

## 長距離超高速インターネットの特性解析

天海良治\*, 村上健一郎\*, 釘本健司\*\*, 岡敦子\*\*, 伊藤正樹\*\*, 後藤滋樹\*\*, 伊藤光恭\*\*

NTT 基礎研究所\*, NTT ソフトウェア研究所\*\*

長距離超高速専用線を使用したインターネット接続実験を行なった。東京・京都間 約880Km (光ファイバ長) を同期デジタルハイアラキ SDH (Synchronous Digital Hierarchy) の STM-1 (Synchronous Transfer Mode) 1, 速度 155.520Mbit/sec の専用線で結び, 計算機プロセス間で TCP/IP によるネットワーク接続を構築し, 長距離超高速ネットワークにおける1本のストリームの (多重化していない) 通信の特性を解析した。その結果, 光速の限界による伝送遅延が通信の速度に大きく影響を与えていることが確認された。本稿では, 高速プロセス間通信のボトルネック, 伝送遅延の影響, 伝送速度に見合った通信速度を得るための指針等, 実験によって得られた知見, 指針を述べる。

### 1. はじめに

計算機ネットワークは, 接続計算機の数と回線速度の両面で急速に拡大している。アプリケーションの面でもナンバーランチャ, ビジューライゼーション, 大規模データベースといった特定処理向けのスーパーコンピュータをネットワーク接続して分散処理し大きな効果を上げている。これは単に資源の最大活用だけでなく, 不可能と考えられていたことが可能になるという本質的展開を意味する。

各所に散在した高性能計算機を相互接続するためのインターネットワークは, 大量のデータを長距離にわたって送受する長距離超高速ネットワーク (Long Fat Pipe: LFP) によるインターネットでなければならぬ。物理回線としては, 155Mbit/sec や 600Mbit/sec の速度をもったものが利用可能となってきた。だが, 長距離超高速インターネットでは, 比較的短距離低速のこれまでのインターネットとは異なり, 回線速度に見合ったプロセス間通信速度を得ることが難しい。こ

れは物理的な応答の遅延の影響が無視できなくなるためである。

われわれは, 長距離超高速インターネット研究の一環として, 長距離超高速専用線を使用したインターネット接続の実験を行なった。これは, 東京大学医科学研究所と京都大学化学研究所の協力を得, 東京・京都間の約880Km (光ファイバ長) を同期デジタルハイアラキ SDH (Synchronous Digital Hierarchy) [1] の STM-1 (Synchronous Transfer Mode) 1, 速度 155.520 Mbit / sec の専用線で結び, 計算機プロセス間で TCP/IP によるネットワーク接続を構築し, 1本のストリームの (多重化していない) 通信の特性を解析した。その結果, 物理的な応答遅延の影響により, 回線速度に見合った通信速度が得られないことが明らかになった。

本論文では, まず長距離超高速の伝送線における伝送遅延の影響について述べ, 次に, われわれが行なった接続実験とその結果について説明し, 伝送遅延の影響を示す。特に LFP シンドロームと呼ぶ現象について説明し, 次にこれを改善するための方式について述べる。

### 2. 伝送遅延が通信速度に与える影響

本章では, 確認応答を伴うような計算機間プロセス通信において, 物理的な伝送遅延が通信速度に大きく

An analysis of Behavior in a High-speed Long-distance Internet  
Yoshiji Amagai\*, Ken-ichiro Murakami\*, Kenji Kugimoto\*\*  
Atsuko Oka\*\*, Masaki Itoh\*\*, Shigeki Gotoh\*\*, Mitsutaka Itoh\*\*  
\*NTT Basic Research Laboratories  
\*\*NTT Software Laboratories

