社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2006－HPC－107 (30)
2006／8／1

再送制御を考慮した通信モデルの設計

野村哲弘† 石川 裕†

通信パターンによるバンド幅の変化及びパケット再送の遅延を考慮した新しい通信モデル PlogP-BR を提案する．PlogP モデルを拡張し，再送タイムアウトによる遅延とスイッチ内のボトルネックリンクによるバンド幅低下に対応させる．このことによりボトルネックがある通信路や再送制御プロトコルによる遅延が入る通信路でのアルゴリズム実行時間を予測できるようになる．複数のアルゴリズムについて PlogP-BR モデルによる実行時間予測と実際の実行時間を比較することによって，このモデルが従来のモデルよりもより正確に通信の振る舞いを模倣できることを示す．

Design of Communication Model Considering Retransmission Control

AKIHIRO NOMURA † and YUTAKA ISHIKAWA†

We propose the PlogP-BR communication model which considers the variation of bandwidth brought by communication pattern and the delay of packet retransmission. PlogP model is extended so that it can treat the delay brought by retransmission timeout and bandwidth decreasing caused by bottleneck link in the switch. Execution time is be predicted in the network with bottleneck link or in the network with delay by retransmission control protocol. We show the PlogP-BR model can simulate the behavior of communication more collect than the existing model, by comparing between the expected time based on the model and actual execution time for several algorithms in practice.

1. はじめに

近年のクラスタ・グリッド関連技術の進歩に伴い，並列計算環境が比較的身近な存在になってきた．それとともに，その計算性能・通信性能のバランスやトポロジに多様性が生まれてきている．このため，MPI などの集団通信の通信順序について，常に決まった最適解が存在するとは言えなくなってきている．そこで，並列通信路の性質を定式化し，その計算機で各通信順序を取ったときの実行時間を予測し，最速のアルゴリズムを求めようと試みられている．

従来からの試みはハードウェアのレベルで到達可能性を保証し，ボトルネックリンクも存在しない並列計算機専用の通信路を仮定している．このため，これらの通信路に対する仮定が成立しない Ethernet 上の TCP/IP などの通信路では正しく実行時間を予測できない．

本論文では，PlogP モデルを拡張した PlogP-BR モデルを提案する．本モデルでは，ボトルネックリンクによる通信速度の低下を表現するとともに，ソフトウェア層での再送制御が行われる通信路を表現する．また，PlogP-BR モデルが実際の TCP-IP 上のネットワークで実際の通信をより精度よく模倣していることを実験により示す．

† 東京大学
The University of Tokyo

2. 関連研究

2.1 既存の通信モデル

2.1.1 logP

logP[1] は，対称な通信路におけるパケットの送信処理時間を通信レイテンシ l，送受信処理時間 o，メッセージ間の最小間隔 (バンド幅)g で表し，これに参加ノード数 P を加えた 4 つのパラメータでノード間の通信の性質を表すモデルである (図 1)．図中の太線で示されている部分が各ノードが通信処理に CPU を割いている時間であり，それ以外の部分は計算等通信以外の処理に CPU を割くことができる時間である．

このモデルは安定した環境下でのサイズが小さいメッセージのやりとりをモデル化するのに向いている．しかし，このモデルではメッセージ長を考慮していないため，複数種類の長さのメッセージを扱う集団通信をモデル化するのには不十分である．

2.1.2 logGP

logGP[2] は，logP の上記の欠点をおぎなうべく考案された．LogGP では，メッセージ間の間隔がメッセージ長に依存しない定数部分とメッセージのサイズに比

