

## (11) スーパー SINET 速度測定

江本 雅彦 核融合科学研究所

### 概要

遠隔ステーション設置直後より、スーパーSINET によるファイルの転送が期待通り速度が出ないことが報告されていた。ここでは、その問題点を探るとともに、解決策を探した。

### 測定方法

核融合科学研究所と京都大学エネルギー理工学研究所双方の Linux PC を利用し、ネットワークの転送速度を測定した。使用した PC のスペックを表 1 に示す。

表 1 測定に使用した Linux PC の仕様

	A	B
設置場所	核融合科学研究所	京都大学エネルギー理工学研究所
CPU	Xeon 2GHz × 2	Pentium 4 2GHz
Memory	2GB	1GB
NIC	BROADCOM BCM5701	Intel Pro 1000
OS	LASER5 Linux 7.2	Turbo Linux 7

### 測定結果

#### 測定 1 : ping による応答時間

ping コマンドにより核融合科学研究所、京都大学間の応答時間を測定する。TCP プロトコルはパケット毎に到達の可否を確認しながら送受信するプロトコルであるので、この時間が長いと通信速度が遅くなる。ping コマンドの応答時間は次の通り。

A - - > B	10.5(ms)
B - - > A	10.5(ms)

この結果、例えば、パケット長が 1500 バイト(Ethernet の MTU)の場合、転送速度は  $1500 / 10.5(\text{ms}) = 14.3 \text{ KB/s} = 1.14 \text{ Mbps}$  が限界となる。

#### 測定 2 : netperf による UDP ストリームの測定

netperf コマンドにより京都大学から核融合科学研究所間の UDP での転送速度を測定した(表 2)。この結果、約 700Mbps のスループットが得られた。これは 1Gbps のネットワークとしては妥当な数字であると思われる。

表 2 B - - > A UDP 転送速度測定

Socket Size (bytes)	Message Size(bytes)	Throughput (M bits/s)
65536	64	136
65536	1024	629.26

65536	1472	688.93
-------	------	--------

### 測定 3： netperf による TCP ストリームの測定

TCP ではパケットの到達を確認しながら通信を行うため、応答時間が長い場合、一パケット毎到達を確認すると著しく速度が遅くなる。そのため、TCP ではスライディングウィンドウという方法を取り、パケットの確認を遅延させ、スループットを上げることができる。この試験ではウィンドウのサイズを変えることにより、TCP の速度の影響を調べた。また、TCP の送受信にはソケット通信が行われるが、ソケットの送受信バッファのサイズを大きくすることで、一度に送るデータを大きくしオーバーヘッドを小さくすることができる。今回の試験ではソケットバッファの依存度も調べた。

TCP のウィンドウサイズを変えた試験では、デフォルトのサイズが最も高速であり、これはシステム側で最適な値に最適化されているものと思われる。一方、ソケットバッファのサイズ変更は大きく速度に寄与し、2MB までとることにより、600Mbps 以上の速度が得られた。ウィンドウサイズがデフォルトの値の時、ソケットサイズとスループットの関係を図 1 に示す。

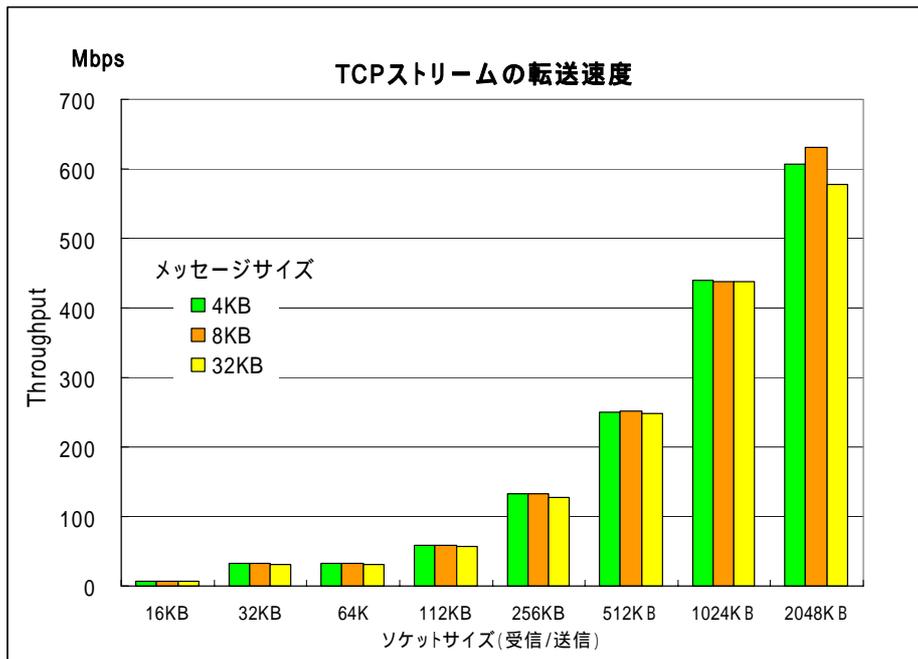


図 1 TCP ストリームのスループット測定

### 測定 4： FTP による転送速度

上記の試験でソケットバッファを大きくすることで速度が向上することがわかった。この設定で ftp により、A の 200MB のファイルを B の RAM ディスク上にコピーし、転送速度を測定した。

ソケットサイズデフォルトの時	45.2 Mbps
システムソケットサイズ 2MB の時	47.1 Mbps

この結果、システムの最大ソケットサイズには影響を受けないことがわかった。これはソフトウェア内部でソケットサイズの設定を行っており、システム側の最大ソケットサイズを利用するためには、プログラムの変更が必要になると思われる。