影響することを示す。

## 2.1 確認応答を伴うプロセス間通信と応答時間

プロセス間通信では、メッセージの送信とその確認応答 (Ack) が一組となって処理が進む。下層レイヤでのバケット単位での送信と応答だけでなく、アプリケーションレイヤでの種々のネゴシエーションプロトコルや、ユーザインタフェースでの計算機からのプロンプト出力とそれに従ったユーザの入力、といったように、互いに同期をとる例は多い。このようにメッセージに依存関係があるときには、応答が返ってくるまでは次の処理には進めない。よって、応答時間 (メッセージを送信してから、確認応答が戻るまでの時間) が全体の処理の速度に大きく影響する。

応答時間は、送信側から受信側へバケットが届く通信時間、受信側での処理とそれに伴うオーバーヘッド、受信側から送信側への通信時間の3つの部分からなる。受信側でのオーバーヘッドは、ヘッダの解析、チェックサムを計算して伝送誤りの有無を確認する、などである。インターネットで使用されているTCP/IP (Transmission Control Protocol / Internet Protocol) プロトコル<sup>[1]</sup>では、CPUの高速化、ソフトウェアの最適化、プロトコル処理の一部のハードウェア化などにより、このオーバーヘッドはごく小さくなっている。具体的には、ヘッダ処理が1msec以下、チェックサム計算が1バイトあたり330 μ sec程度である<sup>[2]</sup>。よって、応答時間に往復の通信時間が占める割合が大きくなる。通信時間には、メッセージを伝送路に送りこむ時間 (メッセージ転送時間) とメッセージが実際に目的地に届く時間 (伝送遅延) の2つの部分がある。前者はメッセージの長さや伝送路の帯域による値だが、後者は2つの伝送路の長さによる。以下、平均応答時間を具体的に求め、伝送遅延の影響を示す。

## 2.2 通信時間と伝送遅延

まず、通信時間を決めるメッセージ転送時間と伝送遅延について考察する。

Nバイトのメッセージを光ファイバーで送るとする。メッセージが光ファイバーの入口にはいり始めてから、もう一方の出口から出終るまでの時間をTとする。

Tは以下のようになる。

$$T = \frac{L}{c} + \frac{8N}{B}$$

Lは光ファイバーの長さ、cは光ファイバー中の光速

でここでは  $2.1 \times 10^8$  m/sec とする。Bは伝送路の帯域 (速度) で bit/sec で表わされる。式の第1項はメッセージの先頭が光ファイバーを通過する時間、第2項は、光ファイバーの出口において、メッセージの先頭が到着してからメッセージの最後がファイバーを出るまでの時間である。

L = 1000Km とすると、第1項、すなわち光ファイバーという伝送路の伝送遅延は約 4.8msec となる。N = 128byte としたとき、メッセージ転送時間は表1に示したようになる。

表 1. 帯域と転送時間

帯域 (Mbit/sec)	時間 (msec)
1.5	0.68
155	$6.61 \times 10^{-3}$
660	$1.55 \times 10^{-3}$
2400	$0.43 \times 10^{-3}$

このように、伝送遅延に比べてメッセージの転送時間はごく小さい。LFPでは、応答時間が転送速度ではなく、伝送遅延、すなわち、光速によって決まること、がわかる。実際には、長距離伝送路では途中に中継器をいれる必要があり、伝送遅延はさらに大きくなる。

## 2.3 平均応答時間と伝送遅延

次に、これにバケット処理オーバーヘッドを加味する。値としてはJ.Kayらの測定値<sup>[2]</sup>を参照するが、200MIPSクラスのCPUを仮定する。メッセージの到着をポアソン分布とすると、処理オーバーヘッドは待ち行列理論のサービス時間にあたる。Nバイトメッセージのサービス時間として、平均  $93 + 0.34N$  μ sec の指数型サービス時間を想定すると、平均応答時間Rは以下のように表わされる。

$$R = \frac{(93 + 0.34N) \times 10^{-6}}{(1 - \rho)} + \tau$$

$\tau$ は往復の遅延時間で、

$$\tau = \frac{2L}{c} + \frac{8N}{B}$$

である。この時間をRTT (Round Trip Time) と呼ぶ。確認応答は極めて短いメッセージなので、メッセージ転送時間は無視できる。L=1000Kmとすると2.2節で示したように $\tau$ は、B、Nによらず約9.6msecである。N=128byteとすると、平均サービス時間は $\rho$ が

