MACC97 資料
1997 年 12 月 18 日

市場モデルによる QoS 制御の実装時における問題点

京都大学大学院工学研究科情報工学専攻
八槇 博史
山内 裕
石田 亨

要旨

これまで経済学における一般均衡理論にもとづくアルゴリズムの性能が, 理論的あるいはシミュレーションを通じて議論されてきた．しかしその一方で，それらのアルゴリズムを分散環境に実装する際に問題となる計算や通信コストについては十分な解析が行なわれてこなかった.

本稿では, マルチメディア通信におけるアプリケーション QoS の制御に一般均衡理論に基づくアルゴリズムを適用し, 実装時に生じる 2 種のトレードオフについて論ずる: 空間的トレードオフは市場を構成するエージェントをネットワーク中に分散あるいは集中して配置した場合の, 通信時間と計算時間のトレードオフである．時間的トレードオフは，QoS に対するデマンドが動的に変化するときに, 各々の均衡解の正確さを重視し収束計算を続ける方法と, 均衡解の計算を途中で打ち切ってデマンドの変化に追随する方法との間の，資源の割当て効率に関するトレードオフである.

FreeWalk 環境 (マルチメディア会議ツール) における帯域割当と, RSVP 環境における帯域割り当てという 2 つの両極端な応用を想定し, 上記のトレードオフを分析した. その結果, (1) 空間的トレードオフにおいては, クライアント数が増大すると分散計算が有利になる, また, (2) 時間的トレードオフにおいてはデマンドの変化が激しいほど, 収束計算を打ち切りデマンドの変化へ追随するメリットが, 時間をかけて解の精度を高めるメリットを上まわることが分かった. これらを総合すると, (1) クライアント数の少ない FreeWalk には集中計算が, 多い RSVP には分散計算がよいこと, (2) 実行時に QoS デマンドが激しく変化する FreeWalk や, デマンドの変化は緩やかであるがクライアント数の多い RSVP 環境では, 共に収束計算を最後まで行なうことは実際的でなく, 適度に打ち切る必要があることが分かった.

1　はじめに

マルチメディア通信においては, より多くの帯域を割り当てるほど, 一般によりよい通信品質が得られる. しかし, インターネットに代表 されるような多くの利用者が資源を共有する環境では, それぞれの要求は衝突をおこすため, それらを調停する資源割当ての機構が必須である. 現状では, 資源確保のための機構についてはネットワーク層の規格として規定されつつあるものの, それらの機構を用いて各通信への割当てを行うポリシーについては十分議論されていない.

我々の研究の目的は, 利用者の選好をもとに資源割当てを行なうポリシーを与えることにあり, これまで市場における価格決定モデルである一般均衡理論をもとにした方式を提案してきた [11].　利用者の選好を消費者エージェントとし, アプリケーションの性質を生産者エージェントとして表現した市場モデルを構築した. このモデルでは各エージェントは市場価格に基づいて入札の内容を変更し, 最終的にパレート最適な資源の割当てに到達する. さらに, 動的に変化するデマンドに対応するため, 未来に渡っての取引きを行う機構を導入し、シミュレーション実験により良好な結果をえた。

このような市場均衡に基づくアプローチに関する研究は過去に多く行われ，いくつかのアルゴリズムについてその特徴や性能が理論的側面やシミュレーションを通じて議論されてきた [1, 3, 4, 5, 7, 9]．しかしその一方で，それらを分散環境に実装していく際に問題となる計算や通信コストについては十分な解析がなされてこなかった. そこで本稿では, 市場モデルを実際のマルチメディア通信に適用するための次のステップとして, 以下に挙げる 2 つの観点から, 実装時に生ずるトレードオフに関して論ずる.

- 空間的トレードオフ
  市場モデルにおける価格探索は分散計算が可能である. 各計算要素をネットワーク上にどの程度分

散させて配置するかによって並列計算による計算時間の短縮と通信時間の増大との間でトレードオフが生じる.

- 時間的トレードオフ
市場均衡にもとづく方式では一般に, 解に収束するまでエージェント間で情報交換を繰り返し行い, その後ではじめて実際の割当てが行われる. しかし, マルチメディア通信では, 計算途中でも市場のデマンドは変動するので, 解の正確にさを求めれば求めるほど, 得られた解はその時点のデマンドと合致しなくなる. このため, 正確さと反応性の間にトレードオフが生じる.

