

デバッグ支援環境におけるパケット受信の測定

2015/9/18

藤田将輝

1 はじめに

本資料では、デバッグ支援環境を用いて、デバッグ支援 OS からデバッグ対象 OS に UDP パケットを連続で送信した際に、デバッグ対象 OS 上で動作する受信プログラムがどの程度の割り込み間隔、およびパケットサイズならば、パケットを全て受信できるかについて測定したことを示す。測定の目的は、本デバッグ支援環境の動作を確認し、どの程度の通信量を実現できるかを調査することである。前回の資料 < No.284-07 > では、デバッグ支援 OS で作成したパケットがデバッグ対象 OS の NIC ドライバでソケットバッファに格納された数を計測した。これに対し、本資料では、デバッグ支援 OS で作成したパケットがデバッグ対象 OS 上で動作する UDP の受信プログラムによって受信された数を計測している。

2 測定

2.1 測定目的

本測定は、本デバッグ支援環境がどの程度の通信量を実現して動作できるかを調査することである。また、前回の資料で示した、NIC ドライバで処理できる割り込み間隔と比較し、差異があればこの差異について考察することである。これらのため、デバッグ対象 OS に繰り返し割り込みを発生させ、パケット受信割り込み処理をさせた際の、受信に失敗した回数を計測する。また、繰り返し割り込みを発生させた際の時間を計測し、計測した時間と、パケットの受信に成功したデータ量から通信量を計算する。なお、受信に失敗するとは、割り込みを発生させたにも関わらず、デバッグ対象 OS 上で動作する UDP の受信プログラムがパケットを受信できなかったことを指す。

2.2 測定環境

本測定における測定環境を表に示す。

2.3 測定方法

本測定では、条件を変更させながら、繰り返し割り込みを発生させることで、デバッグ対象 OS で複数回パケット受信割り込み処理をさせ、パケットが受信できなかった数を測定する。ここで、パケットが受信できなかったとは、デバッグ対象 OS 上で動作する受信プログラムが `recv()` でパケットを受け取れなかったことを指す。具体的な動作は、以下の動作を 3000 回繰り返すことを 1 サイクルとして、条

表 1 実験環境

項目名	環境
OS	Fedora14 x86_64(Mint 3.0.8)
CPU	Intel(R) Core(TM) Core i7-870 @ 2.93GHz
NIC ドライバ	RTL8169
ソースコード	r8169.c

件を変えながら 500 サイクル行う．なお，この動作はデバッグ支援 OS が実装しているデバッグ支援機構によって行う．

- (1) パケットを Mint の共有メモリ上にある NIC の受信バッファに配置する．
- (2) Mint の共有メモリ上にある NIC の受信ディスクリプタを取得し，受信済み状態に更新して，Mint の共有メモリ上に再配置する．
- (3) デバッグ対象 OS のコアへ IPI を送信する．
- (4) 指定したインターバルとなるまで処理を待つ．
- (5) (1) に戻る．

上記 (1)～(4) の動作 1 回につき，デバッグ対象 OS ではパケット受信割り込み処理が発生し，デバッグ対象 OS 上で動作する UDP の受信プログラムが 1 パケットを受信する．

2.4 失敗回数の測定

失敗回数の測定は，デバッグ支援 OS とデバッグ対象 OS の 2 つの OS が参照できるカウンタを用いることで行っている．デバッグ対象 OS 上で動作する `recv()` がパケットを受信すると，共有メモリに配置しているカウンタをインクリメントする．1 サイクルが終了すると，デバッグ支援 OS がカウンタを取得し，ループ回数の 3000 回との差分を取ることで，受信に失敗した回数を測定している．

2.5 測定流れと測定箇所

スループットを算出するため，1 サイクルにかかった時間を計測している．1 サイクルにかかった時間と受信に成功したデータ量からスループットを計算する．本測定における処理フローと時間の測定箇所を図 1 に示す．デバッグ支援 OS における処理流れを以下で説明する．

- (1) デバッグ支援 OS 上でパケットジェネレータを動作させる．
- (2) システムコールによりデバッグ支援機構が呼び出され，パケットを配置する．
- (3) Mint の共有メモリに配置している受信ディスクリプタを更新し，再配置する．
- (4) デバッグ対象 OS の占有するコアへ IPI を送信する．
- (5) 指定した時間の間隔になるまで処理を待たせる．