0.9の場合でも1.37msecであり、平均応答時間は $\tau$ に支配されることがわかる。将来、さらに高速なCPUを得、ソフトウェアを改良し、高速の伝送路が利用できるようになれば、平均サービス時間とメッセージ転送時間は短くなるが、 $\tau$ のうちの $2L/c$ は短くなりようがない。

メッセージ送信とその確認応答という互いに時間的に依存したメッセージのやりとりで進む処理形態は、計算機能力や伝送路の容量に関わらず速度の限界に達している。この限界を乗り越えるには、予測による先行処理、確認応答を要しないエラー回復、互いが協調して予測することによる通信をしない通信、といった新手法を作り出す必要がある。

## 2.4 ウィンドウ制御と伝送遅延

次に、LFPにおけるプロセス間のデータの転送速度(スループット)と伝送遅延の関係を、TCPを例として述べる。

ファイル転送といった大きなデータを送る場合、原理的にはファイル1つ分を1つのメッセージと考えれば、これをすべて送ったのち、確認応答が1つ戻ればよい。だが、下位のレイヤにおいては、ファイルはいくつかのペケットに分かれて送信され、ペケットの送信ごとに確認応答が戻る。すべてのペケットの転送が完了したら、上位のレイヤでは、ファイルの転送が完了した、という確認応答が1つ戻ることになる。このような場面では下位のレイヤのペケット同士には依存関係がないので、確認応答が戻る前に次のペケットを送るバイブライニング処理が可能である。

TCPプロトコルでは、この先送りの制御をウィンドウ制御と呼ぶ。TCPでは送信データにバイト単位に順序番号(Seq番号)をつけて送信する。受信側では、確認応答として受理した順序番号の次の番号(Ack番号)を返す。すなわち、次に受信側が受取りたい番号を返す。送信側はこの確認応答を待つ。一定時間内に確認応答が戻らなければ、エラーがあったとみなして再送する。ここで、受信側が確認応答を返すときに、Ack番号のデータからあるデータ量を応答なしに先送りしてよいかを判断し、この量を確認応答に含めて返す。このデータ量は受信側での空きバッファ量等から決める。この先送りの量をウィンドウサイズと呼ぶ。

ウィンドウ制御は、処理のバイブライニング化によるスループット改善とともに、受信側のバッファの枯渇な

ど場合にウィンドウサイズを小さくすることにより送信を抑制する、フロー制御のためにも使われる。

伝送路の速度を使いきるには、ウィンドウサイズ分のデータで伝送路を満たす必要がある。伝送路には、帯域 $B \times RTT$ 分のデータがのる。よってウィンドウサイズ $W$ (バイト)は少なくともこれより大きい必要がある。すなわち、

$$W \geq \frac{B}{8} \times RTT$$

逆に、RTT時間内にはウィンドウサイズ分のデータしか送れないので、スループットは次のようになる。

$$\min\left(\frac{8W}{RTT}, B\right)$$

伝送路の帯域 $B$ がある程度大きいときは、RTTすなわち伝送路の長さやウィンドウサイズからスループットが決まる。伝送路として155Mbit/secの光ファイバーを使ったとき、ファイバーの長さやスループットの関係を表2に示す。ここでウィンドウサイズは4Kbyteとした。この値は、BSD(Berkeley Software Distribution)UNIXのTCP/IPインプリメンテーションのデフォルト値である。

表2. ファイバー長と転送速度

距離(Km)	RTT(msec)	スループット(Mbit/sec)
1000	9.6	3.4
3000	28.6	1.15
14000(日米間)	133	0.246