これらの解析のため, 本研究では FreeWalk 環境と RSVP 環境という, 性格の異なる典型的な 2 つ応用領域を考える。前者は Internet 上での 3 次元仮想空間を用いた会議システムであり, 比較的小人数によって利用される. QoS は FreeWalk という応用プログラムの中で管理され, QoS に対するデマンドは仮想空間での利用者の移動に従って激しく変化する. 後者は多数の利用者によって開かれたネットワーク環境で用いられる. デマンドの変化は緩やかである. 以上の 2 つのドメインについて空間的、時間的トレードオフに関する解析を行った. 両者は利用者数や QoS 割り当てに対する要求の変化の速さなどが異なり、その結果として上述のトレードオフについて対照的な結論を導くことを示す.

2 市場機構にもとづくアプリケーション QoS の制御

本研究では, 選好の異なる利用者が存在するネットワーク環境における資源の割当てには単一の評価規準に基づく最適割当ては存在せず, したがって各利用者が効用を最大化する中でのパレート最適なものをめざさざるをえないという観点から, 経済学における一般均衡理論の枠組を適用するというアプローチをとる.

我々はすでに, マルチメディアアプリケーションをターゲットとした市場モデルの構成, 計算方式およびシミュレーションによる性能評価について報告を行った [11]. 実装上の問題点を議論するに先立ち, 本章では以降の議論のために, この市場モデルの基本的な考え方と構成方式について簡単にまとめる.

2.1 基本的な考え方

市場モデルを構築するための二つの基本的な考え方を以下に述べる.

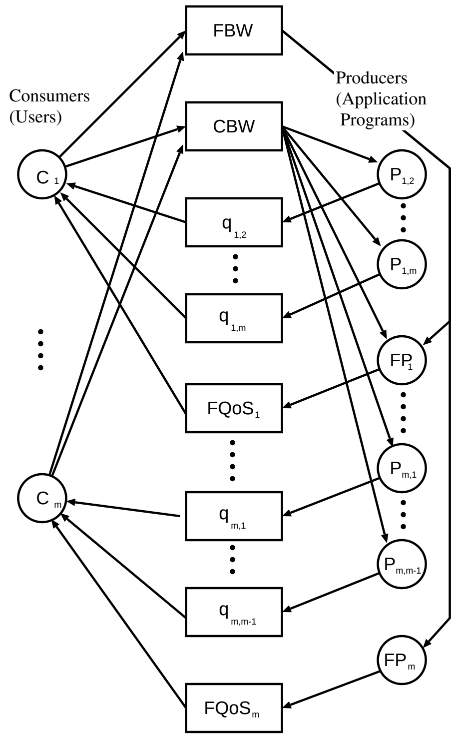


図 1: 市場モデル

1. 利用者はネットワーク資源の性能指標 (帯域幅など) そのものではなく, アプリケーション QoS の方を評価する. 帯域幅などはあくまでも生のネットワーク資源を測るためのものであり, 利用者にとって真に重要なのは, これを用いて実現されたサービスがどれだけ目的とするタスクにとって良いものであるかである.

2. ネットワークに「現在」と「未来」との区別を設けることで, 比較的活動的でない利用者が, ネットワーク資源を活動的な利用者群に対して譲渡する動機付けを行う. 同一の価格システムの中で現在と未来について別々に財を定義することによって, 利用者が効率的に時間方向での財のやりとりが行えるようにする.

単一セグメントのネットワーク上で, 各利用者のアプリケーションプログラムが互いにシングルキャストにより通信を行う場合を想定したときの, アプリケーション QoS 割当てのための市場モデルを図 1に示す.

