RTL8169 の過去のバグについて

2016/6/30

藤田将輝

1 はじめに

開発支援環境を用いて，過去あった NIC ドライバのバグを再現することを考えている．Linux カーネルは Git を用いて開発されており，開発過程が全て記録として残っている．本開発環境での開発対象である NIC ドライバ (RTL8169) の開発過程も記録されており，どのようなバグが修正されてきたかを確認可能である．そこで，この記録から修正されたバグを調査した．本資料は，調査したバグについてまとめたものである．

2 調査目的

本研究における開発支援環境は NIC ドライバを対象としており，NIC を用いずに NIC の動作を再現するものである．これを用いて，報告されたバグを再現できれば，本環境がデバッグにおいて有用である可能性を示せる．そこで，過去に修正されたバグを調査し，現在の開発支援環境で再現できるかを考察する．また，現在の開発支援環境にどのような機能を追加すればバグを再現するかを考察する．これらの目的からバグについて調査した．

3 調査方法

本開発支援環境の開発対象としている RTL8169 は Linux カーネルの一部として Git を用いて開発されている．このため，開発工程が全て記録として残っている．この記録はある機能を追加した，あるバグを修正した，といった作業の粒度で記録されており，そのメッセージが残されている．そこで，本開発環境が対象としているバージョンの最新の更新記録から過去 3 年間のバグを調査した．バグを修正したメッセージには「fix: 」という文字列が含まれているため，この文字列を含むメッセージの記録を調査した．この際，ハードウェアに依存した修正や詳細が記述されていない修正は調査対象外とした．

4 発見したバグ

4.1 一覧

Linux3.0.8 の最新更新記録から過去 3 年間のバグを調査した．なお，詳細の記載されていないバグやハードウェアに関するバグは省いている．その一覧を以下に示す．

(1) チェックサムを確認するタイミングに関するバグ

(2) 大きなサイズのパケットを受信した場合，マシンをクラッシュさせるバグ
(3) 特定のサイズのパケットを受信できないバグ

4.2 チェックサムを確認するタイミングに関するバグ

4.2.1 バグが発生する機能の概要

NIC ドライバはパケットを受信した後，パケットをソケットバッファに格納する．その後，NIC ドライバは受信ディスクリプタを確認する．TCP か UDP パケットを受信していることを確認するとそのパケットが格納されているソケットバッファの構造体のメンバの 1 つである `skb->ip_summed` という IP 層でのチェックサムが必要であるか否かを示すフラグを更新する．これにより，IP 層でチェックサムが必要でない状態にする．

4.2.2 バグ内容

本バグでは，ソケットバッファにパケットを格納する前に `skb->ip_summed` を更新してしまう．この後，ソケットバッファにパケットを格納する．この際，新しくソケットバッファを作成し，新しいソケットバッファにパケットを格納する．これにより，`skb->ip_summed` の更新が無かったことになる．したがって，チェックサムの不要なパケットであっても IP 層でチェックサムを確認することになる．

4.3 大きなサイズのパケットを受信した場合，マシンをクラッシュさせるバグ

4.3.1 バグが発生する機能の概要

NIC は受信パケットの最大サイズとして RxMaxSize というレジスタを持っている．これより大きいパケットはフィルタし，破棄するようになっている．

4.3.2 バグ内容

本バグでは，RxMaxSize を固定的に 16384 としていた．NIC ドライバがアロケートするバッファサイズは `tp->rx_buf_sz` に格納されているが，このサイズは MTU によって決定されるため，16384 より小さくなる可能性がある．このため，アロケートしたバッファサイズよりも大きいパケットを受信できる状態となる．アロケートしたバッファサイズよりも大きいパケットを受信すると他のカーネル領域を侵害し，クラッシュする可能性がある．

4.4 特定のサイズのパケットを受信できないバグ

4.4.1 バグが発生する機能の概要

NIC ドライバはネットワークインタフェースの起動時，MTU を指定できる．この MTU 値は `dev->mtu` に格納されている．NIC ドライバはこの MTU 値を用いて `tp->rx_buf_sz` という受信バッファのサイズを決定する．このメンバはハードウェアのフィルタリングに関係しており，`tp->rx_buf_sz` より大きいサイズのパケットはハードウェアにより破棄される．ユーザは MTU を IP パケットの最大

サイズだと意識して値を指定する．

4.4.2 バグ内容

本バグでは，tp->rx_buf_sz を決定する際，dev->mtu の値とデフォルトの受信バッファサイズである 1536 を比較し，dev->mtu の方が大きければ，dev->mtu の値に ether net ヘッダのサイズを付与したサイズを tp->rx_buf_sz とする．dev->mtu の方が小さければデフォルト値である 1536 とする．ここで，指定した MTU は IP パケットサイズだと意識しているため，tp->rx_buf_sz を決定する際，デフォルト値と比較するのはフレームサイズである必要がある．しかし，本バグで比較しているのは IP パケットサイズとして指定した MTU とデフォルト値である．これでは，指定した MTU に従ってパケットを受信しようとするとハードウェアによって破棄される可能性がある．例えば，ユーザが MTU として 1536 を指定した場合を考える．dev->mtu には 1536 が格納され，tp->rx_buf_sz を決定する際，dev->mtu とデフォルト値である 1536 を比較する．同じ値であるため，tp->rx_buf_sz は 1536 となる．このため，1537B 以上のパケットはハードウェアにより破棄される．しかし，ユーザは 1536B までの IP パケットを受信できると意図している．1536B の IP パケットはフレームにすると 1536 + 22(ether net ヘッダ等) = 1558 となり，受信できない．

5 考察

5.1 チェックサムを確認するタイミングに関するバグ

このバグは，バグが発生するバージョンに戻すことで，容易に再現が可能であると考えられる．なぜなら，全てのパケットにバグの効果が適用されるためである．本研究における開発支援環境を用いなくても，ソケットバッファが作成された段階でソケットバッファ構造体の要素の値を確認するだけでよい．このため，このバグは，本開発支援環境のデバッグ支援対象としては使用できないと考えられる．

5.2 大きなサイズのパケットを受信した場合，マシンをクラッシュさせるバグ

このバグを再現しようとすると，2 台の計算機を用いる必要がある．また，アロケートしたバッファサイズ以上のパケットを受信する必要がある．本開発支援環境を用いると 1 台の計算機で 2 つの OS を動作できる．また，任意にパケットのサイズを指定できる．これにより，このバグを再現できると考えられる．このため，このバグは本開発支援環境で再現が容易になるバグだと考えられる．ただし，このバグが発生するドライバのバージョンは本研究で実装したバージョンとバッファのアロケート方法が違っているため，これを考慮する必要がある．

5.3 特定のサイズのパケットを受信できないバグ

このバグを再現しようとすると，2 台の計算機を用いる必要がある．また，MTU を指定してこれに近いサイズのパケットを受信する必要がある．本開発支援環境を用いると，1 台の計算機上で 2 つの

OS を動作できる．また，パケットのサイズの指定も可能である．このため，このバグは本開発支援環境で再現が容易になるバグだと考えられる．ただし，このバグが発生するドライバのバージョンは本研究で実装したバージョンとバッファのアロケート方法が違っているため，これを考慮する必要がある．

6 おわりに

本資料では，過去に修正された NIC ドライバのバグについてまとめた．今後は，バグを再現するための具体的な課題や対処について検討する．