これからわかるように、プロセス間の距離が1000Kmのとき、155Mbit/secの伝送路を使っても3.4Mbit/secのスループットしか得られない。

逆に、155Mbit/secの速度を使い切るには、190Kbyteのウィンドウサイズが必要である。計算では、ウィンドウサイズに比例してスループットが向上する。だが、LFPでは、ウィンドウサイズを大きくしてもスループットは得られないことが実験によって明らかになった。これは、従来のウィンドウ制御アルゴリズムがLFPには適用できないためである。特にウィンドウサイズがある量より大きくなったところでスループットが特に低くなる現象が起きる。我々はこれをLFPシンドロームと名づけた。

### 3. LFP シンドローム

本章では、実験によって得られたウィンドウサイズとスループットの関係を解析し、LFP シンドロームの原因を追及する。

#### 3.1 長距離超高速インターネット接続実験

##### 3.1.1 実験環境

実験は図1の環境で行なった。東京と京都のルータ間は STM-1 (Synchronous Transfer Mode) 1, 速度 155.520Mbit/sec で結ばれている。バックボーンのルータには、STM-1 を VC-4 (Virtual Circuit 4) のままで利用するものと、VC-4 を3つの VC-3 に仮想的に分割して利用するものの2種類を用意した。今回は後者を利用した。これは試験用トラフィックの生成に 100Mbit/sec の転送速度をもつ FDDI (Fiber Distributed Data Interface) に接続したワークステーションを用いたため、100Mbit/sec 以上の帯域をもったパスでは帯域限界付近の特性が測定できないためである。

FDDI を選択したの理由は、FDDI に適用できる優秀なプロトコルアナライザが利用できること、速度が公称値 100Mbit/sec のうち 90Mbit/sec 以上の速度が実際に得られること、である。ATM (Asynchronous Transfer Mode) インタフェースなど公称値がもっと大きいものがあるが、実効値は FDDI を下まわることが

ある。これは、FDDI の実現技術が成熟しており、実効値が理論性能に近ずいているためである。

各 VC-3 には DS (Digital System) 3 倍号 (約 45Mbit/sec) を通し、ルータのダイナミックロードバランシング (Dynamic Load Balancing) によってこれらの DS3 倍号を IP レイヤで逆多重化している。

##### 3.1.2 測定条件

測定は、2 台のワークステーション間に TCP/IP による接続を確立し、この 1 本のストリーム上でメモリ間データ転送を行なった。実験には我々で作成した rttcp と呼ばれるプログラムを使用した。rttcp は送り側、受け側の計算機でそれぞれプロセスを生成し、いくつかのパラメータを互いで設定したのち、指定されただけの量のデータを転送し、転送速度などの情報を表示する。rttcp 実行例を図2に示す

今回の測定では、約 4Kbyte のデータを 20000 回のシステムコールで送りだした。上位レイヤのプロトコルはないので、ウィンドウサイズによるフロー制御に従ってパケットが送り出される。rttcp が表示する統計情報と共に、rttcp による転送パケットを FDDI プロトコルアナライザによって収集し、RTT の値、Seq 番号、Ack 番号の推移を解析した。今回は以下のパラメータを使って実験を行なった。

