. Aufgrund besonderer, mathematischer Maßnahmen erzielt er damit selbst bei minimaler Netzlast Abweichungen im Bereich von Nanosekunden. Er

1. fragt in ganz bestimmten Intervallen die Zeit verschiedener ntp-Server und sonstiger Hardware (z.B. DCF-77 oder GPS-Empfänger) ab,
2. rechnet die übertragungsbedingten Verzögerungen heraus,
3. sortiert die Zeitquellen nach Zuverlässigkeit,
4. korrigiert kleinere Abweichungen durch beschleunigen oder verzögern der lokalen Uhr (Vermeidung von Zeitsprüngen),
5. justiert die Geschwindigkeit der lokalen Uhr dauerhaft, so dass diese bei konstanter Raumtemperatur auch ohne Netz exakt läuft,
6. bietet die Zeit über ntp im Netz an, wahlweise
 - auf Anfrage (unicast),
 - als regelmaessige Rundsendung im lokalen Segment (Broadcast), oder
 - als Multicast-Strom an alle angemeldeten Teilnehmer.
7. bietet die gegenseitige Authentifizierung an, so dass ein Schaden durch unbefugte Server oder Clients ausgeschlossen werden kann.

Üblicherweise hat man im LAN einen Zeitserver, der die Uhrzeit von öffentlichen Anbietern im Internet holt und allen Clients zur Verfügung stellt.

So sieht die Konfigdatei `/etc/ntp.conf` auf allen Clients aus:

```
driftfile /var/lib/ntp/ntp.drift
server 192.168.42.1 burst
```

Erläuterungen:

- Das `driftfile` speichert die Erkenntnisse bzgl. der Fehler der lokalen Uhr, so dass diese dem `ntpd` auch nach einem Restart oder Reboot vorliegen.
- Der Host `192.168.42.1` dient als Zeitserver im Unicast-Modus.
- Er wird im `burst`-Modus nach seiner Zeit gefragt, was die Messung der Zuverlässigkeit und netzwerkbedingter Verzögerungen sichtlich beschleunigt. Die damit verbundene, achtfache Netzlast in den ersten 30 Sekunden und nach jedem Verbindungsverlust fällt im 100Mbit LAN nicht ins Gewicht.

und so die auf dem Server:

```
driftfile /var/lib/ntp/ntp.drift
server ntp2.ptb.de minpoll 4 maxpoll 10
server xlink1.xlink.net minpoll 4 maxpoll 10
server willow.fernuni-hagen.de minpoll 4 maxpoll 10
server ws-leil.win-ip.dfn.de minpoll 4 maxpoll 10
server tuminfo1.informatik.tu-muenchen.de minpoll 4 maxpoll 10
server NTP.HEH.Uni-Oldenburg.DE minpoll 4 maxpoll 10
server ntps2.gwdg.de minpoll 4 maxpoll 10
server ntp.rz.tu-harburg.de minpoll 4 maxpoll 10
server ntp.nml.csiro.au minpoll 4 maxpoll 10
server ntp0.fau.de minpoll 4 maxpoll 10
server clock.tl.fukuoka-u.ac.jp minpoll 4 maxpoll 10
server goodtime.ijs.si minpoll 4 maxpoll 10
server tick.usno.navy.mil minpoll 4 maxpoll 10
server time-nw.nist.gov minpoll 4 maxpoll 10
```

Erläuterungen:

- der `minpoll`-Parameter gibt an, mit welchem Intervall der `ntpd` zu Beginn und nach einer Störung den Server anfragen soll. Mögliche Werte sind:

Wert	Intervall
4	16 Sekunden
5	32 Sekunden
6	64 Sekunden
...	...
17	36.4 Stunden

- der `maxpoll`-Parameter gibt an, wie groß das Intervall zwischen den Abfragen werden darf, wenn die Kommunikation problemlos ist.

