

netstat -s output: "packets pruned" and "packets collapsed"

netstat -s output: "packets pruned" and "packets collapsed"

Source: <http://linux.derkeiler.com/Newsgroups/comp.os.linux.networking/2007-07/msg00241.html>

- *From:* roybatty <tcp_rob@xxxxxxx>
 - *Date:* Fri, 20 Jul 2007 13:44:12 -0700
-

Hi All,

We seem to have networking related problems running some locally developed applications on a pair of identical Linux (2.6.18-1.2239.fc5) server systems. While trying to track down the source of these problems, I found some output from the "netstat -s" command that looks worrying e.g.

....

4087 packets pruned from receive queue because of socket buffer overrun

3827052 packets collapsed in receive queue due to low socket buffer

....

(complete sample output from netstat -s is included below).

These variables first begin to be reported when when our daemons (applications) are started and their values ramp up from then on (obviously :-)).

Despite some pretty intensive research I have found practically zero information about these variables. Does anyone out there have some tips for me on what these actually mean? For example, I notice that they are both, per their names, explicitly socket related. This lead me to try increasing the size of some of the system networking buffers e.g.

/proc/sys/net/ipv4/tcp_rmem default value = 4096 87380 4194304

/proc/sys/net/ipv4/tcp_wmem default value = 4096 16384 4194304

And:

/proc/sys/net/core/rmem_max default size = 131071

/proc/sys/net/core/wmem_max default size = 131071

I doubled these sizes and then doubled them again, without any apparent improvement in the rate at which the above errors were occurring. Note though that "tcp_moderate_rcvbuf" is also set to 1 on these systems which, as I understand it, means buffer sizes are auto-tuned (could it be then that my changes had no effect - is it necessary to turn this auto-tuning off, before changing the above

netstat -s output: "packets pruned" and "packets collapsed"

netstat -s output: "packets pruned" and "packets collapsed"

sizes manually?)

I also tried a different approach of greping the source tree for any calls to "setsockopt" looking for, for example, small buffer sizes being set, but none were apparent (I did find options such as: SO_SNDTIMEO, SO_RCVTIMEO and TCP_NODELAY being set).

At this point I am really just assuming that these values are not normal and are in fact associated with our actual problems. Please take a look at the output below and let me know what you think.

Thanks in advance!

Yours,
Robb.

```
+-----+
| "I've seen things you people wouldn't believe. |
| Attack ships on fire off the shoulder of Orion. I watched |
| C-Beams glitter in the dark near the Tannhauser Gate. All |
| those moments will be lost in time, like tears in rain. |
| Time to die." -- Roy Batty, Nexus6, N6MAA10816, Combat |
+-----+
```

```
Ip:
4097694726 total packets received
155085 with invalid addresses
2492354975 forwarded
0 incoming packets discarded
1605183310 incoming packets delivered
4042938962 requests sent out
Icmp:
38331 ICMP messages received
9 input ICMP message failed.
ICMP input histogram:
destination unreachable: 36
echo requests: 25554
echo replies: 12741
26659 ICMP messages sent
0 ICMP messages failed
ICMP output histogram:
destination unreachable: 1105
echo replies: 25554
Tcp:
362332 active connections openings
404536 passive connection openings
1580 failed connection attempts
7458 connection resets received
187 connections established
1600328035 segments received
1550250974 segments send out
```

netstat -s output: "packets pruned" and "packets collapsed"

netstat -s output: "packets pruned" and "packets collapsed"

260697 segments retransmitted
0 bad segments received.
7298 resets sent
Udp:
66672 packets received
1080 packets to unknown port received.
0 packet receive errors
66988 packets sent
TcpExt:
5354 invalid SYN cookies received
613 resets received for embryonic SYN_RECV sockets
4087 packets pruned from receive queue because of socket buffer
overrun
398014 TCP sockets finished time wait in fast timer
158 time wait sockets recycled by time stamp
39735643 delayed acks sent
544 delayed acks further delayed because of locked socket
Quick ack mode was activated 3938 times
547091 times the listen queue of a socket overflowed
547091 SYNs to LISTEN sockets ignored
10641526 packets directly queued to rcvmsg prequeue.
19159885 packets directly received from backlog
1523726509 packets directly received from prequeue
1070393298 packets header predicted
8431813 packets header predicted and directly queued to user
14447834 acknowledgments not containing data received
817036278 predicted acknowledgments
111 times recovered from packet loss due to SACK data
3887 congestion windows recovered after partial ack
164 TCP data loss events
71 timeouts after SACK recovery
157 fast retransmits
27 forward retransmits
58 retransmits in slow start
259192 other TCP timeouts
2 sack retransmits failed
3827052 packets collapsed in receive queue due to low socket
buffer
36049 DSACKs sent for old packets
28 DSACKs sent for out of order packets
4 DSACKs received
1852 connections reset due to unexpected data
894 connections reset due to early user close
2 connections aborted due to timeout