

リアルタイム Linux における Ethernet 通信性能の評価

岡部 亮† 山本 遼介† 大木 英俊† 水口 武尚†

三菱電機株式会社 情報技術総合研究所†

1. はじめに

複数の機器を Ethernet で接続した組み込みシステムが増えている。このようなシステムでリアルタイム性を保証するには、各機器内部のリアルタイム性に加えて、機器間の通信のリアルタイム性が重要である。豊富な機能を備えることから、Linux はこのようなシステムでも採用されている。特に、リアルタイム性の保証のために PREEMPT_RT パッチが適用された Linux (リアルタイム Linux) が採用されることが多い。

リアルタイム Linux を利用する場合であっても、Ethernet 通信のリアルタイム性には課題が残る [1][2]。Ethernet 通信の送信処理では、送信を要求したアプリケーションのコンテキストでプロトコルスタック処理から NIC ドライバへの送信依頼までが行われるのが基本である。この場合は、アプリケーションにリアルタイム優先度を割り当てることで送信処理の遅延を抑制できる。しかし、送信要求時に NIC ドライバのキューがフルである場合や、アプリケーション間で操作対象の NIC が競合してキューがロックされている場合、NIC ドライバへの送信依頼はソフト割込みとして実行される。一方、受信処理では、IRQ スレッドから起動されるソフト割込みにおいて、パケットの受信処理およびプロトコルスタック処理が行われる。通常、ソフト割込み処理は、そのソフト割込みを最後に発生させたスレッドのコンテキストで動作する。ただし、ソフト割込み実行中のスレッドがプリエンプトされたり、パケットのポーリング受信の処理時間が閾値を超えたりした場合には、残りのソフト割込み処理は `ksoftirqd` と呼ばれるカーネルスレッドに移譲される。リアルタイム Linux のデフォルトでは、この `ksoftirqd` には非リアルタイム優先度が割り当てられる。このため、送受信のアプリケーションや IRQ スレッドにリアルタイム優先度が割り当てられた場合でも、

`ksoftirqd` への処理の移譲が発生する場合には、リアルタイム性が低下すると考えられる。

本稿の目的は、ソフト割込みの遅延実行を担う `ksoftirqd` カーネルスレッドにリアルタイム優先度を割り当てた場合における、Linux の Ethernet 通信のリアルタイム性を評価することである。評価対象のプロトコルは UDP であり、定周期送信のジッターと、他機器からパケットを受信した際のラウンドトリップタイムをリアルタイム性の指標とする。

2. 評価方法

リアルタイム Linux を搭載する組み込みボードで、定周期送信ジッターとラウンドトリップタイム (RTT) を測定した。これらの測定結果を、`ksoftirqd` の優先度が非リアルタイムである場合と、リアルタイム優先度の場合とで比較した。

定周期送信ジッターは、ボードから一定周期で送信されるパケットが、対向機に到着する時間間隔である。ボード上の高優先度スレッドが 10 ミリ秒周期で起床してパケットを送信し、対向機でその到着間隔を測定した。定周期送信のパケットサイズは 122 Byte である。

RTT は、対向機がボードにパケットを送信してから、ボードからの応答パケットを受信するまでの時間である。ボード上の高優先度スレッドが、対向機からのパケットの受信、および、応答パケットの送信を行った。パケットサイズは 122 Byte であり、測定には `netperf` を利用した。

これらの性能を、無負荷の場合、TCP 送信負荷がある場合、TCP 受信負荷がある場合で測定した。TCP の送受信負荷の生成には `iperf` を利用し、サイズ 512 KB のパケットを最大帯域で流した。

動作周波数 600 MHz のシングルコア CPU と 100 Mbps の Ethernet ポートを搭載する組み込み向け評価ボードを使用した。評価に利用した Ethernet ポートは 1 チャンネルである。OS は Linux 4.19 に PREEMPT_RT を適用したものである。優先度設計として、定周期送信スレッドと `netperf` をリアルタイム優先度 10、Ethernet の IRQ スレッドをリアルタイム優先度 50、TCP 負荷生成用の `iperf` を非リアルタイム優先度とした。`ksoftirqd` につ

Evaluating Ethernet Communication Performance of Real-time Linux

† Ryo Okabe, Ryosuke Yamamoto, Hidetoshi Oki, Takehisa Mizuguchi, Information Technology R&D Center, Mitsubishi Electric Corporation

表 1: 定周期送信ジッターの測定結果 (単位: msec)

	無負荷		送信負荷		受信負荷	
	非 RT 優先度	RT 優先度	非 RT 優先度	RT 優先度	非 RT 優先度	RT 優先度
最大	0.074	0.103	4.264	5.862	0.433	1.765
標準偏差	0.007	0.007	0.494	3.101	0.106	0.215

