

## Re: Automatic TCP send socket buffer sizing

---

*Source:* <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/net/2006-11/msg00211.html>

---

- *From:* Phil Rosenthal <[pr@xxxxxxxxxxx](mailto:pr@xxxxxxxxxxx)>
  - *Date:* Tue, 21 Nov 2006 16:20:17 -0500
- 

Hello,

I've tested this out and had a few comments.

- 1) I've seen in production that some sockets get large very quickly during periods of high latency (eg: when sending to a user downloading from a cablemodem and their headend gets temporarily saturated and has large buffers, which raises the RTT under saturation, which increases the bandwidth delay product), but then as there isn't any code to shrink the buffers. This would probably need to be in the timers to notice the case of the sender temporarily stopping sending – eg in a keepalive http socket (a separate, but related issue).
- 2) In our code, and I believe in other code, we watch the tcp send buffer drain before closing the sockets, which in our implementation works by checking with kqueue how many bytes we can send to that socket. If the buffer size keeps changing, we would now have to call getsockopt() periodically to get the current size, which would obviously have a polling-type performance problem. Is it possible to make a kqueue hook to let us know when the socket buffer size changes?
- 3) There is a potential for a DoS here as you could have a large number of clients simulate a very large bandwidth delay product (eg: ipfw delay 100,000 ms) and start up a lot of downloads. This could cause your total socket buffers to rapidly grow much larger than normal and eat up all the available kernel memory. Perhaps having another tunable for a soft limit on total socket size past which we get more conservative on buffer sizing, and reduce the size of the largest buffers in exchange for making the smallest buffers larger.
- 4) What happens if you call setsockopt() to set the size? Perhaps we can have a new setsockopt() flag to disable automatic sizing on a socket if we have a specific size we want a buffer to be and don't want it dynamically sized.

After some refining, this does have some very good potential to be very useful for us.

-P

---

freebsd-net@xxxxxxxxxxx mailing list  
<http://lists.freebsd.org/mailman/listinfo/freebsd-net>  
To unsubscribe, send any mail to "freebsd-net-unsubscribe@xxxxxxxxxxx"