

Marcin WOCH  
Politechnika Śląska, Instytut Informatyki

## METODY SYNCHRONIZACJI CZASU W SIECIACH KOMPUTEROWYCH

**Streszczenie.** Artykuł prezentuje metody synchronizacji czasu w sieciach komputerowych, tj. metodę Cristiana, algorytm z Berkeley oraz Network Time Protocol (NTP). Omawia także model Lamporta oraz zegary logiczne. Opisane metody są ogólnie stosowane w sieciach komputerowych. Ponadto protokół NTP jest międzynarodowym standardem używanym do synchronizacji czasu przez sieć Internet, za pomocą modemów telefonicznych, radia oraz satelity.

**Słowa kluczowe:** synchronizacja czasu, metoda Cristiana, algorytm z Berkeley, NTP, relacja uprzedniości zdarzeń, zegar logiczny.

## TIME SYNCHRONIZATION METHODS IN COMPUTER NETWORKS

**Summary.** This article presents time synchronization methods used in computer networks. Cristian method, algorithm from Berkeley and Network Time Protocol (NTP) are described. Article also describes Lamport's happened-before relation and logical clocks. Described methods are widely deployed in computer networks. Moreover NTP is international standard to synchronize clocks via Internet, telephone modems, radios and satellites.

**Keywords:** time synchronization, Cristian method, algorithm from Berkeley, NTP, happened-before relation, casual ordering, logical clock.

### 1. Wstęp

Zegar komputera to pewien zbiór komponentów sprzętowych oraz programowych używanych do ustalania czasu. W sieciach komputerowych, systemach rozproszonych czas jest bardzo ważnym elementem. Istnieją procesy, zdarzenia, które muszą być dokładnie mierzone. Wiele algorytmów, np. sprawdzanie czasu zamówienia (systemy bankowe, sklepy

internetowe), zależy od synchronizacji zegarów. Jest też grupa algorytmów, które do poprawnego działania wymagają odpowiedniej kolejności zdarzeń zachodzących w procesach na różnych komputerach.

Metody synchronizacji czasu można podzielić na dwie grupy: synchronizację zewnętrzną oraz wewnętrzną. Synchronizacja zewnętrzna polega na zsynchronizowaniu zegara ze źródłem wzorcowym. Synchronizacja wewnętrzna sprowadza się do pomiaru odstępów czasu pomiędzy zdarzeniami występującymi w różnych komputerach. Warunkiem poprawnej synchronizacji wewnętrznej jest dokładna znajomość stopnia synchronizacji lokalnych zegarów [1].

## 2. Synchronizacja zegarów fizycznych komputerów

Każdy komputer ma wbudowany zegar fizyczny. Zegary nawet o tej samej budowie mogą mieć różne odchylenia, które są związane z różnicami częstotliwości pracy, spowodowanymi czynnikami fizycznymi. Częstotliwość tego samego zegara też może się wahać. Współczynnik odchylenia zegara jest to rozbieżność pomiędzy danym zegarem a zegarem wzorcowym.

Eliminacja zmian częstotliwości pracy jest niemożliwa, stosuje się natomiast kompensację odchyleń. Artykuł omawia metody kompensacji za pomocą odbiorników czasu, metody Cristiana, algorytmu z Berkeley oraz NTP.

### 2.1. Kompensowanie czasu za pomocą odbiorników czasu

Czas astronomiczny to czas, który jest bezpośrednio związany z ruchem Ziemi wokół Słońca oraz obrotem Ziemi wokół swojej osi. Okres osiowego obrotu Ziemi i okrążania Słońca jest zmienny, więc jednostki, które wynikają z czasu astronomicznego, też nie są stałe.

Obecnie najdokładniejszymi zegarami są zegary atomowe, których dokładność jest rzędu  $10^{13}$ . Jednostki wynikające z czasu atomowego są stałe, np. sekunda wynosi 9 192 631 770 okresów przejść pomiędzy dwoma poziomami energetycznymi  $\text{Cs}^{133}$  w stanie podstawowym [1].

