Re: How do I alter values for TCP timeouts (SRTT & RTO)

Source:
http://www.tech−archive.net/Archive/Win2000/microsoft.public.win2000.general/2005−12/msg00181.html

---

- *From*: "hotdogdave" <hotdogdave@xxxxxxxxxxxxxxxxxxxxxxxxx>
- *Date*: Mon, 5 Dec 2005 08:56:03 −0800

Thanks for your reply Dan

Well recompiling is a bit above my level if that is what is required and
drifts out of scope of our task to get 'out of the box' IP apps to work over
HF albeit with a few tweaks. I consider altering registry settings to fall
within the scope.

I would be disappointed if these values cannot be set somewhere as settings
such as TcpInitialRtt , and TcpMaxDataRetransmissions can be.

These values all refer to RFC 793 and all these values are talked about in
suggested ranges − leading me to assume that they can be altered. Possibly MS
has not implemented the RFC with non−alterable default values while say,
Linux has.

Your right in saying there may be times when the radio cannot simply make a
link hence why some maximum timeout is required. What I'm finding is one node
will have a quick TCP response and dynamically alter its RTT to say 25 secs.
The next time this is not long enough and it will retransmit (half duplex)
over the incoming response. The system generally works 90% but occasionally
gets its 'knickers in a knot'. The data wont get through until a TCP protocol
attempts to retransmit (eg SMTP failed mail retry).

There are 2 ways I can think of to overcome this, set LBOUND to a min, or
give less weight of SRTT to the last measured RTT. This algorithm is given
from one of the links below;

Windows 2000 TCP Retransmission Behavior
TCP starts a retransmission timer when each TCP segment is sent and then
waits for an acknowledgment to come back before the retransmission timer
expires. This technique is the Positive Acknowledgment Retransmission (PAR)
scheme.

In Windows 2000 every new TCP connection request uses an initial
retransmission timer value that is set to three seconds. This value can be
changed using the TcpInitialRtt Registry parameter. If a TCP connection
cannot be established, it is retried TcpMaxDataRetransmissions times (the

default is two times).

For established TCP connections, the number of retransmissions is controlled by the TcpMaxDataRetransmissions Registry parameter (set to 5 by default). The retransmission timeout is adjusted dynamically as per RFC 793,"TCP DARPA Internet Program Protocol Specification" to adjust for the delay characteristics of the connection. A Smoothed Round Trip Time (SRTT) calculation is used as described in the following sidebar.

In some situations TCP retransmits before the retransmission timer expires. This situation commonly occurs when TCP implements a fast retransmit feature, described in RFC 2581, "TCP Congestion Control" and explained in the following sidebar.

Calculating Smoothed Round Trip Time (SRTT)

Measure the elapsed time between sending a data octet with a particular sequence number and receiving an acknowledgment that covers that sequence number. This measured elapsed time is the Round Trip Time (RTT). Based on the RTT, compute a Smoothed Round Trip Time (SRTT) as

SRTT = ([alp] * SRTT) + ((1−[alp]) * RTT)

[alp] is a weighting factor whose value is between 0 and 1 (0 <= [alp] < 1). The SRTT is an estimated timeout. The SRTT computation, therefore, takes into account its old SRTT value and the new RTT value and takes a weighted average of these two. When [alp] = 0.5, SRTT is just the simple average of old SRTT and the RTT values:

SRTT = (SRTT + RTT)/2 when [alp] = 0.5

When [alp] is less than 0.5, a greater weight is given to the actual Round Trip Time (RTT). When [alp] is greater than 0.5, a greater weight is given to the previous estimated Round Trip Time (SRTT).

When [alp] = 0 or almost 1, you have the two extremes. When [alp] = 0, SRTT is always set to RTT; and when [alp] is almost 1, a fixed SRTT is used with little deference to the actual RTT that is implied. Needless to say, the two extremes are seldom used in actual practice but are useful for understanding the nature of the weighted average.

