

TCP Connection Life

Asked 16 years, 2 months ago Modified 15 years, 11 months ago

Viewed 31k times



How long can I expect a client/server TCP connection to last in the wild?

28



I want it to stay permanently connected, but things happen, so the client will have to reconnect. At what point do I say that there's a problem in the code rather than there's a problem with some external equipment?



tcp

Share

Improve this question

Follow

asked Oct 1, 2008 at 17:08



Robert Deml

12.5k ● 21 ● 69 ● 92

3 forever if you have the correct KeepAlive interval, related: stackoverflow.com/questions/3907537/... – markmnl Mar 1, 2011 at 7:36 ✎

What people answered in 1999:

ask.slashdot.org/story/99/09/19/2150236/... – user3041714 Jul 1, 2016 at 22:30 ✎

6 Answers

Sorted by:

Highest score (default)





19



I agree with Zan Lynx. There's no guarantee, but you can keep a connection alive almost indefinitely by sending data over it, assuming there are no connectivity or bandwidth issues.

Generally I've gone for the application level keep-alive approach, although this has usually been in the client spec so I've had to do it. But just send some short piece of data every minute or two, to which you expect some sort of acknowledgement.

Whether you count one failure to acknowledge as the connection having failed is up to you. Generally this is what I have done in the past, although there was a case I had wait for three failed responses in a row to drop the connection because the app at the other end of the connection was extremely flaky about responding to "are you there?" requests.

If the connection fails, which at some point it probably will, even with machines on the same network, then just try to reestablish it. If that fails a set number of times then you have a problem. If your connection persistently fails after it's been connected for a while then again, you have a problem. Most likely in both cases it's probably some network issue, rather than your code, or maybe a problem with the TCP/IP stack on your machine (has been known: I encountered issues with this on an old version of QNX--it'd just randomly fall over). Having said that you might have a software problem, and the only way to know for sure is often to attach a debugger, or to get

some logging in there. E.g. if you can always connect successfully, but after a time you stop getting ACKs, even after reconnect, then maybe your server is deadlocking, or getting stuck in a loop or something.

What's really useful is to set up a series of long-running tests under a variety of load conditions, from just sending the keep alive are you there?/ack requests and responses, to absolutely battering the server. This will generally give you more confidence about your software components, and can be really useful in shaking out some really weird problems which won't necessarily cause a problem with your connection, although they might result in problems with the transactions taking place. For example, I was once writing a telecoms application server that provided services such as number translation, and we'd just leave it running for days at a time. The thing was that when Saturday came round, for the whole day, it would reject every call request that came in, which amounted to millions of calls, and we had no idea why. It turned out to be because of a single typo in some date conversion code that only caused a problem on Saturdays.

Hope that helps.

[Share](#) [Improve this answer](#)

[edited Oct 1, 2008 at 17:41](#)

[Follow](#)

[answered Oct 1, 2008 at 17:32](#)



Bart Read

2,757 ● 3 ● 22 ● 32



17



I think the most important idea here is theory vs. practice.

The original theory was that the connections had no lifetimes. If you had a connection, it stayed open forever, even if there was no traffic, until an event caused it to close.

The new theory is that most OS releases have turned on the keep-alive timer. This means that connections will last forever, as long as the system on the other end responds to an occasional TCP-level exchange.

In reality, many connections will be terminated after time, with a variety of criteria and situations.

Two really good examples are: The remote client is using DHCP, the lease expires, and the IP address changes.

Another example is firewalls, which seem to be increasingly intelligent, and can identify keep-alive traffic vs. real data, and close connections based on any high level criteria, especially idle time.

How you want to implement reconnect logic depends a lot on your architecture, the working environment, and your performance goals.

answered [Jan 7, 2009 at 7:39](#)

Share Improve this answer

Follow

community wiki
[benc](#)



It shouldn't really matter, you should design your code to automatically reconnect if that is the desired behavior.

9

Share Improve this answer

answered Oct 1, 2008 at 17:10



Follow



[GEOCHET](#)

21.3k ● 15 ● 77 ● 99



There really is no way to tell. There is nothing inherent to TCP that would cause the connection to just drop after a certain amount of time. Someone on a reliable connection could have years of uptime, while someone on a different connection could have to reconnect every 5 minutes. There is no way to tell or even guess.

6



Share Improve this answer

answered Oct 1, 2008 at 17:13

Follow



[noah](#)

21.5k ● 17 ● 67 ● 88



You will need some data going over the connection periodically to keep it alive - many OS's or firewalls will drop an inactive connection.

2



Share Improve this answer

answered Oct 1, 2008 at 17:18

Follow



[Mark Ransom](#)

308k ● 44 ● 416 ● 647



-2

Pick a value. One drop every hour is probably fine. Ten unexpected connection drops in 5 minutes probably indicates a problem.



TCP connections will generally last about two hours without any traffic. Either end can send keep-alive packets, which are, I think, just an ACK on the last received packet. This can usually be set per socket or by default on every TCP connection.



An application level keep-alive is also possible. For a telnet style protocol like FTP, SMTP, POP or IMAP something like sending return, newline and getting back a command prompt.

Share Improve this answer

answered Oct 1, 2008 at 17:18

Follow



[Zan Lynx](#)

54.3k ● 11 ● 81 ● 134

TCP keepalive is a timer that varies by OS, so the 2 hours might vary in specific environments. – [benc](#) Jan 7, 2009 at 7:21

- 1 I realize that this is old, but since it hasn't been deleted, I'd like to point out that it is incorrect. Connections will not "generally last about two hours without any traffic." They will remain connected forever unless an endpoint goes away (is

rebooted, for example) or an intervening device (firewall) "forgets" the connection. Even then, it will appear to be up until an endpoint attempts to send data, at which point it will be RST by the *firewall*. – [David Hoelzer](#) Sep 27, 2018 at 18:21

- 1 @DavidHoelzer: At the time, it was my experience that *something* was resetting the connections every two hours. NAT, firewall, random DSL bridge resets, PPPoE reconnects, DHCP address changes, who knows what. The point of my answer was that TCP drops after a couple of hours can be expected to be normal operation of the Internet. – [Zan Lynx](#) Sep 27, 2018 at 18:45
-