Uniwersalny czas skoordynowany (UTC) to międzynarodowy standard oparty na zegarach atomowych. Czas ten okresowo kompensuje się, aby uzyskać zgodność z czasem astronomicznym. Sygnały UTC są regularnie rozsyłane przez stacje naziemne oraz satelitarne. Przykładem stacji satelitarnej jest system GPS (Globalny system określania położenia), który jest również stosowany w nawigacji.

Jeżeli jest znana dokładna prędkość rozchodzenia się sygnału radiowego oraz odległość odbiornika od źródła, w łatwy sposób można skompensować zegar fizyczny komputera. O ile drugi parametr można jednoznacznie określić, o tyle prędkość rozchodzenia się sygnału radiowego zależy od wielu czynników, np. warunków atmosferycznych, propagacji. Dlatego dokładność odbiorników tego typu wynosi np. 1 ms dla systemu GEOS (Geostationary Environmental Satellites), czy 10 ms dla stacji naziemnych [1].

Gdy czas podawany przez sygnał synchronizujący jest większy od czasu w komputerze, to przestawienie takiego zegara nie stanowi problemu. Natomiast w sytuacji odwrotnej nie można cofnąć czasu komputera, gdyż istnieją aplikacje zakładające, że czas nadal wzrasta. Rozwiązaniem tego problemu jest spowolnienie pracy zegara komputerowego aż do momentu zrównania czasów. Nie wszystkie zegary sprzętowe mają możliwość zmieniania częstotliwości pracy. W takiej sytuacji zmieniana jest częstotliwość aktualizowania czasu poddawanych aplikacjom [1].

## 2.2. Metoda Cristiana

Z przesyłaniem sygnału synchronizującego związane jest pewne opóźnienie, które wynika z samej przepustowości medium, np. sieci komputerowej, dostępności zasobów itp. Właśnie z tych powodów czas przesyłania komunikatów jest zmienny [2]:

$$T_{transmisji} = T_{\min} + x \quad (1)$$

gdzie  $T_{\min}$  oznacza najmniejszy możliwy czas przesłania sygnału w warunkach idealnych, natomiast zmienny składnik opóźnienia  $x \geq 0$  nie jest dokładnie znany.

Cristian zaproponował metodę synchronizacji zegarów, polegającą na pomiarze czasu, który upłynął od momentu wysłania żądania czasu do otrzymania sygnału. Jeżeli komputer wysłał sygnał z żądaniem czasu do serwera w czasie  $T_1$ , a otrzymuje odpowiedź w czasie  $T_2$ , to całkowity czas obiegu wynosi:

$$T_{obiegu} = T_2 - T_1 \quad (2)$$

Przy założeniu że ruch w obie strony odbywa się w jednakowym czasie, można przyjąć, że zegar komputera można skompensować zgodnie z równaniem:

$$T_{komp} = T_{serwera} + \frac{T_{obiegu}}{2} \quad (3)$$

Dokładność tego oszacowania wynika z faktu, że czas zegara serwera w chwili dotarcia odpowiedzi mieści się w przedziale:

$$\left[ T_{serwera} + T_{\min}, T_{serwera} + T_{obiegu} - T_{\min} \right] \quad (4)$$

Szerokość tego przedziału to:

$$T_{obiegu} - 2T_{\min} \quad (5)$$

Dokładność wynosi zatem:

$$\pm \left( \frac{T_{\text{obiegu}}}{2} - T_{\text{min}} \right) \quad (6)$$

Czas  $T_{\text{min}}$  można dokładnie określić, natomiast czas obiegu  $T_{\text{obiegu}}$  można uśrednić wykonując dużą liczbę prób. Najdokładniejsze wyniki otrzymuje się, gdy czas przesyłania komunikatów jest jak najbardziej zbliżony do  $T_{\text{min}}$  [2].

W sieciach tylko z jednym serwerem czasu istnieje możliwość wystąpienia awarii tego serwera, co uniemożliwi synchronizację. Aby zapobiec takiej sytuacji, Cristian zaproponował użycie grupy serwerów czasu, z których każdy posiada własny odbiornik sygnału UTC.