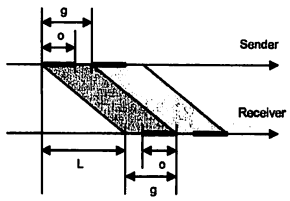

図 1 logP モデルにおけるパラメータの位置付け

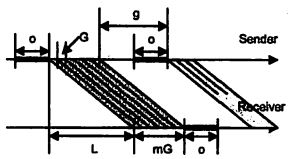

図 2 logGP モデルにおけるパラメータの位置付け

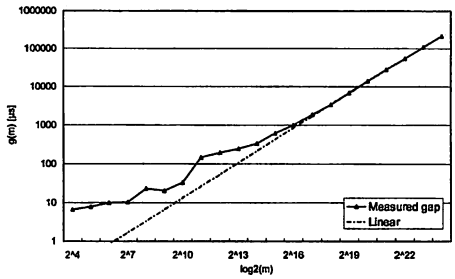

図 3 メッセージ長 m とパケット最小間隔 $g(m)$ との関係

例する部分からなるとモデル化している．パラメータ L の扱いが logP モデルと異なる等の細かい違いはあるが，意味的には logP における定数パラメータ g をメッセージ長 m の一次関数 $g+mG$ の形で置き換えたものといえる (図 2)．

しかしながら通信の下層に IP ネットワークを用いている場合等には，IP の MTU よりもメッセージが大きいかどうかで一般に送信時間のとる性質は変わる．表 1 に示すクラスタでは特にメッセージ長が短いところで図 3 に示すようにメッセージ長と送信遅延の間の線型性が崩れることが観察される．このような環境では LogGP モデルでは小さいメッセージの送信時間を正確に見積もることが出来なくなる．

2.1.3 PlogP

PlogP[3] は，logP モデルの拡張であり，logP における定数パラメータ o, g をメッセージ長 m の関数 $o_s(m), o_r(m), g(m)$ に置き換える．ここで，o_s および o_r はそれぞれ送信時，受信時の処理時間を示す．これらのパラメータを通信に利用される各メッセージ長 m について独立にその値を測定して用いる (図 4)．このことにより logP モデルでは表現できなかった大きいメッセージの送信によるオーバヘッドの増加と小さなメッセージにおけるほぼ定数となるオーバヘッドの両方の性質をひとつのモデルで表現できるようになる．

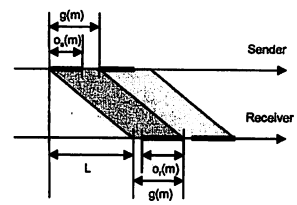

図 4 PlogP モデルにおけるパラメータの位置付け

しかしながら，通信の競合が発生したときの影響はこのモデルでは表現することができない．

2.1.4 loGPC

loGPC[4] は，logP と logGP の適用範囲の違いを考慮し，メッセージ長に応じて両者のモデルを使いわけをするモデルである．また，通信の競合が起こる際の通信遅延を M/G/1 待ち行列モデルによって定式化し，予測実行時間に加算する．ネットワークはメッシュに代表される k-ary n-cube 型の接続を想定しており，Ethernet などとは接続の形態が完全に異なる．

2.1.5 logPQ

logPQ[5] は，PlogP のモデルの中に明示的に送信・転送・受信のキューを置き，それらの間の転送速度とキュー長を明示的にパラメータに組み込んだものである．通信の競合が実際に何処で起こっていてどのように波及しているかを示すことが可能な強力なモデルである．しかし，その複雑性故に任意の通信パターンに対する所要時間の解析は自明ではない．そのため，個々の事例をモデルにあわせて注意深く検討しなければならず，常にこのモデルを使って解析することは容易とは言えない．また，中継地点のキューが満杯になった時に 1 つ前の送信者が直ちに送信を止めることを仮定しており，これによるパケットの喪失と上位層による再送は想定していない．

2.1.6 総　　論

上記の各モデルはいずれもハードウェアのレベルで到達保証を行い，かつボトルネックリンクが存在しない通信路を想定している．そのため，通信パターンの変化によるバンド幅の変化や，中継地点でのキュー溢れによる上位層による再送は想定されてない．

2.2 通信モデルの応用

文献 6) では，これらの通信モデルを元に集団通信関数のひとつである Reduce の複数の実装について Hockney[7], LogP/LogGP, PlogP の各モデル上で実行時間を予測し，最適アルゴリズムを選択することを試みている．実行環境のネットワークは Ethernet であり，本論文で論じている再送制御やスイッチのボトルネックの問題が発生する環境であると思われるが，その点については触れられていない．ノード数が多く，メッセージ長が短い場合についてはモデルの予測による最速アルゴリズムと実際の最速アルゴリズムの実行時間に不一致があるものの，それ以外の条件において

