

リアルタイムネットワーク性能測定

 **株式会社マイクロネット**
<http://www.mnc.co.jp>




TEL: +81(0)299-90-1733

FAX: +81(0)299-92-8557

目次

1	はじめに	4
2	TTCP：ネットワーク通信性能測定ツール	4
3	転送性能測定方法	4
4	転送測定結果	5
4.1	送信: UPLOAD (HOST1->HOST2)	5
4.2	受信: DOWNLOAD(HOST2->HOST1)	5
5	INscope でみるネットワーク通信	6
5.1	UPLOAD	6
5.1.1	送信処理詳細	7
5.2	DOWNLOAD	8
5.2.1	受信処理詳細	8
6	改定履歴	11

本書で使用するマークについて

	ノート: 操作方法や手順等の補足情報や注釈を説明しています。
	情報: 製品を利用する上で有効な豆知識となる説明をしています。
	警告: 製品仕様上注意が必要な事象について説明しています。

Windows、Visual Studio は、米国 Microsoft Corporation の米国およびその他の国における商標または登録商標です。

INtime は、米国 TenAsys Corporation の登録商標です。

TenAsys®, INtime®, eVM® and iRMX® are registered trademarks in USA of the TenAsys Corporation.

その他、本書に記載されている会社名、商品名は、各社の商標または登録商標です。

本書の内容を無断で転載することは禁止されています。

※本ドキュメントの内容は予告なく変更される可能性があります。

また、本ドキュメントの無断転載・使用を固く禁じます。

1 はじめに

本ドキュメントでは、INtime 製品(“INtime SDK”, “INtime for Windows”等)におけるリアルタイムネットワークスタックにおけるパフォーマンス測定結果について記述しています。製品利用によるアプリケーション開発時の参考としてお役立てください。

2 TTCP : ネットワーク通信性能測定ツール

本ドキュメントではネットワーク通信パフォーマンスの測定ツールとして TTCP(Test TCP)を使用しています。TTCP は、UNIX システムでよく利用されるユーティリティで、システム間の TCP、UDP ネットワーク通信スループットを測定します。

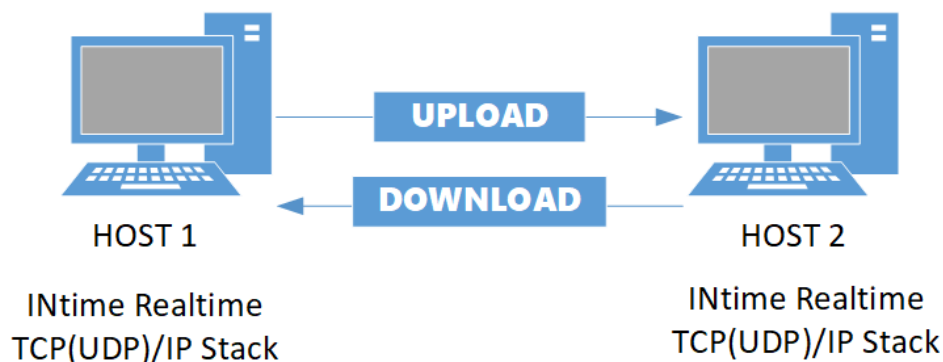
TTCP ユーティリティは様々なプラットフォームに移植されており、本計測では INtime 用に移植した TTCP.RTA 使用しています

TTCP リポジトリ

<https://web.archive.org/web/20091213160955/http://connection.netcordia.com/files/folders/tools/entry103.aspx>

3 転送性能測定方法

測定ターゲットには INtime(INtime for Windows 6.3.19040.2)をインストールし、リアルタイムネットワークスタックにおける送信(UPLOAD)、受信(DOWNLOAD)を計測します:



測定環境

	HOST 1	HOST2
INtime	6.3.19040.2	
Windows	Windows 10 Professional	
CPU	Intel Core i7-7700 3.60GHz	Intel Core i5-2400 3.10GHz
Network	Inte PRO/1000 CT Desktop Adapter	Intel PRO/1000 PT Dual Port Network Connection
	Intel 82574L(0x8086/0x10D3)	Intel 82571EB(0x8086/0x105E)
	PCI Address 6:0.0	PCI Address 1:0.0
	ie1g0	ie1g0
	IP: 100.100.5.80 MASK: 255.255.255.0	IP: 100.100.5.90 MASK: 255.255.255.0

4 転送測定結果

ttcpプログラムにて HOST1(クライアント)->HOST2(サーバ)への通信処理。および HOST2(クライアント)->HOST1(サーバ)への通信処理の測定を行った。

