

Re: read() returns ETIMEDOUT on steady TCP connection

Re: read() returns ETIMEDOUT on steady TCP connection

Source: <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/net/2008-04/msg00221.html>

- *From:* Mark Hills <mark@xxxxxxxxxxxx>
 - *Date:* Mon, 21 Apr 2008 07:51:16 +0100 (BST)
-

On Mon, 21 Apr 2008, Andre Oppermann wrote:

Mark Hills wrote:

On Sun, 20 Apr 2008, Peter Jeremy wrote:

I can't explain the problem but it definitely looks like a resource starvation issue within the kernel.

I've traced the source of the ETIMEDOUT within the kernel to `tcp_timer_rexmt()` in `tcp_timer.c`:

```
if (++tp->t_rxtshift > TCP_MAXRXTSHIFT) {
    tp->t_rxtshift = TCP_MAXRXTSHIFT;
    tcpstat.tcps_timeoutdrop++;
    tp = tcp_drop(tp, tp->t_softerror ?
    tp->t_softerror : ETIMEDOUT);
    goto out;
}
```

Yes, this is related to either lack of mbufs to create a segment or a problem in sending it. That may be full interface queue, a bandwidth manager (dummynet) or some firewall internally rejecting the segment (ipfw, pf). Do you run any firewall in stateful mode?

There's no firewall running.

I'm new to FreeBSD, but it seems to imply that it's reaching a limit of a number of retransmits of sending ACKs on the TCP connection receiving the inbound data? But I checked this using `tcpdump` on the server and could see

Re: read() returns ETIMEDOUT on steady TCP connection

Re: read() returns ETIMEDOUT on steady TCP connection

no retransmissions.

When you have internal problems the segment never makes it to the wire and thus you wont see it in tcpdump.

Please report the output of 'netstat -s -p tcp' and 'netstat -m'.

Posted below. You can see it it in there: "131 connections dropped by rexmit timeout"

As a test, I ran a simulation with the necessary changes to increase TCP_MAXRXTSHIFT (including adding appropriate entries to tcp_sync_backoff[] and tcp_backoff[]) and it appeared I was able to reduce the frequency of the problem occurring, but not to a usable level.

Possible causes are timers that fire too early. Resource starvation (you are doing a lot of traffic). Or of course some bug in the code.

As I said in my original email, the data transfer doesn't stop or splutter, it's simply cut mid-flow. Sounds like something happening prematurely.

Thanks for the help,

Mark

```
$ netstat -m
14632/8543/23175 mbufs in use (current/cache/total)
504/4036/4540/25600 mbuf clusters in use (current/cache/total/max)
504/3976 mbuf+clusters out of packet secondary zone in use (current/cache)
12550/250/12800/12800 4k (page size) jumbo clusters in use
(current/cache/total/max)
0/0/0/6400 9k jumbo clusters in use (current/cache/total/max)
0/0/0/3200 16k jumbo clusters in use (current/cache/total/max)
54866K/11207K/66073K bytes allocated to network (current/cache/total)
0/0/0 requests for mbufs denied (mbufs/clusters/mbuf+clusters)
0/0/0 requests for jumbo clusters denied (4k/9k/16k)
0/6/6656 sbufs in use (current/peak/max)
0 requests for sbufs denied
0 requests for sbufs delayed
0 requests for I/O initiated by sendfile
0 calls to protocol drain routines
```

```
$ netstat -s -p tcp
tcp:
3408601864 packets sent
```

Re: read() returns ETIMEDOUT on steady TCP connection

Re: read() returns ETIMEDOUT on steady TCP connection

3382078274 data packets (1431587515 bytes)
454189057 data packets (1209708476 bytes) retransmitted
14969051 data packets unnecessarily retransmitted
0 resends initiated by MTU discovery
2216740 ack-only packets (9863 delayed)
0 URG only packets
0 window probe packets
273815 window update packets
35946 control packets
2372591976 packets received
1991632669 acks (for 2122913190 bytes)
16032443 duplicate acks
0 acks for unsent data
1719033 packets (1781984933 bytes) received in-sequence
1404 completely duplicate packets (197042 bytes)
1 old duplicate packet
54 packets with some dup. data (6403 bytes duped)
9858 out-of-order packets (9314285 bytes)
0 packets (0 bytes) of data after window
0 window probes
363132176 window update packets
3 packets received after close
0 discarded for bad checksums
0 discarded for bad header offset fields
0 discarded because packet too short
635 discarded due to memory problems
39 connection requests
86333 connection accepts
0 bad connection attempts
2256 listen queue overflows
8557 ignored RSTs in the windows
86369 connections established (including accepts)
83380 connections closed (including 31174 drops)
74004 connections updated cached RTT on close
74612 connections updated cached RTT variance on close
74591 connections updated cached ssthresh on close
3 embryonic connections dropped
1979184038 segments updated rtt (of 1729113221 attempts)
110108313 retransmit timeouts
131 connections dropped by rexmit timeout
1 persist timeout
0 connections dropped by persist timeout
0 Connections (fin_wait_2) dropped because of timeout
23 keepalive timeouts
22 keepalive probes sent
1 connection dropped by keepalive
320 correct ACK header predictions
1638923 correct data packet header predictions
87976 syncache entries added
182 retransmitted
111 dupsyn

Re: read() returns ETIMEDOUT on steady TCP connection

Re: read() returns ETIMEDOUT on steady TCP connection

13061 dropped
86333 completed
8907 bucket overflow
0 cache overflow
1 reset
0 stale
2256 aborted
0 badack
0 unreach
0 zone failures
101037 cookies sent
9521 cookies received
36888 SACK recovery episodes
68139 segment rexmits in SACK recovery episodes
98390670 byte rexmits in SACK recovery episodes
243196 SACK options (SACK blocks) received
2853 SACK options (SACK blocks) sent
0 SACK scoreboard overflow

freebsd-net@xxxxxxxxxxx mailing list

<http://lists.freebsd.org/mailman/listinfo/freebsd-net>

To unsubscribe, send any mail to "freebsd-net-unsubscribe@xxxxxxxxxxx"