

Re: [RFC,PATCH] loopback: calls netif_receive_skb() instead of netif_rx()

Re: [RFC,PATCH] loopback: calls netif_receive_skb() instead of netif_rx()

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2008-03/msg11925.html>

- *From:* Eric Dumazet <dada1@xxxxxxxxxxxxxx>
 - *Date:* Mon, 31 Mar 2008 12:01:16 +0200
-

Ingo Molnar a écrit :

* Eric Dumazet <dada1@xxxxxxxxxxxxxx> wrote:

I noticed some paths in kernel are very stack aggressive, and on i386 with CONFIG_4KSTACKS we were really in a dangerous land, even without my patch.

What we call 4K stacks is in fact 4K – sizeof(struct task_struct), so a little bit more than 2K. [...]

that's just wrong – 4K stacks on x86 are 4K–sizeof(thread_info) – the task struct is allocated elsewhere. The patch below runs just fine on 4K–stack x86.

Yes, this error was corrected by Andi already :)

Thank you Ingo but this patch was already suggested by me previously (<http://marc.info/?l=linux-netdev&m=120361996713007&w=2>) and was rejected, since we can very easily consume all stack space, especially with 4K stacks. (try with NFS mounts and XFS for example)

Only safe way is to check available free stack space, since we can nest loopback_xmit() several time. In case of protocol errors (like in TCP, if we answer to an ACK by another ACK, or ICMP loops), we would exhaust stack instead of delaying packets for next softirq run.

Problem is to check available space :

It depends on stack growing UP or DOWN, and depends on caller running on process stack, or softirq stack, or even hardirq stack.

Re: [RFC,PATCH] loopback: calls netif_receive_skb() instead of netif_rx()

Ingo

----->

Subject: net: loopback speedup
From: Ingo Molnar <mingo@xxxxxxx>
Date: Mon Mar 31 11:23:21 CEST 2008

Signed-off-by: Ingo Molnar <mingo@xxxxxxx>

drivers/net/loopback.c | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)

Index: linux/drivers/net/loopback.c

=====

--- linux.orig/drivers/net/loopback.c
+++ linux/drivers/net/loopback.c
@@ -158,7 +158,7 @@ static int loopback_xmit(struct sk_buff lb_stats->bytes +=
skb->len;
lb_stats->packets++;
- netif_rx(skb);
+ netif_receive_skb(skb);
return 0;
}

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in
the body of a message to majordomo@xxxxxxxxxxxxxxxxxxx
More majordomo info at <http://vger.kernel.org/majordomo-info.html>
Please read the FAQ at <http://www.tux.org/lkml/>

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in
the body of a message to majordomo@xxxxxxxxxxxxxxxxxxx
More majordomo info at <http://vger.kernel.org/majordomo-info.html>
Please read the FAQ at <http://www.tux.org/lkml/>