中央の四角形は市場中でやりとりされる財 (good) を表す. 財には帯域幅 (Bandwidth) とサービス品質 (QoS) とがあり, これらの財はさらに時間軸方向で現在 (current) と未来 (future) の二つに分けられる. 財としての帯域幅は各利用者が保持している帯域の使用権

を代表し，BW と FBW はそれぞれ各現在および未来の帯域幅を表す．利用者 i から利用者 j へ向かう通信の現在のサービス品質は q_{ij} で表わされ，$FQoS_i$ は利用者 i が受ける未来のサービス品質である．未来のサービス品質についてはモデルの簡単化のため，通信毎には定義せずに各利用者 1 つずつの財としている．

図中の円はエージェントを表す．これには各アプリケーションにおける利用者の効用を代表する消費者エージェント（左列）と，使用するネットワーク資源からアプリケーション QoS へのハードウェアおよびソフトウェアによる変換操作を代表する生産者エージェント（右列）とがある．図中の矢印はモデル中での財の流れを表し，利用者が持つ帯域の使用権がアプリケーションによりサービス品質へと変換され，利用者がそれを享受することを表している．

2.2 消費者 - 生産者モデル

消費者エージェントはアプリケーションの利用者と一対一に対応する．利用者 i の帯域幅およびサービス品質に対する選好は効用関数として表現される．効用関数はいくつかの条件を満足する限りにおいて任意であるが，以下の議論では仮に，一般均衡のモデル構築の際によく用いられる [8], CES（constant elasticity of substitution）関数を採用する．利用者 j から利用者 i への通信のサービス品質を q_{ij}, 利用者 i にわりあてられる未来の帯域幅，未来のサービス品質，および現在の帯域幅をそれぞれ x_i^{fbw}, x_i^{fqos}，および x_i^{cbw} とすると，利用者 i の CES 効用関数は次式で与えられる．

$$\begin{aligned} & u_i(x_i^{fbw}, x_i^{fqos}, x_i^{cbw}, q_{i1}, \ldots, q_{im}) \\ &= \left(\alpha_i^{fbw} x_i^{fbw\,\frac{\sigma}{\sigma-1}} + \alpha_i^{fqos} x_i^{fqos\,\frac{\sigma}{\sigma-1}} \right. \\ & \qquad \left. + \alpha_i^{cbw} x_i^{cbw\,\frac{\sigma}{\sigma-1}} + \sum_{j=1}^{m} \alpha_{ij} q_{ij}^{\frac{\sigma}{\sigma-1}} \right)^{\frac{\sigma-1}{\sigma}} \end{aligned} \tag{1}$$

係数 α は，対応する財が消費者にとって他の財との比較においてどの程度の価値があるかを示す．また σ は，財の間の代替率を決定する定数である．

消費者は市場による資源割当て以前に各財について初期保有量（endowment）を所有する．一般的に，K 種の財に対して消費者の初期保有量がベクトル $\boldsymbol{e} = (e_1, \ldots, e_K)$, 各財の価格がベクトル $\boldsymbol{p} = (p_1, \ldots, p_K)$ でそれぞれ表わされる場合，消費者の選ぶ消費レベル $\boldsymbol{x} = (x_1, \ldots, x_K)$ は次の最適化問題の解として求めることができる．

$$\max_{\boldsymbol{x}} u(\boldsymbol{x}) \quad \text{subject to} \quad \boldsymbol{p} \cdot \boldsymbol{x} \leq \boldsymbol{p} \cdot \boldsymbol{e}.$$

他方の生産者エージェントは，割当てられたネットワーク資源から，利用者が受けるサービス品質への，アプリケーションソフトウェアによる変換を代表するエージェントであり，得られるサービス品質をネットワーク資源（ここでは帯域幅）の関数として表わされる QoS 生産関数を持つ．関数の形式は一定の条件を満たすかぎりは任意であるが，以下の議論では，現在のサービス品質 q_{ij} は次式に示す QoS 生産関数の出力として得られるものとする．

$$f_{ij}(x) = Q_i - \frac{Q_i}{1 + \gamma_{ij} x} \tag{2}$$

ここで，Q_i は利用者 i に対応するアプリケーションプログラムによって達成可能なサービス品質の最大値である．また，γ_{ij} は帯域の投入によってどの程度サービス品質が向上するかを決定するパラメータである．未来のサービス品質の生産関数 f は式 (2) と同じ形で，パラメータ γ を 1 に固定したものと仮定する．

生産者は次式で示されるように，利潤の最大化をめざして生産レベルを選択する．

$$\max \boldsymbol{p} \cdot \boldsymbol{y} \quad \text{subject to} \quad \boldsymbol{y} \in Y$$