The SRTT is then used to compute the retransmission timeout (RTO) using the following:

RTO = min[UBOUND,max[LBOUND,([beta]*SRTT)]]

UBOUND is an upper bound on the timeout (for example, one minute), and LBOUND is a lower bound on the timeout (for example, one second). These

values ensure that the RTO will fall between LBOUND and UBOUND. [beta] is a
delay variance factor to change the sensitivity to the delay. If [beta] is
set to 1, RTO will be set to SRTT unless the SRTT value is below LBOUND or
above UBOUND. This value ([beta] = 1) makes TCP sensitive to packet loss and
does not cause TCP to wait for a long time before retransmitting. However,
small delays might cause unnecessary retransmissions. To make TCP less
susceptible to small delays, a larger [beta] value might be used. RFC 793
suggests using a value from 1.3 to 2.0.

above extract from
http://www.informit.com/articles/article.asp?p=130716&seqNum=4&rl=1

"Dan Seur" wrote:

> Not positive, but I believe you'll discover that lbound & ubound are set
> in the source code of whatever TCP you're using, and resetting would
> involve a recompile of that module. Not highlevel stuff, likely not in
> registry.
>
> Further I believe the max (ubound) setting is dynamically adjusted
> according to ongoing retransmit durations, and in most implementations
> can exceed your 35 second ping by close to a minute and a half.
>
> Is it possible your radio link is occasionally completely blocked for
> periods exceedingf a few hundred seconds? Jammed? Hdw/sftw glitch?
>
> As I say, cum grano salis etc.
>
> hotdogdave wrote:
> > Hi
> >
> > Can anyone please help?! :)
> >
> > I am working with a Win2000 IP network over a HF radio system with high
> > latencies (ping = 35 secs). Putting the jokes aside of 'why bother'!, we
> > occasionally get a TCP timeout occurring. I know this is because the reply
> > takes longer than the TcpTimeout setting. The MS info on this is at
> >
> >
http://www.microsoft.com/resources/documentation/Windows/2000/server/reskit/en−us/Default.asp?url=/resources/do
> >
> > I have changed the TcpInitialRtt value in the registry but this is an
> > initial value to start the Smoothed RTT (SRTT) which is then dynamically
> > calculated
> > using a given formula discussed in
> >
> > http://www.informit.com/articles/article.asp?p=130716&seqNum=4&rl=1
> >
> > SRTT is calculated by values such as [alp], [beta]. Retransmit TimeOut (RTO)
> > is then calculated from SRTT and also has values such as upper and lower
> > bounds (UBOUND, LBOUND). I have done a lot of searching and can find lots of

> > information on what I have discussed above, but nothing about where I can set
> > the values above. The articles indicate that these values may be altered. I
> > assume these are set in the registry, like TcpInitialRTT, but I cant find out
> > where.
> >
> > cheers David
> > sunny England
> >
> >
> > −−−
> > avast! Antivirus: Inbound message clean.
> > Virus Database (VPS): 0549−1, 12/05/2005
> > Tested on: 12/5/2005 10:45:17 AM
> > avast! − copyright (c) 1988−2004 ALWIL Software.
> > http://www.avast.com
> >
> >
> >
>
>
>
> −−−
> avast! Antivirus: Outbound message clean.
> Virus Database (VPS): 0549−1, 12/05/2005
> Tested on: 12/5/2005 11:25:28 AM
> avast! − copyright (c) 1988−2004 ALWIL Software.
> http://www.avast.com
>
>
>
>
.

- ***Follow−Ups***:
  - ***Re: How do I alter values for TCP timeouts (SRTT & RTO)***
    - ◊ *From:* Dan Seur

- ***References***:
  - ***Re: How do I alter values for TCP timeouts (SRTT & RTO)***
    - ◊ *From:* Dan Seur

- Prev by Date: ***Re: How do I alter values for TCP timeouts (SRTT & RTO)***
- Next by Date: ***moving from a single server to multiple servers***
- Previous by thread: ***Re: How do I alter values for TCP timeouts (SRTT & RTO)***
- Next by thread: ***Re: How do I alter values for TCP timeouts (SRTT & RTO)***
- Index(es):
  - ***Date***
  - ***Thread***