

# Re: [RFC][PATCH 0/9] Network receive deadlock prevention for NBD

---

*Source:* <http://linux.derkeiler.com/Mailing-Lists/Kernel/2006-08/msg04293.html>

---

- *From:* Rik van Riel <[riel@xxxxxxxxxx](mailto:riel@xxxxxxxxxx)>
  - *Date:* Fri, 11 Aug 2006 23:42:50 -0400
- 

Peter Zijlstra wrote:

You say "critical resource isolation", but it is not the case – consider NFS over UDP – remote side will not stop sending just because receiving socket code drops data due to OOM, or IPsec or compression, which can require reallocation. There is no "critical resource isolation", since reserved pool `_must_` be used by everyone in the kernel network stack.

The idea is to drop all !NFS packets (or even more specific only keep those NFS packets that belong to the critical mount), and everybody doing critical IO over layered networks like IPsec or other tunnel constructs asks for trouble – Just DON'T do that.

The only problem with things like IPsec is renegotiation, which can take up memory right at the time you don't have any extra memory available.

Decrypting individual IPsec packets during normal operation and then dropping the ones for non-critical sockets should work just fine.

The problem is layered networks over TCP, where you have to process the packets in-order and may have no choice but to hold onto data for non-critical sockets, at least for a while.

Dropping these non-essential packets makes sure the reserve memory doesn't get stuck in some random blocked user-space process, hence you can make progress.

In short:

- every incoming packet needs to be received at the packet level
- when memory is low, we only deliver data to memory critical sockets

Re: [RFC][PATCH 0/9] Network receive deadlock prevention for NBD

– packets to other sockets get dropped, so the memory can be reused for receiving other packets, including the packets needed for the memory critical sockets to make progress

Forwarding packets while in low memory mode should not be a problem at all, since forwarded packets get freed quickly.

The memory pool for receiving packets does not need much accounting of any kind, since every packet will end up coming from that pool when normal allocations start failing. May