Metoda Cristiana zakłada, że serwery czasu są samokorygujące, tzn. nie uwzględniają one sytuacji, w której serwer przesyła błędny czas, np. z powodu włamania do systemu komputerowego lub błędów aplikacji.

### 2.3. Algorytm z Berkeley

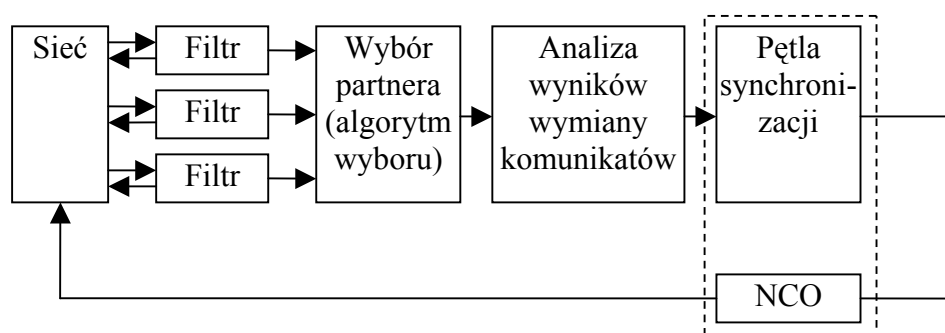
W algorytmie z Berkeley zaproponowanym przez Gusella i Zatti [3] serwer czasu, tzw. master, okresowo odpytuje inne komputery, które odsyłają mu swoje wartości. Master szacuje wartości ich zegarów, uwzględniając czasy obiegu. Serwer wysyła kilka sygnałów, a wartości czasu są uśredniane. Uśrednianie eliminuje tendencje zegarów do przyspieszania lub opóźniania. Dokładność tej metody zależy od maksymalnego czasu obiegu komunikatu pomiędzy serwerem a synchronizowanymi komputerami. Jeżeli czas obiegu w dowolnym przypadku przekroczy czas maksymalny, to taka próba nie jest brana pod uwagę.

Serwer wysyła wielkości, o które czas lokalny musi być skorygowany.

Algorytm z Berkeley, w odróżnieniu od metody Cristiana, uwzględnia awarie zegarów fizycznych. Zegary, które wykazują fałszywe wartości czasów, nie są uwzględniane podczas obliczaniu średniej. Metoda używa tzw. średniej tolerującej awarie, czyli bierze pod uwagę tylko te odczyty, które nie różnią się od siebie więcej niż zadany parametr.

### 2.4. Network Time Protocol (NTP)

Protokół NTP jest standardem synchronizacji zegarów w sieci. Wywodzi się z protokołu ICMP (Internet Control Message Protocol), a został zaprojektowany dla uzyskania wysokiej dokładności oraz stabilności. NTP określa architekturę usług czasu oraz protokół rozpowszechniania informacji o czasie w obrębie sieci Internet [5]. NTP bazuje na protokole IP oraz UDP, co wiąże się z bezpołączeniowym dostarczaniem datagramów. Architektura protokołu NTP jest przedstawiona na rys. 1.



Rys. 1. Schemat protokołu NTP [7]

Fig. 1. Overall organization of NTP [7]

Z dostarczaniem informacji w sieci Internet związanych jest kilka problemów, które protokół NTP rozwiązuje [7]:

- opóźnienia w sieci - protokół NTP wykorzystuje metody statystyczne do filtrowania danych i dzięki temu niwelowane są błędy związane z opóźnieniami w sieci Internet;
- długotrwałe braki połączeń – stosowane są nadmiarowe serwery oraz połączenia pomiędzy nimi. Gdy któryś z nich ulegnie awarii, pozostałe przejmują jego zadania;
- przypadkowe lub celowe zaburzanie usług – w NTP stosowane są techniki uwierzytelniania w celu sprawdzenia, czy źródła są wiarygodne. Ponadto, sprawdzane są ważności adresów zwrotnych przesyłanych komunikatów.