### 測定 5 : bbftp による転送速度

bbftp は巨大なファイルの高速転送を行うためのファイル転送ソフトウェアで、複数のストリームを同時に使用し、ファイルを転送することができる。この試験ではストリーム数と転送速度の影響を調べた。使用したファイルは 200MB で、ftp のテストと同様に RAM ディスク上に書き込んだ。結果を表 3 と図 2 に示す。

表 3 転送方向 B - - > A、ソケットサイズ=2MB

ストリーム数	速度(Mbps)
1	227.00
2	388.06
4	499.63
8	407.35

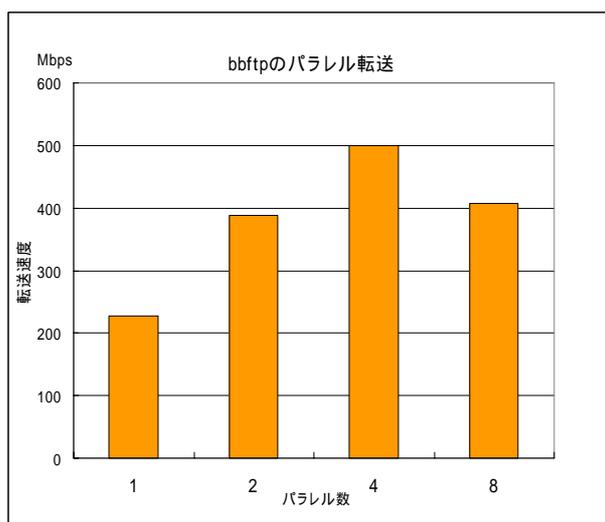


図 2 bbftp のパラレル数と転送速度

この結果から、ストリーム数を 4 にすることで 500Mbps 近い速度が得られた。ストリーム数を 8 にすると逆に速度が低下した。オーバーヘッドが原因かもしれない。最適なストリーム数はファイルの大きさ等によると思われる。

### 測定 6 : ftp の複数セッションによる転送速度

bbftp の結果から複数セッションを張ることにより、全体としての速度が上がるのがわかった。さらに、ftp を同一 PC で複数実行することにより全体との速度がどのように変化するかを測定した。測定は 200MB のファイルの転送を 5 回繰り返すというプロセスを同時に複数プロセス実行し、全プロセスの実行時間から全体としての転送速度を測定した。測定結果を表 4 と図 3 に示す。ftp はバッチ的に起動し、バックグラウンドで実行させている

ことから、全プロセスはほぼ同時に実行されていると期待される。

結果：ftp を複数実行することで全体としての転送速度が上げられることが確認できた。  
なお、inetd や ftp により、コネクション数等制限があるので、複数セッション実行できるように変更しなければならない。

表4 ftp のプロセス数と転送速度

プロセス数	転送速度 B - > A (Mbps)	転送速度 A - > B (Mbps)
1	45.1	24.4
2	90.6	48.3
4	180.5	92.9
8	350.9	155.8
16	635.6	265.3
20	412.3	287.0
32	NA	311.5
64	NA	335.4

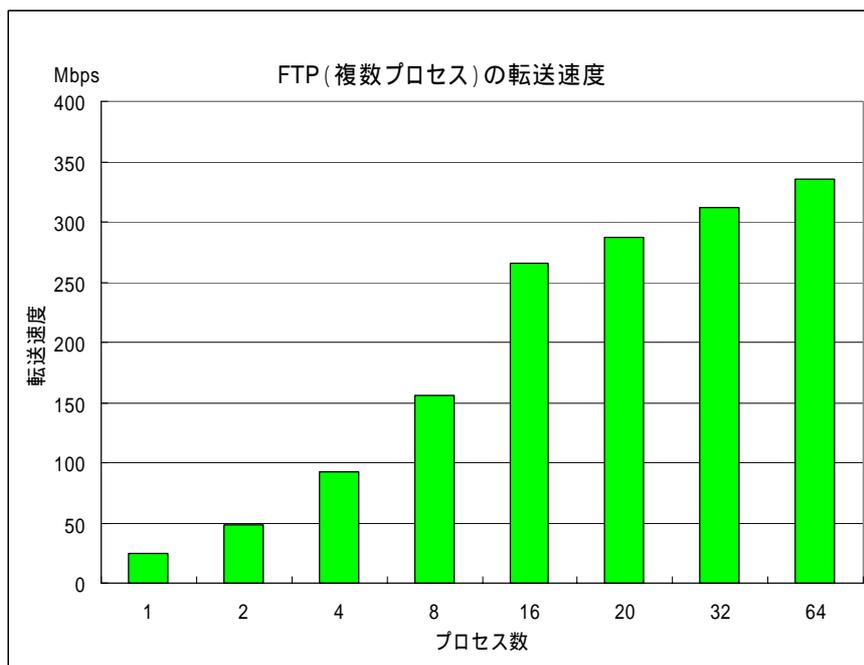


図3 FTP (プロセス数) 転送速度

## 結論

UDP では 700Mbps 程度の速度が得られ、この値は 1Gbps のネットワークでは妥当なものと思われる。TCP を利用した FTP 等で速度が得られないのは、TCP の原理上不可避であると考えられる。TCP のウィンドウサイズを変更することで速度が上がることを期待されたが、十分な速度を得ることができなかった。しかしながら、短期的な方策として、

- bbftp でストリーム数を増やす
- ftp を複数起動する

等の方法で、ストリームを増やすことにより全体としての速度を上げることが可能である。  
より長期的には、ネットワーク事情を考慮した転送プロトコルの開発が必要であろう。