Y は生産者の実行可能な生産計画の集合であり，ここで扱う市場モデルでは可能なサービス品質生産レベルの集合と一致する．

2.3 現在 - 未来モデル

現在と未来という二つの時間の間の関係を定義するためには，時間の経過とともに未来の資源をどのように現在の資源へと変換するかを指定する必要がある．この問題に対する本研究でのアプローチを図 2に示す．

このアプローチでは，時間を一定間隔に分割し，現時点からの 1 単位時間を現在（current），それ以降一定の時間（$T-1$ 単位時間とする）を未来（future）と定義する．ネットワーク全体での帯域幅を β とすると，現在の帯域幅の初期保有量の総計は β，未来の帯域幅の総和は $(T-1)\beta$ となる．市場価格決定を一度実行（図中 1）し，得られた現在の帯域およびサービス品質をネットワーク資源の割当て分として使用する（図中 2）．次に時刻を一単位時間進めて同じプロセスを繰返すが，この時に前回の未来の帯域および未来のサービス品質を次の割当ての際の初期割当てに反映させる（図中 3）．具体的には，まず消費者 i の未来の帯域幅の割当て x_i^{fbw} と，未来のサービス品質の割当て量をそれを生産するのにかかる帯域幅に変換したものとを合計し，r_i とする．割りあて r_i は次式で与えられる．

$$r_i = x_i^{fbw} + f^{-1}(x_i^{fqos})$$

$\sum_i r_i = (T-1)\beta$ なることに注意して，さらに m 個の消費者エージェント各々に対して均等に β/m の帯域

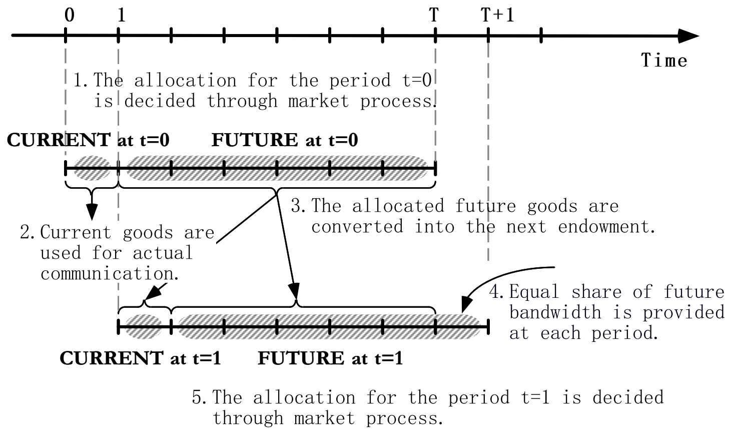


図 2: 現在 - 未来モデル

幅を与え（図中 4）, 全体として $T\beta$ の初期保有とする. これにより結局, 消費者 i に対する割当ては $r_i+\beta/m$ となり, その上で各消費者についてこれを $1:(T-1)$ に分け, それぞれ次回の現在および未来の帯域の初期保有量とする.

$$
\begin{aligned}
e_i^{cbw} &= \frac{1}{T}\left(r_i+\frac{\beta}{m}\right) \\
e_i^{fbw} &= \frac{T-1}{T}\left(r_i+\frac{\beta}{m}\right)
\end{aligned}
$$

以上の初期保有量決定の後, 各クライアントにおける効用関数および生産関数の諸係数を更新した上で, 再び一般均衡のモデルに従って割当てを実行する（図中 5）. これによって新しい期間内での帯域割当てが決定される.

3 応用分野の分析

上述の市場モデルを実際のネットワークに適用する場合、分散計算を行う規模や対応する applicaton の種類などの条件により、性能や機能への要求が異ってくることが考えられる。本稿では以下に示す 2 つの典型的な場合について議論する.