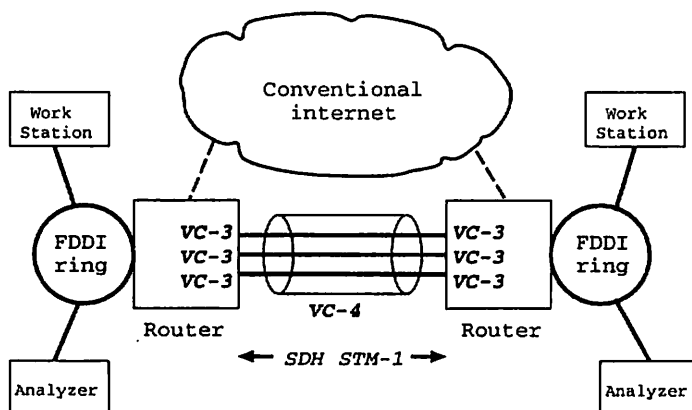


図1. 実験環境

```

aristate 2% rttcp
rttcp-t:  buflen=8192, nbuf=100000, align=16384/0, port=5001, sockbufsize=524288
rttcp-t:  sndbuf
sockname:  family=AF_INET addr=fddi-aristate port=1107
peername:  family=AF_INET addr=fddi-damaskrose port=5001
SO_DEBUG : off SO_REUSEADDR : off SO_KEEPALIVE : off
SO_DONTROUTE : off SO_BROADCAST : off SO_USELOOPBACK: off
SO_OOBINLINE : off SO_REUSEPORT : off SO_LINGER : off
SO_SNDBUF : 524288 SO_RCVBUF : 61440 SO_SNDLOWAT : 0
SO_RCVLOWAT : 0 SO_SNDTIMEO : 0 SO_RCVTIMEO : 0
SO_ERROR : 0 SO_TYPE : 2
rttcp-t:  819200000 bytes in 68.89 real seconds = 90.72 Mbit/sec +++
rttcp-t:  819200000 bytes in 15.37 CPU seconds = 406.74 Mbit/cpu sec
rttcp-t:  100000 I/O calls, msec/call = 0.71, calls/sec = 1451.49
rttcp-t:  0.8user 14.5sys 1:08real 22% 94maxrss 0+0pf 38091+0csw

```

図 2. RTTCP 実行例

1. ウィンドウサイズ、8Kbyte から 520Kbyte
2. MTU (Maximum Transmission Unit) は 4352byte
3. 1 回の転送量は約 80Mbyte
4. TCP ストリームが使える帯域は、最大 45Mbit/sec (DS3)
5. 光ファイバー長は約 880Km

負荷の少ない状態でのルータ間の RTT の実測値は 8～10msec であった。光ファイバー長が約 880Km であるから、光速を  $2.1 \times 10^8$  m/sec とすると RTT は約 8.4msec となり、実測値と合っている。

### 3.1.3 測定結果と解析

ウィンドウサイズとスループットの関係を図 3 に示す。これから、ウィンドウサイズが 100Kbyte 以下の領域、100Kbyte から 170Kbyte の領域、170Kbyte 以上の領域の 3 つの領域で状態が異なることがわかる。

1. ウィンドウサイズが 100Kbyte 以下の領域

この領域では、ウィンドウサイズが帯域と RTT の積より小さいので、ウィンドウサイズの増加に従ってスループットは線形に大きくなる。スループット増大の割合は、2.4 で述べたように  $1/RTT$  である。プロトコルアナライザで測定したウィンドウサイズ間の RTT の実測値は約 12msec であった。この値による計算値を図の破線で示した。ウィンドウサイズが 16Kbyte 以下の領域ではスループットが特に悪く、計算値とも大きく異なる。この原因は解析中である。

2. ウィンドウサイズが 100Kbyte から 170Kbyte の領域

この領域では安定した転送が行なわれている。転送速度もほぼ回線の帯域と同じである。確認応答の packets がちょうど送信 packets と同じタイミングで戻ってくる。パイプライン効果が最大限に働いている。

3. ウィンドウサイズが 170Kbyte 以上の領域 (LFP シンドロームの発生)

