

Windows: ?????????? ????????????? ??????????

В какой-то момент я обнаружил, что на одной из моих домашних машин с Windows 10 установлено некорректное время, часы отставали. Я проверил службу Windows Time, она была запущена, но, судя по всему, не работала. Затем я проверил вторую машину (ноутбук) и там ситуация оказалась аналогичной, то есть время не устанавливалось.

Сначала я попробовал решить проблему с помощью Gemini, тот довольно быстро сдался и предложил устанавливать время с помощью скрипта, время предполагалось получать через HTTP с одной из общедоступных служб. Тогда я обратился к ChatGPT, тот оказался гораздо более умным и мы провели исследования в результате которых он пришел к выводу, что не работает что-то именно в самой службе Windows. Мы проверяли обе машины, отпали блокировки портов и протокола UDP, был проверен мой роутер, все было в рабочем состоянии. После того как я потратил на решение проблемы целый день, я решил остановиться и таки да, подключил скрипт в планировщик для установки времени при запуске компьютера.

????????? ??? ?????????? ?? ????? ????????

В Windows за синхронизацию времени отвечает служба **Windows Time** (`W32Time`). В упрощённом виде она работает так:



▼
расчёт смещения

|

▼

корректировка часов

Но внутри есть несколько важных механизмов.

1. ??? ???????? Windows

Компьютер получает первоначальное время из **RTC/CMOS** — аппаратных часов материнской платы.

Именно поэтому при:

```
w32tm /query /source
```

можно увидеть:

```
Local CMOS Clock
```

Это **не означает, что CMOS является настроенным NTP-сервером**. Это означает: *последнего пригодного внешнего источника времени нет, поэтому система использует локальные часы.*

2. W32Time ???????? ????????

Настройка:

```
w32tm /query /configuration
```

показывает, например:

```
Type: NTP
```

```
NtpServer: time.windows.com,0x8
```

То есть W32Time должна обращаться к:

```
time.windows.com
```

по протоколу **NTP**.

Можно указать несколько серверов:

time.windows.com,0x8 pool.ntp.org,0x8 time.nist.gov,0x8

3. ???????????? UDP/123

Классический NTP использует:

UDP source port: 123

UDP destination port: 123

То есть W32Time создаёт NTP-сокеты и отправляет примерно:

PC 192.168.0.103:123

|

| UDP



NTP 213.239.234.28:123

Сервер отвечает:

NTP 213.239.234.28:123

|



PC 192.168.0.103:123

Обычный NTP-клиент может работать иначе:

192.168.0.103:59882 → 185.125.190.57:123

185.125.190.57:123 → 192.168.0.103:59882

А W32Time использует **локальный порт 123**.

4. W32Time ?? ?????? ???????????? «????????? ?????»

NTP-пакет содержит временные метки.

Упрощённо:

T1 = время отправки запроса клиентом

T2 = время получения запроса сервером

T3 = время отправки ответа сервером

T4 = время получения ответа клиентом

По этим четырём значениям можно оценить:

задержку сети:

$$\text{delay} \approx (T4 - T1) - (T3 - T2)$$

и смещение часов клиента:

$$\text{offset} \approx ((T2 - T1) + (T3 - T4)) / 2$$

Поэтому NTP значительно сложнее простого:

“отправил запрос → получил время сервера → поставил его себе».

5. W32Time ?????????? ?????????? ??????????

Windows не обязательно корректирует часы после одного ответа.

Она собирает **NTP samples**:

```
sample 1
sample 2
sample 3
...
```

Затем отбрасывает плохие/аномальные измерения и пытается получить надёжную оценку смещения.

6. ????? Windows ?????????????? ?????

Если расхождение небольшое, Windows может **плавно изменить скорость хода часов**, чтобы постепенно привести их к правильному времени.

Если расхождение достаточно большое, возможна **ступенчатая коррекция (step)**.

Поэтому W32Time — это не просто:

```
получил время → SetSystemTime()
```

а полноценный механизм регулирования системных часов.

7. ????????????? ??????? ???????

В конфигурации можно увидеть:

```
MinPollInterval: 10
MaxPollInterval: 15
```

Это степени двойки:

```
2^10 = 1024 секунд
2^15 = 32768 секунд
```

То есть интервал может динамически изменяться примерно от:

```
17 минут
```

до:

```
9 часов
```

Кроме того, может быть:

```
SpecialPollInterval: 16384
```

Это примерно:

```
4 часа 33 минуты
```

Конкретное поведение зависит от режима и флагов peer'a.

