



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 
 [Add links](#)

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

Article [Talk](#)

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



Intersection algorithm

From Wikipedia, the free encyclopedia

The **intersection algorithm** is an **agreement algorithm** used to select sources for estimating accurate time from a number of **noisy** time sources, it forms part of the modern **Network Time Protocol**. It is a modified form of **Marzullo's algorithm**.^{[1][2]}

While Marzullo's algorithm will return the smallest interval consistent with the largest number of sources, the returned interval does not necessarily include the center point (calculated offset) of all the sources in the intersection. The Intersection algorithm returns an interval that includes that returned by Marzullo's algorithm but may be larger since it will include the center points. This larger interval allows using additional statistical data to select a point within the interval, reducing the **jitter** in repeated execution.

Method [\[edit\]](#)


Given *M* intervals of the form $c \pm r$ (which means $[c-r, c+r]$), the algorithm seeks to find an interval with *M*−*f* sources. The value *f* is referred to as the number of falsetickers, those sources which are in error (the actual value is outside the **confidence band**). The best estimate is that which assumes the least number of falsetickers, *f*. The results will be considered valid if $f < M/2$, otherwise the algorithm will return failure instead of an interval.

The intersection algorithm begins by creating a table of tuples <offset,type>. For each interval there are three entries: the lower endpoint, the midpoint and the upper endpoint, labelled with types −1, 0 and +1 respectively. Thus the interval $c \pm r$ results in the entries <*c*−*r*,−1>, <*c*,0> and <*c*+*r*,+1>. These entries are then sorted by offset.

Variables: This algorithm uses *f* as number of false tickers, *endcount* and *midcount* are integers. *Lower* and *upper* are values of offsets.

- [initialize best f] Start with *f*=0, assuming all input intervals are valid. Each time no interval is found *f* will be incremented until either an interval is found or $f \geq M/2$.
- [initialize] *endcount*=0 and *midcount*=0.
- [find lower endpoint] Start at beginning of the list (lowest offset) consider each tuple in order. *endcount* = *endcount*−*type*. If *endcount* ≥ *M*−*f* then *lower* = *offset* and goto step 3 because the (possible) lower endpoint has been found. If the *type* = 0 then *midcount* = *midcount*+1. Repeat with next tuple. If reach end of list then goto step 6.
- [tentative lower endpoint found, initialize to find upper endpoint] set *endcount*=0.
- [determine number of midpoints] Start from end of list and work towards lower offsets. *endcount* = *endcount*+*type*. If *endcount* ≥ *M*−*f* then *upper* = *offset*, goto step 5. If *type* = 0 then *midcount* = *midcount*+1. Repeat for next tuple. If reach end of list then goto step 6.
- if *lower* ≤ *upper* and *midcount* ≤ *f* then return interval [*lowerendpoint*,*upperendpoint*] as resulting confidence interval.
- [increment number of falsetickers] *f* = *f*+1. If $f \geq M/2$ then terminate and return FAILED, otherwise goto step 1.

References [\[edit\]](#)

- ↑ "RFC 1305 - Network Time Protocol (Version 3) Specification, Implementation and Analysis" . *tools.ietf.org*. 2013. Retrieved October 6, 2013. "Digital Time Service Functional Specification Version T.1.0.5. Digital Equipment Corporation, 1989."
- ↑ Digital Time Service Functional Specification Version T.1.0.5. Digital Equipment Corporation, 1989.

Categories: [Agreement algorithms](#)

This page was last modified on 5 May 2014, at 09:54.

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](#)



