



WIKIPEDIA
The Free Encyclopedia

[Main page](#)
[Contents](#)
[Featured content](#)
[Current events](#)
[Random article](#)
[Donate to Wikipedia](#)
[Wikipedia store](#)

Interaction

[Help](#)
[About Wikipedia](#)
[Community portal](#)
[Recent changes](#)
[Contact page](#)

Tools

[What links here](#)
[Related changes](#)
[Upload file](#)
[Special pages](#)
[Permanent link](#)
[Page information](#)
[Wikidata item](#)
[Cite this page](#)

Print/export

[Create a book](#)
[Download as PDF](#)
[Printable version](#)

Languages

[Deutsch](#)
[فارسی](#)
[Português](#)

 [Edit links](#)

[Create account](#) [Log in](#)

Article [Talk](#)

[Read](#) [Edit](#) [View history](#)

Berkeley algorithm

From Wikipedia, the free encyclopedia

The **Berkeley algorithm** is a method of [clock synchronisation](#) in [distributed computing](#) which assumes no machine has an accurate time source. It was developed by Gusella and Zatti at the University of California, Berkeley in 1989 ^[1] and like [Cristian's algorithm](#) is intended for use within [intranets](#).

The algorithm [\[edit\]](#)

Unlike [Cristian's algorithm](#), the server process in the Berkeley algorithm, called the *master*, periodically polls other *slave* processes. Generally speaking, the algorithm is:

1. A *master* is chosen via an [election process](#) such as [Chang and Roberts algorithm](#).
2. The *master* polls the *slaves* who reply with their time in a similar way to [Cristian's algorithm](#).
3. The *master* observes the [round-trip time](#) (RTT) of the messages and estimates the time of each *slave* and its own.
4. The *master* then averages the clock times, ignoring any values it receives far outside the values of the others.
5. Instead of sending the updated current time back to the other process, the *master* then sends out the amount (positive or negative) that each *slave* must adjust its clock. This avoids further uncertainty due to RTT at the *slave* processes.

With this method the average cancels out individual clock's tendencies to drift. Gusella and Zatti released results involving 15 computers whose clocks were synchronised to within about 20-25 milliseconds using their protocol.

Computer systems normally avoid rewinding their clock when they receive a negative clock alteration from the master. Doing so would break the property of monotonic time, which is a fundamental assumption in certain algorithms in the system itself or in programs such as [make](#). A simple solution to this problem is to halt the clock for the duration specified by the master, but this simplistic solution can also cause problems, although they are less severe. For minor corrections, most systems slow the clock (known as "clock slew"), applying the correction over a longer period of time.

References [\[edit\]](#)

- ↑ Gusella, R.; Zatti, S. (1989), "The accuracy of the clock synchronization achieved by TEMPO in Berkeley UNIX 4.3BSD" [↗](#), *Software Engineering, IEEE Transactions on* (IEEE) **15** (7): 847–853, doi:10.1109/32.29484 [↗](#)

Categories: [Distributed algorithms](#)

This page was last modified on 2 March 2015, at 17:21.

Text is available under the [Creative Commons Attribution-ShareAlike License](#); additional terms may apply. By using this site, you agree to the [Terms of Use](#) and [Privacy Policy](#). Wikipedia® is a registered trademark of the [Wikimedia Foundation, Inc.](#), a non-profit organization.

[Privacy policy](#) [About Wikipedia](#) [Disclaimers](#) [Contact Wikipedia](#) [Developers](#) [Mobile view](#)