MTU1500 バイト TCP/IP 送信 約 **116 MB/S** 受信 約 **116 MB/S**
 UDP/IP 送信 約 **120 MB/S** 受信 約 **94 MB/S**

4.1 送信: UPLOAD (HOST1->HOST2)

HOST1: piperta ttcp.rta -t 100.100.5.90

HOST2: piperta ttcp.rta -r 100.100.5.90

プロトコル	TCP	UDP
送信サイズ	16MB	
所要時間	140ms	138ms
スループット	115.8 MB/S	121.4 MB/S

4.2 受信: DOWNLOAD(HOST2->HOST1)

HOST1: piperta ttcp.rta -r 100.100.5.80

HOST2: piperta ttcp.rta -t 100.100.5.80

プロトコル	TCP	UDP
送信サイズ	16MB	
所要時間	140ms	150ms
スループット	115.8 MB/S	91.7 MB/S

5 INScope でみるネットワーク通信

INScope Task Analyzer を使用し、tcp によるネットワーク性能測定中のスレッディング状況を確認してみました。INScope を利用することで tcp ツールのネットワーク処理内部状況がわかります:

5.1 UPLOAD

UPLOAD(送信処理)では、連続的に `sendto`(データ送信)処理を行っている様子が伺えます:

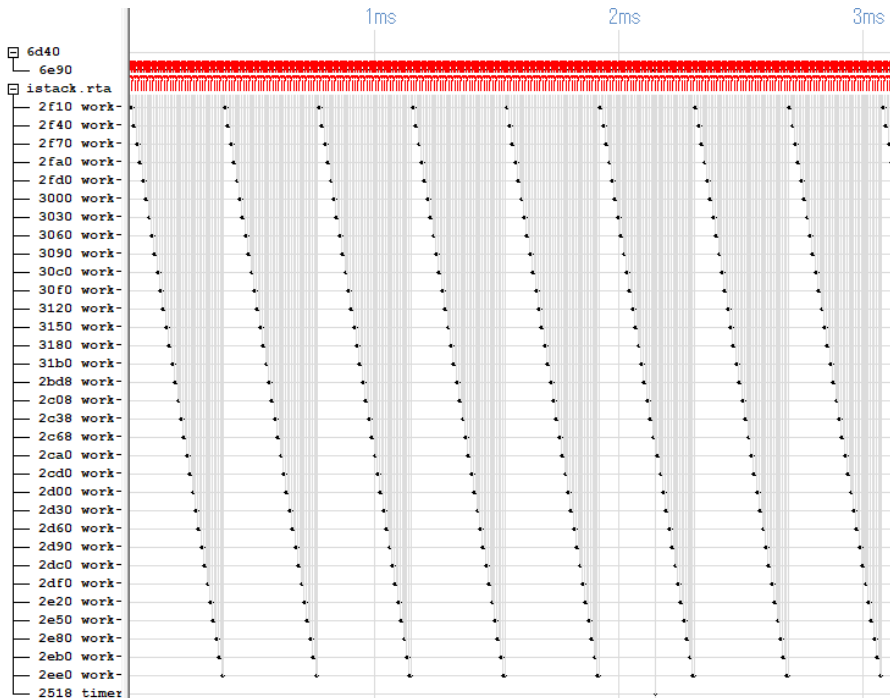
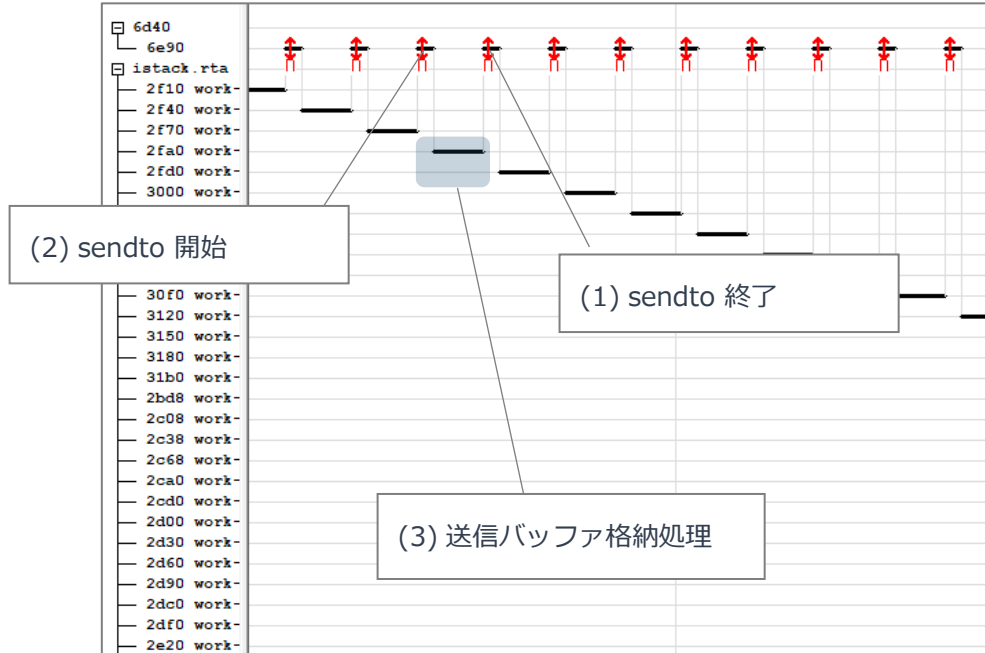