Sieć NTP tworzy pewną architekturę logiczną, zbudowaną warstwowo. W pierwszej warstwie serwery pobierają sygnał czasu bezpośrednio z odbiorników UTC. Każda kolejna warstwa jest synchronizowana przez swoją warstwę nadrzędną. Najniższą warstwę tworzą komputery użytkowników. Budowa ta umożliwia elastyczne dodawanie większej liczby klientów oraz serwerów.

Błąd określenia czasu zależy od warstwy, na której znajduje się dany komputer, gdyż błędy na poszczególnych poziomach sumują się.

Istnieje pięć trybów synchronizacji serwerów NTP [9]:

- tryb 1 - symetryczny aktywny – hosty pracujące w tym trybie rozgłaszają swoją gotowość do wzajemnej synchronizacji w obrębie sieci. Wysyłane komunikaty są niezależne od dostępności komputera i jego warstwy;
- tryb 2 - symetryczny pasywny – ten tryb związany jest z trybem aktywnym. Jest on tworzony po otrzymaniu komunikatu od hosta pracującego w trybie aktywnym. Połączenie istnieje tak długo, jak długo aktywny jest komputer wysyłający komunikat. Hosty pracujące w tym trybie rozgłaszają swoją gotowość do wzajemnej synchronizacji;

- tryb 3 - klienta - hosty pracujące w tym trybie rozgłaszają swoją chęć, aby zostać zsynchronizowanym przez komputer z sieci. Wysyłane komunikaty są niezależne od dostępności komputera i jego warstwy;
- tryb 4 - serwera – połączenie tego typu nawiązywane jest po otrzymaniu komunikatu od hosta pracującego w trybie klienta. Host pracujący w tym trybie, zazwyczaj jest to serwer, służy do synchronizowania innych komputerów, najczęściej niższych warstw;
- tryb 5 - rozsyłania – w tym trybie serwer lub grupa serwerów rozsyła okresowo sygnał czasu do komputerów niższych warstw. Metoda ta wykorzystywana jest w szybkich sieciach, a charakteryzuje się stosunkowo małą dokładnością.

Tryby 0, 6 oraz 7 są trybami zarezerwowanymi.

Żaden z tych algorytmów nie gwarantuje niezawodnego dostarczania komunikatów przesyłanych za pomocą protokołu UDP. W trybach symetrycznych, klienta oraz serwera komunikaty wymieniane są parami. Każdy wysłany komunikat zawiera lokalny czas wysłania oraz odbioru poprzedniego komunikatu, a także lokalny czas wysłania komunikatu bieżącego. Odbiorca zapisuje czas przyjścia sygnału. W obu trybach komunikaty mogą ginać, może też nastąpić pewne opóźnienie pomiędzy nadejściem komunikatu a wysłaniem kolejnego [7].

Protokół NTP oblicza opóźnienie na podstawie łącznego czasu transmisji dla każdej pary komunikatów. Serwery NTP stosują algorytm filtrowania danych, który szacuje opóźnienie i oblicza dokładność tego oszacowania. Dokładność ta nazywana jest dyspersją filtru [1]. Im większa dyspersja, tym dane są mniej dokładne, a tym samym mniej pewne.

Opóźnienie oraz dokładność jego oszacowania tworzą pary, z których osiem ostatnich zostaje zachowanych.

Serwer NTP komunikuje się z kilkoma równorzędnymi partnerami. Przy tej komunikacji protokół NTP stosuje algorytm wyboru partnera. W algorytmie tym analizowane są wyniki wymiany komunikatów z każdym z partnerów i na tej podstawie określany jest najlepszy z kooperujących serwerów. W wyniku tej metody dany serwer może dowolnie zmieniać partnerów.

Partnerzy z warstw bliższych źródła są częściej wybierani z powodu mniejszego błędu. Innym elementem brany pod uwagę przy wyborze partnera jest tzw. współczynnik dyspersji synchronizacji, czyli suma dyspersji filtrów mierzonych pomiędzy danym serwerem a korzeniem podsieci synchronizacji. Elementy tej sumy wchodzi w skład komunikatów, więc jej obliczenie nie przedstawia problemu. Preferowani są partnerzy o najniższej dyspersji synchronizacji [1].