3.1 FreeWalk 環境

仮想 3 次元共有空間によるマルチメディア会合システム FreeWalk [6](図 3) では，利用者は各自のワークステーション上のクライアントプログラムを通じてサーバに接続することにより，仮想共有空間に入る． その中では，各参加者は 3 次元ポリゴンによって構成された四角錐で表現される．その 5 つの面のうちの四角形の面に，対応する利用者のワークステーションに付属するカ

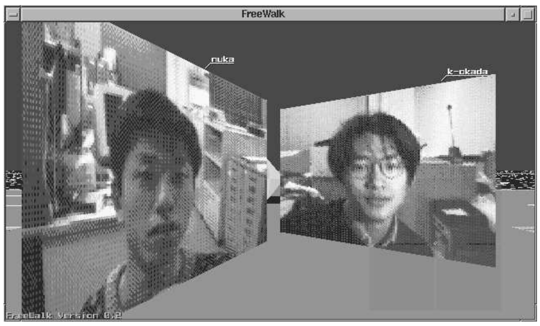


図 3: マルチメディア会合システム FreeWalk

メラにより入力された動画像がマップされる． それぞれの視点はこの面の中心に置かれ， そこから見た共有空間の様子が画面上に表示される． したがって， 遠くの参加者は小さく，近くの参加者は大きく表示され，また一定以上離れた場合や視野の範囲外に出たものについては表示は行われない． 同様に， 音声の大きさも送信側と受信側との距離に比例して変化する． 参加者は空間内の平面上を， マウスを操作して動きまわることができる．

このような環境では利用者の位置が刻々と変化し, それにともなって表示される他の利用者からの動画像の大きさも位置関係によって大きく変化する. したがって利用者の各動画像通信に対する選好も激しく変動するため, QoS 割当てをより高速に行う必要がある.

一方, 一つの応用システムの中での QoS 制御であるので, クライアント (QoS 消費者) 数が比較的少なく, また選好関数が均質で, 選好関数は設計時に決まっている. また, QoS 生産者の数も同様に少なく, 生産関数も既知である. 従って, 市場の構成は均質で扱いやすいと言え

表 1: 二つの典型的なネットワーク環境の比較

市場	FreeWalk 環境	RSVP 環境
消費者	• 効用関数は均質 • 選好はサービス中に変化する。 • 消費者数は小 (<10)	• 効用関数が非均質. • 選好はサービス中あまり変化しない. • 消費者数は大 (>10).
生産者	• 生産関数は均質. • 割当て機構決定時に確定. • 生産者数は小 (<100).	• 生産関数は非均質 • 実行時に新規に入ってくる. • 生産者数は大 (>100).
財	• 財の数は小 (<100).	• 財の数は大 (>100).

る.

3.2 RSVP 環境

現在のインターネットでは, 通信のための資源予約は通常行われず, 各ルータにおける best effort による通信が行われている. インターネットプロトコルの次期規格である IPv6 では, 資源予約に対して RSVP と呼ばれるルータ間でのシグナリングプロトコルとして提案されている. 冒頭でも述べたように, これはあくまでも資源予約要求のためのプロトコルであり, 実際に資源が各通信にどのようにわりあてられるかは規定されていない.

インターネット環境では, 利用者が様々な使い方をするためにアプリケーションの種類は実行時にしか決定せず, かつ利用者の種類も特定できない. したがって, そこでの QoS 割当て市場においては消費者の選好モデルは選好関数が非均質となり, 生産者 の生産関数も実行時まで定まらない. また, インターネット環境では一般に利用者数が大きくなるため, 消費者, 生産者および QoS 財の数もそれにしたがって大きくなる.

一方, 各利用者における選好の変化について考えると, RSVP を用いた資源確保はある通信の品質を一定期間継続して確保するために行われるということから, その品質に対する要求の変化は FreeWalk の場合に比べてはるかに遅い.

4 空間的トレードオフ

本研究で扱うような一般均衡に基づくアプローチでは, 2 章で述べた市場モデルにおける価格計算を, 計算要素である消費者, 生産者 をネットワーク中の複数のプロセスに分散して計算を行うことが一般に考えられている. この際, 各計算要素をどのプロセスの内部に実装し, 要素間での通信をどこまでネットワーク上での通信として実現するかによって, システムの効率に大きな差が生ずる.

本章では, 市場モデルを実際のネットワークシステムへの実装する際に問題となる各計算要素の配置について, 考えうる方式とそこでのトレードオフについて考察する.