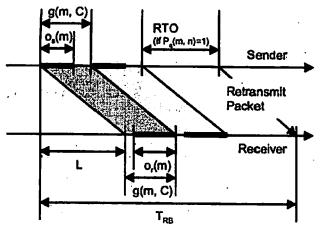

図 5 PlogP-BR モデルにおけるパラメータの位置付け

は比較的正しく最速に近いアルゴリズムを選択できている．メッセージ長が短い場合，どのモデルでも最速のアルゴリズムを正確に当てることはできず，別のアルゴリズムを最速と判断している．これは，通信モデルが想定している通信路と実際の通信路の振る舞いの違いによるものであると見ることができる．

2.3 パラメータの測定手法

PlogP モデルでは，メッセージ長 m に依存して変化する比較的多数のパラメータを短時間で計測する手法が文献 3) で確立されており，MPI 上でのパラメータ計測のツールも公開されている．この方法では，まず $g(0)$ をメッセージの連続送信時のパケット送出速度から求める．次に，メッセージ長 0 のメッセージのラウンドトリップ時間 (RTT) が $2(L+g(0))$ であることより L を求める．その後で，他のメッセージ長 m でのパラメータ $g(m)$ を，メッセージ長 m のパケットを送信してメッセージ長 0 のパケットを受信する時の RTT をもとに決定していく．その過程で $o_s(m), o_r(m)$ も同時に計測する．$g(m)$ を計測する際に通信路をパケットで満たす必要が無く，少ないパケットでパラメータを決定できる利点がある．

この方式において，測定に使用している通信はすべて単方向の通信である．双方向通信でも PlogP のパラメータは変化しないという仮定の下で，この方法で求めたパラメータを双方向通信のモデルにも適用している．しかし，一般に単方向通信時と双方向通信時のネットワークの性質は一致していない．

3. 提案手法

本論文では PlogP モデルを拡張して，以下の 2 つの問題にも対応できる通信モデル PlogP-BR(PlogP with Bottleneck links and Retransmission time-outs) を提案する．PlogP-BR における各パラメータの位置づけは**図 5** に示すとおりである．

3.1 RTO 待ちのペナルティ

TCP/IP[8] のようにソフトウェアレベルで信頼性を確保する通信プロトコルでは，データの送信後に相手から受信確認を受け取ることによって送信完了を確認する．TCP/IP の fast-retransmittion[9] のように，送信パケットの不達を受信確認のタイムアウトを待たずに検出する手法はあるものの，このような機構は後続の通信に依存している．後続の通信が無い状況では，送信後一定時間 (RTO 時間) 受信確認のパケットが届かないことによって初めて送信パケットが届いていないと分かる．MPI などの集団通信では，データの交換が複数段のフェーズに分かれているものが多い．このような通信では，あるフェーズで一斉に通信を行い，終了したら次のフェーズでは別の相手と一斉に通信を行うという通信パターンがとられる．スイッチのフロー制御が機能している場合は，このような場合でもキューにある程度余裕が出来ているのでパケットロスにつながらない．逆に，そのような機能がないスイッチでは出口キューからパケットが溢れて消失してしまう．このような場合，ひとつ前の通信の最終パケットが次の通信の最初のパケットに遮られて不達となると，遮られた通信が RTO 時間だけ停止する[10]．一般に RTO 待ちによる遅延は正常なパケットの送受信遅延に比べてきわめて大きいため，RTO のあるプロトコルではこの現象のモデル化が不可欠となる．