Z badań eksperymentalnych wynika, że błąd synchronizacji jest mniejszy niż 30 [ms]. W celu zwiększenia dokładności protokół NTP stosuje tzw. pętlę synchronizacji [1], czyli

modyfikowanie częstotliwości lokalnych zegarów na podstawie obserwowanych odchyleń jego wskazań.

Z pętlą synchronizacji związany jest pewien mechanizm zwany NCO (Numeric-Controlled Oscylator). Jest to rodzaj licznika stosowanego po to, aby niwelować duże odchyłki czasu.

### 3. Model Lamporta

Wszystkie metody synchronizacji czasu obarczone są pewnym błędem. W związku z tym w sieciach komputerowych niemożliwe jest jednoznaczne ustalenie porządku dowolnej liczby zdarzeń [4], co ma bardzo duże znaczenie w zastosowaniach rozproszonych.

Lamport [4] zaproponował model służący do porządkowania zdarzeń oparty na przyczynowości fizycznej. Metoda ta opiera się na dwóch założeniach:

- kolejność zdarzeń występujących w jednym procesie odpowiada ich zaobserwowaniu,
- dla każdego komunikatu przesyłanego siecią zdarzenie jego wysłania poprzedza zdarzenie odbioru.

Model zaproponowany przez Lamporta nazywa się relacją uprzedniości zdarzeń, relacją uporządkowania przyczynowego lub relacją potencjalnego uporządkowania przyczynowego.

Niech  $x \xrightarrow{p} y$  oznacza fakt, że zdarzenia  $x$  oraz  $y$  wystąpiły w jednym procesie  $p$ , a  $x$  poprzedza  $y$ . Pomiedzy zdarzeniami  $x$  oraz  $y$  zachodzi relacja uporządkowania przyczynowego, jeżeli spełnione są następujące warunki:

$$\begin{aligned} \exists_p^p : x \rightarrow y &\Rightarrow x \rightarrow y \\ \forall_k : send(k) \rightarrow receive(k) & \\ x \rightarrow y \wedge y \rightarrow z &\Rightarrow x \rightarrow z \end{aligned} \quad (7)$$

Gdzie symbol  $x \rightarrow y$  oznacza relację uporządkowania przyczynowego,  $k$  – komunikat,  $z$  – zdarzenie,  $send(k)$  jest zdarzeniem wysłania komunikatu  $k$ , a  $receive(k)$  jest zdarzeniem odbioru tego komunikatu.

Relacja uporządkowania przyczynowego nie występuje pomiędzy zdarzeniami zachodzącymi w różnych procesach, które nie są powiązane ze sobą komunikatami. Takie zdarzenia nazywa się współbieżnymi i oznacza symbolem  $x || y$ .

## 4. Zegary logiczne

Zegar logiczny to mechanizm wymyślony przez Lamporta, porządkujący zdarzenia wg ich następowania. Jest on rosnącym licznikiem przypisanym do każdego procesu, a jego wartość nie musi mieć żadnego związku z zegarem fizycznym.

Aby relacja uprzedniości była spełniona, wartości zegarów logicznych muszą być uaktualniane i przesyłane wg następujących reguł [4]:

- przed wystąpieniem każdego zdarzenia w procesie licznik jest zwiększany o pewną stałą wartość,
- komunikat zawiera w sobie wartość zegara logicznego,
- proces  $q$  odbierający taki komunikat uaktualnia swój zegar logiczny wg reguły:  $Z = \max(Z_p, Z_q)$ , gdzie  $Z_q$  to wartość zegara procesu  $q$ , a  $Z_p$  jest wartością zegara procesu  $p$  wysyłającego komunikat. Następnie zegar  $Z$  jest zwiększany o uprzednio zdefiniowaną stałą.

Metoda zegarów logicznych pozwala na dokładne określenie kolejności zdarzeń w procesach.

## 5. Podsumowanie

Precyzyjne określanie czasu w sieciach komputerowych jest bardzo ważnym procesem, dlatego prowadzonych jest wiele badań, które mają za zadanie zwiększenie dokładności synchronizacji oraz minimalizację błędów.