この領域では、ウィンドウサイズが帯域と RTT の積より大きくなる。ホストは回線帯域以上にデータを送りこもうとする。だが、TCP プロトコルは回線の帯域を知らない。今回の実験は、ワークステーションは 100Mbit/sec の FDDI に接続されているが、経路には 45Mbit/sec の回線が含まれている。また、TCP の下位の IP レイヤでもって IP packets 単位で複数の経路を通して送ることもある。さらに、同一回線上の他の接続によるトラフィックもあるので、TCP がある瞬間の利用可能な帯域を知ることは不可能である。そこで、TCP は packets の消失の検知をもって、利用可能な帯域を超えたとみなす。TCP はストリームごとに利用可能な帯域を保持している。これを輻輳ウィンドウ (Congestion window) と呼ぶ。packets 消失を検知すると、輻輳ウィンドウを絞る (Congestion Avoidance アルゴリズム<sup>[5][6]</sup>)、回線に送りこむデータ量を抑制する。TCP には輻輳の解消を知る手だてもないので、輻輳ウィンドウを徐々に広げ (Slow-start アルゴリズム<sup>[5]</sup>)、利用可能帯域の最大値を探る。この繰り返しにより、回線の状況に応じた最大の帯域を求め、効率を上げる。だが、実験の結果が示すように LFP ではこれが

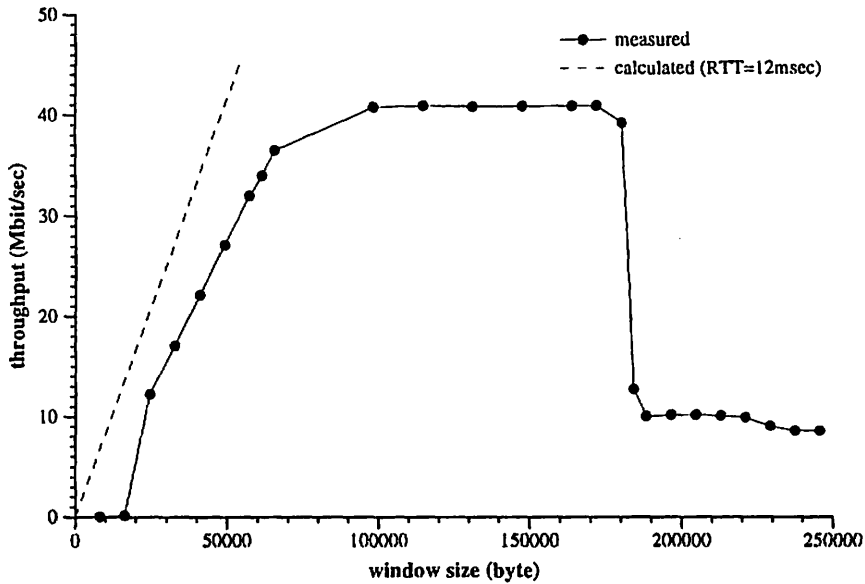


図 3. ウィンドウサイズとスループット

うまく働かず、スループットが 8Mbit/sec 程度にまで低下した。

### 3.2 LFP シンドロームの解析

LFP シンドロームの発生している状況をアナライザで収集した情報によって解析した。送信パケットの TCP seq 番号と確認応答パケットに含まれる Ack 番号の推移を図 4 に示す。図 4 で経過時間 0.2 秒の点から 1.5 秒の点まではパケットはまったく送られていない。このためトータルのスループットが低下した。これは、Congestion Avoidance アルゴリズムと Slow-start アルゴリズムがうまく働いていないことを表わしている。アナライザによる情報によれば、0.2 秒の時点で、seq 番号と ack 番号の差は約 270000 である。この時点で RTT は約 300msec になっていた。これだけのデータ量が伝送路、および、途中のルータのバッファに保持されていた。また、seq 番号の増加の傾きは約 65Mbit/sec、Ack 番号の増加の傾きは約 44Mbit/sec であった。伝送路の途中の帯域は 45Mbit/sec であるが、ワークステーションには 100Mbit/sec の FDDI が接続されているので、送り側がデータを大量に送りこみ、輻輳を招いた。問題は、輻輳が起きたあとの送信の戦略アルゴリズムが LFP