4.1 トレードオフの分析

本研究で扱う市場モデルには利用者の選好を代表する消費者，アプリケーションによる帯域から QoS への変換過程を代表する生産者，およびそれらからの入札を受けて各財についてその価格を決定する 競売人 の 3 種類がある．消費者，生産者はそれぞれ利用者が使用するアプリケーションプロセス上に実装され，それぞれネットワーク上での通信により自らが興味を持つ財の 競売人へと入札を送信する．競売人 はそれぞれが担当する財について，需要と供給の均衡点となる価格 (clearing price) を求め，その財の価格を更新する．すべて財の価格はどのエージェントからも参照可能であり, これらの価格を参照して各生産者 / 消費者は入札の更新を行う．各市場において価格の変動が生じなくなった時点の価格が市場モデル全体の均衡価格であり，それにもとづく生産量 / 消費量はパレート最適となる．

ここで問題となるのは，競売人 による計算をどのように分散して行うかという点である．両極端には (a) 競売人 がそれぞれ独立のプロセス (以下では市場サーバと呼ぶ) として分散して実装され, エージェントが入札や価格取得をそれぞれの競売人との間で行う場合と，(b) 競売人 がすべて単独の市場サーバのプロセス上に実装され, 入札や価格取得はこのプロセスとの間でのみ行う

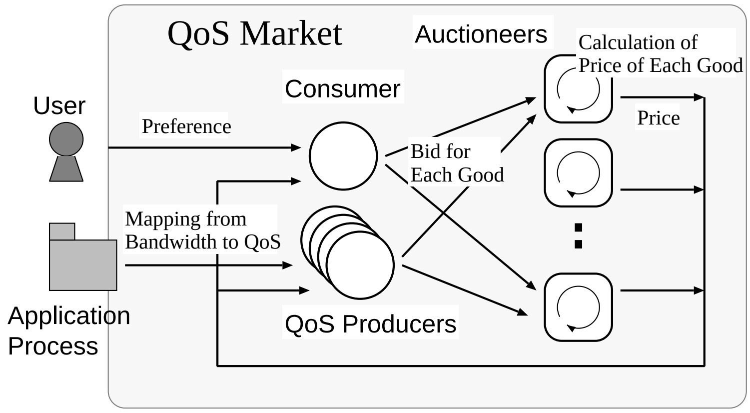


図 4: 市場による QoS 制御機構のシステム構成

表 2: 市場内の各構成要素の数

財の数	現在の帯域	1
	未来の帯域	1
	現在の QoS	nk
	未来の QoS	n
エージェントの数	消費者	n
	生産者	
	現在の QoS	nk
	未来の QoS	n
	競売人	$nk+n+2$

場合とが考えられる．以下ではこの 2 つをそれぞれ分散実装, 集中実装と呼び，それぞれの場合の得失を計算量およびメッセージ数の観点から論ずる．なお, 以下の計算は全て, 図 1に示した QoS 市場モデルを基に算出している.

まず市場モデル内の各要素の個数について検討する. 利用者数を n とする. 財の数は現在の QoS についてひとりあたり k 本 $(0 \le k \le n-1)$ の通信を行っていると仮定すれば表 2上側のように, 現在および未来の帯域については 1 つずつ, 現在の QoS については通信の本数に等しく nk 個, 未来の QoS については各利用者について 1 つずつ存在するため n 個となる. [1]

同表の下側は 消費者 / 生産者 / 競売人 の数を示す. 消費者 の数は利用者数に等しく n となる. 生産者 の数は QoS 財の数に等しく, 現在の QoS について nk, 未来の QoS について n である. さらに 競売人 の個数は財の総数に等しいので, $nk+n+2$ となる.

[1]全ての利用者間に接続があると仮定すると $k = n-1$ となるが, これは実際的ではない.