Algorytmy synchronizacji czasu są nieustannie rozwijane, polepszają swoje parametry jakościowe oraz wydajnościowe. Przykładem może być protokół NTP.

Osiągany w praktyce stopień dokładności spełnia wiele wymagań, mimo to nie zawsze jest wystarczający do określenia porządku zdarzeń na różnych komputerach. Niektóre algorytmy wymagają dokładnego określenia kolejności zdarzeń. Relacja uprzedniości zdarzeń porządkuje je badając przepływ komunikatów pomiędzy nimi. W celu odzwierciedlenia kolejności zdarzeń stosuje się liczniki zwane zegarami logicznymi.

## LITERATURA

1. Coulouris G., Dollimore J., Kindberg T.: Systemy rozproszone, podstawy i projektowanie. Wydawnictwo Naukowo-Techniczne, Warszawa 1998.



2. Cristian F.: A probabilistic approach to distributed clock synchronization. Proc. Ninth IEEE International Conference on Distributed Computing Systems, 1989, p. 288 - 296.
3. Gusella R., Zatti S.: The accuracy of the clock synchronization achieved by TEMPO in Berkeley UNIX 4.3 BSD. IEEE Trans. on Software Engineering, 1989, 15, 7, p. 847 – 853.
4. Lamport L.: Time, clocks and the ordering of events in distributed systems. Comm. ACM, 1978, 21, p. 558 – 565.
5. Lamport L., Melliar-Smith P. M.: Synchronizing clocks in the presence of faults. JACM, 1985, 32, p. 52 – 78.
6. Mills D. L.: Algorithms for synchronizing network clocks. DARPA Network Working Group Report RFC - 956, 1985.
7. Mills D. L.: Internet Time Synchronization: the Network Time Protocol. DARPA Network Working Group Report RFC - 1129, 1989.
8. Mills D. L.: On the accuracy and stability of clocks synchronized by Network Time Protocol in the Internet system. ACM Computer Communication Review, 1990, 20, p. 65 – 75.
9. Mills D. L.: Network Time Protocol (Version 3) Specification, Implementation and Analysis. DARPA Network Working Group Report RFC - 1305, 1992.
10. Marzullo K., Owicki S.: Maintaining the time in distributed system. ACM Operating Systems, 1985, 19, 44 – 54.

Recenzent: Dr inż Ryszard Winiarczyk

Wpłynęło do Redakcji 12 lutego 2004 r.

### **Abstract**

Computer clock is a set of software and hardware components used to provide a time for computers. In computer networks and distributed systems time is a very important element. Correct work of many algorithms (bank systems, internet shops) depends on very accurate time measurement. There is a group of algorithms that need all events and processes to occur in proper order.

Article gives a brief survey of time synchronization methods used in computer networks. The paper describes use of Universal Time Coordinated to compensate of clocks deviation,

Cristian method, Berkeley algorithm, Network Time Protocol, happened-before relation (casual ordering, potential casual ordering), and logical clocks.

Very precise time - UTC is an international standard based on atomic time. UTC signal is regularly synchronized and broadcasted by ground and satellite stations [1]. Time servers synchronized with the help of UTC time constitute a main layer of NTP based networks. Cristian has proposed a method in which time servers receive signals from UTS sets, too. In this algorithm server sends its time to all clients.

In Berkeley algorithm clients called “slaves” send their times to server. Server called “master” sends back only positive or negative values to be added to local time.

Network Time Protocol is widely deployed international standard in the Internet [7]. It automatically organizes and maintains the time synchronization subnet itself. NTP uses algorithms like clocks filtering, selection, combining, certification, etc. to improve its performances.

In many cases precise time measurement is not indispensable but acquaintance of sequence of events is very important. Happened-before relation and logical clocks proposed by Lamport [4] are mechanisms providing information about events sequence.

## **Adres**

Marcin WOCH: Politechnika Śląska, Instytut Informatyki, ul. Akademicka 16,  
44-101 Gliwice, Polska, [marcinwoch@wp.pl](mailto:marcinwoch@wp.pl).