Forward packet code causes WARNING (followed by reboot)

@Linus, the syscall project found following problem:

Hello,

syzbot found the following crash on:

HEAD commit: da940012 Merge tag 'char-misc-5.4-rc3' of git://git.kernel..
get tree: upstream
crash output: https://syzkaller.appspot.com/x/log.txt?x=13ffd808e0000
kernel config: https://syzkaller.appspot.com/x/config?x=2d2f92a28d3e50
dashboard link: https://syzkaller.appspot.com/bug?extid=0b807de416427ff3ddd1
compiler: clang version 9.0.0 (/home/glider/llvm/clang
syz repro: https://syzkaller.appspot.com/x/repro.syz?x=141fffd77600000
C reproducer: https://syzkaller.appspot.com/x/repro.c?x=11edd580e0000

IMPORTANT: if you fix the bug, please add the following tag to the commit:
Reported-by: syzbot+c0b807de416427ff3ddd1@syzkaller.appspotmail.com

------------[ cut here ]------------

WARNING: CPU: 1 PID: 30 at net/batman-adv/bat_iv_ogm.c:382
batadv_iv_ogm_emit net/batman-adv/bat_iv_ogm.c:382 [inline]
WARNING: CPU: 1 PID: 30 at net/batman-adv/bat_iv_ogm.c:382
batadv_iv_send_outstanding_bat_ogm_packet+0x6b4/0x770
Kernel panic - not syncing: panic_on_warn set ...
CPU: 1 PID: 30 Comm: kworker/u4:2 Not tainted 5.4.0-rc2+ #0
Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS

Workqueue: bat_events batadv_iv_send_outstanding_bat_ogm_packet

Call Trace:
__dump_stack lib/dump_stack.c:77 [inline]
dump_stack+0x1d8/0x2f8 lib/dump_stack.c:113
panic+0x264/0x7a9 kernel/panic.c:221
__warn+0x2e0/0x210 kernel/panic.c:582
report_bug+0x1b6/0x2f0 lib/bug.c:195
fixup_bug arch/x86/kernel/traps.c:179 [inline]
do_error_trap+0x4d7/0x440 arch/x86/kernel/traps.c:272
do_invalid_op+0x3e6/0x40 arch/x86/kernel/traps.c:291
invalid_op+0x230 arch/x86/entry/entry_64.S:1028
RIP: 0010:batadv_iv_ogm_emit net/batman-adv/bat_iv_ogm.c:382 [inline]
RIP: 0010:batadv_iv_send_outstanding_bat_ogm_packet+0x6b4/0x770
net/batman-adv/bat_iv_ogm.c:1663
Code: 66 05 00 eb 05 00 0f 0e 09 34 ff ff ff ff ff ff ff 89 d9 80 e1 03 38 01 0f 8c 01 0f ff ff
RSP: 0018:ffff8880a9abf4 8 EFLAGS: 00010293
RAX: ffffffff874e8a4 RBX: ffffffff894160870 RCX: ffffffff880a9ab2080
RDX: 0000000000000000 RSI: 0000000000000001 RDI: 0000000000000002
RRB: ffffffff880a9abfcd8 R08: ffffffff874e28e R09: ffffffff81d0123e6969
R10: ffffffff80c3e6969 R11: 0000000000000000 R12: ffffffff88091f34000
R13: dfffffff880a9abf4c8 1/3

2/22/2020
---
This bug is generated by a bot. It may contain errors. See https://goo.gl/tpsmEJ for more information about syzbot. syzbot engineers can be reached at syzkaller@googlegroups.com.

syzbot will keep track of this bug report. See: https://goo.gl/tpsmEJ#status for how to communicate with syzbot. syzbot can test patches for this bug, for details see: https://goo.gl/tpsmEJ#testing-patches

History

#1 - 10/14/2019 09:16 AM - Sven Eckelmann

They also send following information (which might be bogus because they tried to bisect a race condition):

syzbot has bisected this bug to:

commit 26d051e301f67cdd2ea3404abb43902f13214efa
Author: Arvind Yadav <arvind.yadav.cs@gmail.com>
Date: Thu Jun 29 08:21:35 2017 +0000

media: exynos4-is: fimc-is-i2c: constify dev_pm_ops structures

bisection log: https://syzkaller.appspot.com/x/bisect.txt?x=10a0aff0e00000
start commit: da940012 Merge tag 'char-misc-5.4-rc3' of git://git.kernel...
final crash: https://syzkaller.appspot.com/x/report.txt?x=12a0aff0e00000
console output: https://syzkaller.appspot.com/x/log.txt?x=14a0aff0e00000
kernel config: https://syzkaller.appspot.com/x/.config?x=2d2fd92a28d3e50
dashboard link: https://syzkaller.appspot.com/bug?extid=c0b807de416427ff3dd1
running repro: https://syzkaller.appspot.com/x/repro.c7x=11edd580e00000
C reproducer: https://syzkaller.appspot.com/x/repro.c?

Reported-by: syzbot+c0b807de416427ff3dd1@syzkaller.appspotmail.com
Fixes: 26d051e301f6 ("media: exynos4-is: fimc-is-i2c: constify dev_pm_ops structures")

For information about bisection process see: https://goo.gl/tpsmEJ#bisection
This is the relevant code:

```c
static void batadv_iv_ogm_emit(struct batadv_forw_packet *forw_packet)
{
    struct net_dev *soft_iface;
    if (!forw_packet->if_incoming)
    {
        pr_err("Error - can't forward packet: incoming iface not specified\n");
        return;
    }
    soft_iface = forw_packet->if_incoming->soft_iface;
    if (WARN_ON(!forw_packet->if_outgoing))
        return;
    if (WARN_ON(forw_packet->if_outgoing->soft_iface != soft_iface))
        return;
    if (forw_packet->if_incoming->if_status != BATADV_IF_ACTIVE)
        return;
    /* only for on specific outgoing interface */
    batadv_iv_ogm_send_to_if(forw_packet, forw_packet->if_outgoing);
}
```

We should check whether the softif of the outgoing hardif was really changed by the reproducer. And in this case figure out what we should do in this case. If it is expected then don't do a WARN_ON. If it is not then fix the race condition properly.