Faisal, Mir

From: Faisal, Mir
Sent: Thursday, July 18, 2024 10:42 AM
To: Ninganagoudar, Laxman; Croitoru, Gabriel; Popescu, IoneliaAniela; Liteanu, Alin; Veer, Rajkumar
Subject: RE: IDCEvo:Solution discussion for phy access issue with MDIO read/write access

Hi,

We have found fix of this issue. Root cause analysis is done, and the issue is found in our sysfs implememtation in STMMAC driver to read and write MDIO PHY registers.

Cause of problem: -

1. PHY framework schedules invoking "read_status" callback to check PHY link status.

```
File: - drivers/net/phy/phy_device.c

=> Create workqueue for PHY state machine and PHY link check
        struct phy_device phy_device_create(...) {
                //...
                INIT_DELAYED_WORK(&dev->state_queue, phy_state_machine);
                //...
        } -> worker_thread -> process_one_work -> phy_state_machine -> _phy_state_machine
                                        -> phy_check_link_status -> q222x_read_status

=> Schedule PHY link check at every 1 second
        void _phy_state_machine(struct work_struct *work)
        {
                //...
                err = phy_check_link_status(phydev); --> Invokes Marvell PHY driver's q222x_read_status() API
                //...

                phy_queue_state_machine(phydev, PHY_STATE_TIME);
                //...
        }

=> Initialize delayed workqueue to schedule at every 1 second
        void phy_queue_state_machine(struct phy_device *phydev, unsigned long jiffies)
        {
                mod_delayed_work(system_power_efficient_wq, &phydev->state_queue,
                        jiffies);
        }

        @arg jiffies = PHY_STATE_TIME --> #define PHY_STATE_TIME  HZ

=> Invoke Marvel PHY's read_status() callback per second

phy_check_link_status(...) {
```

```
        //...
        phy_read_status() {
                        //...
                        phydev->drv->read_status(phydev); --> Invoke q222x_read_status()
                        //....
        }
        //...
}
```

Due to workqueue scheduled per second, Marvel PHY driver's ".read_status" registered callback "q222x_read_status()" is
also invoked at every 1 second. This keeps Network stack updated with Ethernet PHY link status.

2. Issue with sysfs file used to read and write MDIO PHY register: -

The sysfs entry used to read and write MDIO PHY register, used in diagnostic script, causes collision of MDIO packet,
as it uses STMMAC MDIO read/write API directly without the mutex lock "mutex_lock(&phydev->mdio.bus->mdio_lock)".

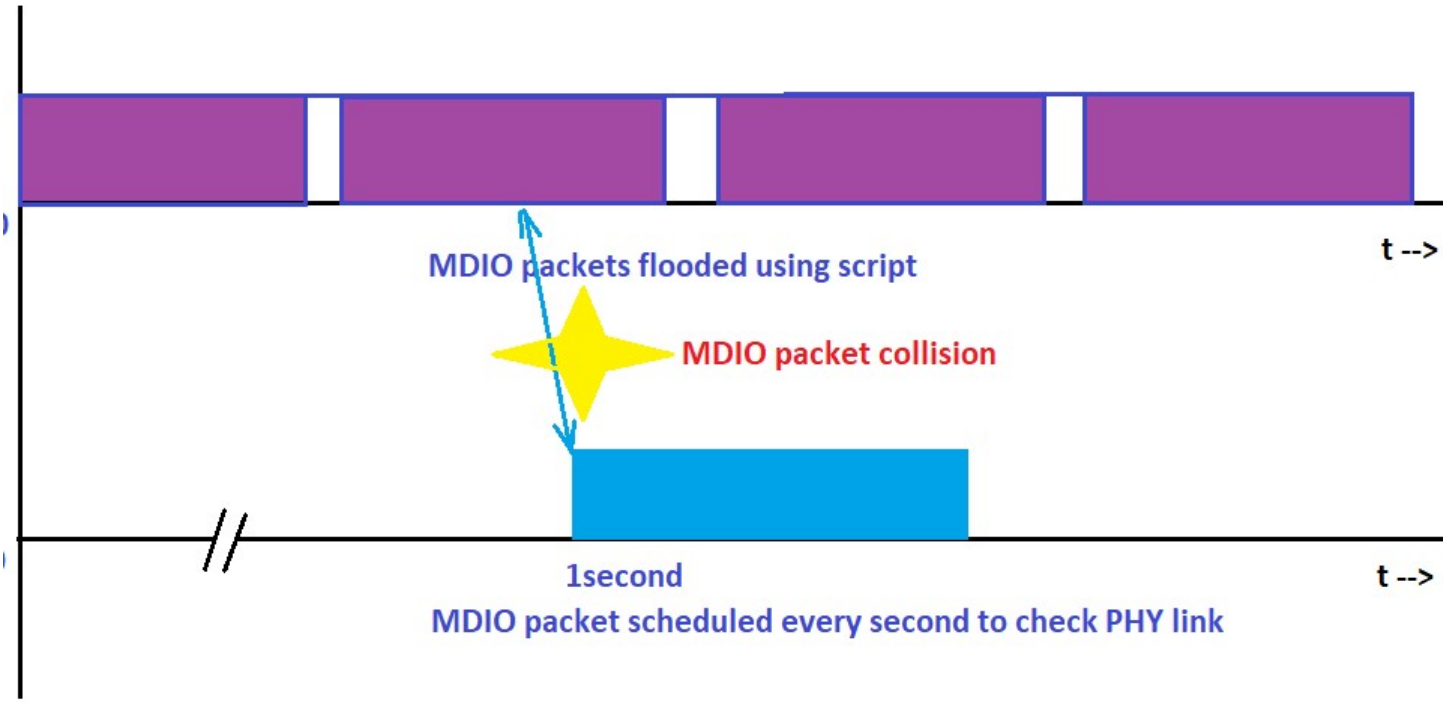
PHY faramework Workqueue invokes phy_check_link_status() every second uses this mutex lock.