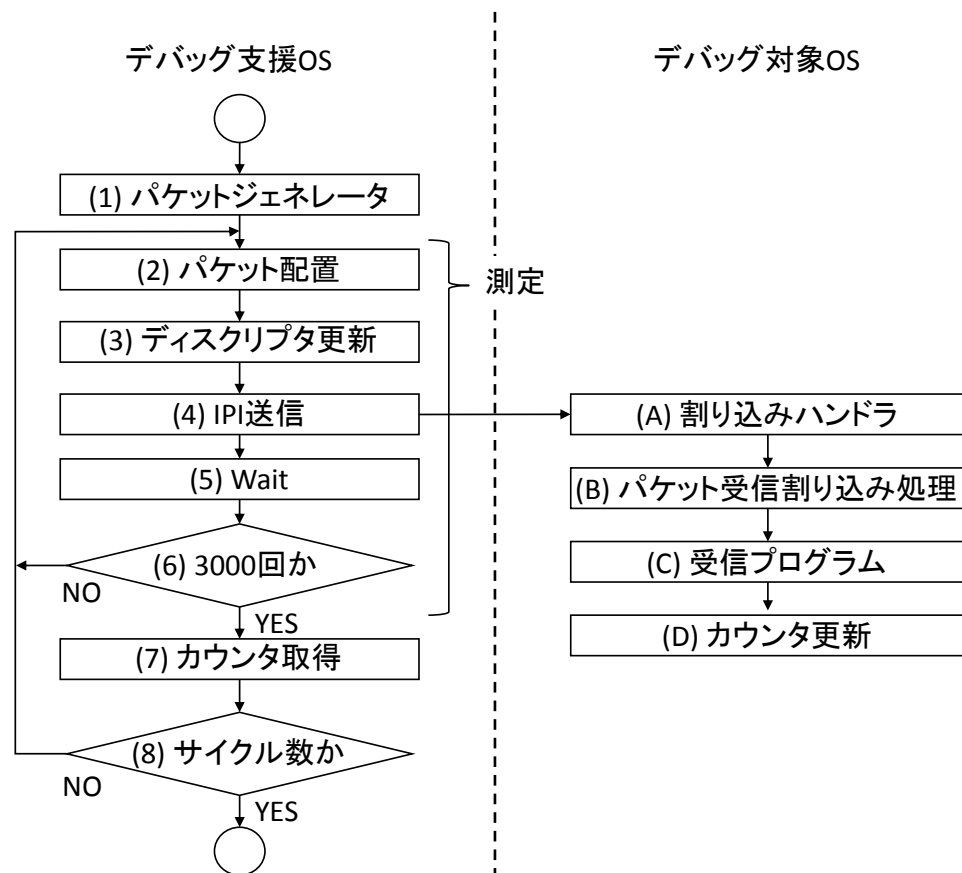


図 1 処理フロー

(6) 3000 回ループすれば、次に進み、3000 回に満たなければ (2) に戻る。

(7) 共有メモリに配置しているカウンタを取得し、3000 回と比較し、失敗回数を求める。

(8) 指定したサイクル数に達していれば終了し、指定したサイクル数に満たなければ (2) に戻る。

デバッグ対象 OS における処理流れを以下で説明する。

(A) IPI を受けて割り込みハンドラが動作する。

(B) パケット受信割り込み処理が動作する。

(C) 受信プログラムの `recv()` がパケットを受信する。

(D) システムコールにより、カウンタを更新し、共有メモリに配置する。

以上のフローの中で、時間を計測した箇所は (2) から (6) のループを抜けた直後である。

2.6 測定手順

測定手順について以下に示し、説明する。

(1) Mint を用いて、デバッグ支援 OS とデバッグ対象 OS を動作させる。

(2) デバッグ対象 OS 上で UDP の受信プログラムを動作させる。

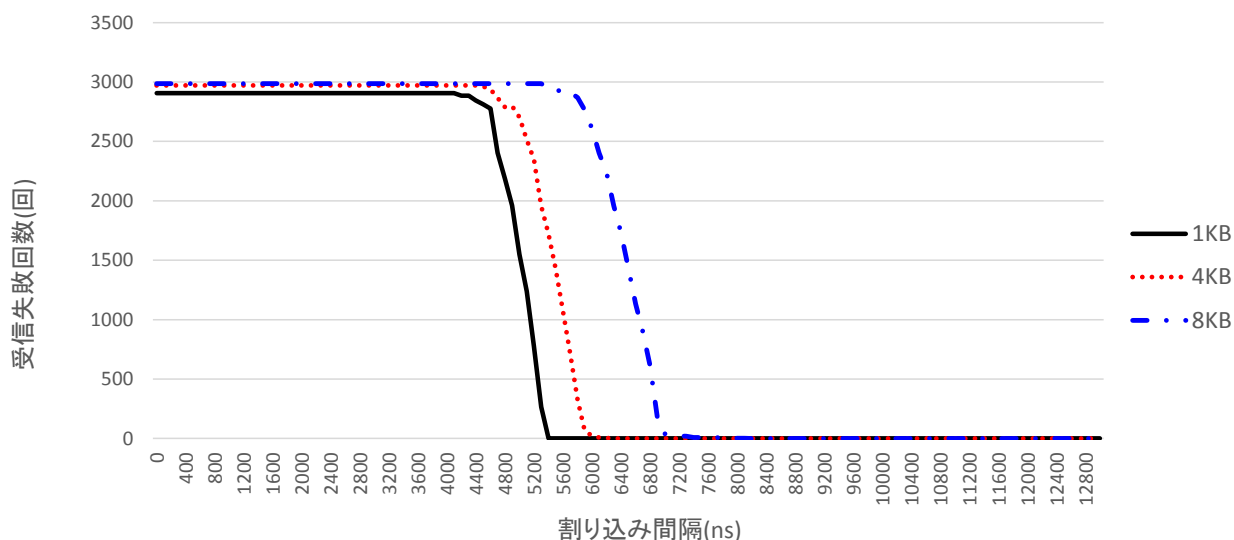


図 2 受信プログラムのパケット受信におけるサイズと割り込み間隔の関係

- (3) デバッグ支援 OS 上で 1 サイクルのループ回数，1 サイクルごとに増加させる時間，サイクル数，およびパケットのサイズを引数に取り，パケットジェネレータを動作させる．
- (4) (割り込みの間隔，失敗回数，1 サイクルに要した時間) という形式でデータをファイルに出力する．

2.7 測定結果

パケットサイズ毎のパケット受信失敗回数と割り込み間隔の測定結果を図 2 に示す．また，各サイズ毎の最初に受信失敗回数が 0 回になった時の割り込み間隔，1 サイクルにかかった時間，およびスループットについて，表 2 に示す．

表 2 パケットのサイズと割り込み間隔の関係

サイズ (KB)	割り込み間隔 (ns)	1 サイクルの時間 (us)	スループット (Gbps)
1	5400	16319	1.4
4	6300	19037	4.8