に適応できなかったことである。

LFP シンドローム発生の原因は以下のように、LFP においてはパケット消失の検知、これに伴う再転送に時間がかかることである。

- 1 LFP では RTT が大きいため、パケットの消失を検知するまでのタイムアウトが長くなり、長い間パケットの送信が停止してしまう。図 4 では転送が 0.2 秒継続したあと、確認応答が返ってこないため、ウィンドウが尽きて送信を停止するが、消失を検知して転送を再開するまで 1.3 秒かかっている。
- 2 RTT が大きいため消失を検知するまでに多くのパケットを送り出してしまう。
- 3 パケット消失が発生すると、TCP プロトコルの再転送アルゴリズムにより、消失したパケット以降のすべてのパケットが再転送される可能性がある。LFP では、ウィンドウサイズが大きいため、再転送されるパケットも多くなり、スループットが低下する (TCP の選択的確認応答オプションはここでは仮定しない)。

これらの問題を根本的に解決するには、

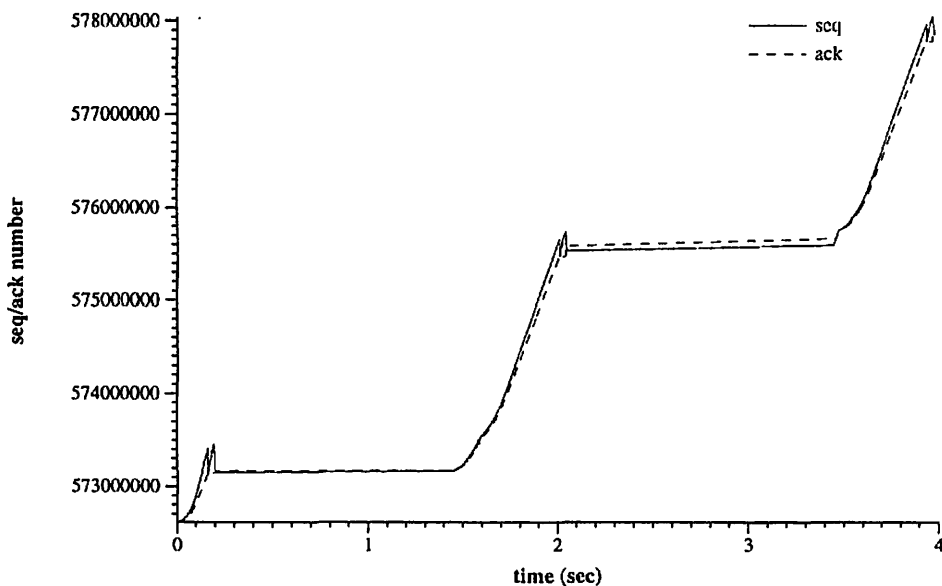


図 4. Seq/Ack 番号の推移

予測による先行処理, 確認応答を要しないエラー回復, 互いが協調して予測することによる通信をしない通信, といった新手法を作り出す必要がある。これは今後の課題である。

### 3.3 LFP シンドロームの抑制

Congestion Avoidance アルゴリズムと Slow-start アルゴリズムでは, パケット消失を検知して輻輳ウィンドウを絞ったあと, もとの帯域の半分にあたるまでは輻輳ウィンドウを指数的に広げ, 半分以上を超えてからは線形に広げていくようになっている。これに対し, 以下のように2つのパラメータ  $x$  と  $y$  を導入する。

まず, もとのアルゴリズムでは, パケットの消失を検知した場合には, まず輻輳ウィンドウを絞る。具体的には TCP の最大セグメントサイズ MSS (Max Segment Size) [4] とする。その輻輳ウィンドウ  $cwnd$  の半分以上を閾値  $ssthresh$  としていた (ただし, 受信側から通信してきた先送り用ウィンドウサイズ  $window$  と比較して小さいほうを選択する。また, 最小値は MSS の2倍である)。