??? ?????????? ??? `w32tm /resync`

Когда мы выполняем:

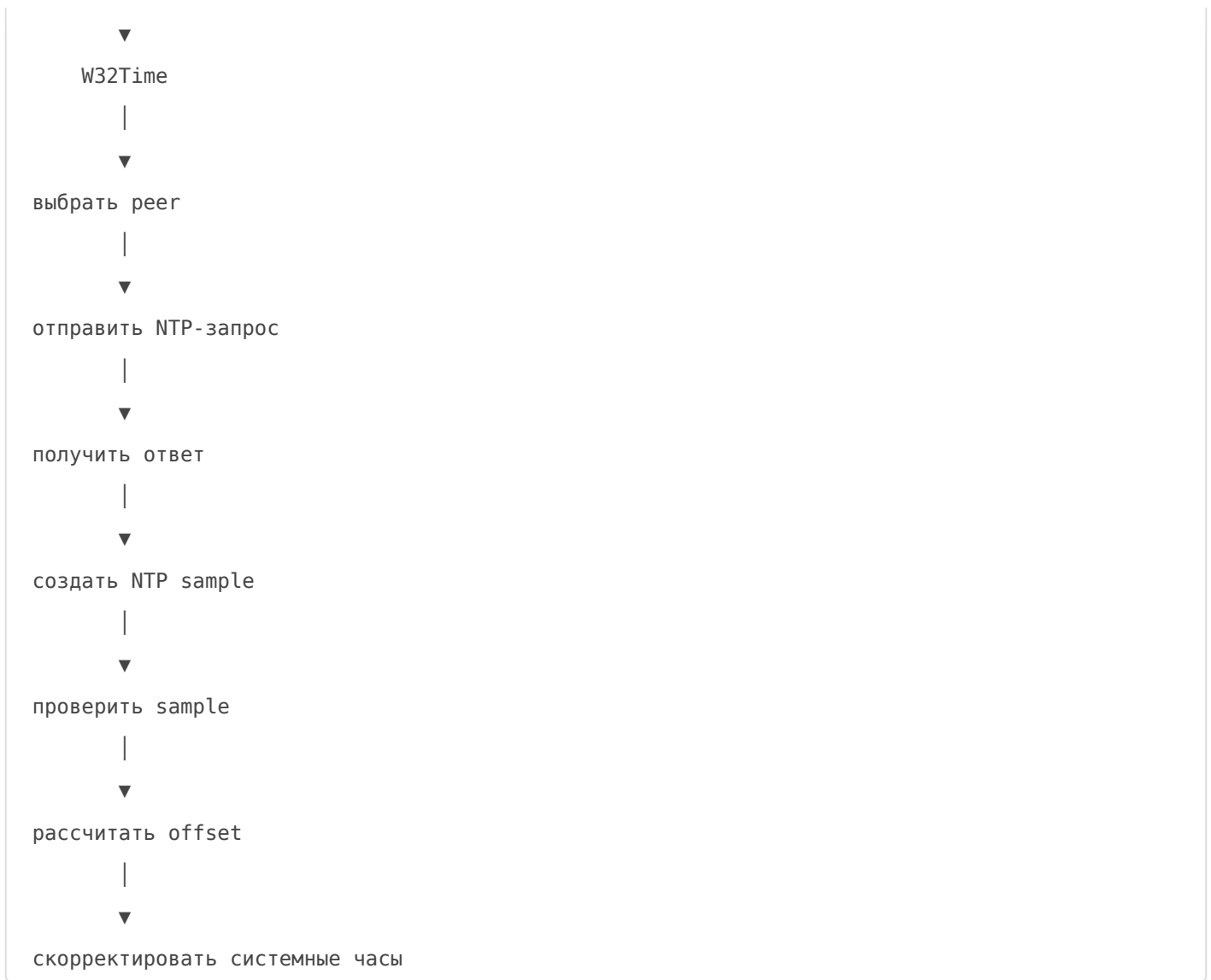
```
w32tm /resync
```

это **не просто команда "скажи серверу время"**.

Она заставляет службу попытаться выполнить новую синхронизацию.

Упрощённо:

```
w32tm /resync
|
```



?????? `stripchart` ?????? ?????? ??????