図 1. UPLOAD(送信)処理

5.1.1 送信処理詳細

細かく分割された処理ブロックは、全て `sendto` で構成されており、所定のバイトデータ送信完了まで連続的に `sendto` コールを実施しています。



INscope ログから、以下のようなデータが読み取れます:

測定する時間	測定方法	処理時間
sendto にかかる時間	Sendto コールイン・コールアウト間を測定する	11.0us



送信処理は16MBのデータをMTUに分割し連続的に送信します。sendto 処理ではネットワークスタックの提供する送信バッファにデータを格納して返却するため、INscope 上での送信処理は送信バッファがフル状態にならない限り、連続的に sendto 処理が実行されていることが分かります。



図 2. INscope からわかる送信処理

5.2 DOWNLOAD

DOWNLOAD(受信処理)では、**select** コールによるソケットデータ受信状況を確認から、**recvfrom**(データ受信)処理を行っている様子が伺えます:

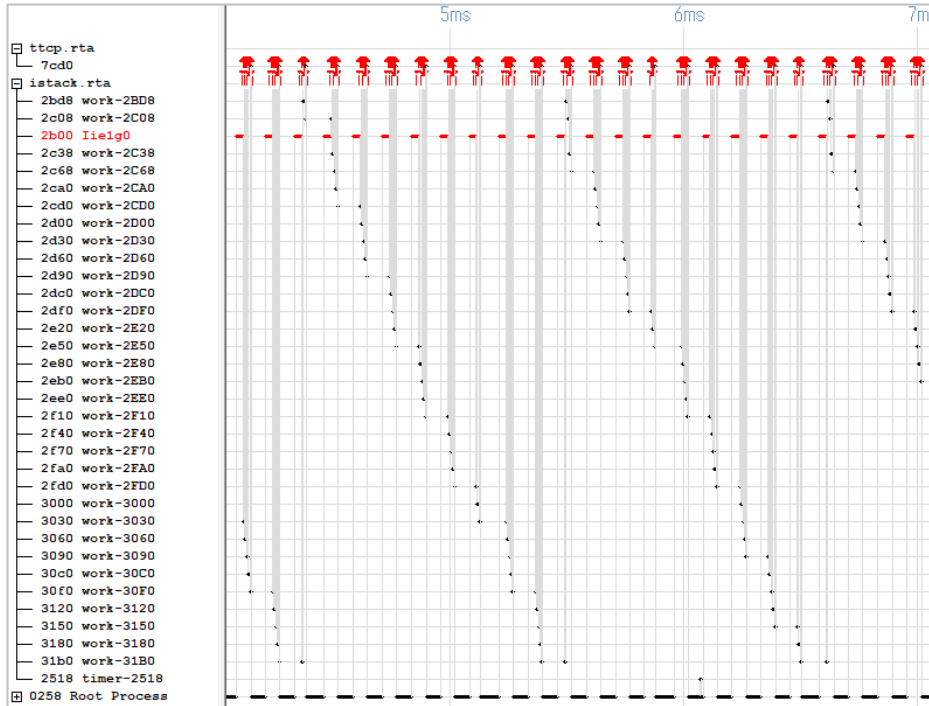


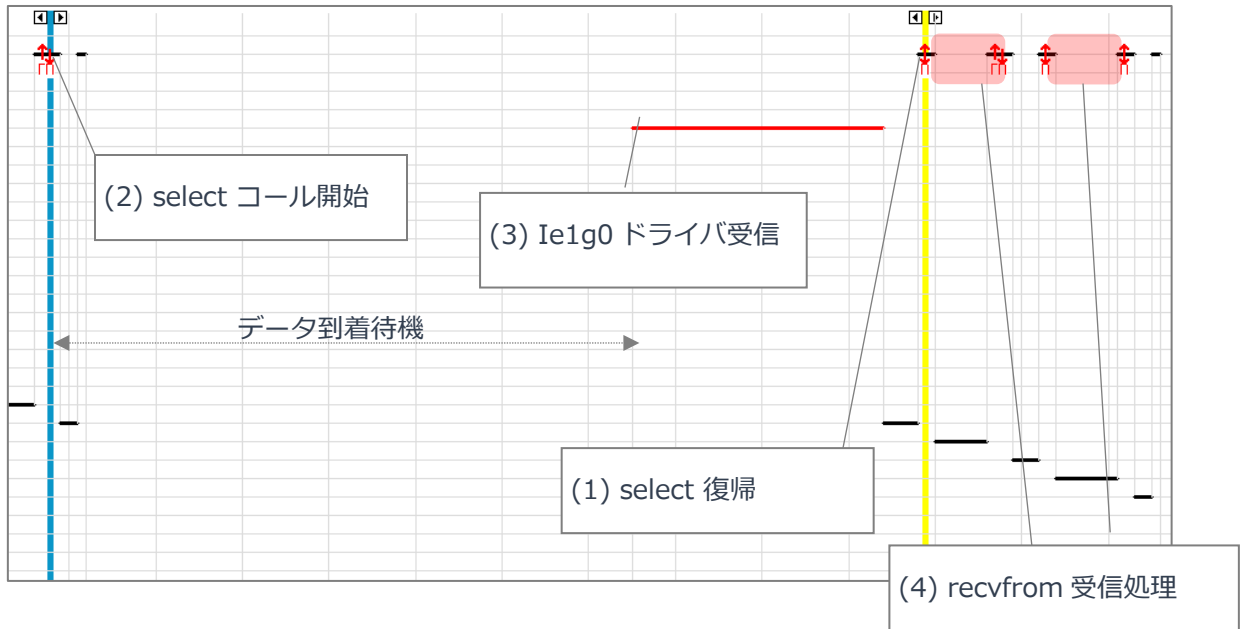
図 3. DOWNLOAD(受信)

5.2.1 受信処理詳細

細かく分割された処理ブロックは、**select - recvfrom** で構成されており、所定のバイトデータを受信するまで以下の処理を繰り返し処理します:

- 1) 一定タイムアウト時間を指定し **select** によりソケットバッファを確認
- 2) ネットワークデバイスドライバ(ie1g0)から受信データを取得