PlogP-BR では，通信路中のスイッチのキューが満杯の状態で局所的に多対 1 の通信が発生するときに，パケットロスが発生しうると考える．局所的に多対 1 の通信が発生する可能性のある状況とは，すなわち通信の相手を切り替える瞬間であり，隣接する 2 つ以上のフェーズの同一ノード宛の通信が同時に発生する状況といえる．初期状態では，フェーズの境界がほぼそろっているために，1 つ前のフェーズの最後のパケットが次のフェーズに割り込まれる可能性が極めて高く，結果 RTO 時間分通信が停止する．逆に一度 RTO のペナルティが発生した後はフェーズの境界が分散するので，不達になるパケットが一連の送信の最後のパケットになる可能性は極めて低くなる．なお，RTO のペナルティは当然複数の宛先ノードで発生しうるが，それらはフェーズのオーバーラップが起こる通信パターンでは一般に独立しており，干渉しない．よって，PlogP-BR ではこのような再送待ちが起こり得る場合にのみ通常の PlogP で算出される実行時間に RTO 時間 1 回分を上乗せする．一般にスイッチの出口キュー長を q とすると，メッセージ長 m のメッセージを n 回のオーバーラップしうるフェーズにわたって送信するとき，q バイト以上送信後にフェーズ境界が来る場合に RTO のペナルティ T_{RTO} が発生するといえる．従って，この場合の送信コスト T_{R} は次式で表される．

$$E_q(m,n) = \begin{cases} 0, & \text{if flow control available} \\ 1, & \text{if } q < (n-1)m \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

$$T_{\mathrm{R}} = ng(m) + L + E_q(m,n)T_{\mathrm{RTO}} \quad (2)$$

ここで，$E_q(m,n)$ は通信中に RTO によるペナルティが発生する回数の期待値を表している．スイッチのキュー長は，対象とする受信ポートが空いている状

態のときのスイッチによるパケットの遅延時間と，対象ポートにパケットが満たされている状態でのスイッチによるパケットの遅延時間の差にリンクのバンド幅をかけることによって推定できる．

3.2 ボトルネックリンクの発生

クラスタをつなぐネットワークには，バイセクションバンド幅が確保されていないネットワークが存在する．このようなネットワークでは，特定のノードの組み合わせで同時に送受信した際にボトルネックリンクになって相対的にバンド幅が低下する．この性質は2点間の転送遅延のみからパラメータを得る PlogP では表現できない．PlogP におけるパラメータ $g(m)$ は本質的にそのリンク間のバンド幅を示しており，このパラメータのみがこの現象下で変化するといえる．そこで，PlogP-BR ではパラメータ g をメッセージ長 m に加えて通信パターン C の関数 $g(m,C)$ とする．$C = s_1 \to d_1|...|s_k \to d_k|...$ と表したとき，C では各 k についてノード s_k からノード d_k への通信が同時に行われることを示す．以下，便宜的に $C_e = N_0 \to N_1$ とし，N_0, N_1 は PlogP パラメータの測定に用いたノードとする．$g(m,C)$ の値は，PlogP の $g(m)$ にあたる $g(m,C_e)$ から文献 3) にあるものと同様の方程式で求めることができる．C 及び C_e の通信パターンで長さ m のメッセージを同時に往復させたときの各ノードにおける遅延の最大値 T_C, T_{C_e} は $2(L+g)$ に等しいことを用いて，次式のように表せる．

$$g(m,C) = \frac{T_C}{T_{C_e}}(g(m,C_e) + L) - L \qquad (3)$$

一般にメッセージ長 m で通信パターン $C_1 \cdots C_n$ の n フェーズの通信を行うとき，T_{BR} は次式で表される．この式が PlogP-BR モデルでの一般的な通信の通信コストである．

$$T_{\mathrm{BR}} = \sum_{k=1}^{n}(g(m,C_k)) + L + E_q(m,n)T_{\mathrm{RTO}} \qquad (4)$$

C の中身は，送信ノードと受信ノードの組のリストであり，通信パターンの数はノード数について指数的に爆発するため，網羅することは現実的には不可能である．実際には利用されうる通信パターンについてだけ求めれば十分であり，次節の評価も利用される通信パターンについてだけデータを取っている．さらに，スイッチの構造とボトルネックリンクが明らかであれば，さらに通信パターン数を減らすことが論理的に可能である．

3.3 パラメータの測定方法

2.3 で述べた文献 3) の方法には双方向通信時の振る舞いを考慮していないという問題がある．Alltoall アルゴリズムを含め実際の通信の殆どは双方向通信であり，単方向通信のパラメータよりも双方向通信のパラメータを得たほうが良い．そこで，より実際の通信状況に近づけるため，測定を双方向で同時に複数回行い，その平均値をパラメータとして扱うよう測定方法を変更した．