But when script runs, then sysfs entry used to read MDIO PHY register (using command below), do not take care of atomicity
of MDIO line, as it does not use MDIO bus's mutex lock, which collides with Workqueue's MDIO packet scheduled every second.

Sysfs command to read MDIO register: -

```
echo -n "cl45r 0x1 0x0003" > /sys/devices/platform/16cc8000.ethernet/stmmac
```

MDIO packet collision: -



==> Our sysfs code having issue is shown below for reference: -

```
static ssize_t stmmac_store(struct device *dev,
                    struct device_attribute *attr,
                    const char *buf,
                    size_t count)
{

          //...
-          sprintf(priv->mdio_buff, "Cl22-R: %x",
-                priv->mii->read(priv->mii, phy_dev->mdio.addr, reg));
+
+          phy_lock_mdio_bus(phy_dev);
+          read_data = priv->mii->read(priv->mii, phy_dev->mdio.addr, reg);
+          phy_unlock_mdio_bus(phy_dev);
+
+          sprintf(priv->mdio_buff, "Cl22-R: %x", read_data);
          //...
}
```

Lines starting with “+” shows fix, which will be pushed.

By adding delay of few milliseconds with every "echo ..." command is solving problem temporarily, as it only reduces
probability of MDIO packet collision with PHY framwork's PHY link check MDIO packet scheduled every second.

Ciollision simulation Script: -

```
while :
        do

        # sleep 0.01 -> By adding this only probability of collision reduces, but not a fix.

        echo -n "cl45r 0x1 0x0003" > /sys/devices/platform/16cc8000.ethernet/stmmac
        done & BGPID1=$!
```

FIX: -

In our sysfs implementation to read/write MDIO PHY register, we will use mutex lock "phydev->mdio.bus->mdio_lock"
or phy_read_mmd() or phy_write_mmd() API as it contains the mutex lock. This fix will make MDIO access atomic, to avoid any collision of MDIO packet. The required gerrit will be pushed.

To verify fix, above test script was used without delay, and there is no issue observed with ping to the target.

Regards,
Mir Faisal

-----Original Appointment-----
From: Ninganagoudar, Laxman <Laxman.Ninganagoudar@harman.com>
Sent: Wednesday, July 17, 2024 7:24 PM
To: Croitoru, Gabriel; Popescu, IoneliaAniela; Liteanu, Alin; Veer, Rajkumar; Faisal, Mir
Subject: IDCEvo:Solution discussion for phy access issue with MDIO read/write access
When: Friday, July 19, 2024 9:00 AM-9:30 AM (UTC+01:00) Amsterdam, Berlin, Bern, Rome, Stockholm, Vienna.
Where: Microsoft Teams Meeting

Hi All,

Agneda : Solution discussion for phy access issue with MDIO read/write access.
https://elvis.harman.com/cgi-bin/ticket?TID=3580615

Best Regards,
Laxman

Microsoft Teams Need help?

Join the meeting now

Meeting ID: 282 716 257 207

Passcode: T6NQoe

Dial in by phone

+49 69 589961530,,490720196# Germany, Frankfurt am Main

Find a local number

Phone conference ID: 490 720 196#

Join on a video conferencing device

Tenant key: harman@m.webex.com

Video ID: 116 161 246 2

More info

For organizers: Meeting options | Reset dial-in PIN