$$ssthresh = \max(\min(cwnd/2, window), 2 * MSS);$$

$cwnd = MSS;$

これに以下のように  $ssthresh$  を調整するパラメータ  $x$  を導入する。

$$ssthresh = \max(\min(cwnd/x, window), 2 * MSS);$$

$$cwnd = MSS;$$

次に, 確認応答パケットを受信したとき, 閾値よりも現在の輻輳ウィンドウが小さければ, 輻輳ウィンドウを MSS だけ増やす。これによって, 指数的に輻輳ウィンドウが広がる。さもなければ, 輻輳ウィンドウを線形に広げる。この場合, RTT あたり MSS 分だけ広がるように, 1 個の確認応答あたり  $MSS/cwnd$  だけ増加させる。もとのアルゴリズムは以下のようになっている。

```
if (cwnd < ssthresh)
  cwnd += MSS;
else
  cwnd += MSS**2/cwnd + MSS/8;
```

この傾きを調整するためにパラメータ  $y$  を導入する。

```
if (cwnd < ssthresh)
  cwnd += MSS;
else
  cwnd += MSS**2/cwnd + MSS/y;
```

さらに、この2つのパラメータに加え、転送が停止した時間が経過時間に占める割合を示す評価用パラメータ  $S$  を導入する。これが 0.5 以上のときは上記の  $x$  と  $y$  が不適切なために、輻輳ウィンドウを広げる割合が大き過ぎるものとみなす。そして、まず  $y$  を増加させる。これにより、輻輳ウィンドウを線形に広げる領域での傾きを小さくする。それでも  $S$  が 0.5 を越えるときには  $ssthresh$  の値、すなわち輻輳ウィンドウを指数的に増加させる期間が大き過ぎるとみなし、 $x$  を大きくして閾値を下げる。逆に  $S$  が 0 に近いときには輻輳ウィンドウの増加を抑制しすぎているとみなし、 $x$  と  $y$  を標準の 2 と 8 に近づける。

RTT は転送が停止している時間に反映されるので、LFP における RTT が大きいという特性は  $S$  に反映される。また LFP の回線帯域はウィンドウサイズの大きさを判断できる。これが小さいときにはパラメータを標準の値のままとして、LFP 用の調整が働かないようにする。今後はこのパラメータの有効性の測定や最適化を行なう予定である。

#### 4. おわりに

本論文では、長距離超高速インターネットでは、これまでのインターネットと異なり、光速度の限界による伝送遅延が通信の速度に大きく影響を与えていることを示した。また、従来の TCP の送信戦略では長距離超高速の通信に対応できず、スループットが大幅に低下する LFP シンドロームが発生することを示し、その原因について考察した。さらに LFP シンドロームを抑制する方式を提案した。現在、実験を継続しており、長距離超高速インターネットに適したアルゴリズムとその効果を検証中である。

#### 謝辞

実験にご協力いただいた東京大学高木利久教授、萩原淳助手、京都大学金久實教授、秋山泰助教授、そして伝送システムについてご指導いただいた NTT 長距離事業本部永井光明担当課長、NTT インターナショナル(株)の片岡陽一係長の各氏に感謝いたします。

#### [文献]

- [1] C. G. Omidyar and A. Aldridge: Introduction to SDH/SONET, IEEE Communications, Vol.31, No.9, pp.30-33, 1993 Processing Overhead in TCP/IP, UCSD, 1993
- [2] J. Kay and J. Pasquale: The Important of Non-

#### Data Touching

- [3] L. Kleinrock: The Latency / Bandwidth tradeoff in Gigabit Networks, IEEE Commn. Magazine, Vol.30, No.4, pp.36-40, 1992
- [4] J. Postel: Transmission Control Protocol, RFC793, 1981
- [5] V. Jacobson: Congestion Avoidance and Control, SIGCOMM'88, pp.314-329, 1988
- [6] L. Zang, S. Shenker and D. D. Clark: Observations on the Dynamics of a Congestion Control Algorithm - The Effect of Two-Way Traffic -, SIGCOMM'91, pp.133-147,1991