ノード数を P, 自身のランクを r とする
- 自分以外のノードに対して MPI_Irecv() を発行
- $i \in 1...P$ について MPI_Send() をランク $r(+)i$ に対してこの順で発行
  但し (+) は排他的論理和を示す
- 全受信待ち通信に対して MPI_Waitall() を発行

図 6 Alltoall アルゴリズム A

ノード数を P, 自身のランクを r とする
$i \in 1...P$ について
- MPI_Irecv()
- MPI_Send()
- MPI_Wait()

をランク $r(+)i$ に対してこの順で発行する

図 7 Alltoall アルゴリズム B

表 1 評価環境

ノード数	64
CPU	AMD Opteron 2GHz Dual
メモリ	3.5GB
OS	Linux 2.6.11 SMP
ネットワークアダプタ	Broadcom Tigon 3 1Gbps
MPI 環境	YAMPII 1.0

4. 評　　価

図 6, **図 7** に示す Alltoall 関数のアルゴリズム A, B の実行時間と PlogP，PlogP-BR の実行時間予測の結果と比較した．なお，両アルゴリズムとも問題を単純化するため，ノード数は 2 の累乗個であることを仮定している．また，実際にはローカルのコピーも行っているが，通信遅延に隠蔽させて実行されるため，実行時間解析には影響しないので省略している．

4.1 評価環境

本評価に使用したクラスタは表 1 に示す通りである．MPI ライブラリには YAMPII[11] を用いた．YAMPII にはランデブー通信機能があり，受信系の命令が発行されていない送信を抑制することができる．アルゴリズム B ではこの機能を使ってパケットの集中を回避するため，本実験では当該機能を有効にしている．

