Subject: [PATCH] possible deadlock in tulip driver
Posted by den on Tue, 24 Jul 2007 07:49:08 GMT
View Forum Message <> Reply to Message

Calling flush_scheduled_work() may deadlock if called under rtnl_lock
(from dev->stop) as linkwatch_event() may be on the workqueue and it will try
to get the rtnl_lock

Signed-off-by: Denis V. Lunev <den@openvz.org>
---

 tulip_core.c | 4 ++--
 1 file changed, 2 insertions(+), 2 deletions(-)

--- ./drivers/net/tulip/tulip_core.c.tulip 2007-07-16 12:54:29.000000000 +0400
+++ ./drivers/net/tulip/tulip_core.c 2007-07-23 19:06:24.000000000 +0400
@@ -726,8 +726,6 @@ static void tulip_down (struct net_devic
  void __iomem *ioaddr = tp->base_addr;
  unsigned long flags;

- flush_scheduled_work();
-
  del_timer_sync (&tp->timer);
 #ifdef CONFIG_TULIP_NAPI
  del_timer_sync (&tp->oom_timer);
@@ -1788,6 +1786,8 @@ static void __devexit tulip_remove_one (
  if (!dev)
   return;

+ flush_scheduled_work();
+
  tp = netdev_priv(dev);
  unregister_netdev(dev);
  pci_free_consistent (pdev,

Subject: Re: [PATCH] possible deadlock in tulip driver
Posted by Valerie Henson on Mon, 30 Jul 2007 19:12:32 GMT
View Forum Message <> Reply to Message

(No longer maintainer, btw.)

What situation have you tested this under? Thanks,

-VAL

On Tue, Jul 24, 2007 at 11:49:08AM +0400, Denis V. Lunev wrote:
> Calling flush_scheduled_work() may deadlock if called under rtnl_lock

> (from dev->stop) as linkwatch_event() may be on the workqueue and it will try
> to get the rtnl_lock
>
> Signed-off-by: Denis V. Lunev <den@openvz.org>
> ---
>
> tulip_core.c | 4 ++--
> 1 file changed, 2 insertions(+), 2 deletions(-)
>
> --- ./drivers/net/tulip/tulip_core.c.tulip 2007-07-16 12:54:29.000000000 +0400
> +++ ./drivers/net/tulip/tulip_core.c 2007-07-23 19:06:24.000000000 +0400
> @@ -726,8 +726,6 @@ static void tulip_down (struct net_devic
>   void __iomem *ioaddr = tp->base_addr;
>   unsigned long flags;
>
> - flush_scheduled_work();
> -
>   del_timer_sync (&tp->timer);
> #ifdef CONFIG_TULIP_NAPI
>   del_timer_sync (&tp->oom_timer);
> @@ -1788,6 +1786,8 @@ static void __devexit tulip_remove_one (
>   if (!dev)
>   return;
>
> + flush_scheduled_work();
> +
>   tp = netdev_priv(dev);
>   unregister_netdev(dev);
>   pci_free_consistent (pdev,
> -
> To unsubscribe from this list: send the line "unsubscribe netdev" in
> the body of a message to majordomo@vger.kernel.org
> More majordomo info at http://vger.kernel.org/majordomo-info.html

---

Subject: Re: [PATCH] possible deadlock in tulip driver
Posted by den on Tue, 31 Jul 2007 06:52:34 GMT

View Forum Message <> Reply to Message

Manual code check. The similar fixes are present in almost all drivers,
f.e. tg3 one. I have an unrelated deadlock with rtnl.

Regards,
    Den

Valerie Henson wrote:
> (No longer maintainer, btw.)
>

> What situation have you tested this under? Thanks,
>
> -VAL
>
> On Tue, Jul 24, 2007 at 11:49:08AM +0400, Denis V. Lunev wrote:
>> Calling flush_scheduled_work() may deadlock if called under rtnl_lock
>> (from dev->stop) as linkwatch_event() may be on the workqueue and it will try
>> to get the rtnl_lock
>>
>> Signed-off-by: Denis V. Lunev <den@openvz.org>
>> ---
>>
>> tulip_core.c | 4 ++--
>> 1 file changed, 2 insertions(+), 2 deletions(-)
>>
>> --- ./drivers/net/tulip/tulip_core.c.tulip 2007-07-16 12:54:29.000000000 +0400
>> +++ ./drivers/net/tulip/tulip_core.c 2007-07-23 19:06:24.000000000 +0400
>> @@ -726,8 +726,6 @@ static void tulip_down (struct net_devic
>>   void __iomem *ioaddr = tp->base_addr;
>>   unsigned long flags;
>>
>> - flush_scheduled_work();
>> -
>>   del_timer_sync (&tp->timer);
>> #ifdef CONFIG_TULIP_NAPI
>>   del_timer_sync (&tp->oom_timer);
>> @@ -1788,6 +1786,8 @@ static void __devexit tulip_remove_one (
>>   if (!dev)
>>   return;
>>
>> + flush_scheduled_work();
>> +
>>   tp = netdev_priv(dev);
>>   unregister_netdev(dev);
>>   pci_free_consistent (pdev,
>> -
>> To unsubscribe from this list: send the line "unsubscribe netdev" in
>> the body of a message to majordomo@vger.kernel.org
>> More majordomo info at http://vger.kernel.org/majordomo-info.html
>