Mehr zum Thema:
siehe [ntp\(1\) manpage](#)

ntpq

Das Tool **ntpq** bietet eine Shell zur Abfrage von Informationen zu dem übergebenen Zeitserver an. Wenn man keinen Servernamen übergibt, wird **localhost** angenommen:

```
$ ntpq
ntpq> help
Commands available:
addvars      associations  authenticate  cl            clearvars
clocklist    clockvar     cooked        cv            debug
delay        exit         help          host          hostnames
keyid        keytype     lassociations lopeers       lpassociations
lpeers       mreadlist   mreadvar     mrl           mrv
ntpversion   opeers      passociations passwd        peers
poll         pstatus     quit          raw           readlist
readvar      rl          rmvars       rv            showvars
timeout      version     writelist    writevar
ntpq> quit
```

Der interessanteste Befehl darin ist **peers**. Man erreicht ihn auch ohne die Shell, indem man **ntpq** mit dem Parameter **-p** aufruft:

```
$ ntpq -pn gate
      remote          refid          st t when poll reach  delay  offset  jitter
=====
*192.53.103.104      .PTB.          1 u  519 1024  377  44.966  0.762  0.689
-193.141.40.1       192.53.103.104  2 u   80 1024  377  53.898 -4.443  8.575
+132.176.114.23     192.53.103.104  2 u   83 1024  377  74.690  0.338  0.024
#193.174.75.162     192.76.176.253  3 u   72 1024  377  46.868  1.128  0.142
#131.159.0.1        131.159.0.76   3 u   84 1024  377  61.069 -3.552  0.441
-134.106.148.1      131.188.3.222  2 u   77 1024  377 129.458 -11.106 11.891
-134.76.98.232     192.53.103.103  2 u   71 1024  367  46.386  1.660  1.055
 134.28.202.15     0.0.0.0        0 u   - 1024   0  0.000  0.000 4000.00
+130.155.98.1      .ATOM.         1 u   81 1024  371 348.563  1.065  0.051
 131.188.3.220     .GPS.          1 u  12h 1024   0  0.000  0.000 4000.00
-133.100.11.8      .GPS.          1 u  133 1024  357 385.072 -7.701  0.799
-193.2.4.2         .GPS.          1 u 1106 1024  376  83.967 -3.170  0.057
-192.5.41.40       .PSC.          1 u   27 1024  377 157.448 11.293 10.819
-131.107.1.10      .ACTS.         1 u   78 1024  377 280.552 -26.444 37.160
```

Erläuterungen: Die Tabelle zeigt jeweils eine Zeile für jeden konfigurierten Server an. Die Spalten haben folgende Bedeutung:

Spalte	Erläuterung								
Das erste Zeichen	gibt die Qualität des Servers an: <table border="1"> <tr> <td>*</td> <td>Der beste. Seine Zeit wird gerade als Referenz angenommen.</td> </tr> <tr> <td>+ # - .</td> <td>akzeptable Qualität (sortiert nach absteigender Qualität)</td> </tr> <tr> <td>x</td> <td>die Zeit dieses Servers ist nicht schlüssig</td> </tr> <tr> <td>(Leerzeichen)</td> <td>keine Antwort oder vom lokalen Server abhängig</td> </tr> </table>	*	Der beste. Seine Zeit wird gerade als Referenz angenommen.	+ # - .	akzeptable Qualität (sortiert nach absteigender Qualität)	x	die Zeit dieses Servers ist nicht schlüssig	(Leerzeichen)	keine Antwort oder vom lokalen Server abhängig
*	Der beste. Seine Zeit wird gerade als Referenz angenommen.								
+ # - .	akzeptable Qualität (sortiert nach absteigender Qualität)								
x	die Zeit dieses Servers ist nicht schlüssig								
(Leerzeichen)	keine Antwort oder vom lokalen Server abhängig								
remote	die IP-Adresse oder (wenn man <code>ntpq</code> ohne <code>-n</code> aufruft:) der Hostname des Servers.								
refid	die IP-Adresse oder (wenn man <code>ntpq</code> ohne <code>-n</code> aufruft:) der Hostname des aktuellen Referenzservers dieses Servers.								
st	(stratum, engl. f. Schicht): wie viele Schritte der Server von der primären Zeitquelle (z.B. der Atomuhr) entfernt ist.								
t	Typ der Verbindung zum Server: l =local, u =unicast, m =multicast, b =broadcast								
when	vor wie vielen Sekunden der Server zuletzt befragt wurde.								
poll	in welchem Intervall (Sekunden) der Server derzeit befragt wird.								
reach	die Erreichbarkeit des Servers als Oktalzahl. Die Skala geht von 0 (=nie) bis 377 (=immer).								
delay, offset, jitter	Netzlaufzeit, Abweichung und Streuung der Antworten dieses Servers in Millisekunden. Je weniger, desto besser.								