4.2 通信モデルによる実行時間予測

両アルゴリズムともノード番号によらず同様の通信パターンを持っているので，任意の 1 ノードにおける通信パターンを解析することで実行時間予測関数をた

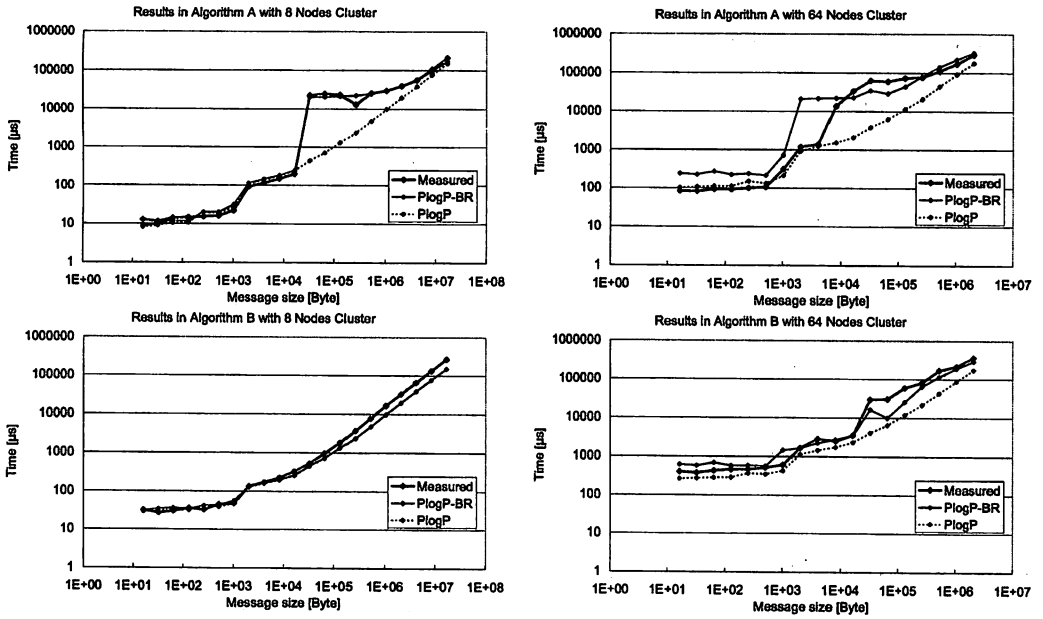

図 8 実行時間予測と実際の実行時間

てることが出来る．

アルゴリズム A では，送信処理はウェイトせずに連続して実行され，そのバックグラウンドで全ての受信処理が行われる．通信相手がフェーズごとに変わることを除けば通信パターンは長さ m のメッセージを $P-1$ 個連続で送信する処理と全く変わらない．よって，PlogP の予測関数 T^A_{PlogP}，PlogP-BR の予測関数 T^A_{BR} は次式の通りとなる．但し，C_k はアルゴリズム A の k 回目のフェーズの通信パターンとする．

$$T^A_{\mathrm{PlogP}} = (P-1)g(m) + L \tag{5}$$

$$T^A_{\mathrm{BR}} = \sum_{k=1}^{P-1}(g(m,C_k)) + L + E_q(m,P-1)T_{\mathrm{RTO}} \tag{6}$$

アルゴリズム B では，各フェーズごとに前フェイズの受信処理が終わるまで次フェイズの受信処理を始めない．ランデブー通信が有効になっている YAMPII では，送信先が同じフェイズに入るまで送信を待つため，同一ノードで複数フェーズのパケットを同時に受信することは無い．結果，原理的にスイッチのキュー溢れが原因となる RTO 待ちを必要とするパケットロスは制御用パケットが喪失する場合を除いて発生しない．その代わり，受信が終わってから次のフェーズに移るため，アルゴリズム A と違い毎回 PlogP における L の通信コストを被ることとなる．このアルゴリズムにおける PlogP の予測関数 T^B_{PlogP}，PlogP-BR の予測関数 T^B_{BR} は次式の通りとなる．

$$T^B_{\mathrm{PlogP}} = (P-1)(g(m)+L) \tag{7}$$

$$T^B_{\mathrm{BR}} = \sum_{k=1}^{P-1}((g(m,C_k)) + L + E_q(m,1)T_{\mathrm{RTO}}) \tag{8}$$

$$= \sum_{k=1}^{P-1}((g(m,C_k)) + L) \tag{9}$$

これらの予測関数を T_{RTO} が L に対して十分大きいことに注目して比較すると，以下のような不等式が得られる

$$T^A_{\mathrm{PlogP}} < T^B_{\mathrm{PlogP}} \tag{10}$$

$$T^A_{\mathrm{BR}} < T^B_{\mathrm{BR}}, \quad \text{if } E_q(m,P-1) = 0 \tag{11}$$

$$T^A_{\mathrm{BR}} > T^B_{\mathrm{BR}}, \quad \text{if } E_q(m,P-1) = 1 \tag{12}$$

この不等式は，PlogP モデル上ではアルゴリズム A はアルゴリズム B より常に短時間で終了することと，PlogP-BR モデル上ではアルゴリズム A に RTO による遅延が発生するとアルゴリズム間の優位性が逆転すると予測していることを示している．

4.3 結　　果

図 8 にノード数が 8 のときと 64 のときの両アルゴリズムの実行時間予測結果と実際の実行時間を示す．PlogP モデルでの予測結果に比べて PlogP-BR モデルでの予測結果のほうがより実際の実行時間に近い値を予測できていた．

実際の実行時間と PlogP-BR による予測結果には次のような不一致が見られる．

1つは，ノード数が多くなったときに，PlogP-BRにおいてRTOのペナルティの発生を予測するメッセージ長が実際よりも短く予測されている点である．パケットの再送が発生し始めるまでに経過するフェーズが多く，タイミングのずれの蓄積が大きくなると，結果的にRTOを待たずに再送できることが多くなる．このような状態になる場合をRTOペナルティの予測においてPlogP-BRでは考慮していないため，ペナルティが発生するメッセージ長の予測にずれが起こる．

もう1つの不一致は，メッセージ長が短いときに本モデルでの予測時間と実際の実行時間にアルゴリズムによらない定数倍のずれが発生する点である．これは実際のネットワークではPlogPのパラメータ測定方法やPlogP-BRの$g(m,C)$の測定方法が仮定しているネットワークの速度に対する安定性がないことが原因と考えられる．但し，このずれはアルゴリズムの実行時間を比較するという目的に使う分にはあまり問題にならない．