いては非リアルタイム優先度とリアルタイム優先度 1 の二通りで計測し、測定結果を比較した。

3. 評価結果

表 1は定周期送信ジッターの測定結果である。無負荷時は、ksoftirqd の優先度の違いによる性能差はほぼ見られなかった。TCP 負荷がある場合、ksoftirqd をリアルタイム優先度にするると最大ジッターと標準偏差が悪化した。送信負荷時の性能悪化について、リアルタイム優先度化により ksoftirqd の処理時間が増えたため、ksoftirqd と定周期送信スレッドの間で送信キューのロック競合が増加した可能性が原因として考えられる。受信負荷時の性能悪化原因は、受信ソフト割込みだけが保留された状況における ksoftirqd の起床が増え、この際に ksoftirqd が当該割込みの処理完了まで送信ソフト割込み処理を開始できなかったためと推測する。優先度変更前は ksoftirqd の起床頻度が低かったため、送受信両方のソフト割込みが保留された状況で起床し、この際に送信処理の方を優先したと考えられる。

図 1は RTT の測定結果である。ksoftirqd をリアルタイム優先度にするると、無負荷時の最大 RTT が大きく改善した。TCP 送信負荷がある場合には、ksoftirqd をリアルタイム優先度にすることで最大 RTT は改善したが、平均と標準偏差が悪化した。ksoftirqd のリアルタイム優先度化により受信処理の最悪実行時間が改善された一方で、定周期送信ジッターの測定結果で見られたように、送信処理の実行時間が平均的に増加したのではないかと考える。TCP 受信負荷がある場合には、ksoftirqd の優先度にかかわらずパケットロスが発生して測定が中断したが、リアルタイム優先度の場合にはパケットロスに至るまでの時間が延び、最大 RTT も小さくなった。

4. まとめと今後の課題

Linux において ksoftirqd をリアルタイム優先度にした場合の通信性能を評価した。ksoftirqd のリアルタイム優先度化により最大 RTT が改善した。ただし、TCP 送信負荷を掛けた場合に平均 RTT が増加した。加えて、ksoftirqd のリアルタイム優先度化は、定周期送信のジッターを悪化させることが分かった。今後、これらの性能悪化要因を解析し、対策を検討する必要がある。

本稿では、シングルコア CPU かつ単一 NIC を

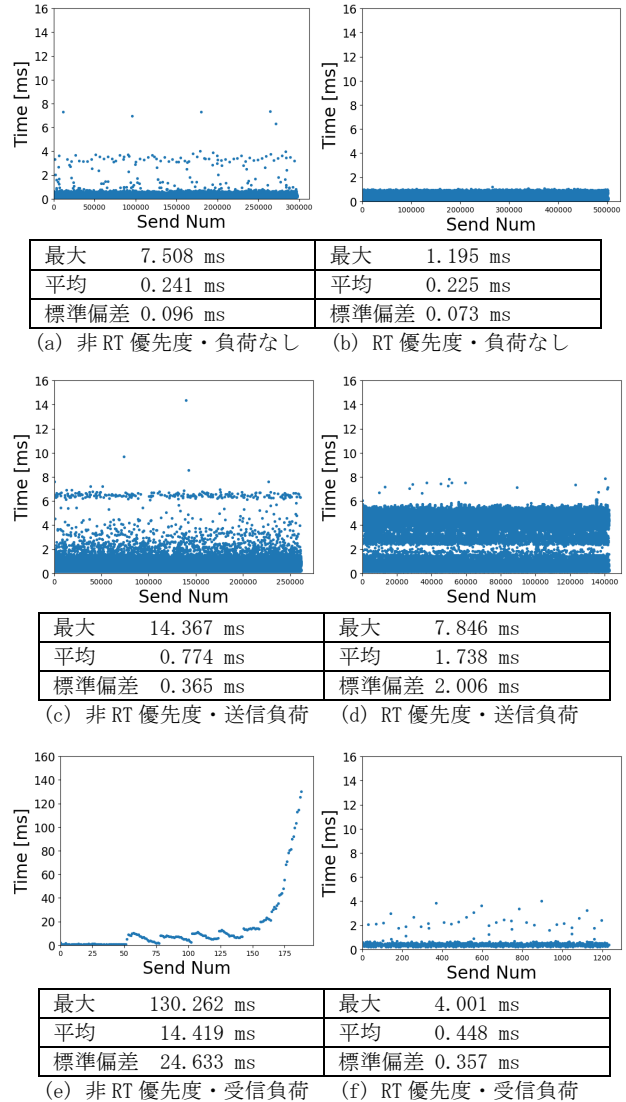


図 1: RTT の測定結果

対象として評価した。ほかに、マルチコア CPU の搭載や、複数 NIC の利用などのシステム構成も考えられる。このような様々な構成で評価することも今後の課題である。

参考文献

[1] Carlos San Vicente Gutiérrez, Lander Usategui San Juan, Irati Zamalloa Ugarte and Víctor Mayoral Vilches: Real-time Linux communications: an evaluation of the Linux communication stack for real-time robotic applications, arXiv:1808.10821v1 [cs.OS] (2018).

[2] Michael M. Madden: Challenges Using the Linux Network Stack for Real-Time Communication, AIAA Scitech 2019 Forum, DOI: 10.2514/6.2019-0503 (2019).