- 3) **select** 処理から復帰
- 4) **recvfrom** 実行、データ取得

リアルタイムネットワーク性能測定



INscope ログから、以下のようなデータが読み取れます：

測定する時間	測定方法	処理時間
select によるデータチェック時間 (バッファにデータがない)	select コールイン・コールアウトを測定する(ドライバ処理を含むケース)	100 us
select によるデータチェック時間 (バッファにデータがある)	select コールイン・コールアウトを測定する(即座に返却するケース)	0.55 us
recvfrom 処理時間	recvfrom コールイン・コールアウトを測定する。	8.1 us
ドライバによる受信処理周期時間	Iie1g0 受信処理スレッドの起動周期を測定する	125 us



送信処理は 16MB のデータを MTU に分割し連続的に送信します。受信処理は既にソケットバッファに格納されたデータを受信(recvfrom)しますが、バッファに格納されていない場合、select にて到着を待機します。このため select 処理時間は、バッファ格納時と未格納時で大きく開きがあります。

およそ 125us 周期でドライバが動作するという事は送信された連続的のデータを受信し、ソケットバッファに格納していることとなります。

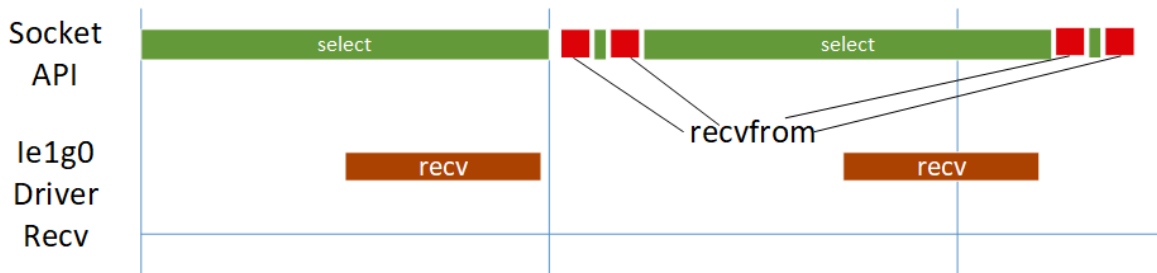


図 4.INscope からわかる受信処理

6 改定履歴

版数	発行日	改定内容
第 1 版	2019 年 8 月 12 日	初版発行
第 2 版	2019 年 10 月 11 日	転載禁止追記