5. まとめと今後の課題

ハードウェアによる到達性保証機構があり，集団通信時にボトルネックとなるリンクがスイッチに無いネットワークを仮定しているPlogPモデルから，EthernetとTCP/IPの組み合わせのように再送制御をしており，スイッチにボトルネックリンクが発生しやすいネットワークでの使用に耐える通信モデルPlogP-BRを提案した．この通信モデル上で集団通信アルゴリズムの通信コストを予測することによって，PlogPモデル上で予測した場合よりも現実に近い予測結果を得ることが出来る．このことにより，再送制御をするネットワーク固有のパフォーマンス低下要因に強いアルゴリズムとそうでないアルゴリズムを判別することができる．

このモデルにとって未解決の問題として，次のような点があげられる．パラメータgを複雑化したために測定すべきgの値の数がノードの組み合わせの数だけあり，クラスタのノード数に対して指数爆発を起こしている．モデルの適用に使われるgの値のみを選択的に計測することにしても，依然測定すべきgの数は多く，計測は困難である．このため，トポロジの対称性を主とする規則性を元に出来るだけ少ない測定で必要なパラメータを確定する手法を確立する必要がある．さらに前節で述べたように，通信モデルやそのパラメータ測定方法にある仮定と実際の通信路の間に残る不一致が原因と考えられる不正確性を改善する必要がある．

謝辞 本研究の一部は，日本AMD株式会社のクラスタをお借りして行いました．

参考文献

1) Culler, D.E., Karp, R.M., Patterson, D.A., Sahay, A., Schauser, K. E., Santos, E., Subramonian, R. and von Eicken, T.: LogP: Towards a Realistic Model of Parallel Computation, *Principles Practice of Parallel Programming*, pp. 1–12 (1993).
2) Alexandrov, A., Ionescu, M. F., Schauser, K. E. and Scheiman, C.: LogGP: Incorporating Long Messages into the LogP Model for Parallel Computation, *Journal of Parallel and Distributed Computing*, Vol.44, No.1, pp.71–79 (1997).
3) Kielmann, T., Bal, H. E. and Verstoep, K.: Fast Measurement of LogP Parameters for Message Passing Platforms, *Lecture Notes in Computer Science*, Vol. 1800, pp. 1176–1178 (2000).
4) Moritz, C. A. and Frank, M. I.: LoGPC: Modeling Network Contention in Message-Passing Programs, *IEEE Transactions on Parallel and Distributed Systems*, Vol. 12, No. 4, pp. 404–415 (2001).
5) 當山孝義: 超並列・分散コンピュータネットワークにおける並列計算機モデルと設備配置に関する研究, 博士論文, 北陸先端科学技術大学院大学 (1998).
6) Pjesivac-Grbovic, J., Angskun, T., Bosilca, G., Fagg, G.E., Gabriel, E. and Dongarra, J.J.: Performance Analysis of MPI Collective Operations, *Proceedings of the 2005 IEEE International Conference on Cluster Computing (Cluster 2005)* (2005).
7) Hockney, R.W.: The communication challenge for MPP: Intel Paragon and Meiko CS-2, *Parallel Computing*, Vol. 20, No. 3, pp. 389–398 (1994).
8) Postel, J.: Transmission Control Protocol, RFC 793 (Standard) (1981).
9) Allman, M., Paxson, V. and Stevens, W.: TCP Congestion Control , RFC 2581 (Proposed Standard) (1999).
10) 松田元彦, 高野了成, 石川裕, 工藤知宏, 児玉祐悦, 岡崎史裕, 手塚宏史: MPIライブラリと協調するTCP通信の実現, 情報処理学会論文誌コンピューティングシステム Vol. 46 No. SIG 12(ACS 11), pp. 362–372 (2005).
11) 石川裕: YAMPII もう一つのMPI実装, 情報処理学会研究報告 2004-HPC-99(SWOPP04), pp. 115–120 (2004).