Команда:

```
w32tm /stripchart /computer:213.239.234.28 /dataonly /samples:3
```

тоже использует NTP, но это **не тот же самый механизм синхронизации, что W32Time service**.

Например:

```
19:58:08, +09.6895857s
19:58:10, +09.6929900s
19:58:12, +09.6926500s
```

Это доказывает:

DNS	✓
маршрутизация	✓
UDP/123	✓
NTP-сервер	✓
ответ сервера	✓

Но это **не доказывает**, что W32Time сможет использовать этот ответ для собственной синхронизации.

? ??? ?????? ???????

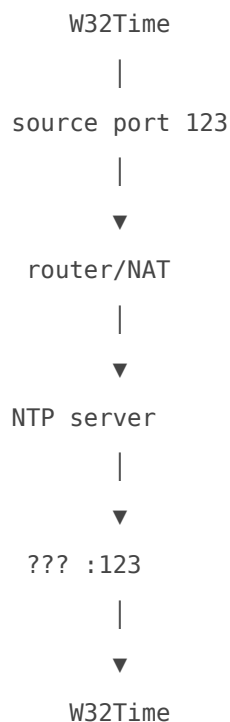
W32Time создаёт сокеты:

```
[::]:123
0.0.0.0:123
```

То есть сама служба слушает **порт UDP 123 на компьютере**.

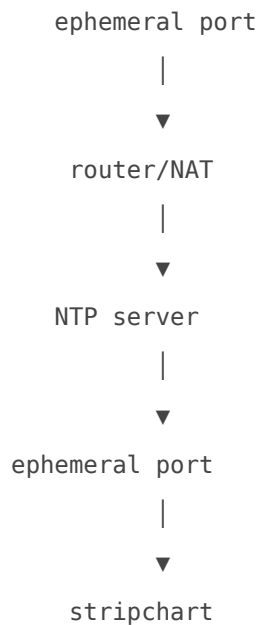
А `stripchart` может использовать другой сетевой путь/сокет.

Поэтому принципиальная разница сейчас выглядит так:



против:

```
stripchart
|
```



????????? ???????? ??????????.

Создайте папку для скриптов, например C:\Scripts и в ней файл SyncTime.ps1 со следующим содержимым:

```
try {
    $res = Invoke-WebRequest -Uri "https://ya.ru" -Method Head -TimeoutSec 5 -UseBasicParsing
    $serverTime = [datetime]$res.Headers.Date
    Set-Date $serverTime
} catch {
    $res = Invoke-WebRequest -Uri "https://www.google.com" -Method Head -TimeoutSec 5 -
UseBasicParsing
    $serverTime = [datetime]$res.Headers.Date
    Set-Date $serverTime
}
```

Сначала будет попытка получить время от Яндекса, если не получится, то от Google.

- Нажмите **Win + R**, введите **taskschd.msc** и нажмите **Enter** (Планировщик заданий).
- Справа нажмите **«Создать задачу...»** (Create Task).
- **На вкладке «Общие»:**
 - Имя: **HTTP Time Sync**
 - Обязательно выберите галочку: **«Выполнить с наивысшими правами»** (Run with highest privileges) — это нужно, чтобы у скрипта были права на изменение системного времени.
 - В поле «Настроить для» выберите **Windows 10** (или Windows 11).
- **На вкладке «Триггеры»:**

- Нажмите **«Создать...»** -> В поле «Начать задачу» выберите **«При запуске»** (At startup) -> OK.
- **«При подключении к сети»** (On workstation unlock / или по событию сна) — чтобы время корректировалось, когда вы открываете крышку ноутбука или разблокируете экран.
- **На вкладке «Действия»:**
 - Нажмите **«Создать...»** -> Действие: **«Запуск программы»** (Start a program).
 - В поле «Программа или сценарий» введите: powershell.exe
 - В поле «Добавить аргументы (необязательно)» введите: -ExecutionPolicy Bypass
-WindowStyle Hidden -File "C:\Scripts\SyncTime.ps1"
- **На вкладке «Условия» и «Параметры»:**
 - Снимите галочку «Запускать задачу только при питании от электросети» (если это ноутбук).
 - Поставьте галочку «Запускать задачу как можно скорее после пропуска планового запуска».

Revision #3

Created 2026-10-05 14:12:17 UTC by Nikolay

Updated 2026-10-06 08:04:54 UTC by Nikolay