8	7800	23536	7.7

3 考察

測定結果からわかることを以下に示す．

- (1) パケットのサイズによらず，ある時点までは受信に失敗した回数は一定である．この時点をもと

する．

- (2) 時点 t を超えるまでは，ほとんど受信に失敗している．
- (3) t を超えると受信失敗回数は一次関数的に減少し，割り込み間隔を増加させると受信失敗回数は 0 回になる．一次関数的に減少する際，パケットのサイズに関わらず，傾きは一定である．
- (4) パケットのサイズを増加させると， t の値も増加する．この際，パケットのサイズとこれに対応する t の値は一次関数的な関係を持っている．
- (5) 実現できる最大のスループットは，パケットサイズが 8KB の時で，7.7Gbps である．

本測定の測定結果は，デバッグ対象 OS 上で動作する UDP の受信プログラム内で `recv()` が受信したパケットの数から受信失敗回数を求めている．これに対し，前回の測定では，デバッグ対象 OS が保持する NIC ドライバ内で，ソケットバッファに格納したパケットの数から受信失敗回数を求めた．これらの測定結果を比較して，わかったことを以下に示す．

- (1) NIC ドライバで測定した各パケットサイズの初めて受信失敗回数が 0 回になった際の割り込み間隔と，受信プログラムにおける時点 t がほぼ一致する．これを示すため，NIC ドライバで測定した各パケットサイズの初めて受信失敗回数が 0 回になった際の割り込み間隔と，受信プログラムにおける時点 t を表に示す．

表 3 初めて受信失敗回数が 0 回になった際の割り込み間隔と時点 t の値

サイズ (KB)	割り込み間隔 (ns)	時点 t (ns)
1	4300	4200
4	4500	4500
8	5400	5400

4 おわりに

本資料では，デバッグ対象 OS 上で動作する UDP の受信プログラムにおいて，どの程度の間隔，パケットサイズならばパケットを受信できるかについて測定したことを示した．前回の測定と今回の測定から，本デバッグ環境を用いて，NIC ドライバでは最大 10Gbps を，受信プログラムでは最大 7.7Gbps で動作することを確認した．