Mehr zum Thema:

siehe [ntpq\(1\) manpage](#)

ntpd

Das Tool `ntpd` dient zur remote Konfiguration eines laufenden ntp-Servers.

Mehr zum Thema:

siehe [ntpd\(1\) manpage](#)

Weiterführendes

- [Network Time Synchronization Project](#)

Das Simple Network Time Protocol (SNTP)

Im Jahr 1996 erklärte Dr. Mills, dass man in vielen Fällen auf die hohe Präzision von NTP verzichten und das NTP-Protokoll auf eine einzelne Serverabfrage beschränken kann. Dieses Verfahren bezeichnet er in [RFC2030](#) als "Simple Network Time Protocol" (SNTP).

net time, w32time

Unter Windows (2000, 2003 und XP) gibt es zwei SNTP-Clients:

- **w32time** ist ein Systemdienst, der regelmäßig die Zeit kopiert.
- **net time** ist ein Programm zum manuellen Abfragen der Zeit und zum Konfigurieren des w32time.

Befehl	Auswirkung
<code>net time /setsntp:server1,server2,...</code>	zum Auswählen der Zeitserver
<code>net time /querysnTP</code>	zeigt die ausgewählten Zeitserver an
<code>net stop w32time</code> <code>net start w32time</code>	stoppt und startet die regelmäßige Abfrage der Zeitserver

Das SMB-Protokoll

Unter LanManager (net time)

OS/2, MS-DOS, Windows 95, 98, NT3 und NT4 enthalten das Programm **NET**, mit dem Sie die Uhrzeit von Windows-Rechnern, bei denen die Serverdienste (netbios-ssn, TCP-Port 139) aktiviert sind, über das SMB-Protokoll abfragen können:

```
C:\PROGRA~1> net time \\mausi /set /yes
```

Achtung: Ab Windows 2000 verwendet der **NET**-Befehl nicht mehr das SMB-, sondern das [SNT-Protokoll](#).

Unter Linux (nettime)

Der **net**-Befehl ist im Lieferumfang von Samba ab Version 3.0 enthalten.

Wer eine Standalone-Variante von **NET TIME** sucht, findet mit **nettime** eine Lösung:

```
$ nettime //mausi
Current system time set to Wed Sep  4 21:07:18 2002
```

Hier ein statisch gelinktes Binary für Linux:

■ [nettime2.bz2](#) [285 kB]

und hier den Sourcecode als Samba 2.0.7-Modul:

■ [nettime.c](#) [8 kB]

Weitere Möglichkeiten

Das Transmission Control Protocol (TCP)

Linux schreibt in die Optionen am Ende des TCP headers die Hundertstelsekunden seit dem Start des Kernels (5 Bytes)

```
$ tcpdump host mausi & telnet mausi 22
[1] 25912
tcpdump: listening on eth0
12:16:15.513290 hamster.jfranken.de.ssh > gate.jfranken.de.2156:
P 3146288727:3146288759(32) ack 3154615717 win 8576
<nop,nop,timestamp 104273773 49599748> (DF) [tos 0x10]
^C
$ bc
scale=5
104273773/100/60/60/24
12.06872

0.06872*24
1.64928

0.64928*60
38.95680
quit
$ uptime
12:22:27 up 12 days, 1:38, 4 users, load average: 0.00, 0.00, 0.00
```

Das Internet Control Message Protocol (ICMP)

Professor David L. Mills beschrieb 1981 in [RFC778](#) den Internet Clock Service, der über ICMP-Pakete vom Typ 14 (siehe [RFC792](#)) die Uhrzeit (ohne Datum, genaugenommen die Millisekunden seit Mitternacht) überträgt. W. Richard Stevens hat hierzu das Program `icmptime` geschrieben, welches die lokale Uhrzeit mit der eines entfernten Hosts vergleicht. Es ist erhältlich auf <ftp://ftp.uu.net/published/books/stevens.tcpipv1.tar.Z>

Das Internet Protocol (IP)

[RFC781](#) aus dem Jahr 1981 beschreibt, dass man das ja bereits im IP-Header tun könne. Mir ist allerdings keine praktische Anwendung hierzu bekannt.