ここで, 集中 / 分散それぞれの実装方式における得失を議論するために, 計算の各段階に要する時間を次のように定義する. まず 集中実装において, 価格ベクトルを市場サーバからクライアントへ送信するのにかかる時間を T_{pr}, クライアントから市場サーバへ入札メッセージを送信するのにかかる時間を T_{pm} とする. さらに市場サーバ側で一つの財について 市場価格を求めるのに要する時間を T_c とすれば, 価格の更新一回に要する時間 $T_{centralized}$ は次のように求めることができる. まず, 価格メッセージの送信に要する時間は市場サーバが n 個のメッセージを送信する処理がボトルネックとなり nT_{pr}, 入札メッセージの送信に要する時間は市場サーバ側での受信がボトルネックとなり nT_{pm}, さらに価格の計算はすべての財について serial に行うため $(nk+n+2)T_c$ となる. 従って, 結局 $T_{centralized}$ は次式で表されることになる.

$$
\begin{aligned}
&T_{centralized} & (3)\\
&\quad = \quad nT_{pr} + nT_{pm} + (nk+n+2)T_c & (4)\\
&\quad = \quad (T_{pr} + T_{pm} + (k+1)T_c)n + 2T_c & (5)
\end{aligned}
$$

一方, 分散実装において同様に, 価格ベクトルを市場サーバからクライアントへ送信するのにかかる時間を T'_{pr}, クライアントから市場サーバへ 入札メッセージを送信するのにかかる時間を T'_{pm} とする. 各財の市場価格の計算にかかる時間は上記と同様に T_c と考えられる. 価格の送信, 入札メッセージの送信にともにクライアント側での送受信がボトルネックとなり, クライアント一つが関連する市場サーバの数は現在の QoS について $n-1$, 未来の QoS, 現在の帯域, 未来の帯域 が各 1 により, $n+2$ となる. 市場価格の計算は並列に行われるから, 分散実装における一回の価格更新時間 $T_{distributed}$ は,

$$
\begin{aligned}
&T_{distributed} & (6)\\
&\quad = \quad (n+2)T'_{pr} + (n+2)T'_{pm} + T_c & (7)\\
&\quad = \quad (T'_{pr} + T'_{pm})n + T_c + 2(T'_{pr} + T'_{pm}) & (8)
\end{aligned}
$$

と書くことができる.

(5)(8) の両式を比較すると, いずれも n に関して線形となっている. 注意する必要があるのは, 分散実装では各財に対する価格および入札が 1 メッセージに 1 つずつ乗っているの対し, 集中実装では複数の価格および入札が 1 メッセージ上に乗っているため, 同じ送信時間でもその内容が異なる点である. このため一般的に,

$$
T_{pr} + T_{pm} > T'_{pr} + T'_{pm} \qquad (9)
$$

が成り立つ. 両者の傾きを比較すると, 式 (9) により前者の傾きの方が大きいことがわかる. また, y 切片を比較したとき, T_c は計算時間なのに対して T'_{pr}, T'_{pm} は通信時間であり, 以下の実験でも示されるように一般に後者の方がはるかに大きくなるため, 両者を平面上にプロットした直線が交差し, 利用者数に応じたトレードオフが存在することがわかる.

4.2 議論

前節の議論を検証するため, 分散 / 集中処理の各方式で計算を行うシステムを実装し，評価実験を行った．計算アルゴリズムは, ワルラスの模索過程において, 効率化のため需要関数を入札するようにしたものである [10]. このシステム上で n の大きさを変えながら価格や入札の通信を行って, 前節の議論で用いた $T_{pr}, T_{pm}, T'_{pr}, T'_{pm}$ の値を $2.0ms$, T_c の値を $40\mu sec$, また $k = 5$ として計算すると, クライアントの数が 7 の辺りが, トレードオフの分岐点が生じた.

一般に, FreeWalk 環境のようにクライアント数がこの分岐点より小さいところでは集中方式が良く, RSVP 環境のようにクライアント数が大きい所では分散方式が良いと言える. 但しこの評価は, 広域のネットワークのように, 分散環境で通信が並列に実行できること, 通信の増加が他の通信速度に影響を与えないことを前提としている. LAN の上では, パケットはシリアルに転送され, 通信の増加が他の通信速度に影響を与えるため, 集中処理方式に有利に働くと考えられる.

5 時間的トレードオフ

一般均衡に基づくモデルでは, 価格の探索機構として各エージェントによる入札と, それに従った 競売人 による価格変更とを繰り返すことで, 競争均衡へ収束するといったアルゴリズムが用いられる. そこでは通常均衡解が求まるまでは財の交換は行われないという原則がある.

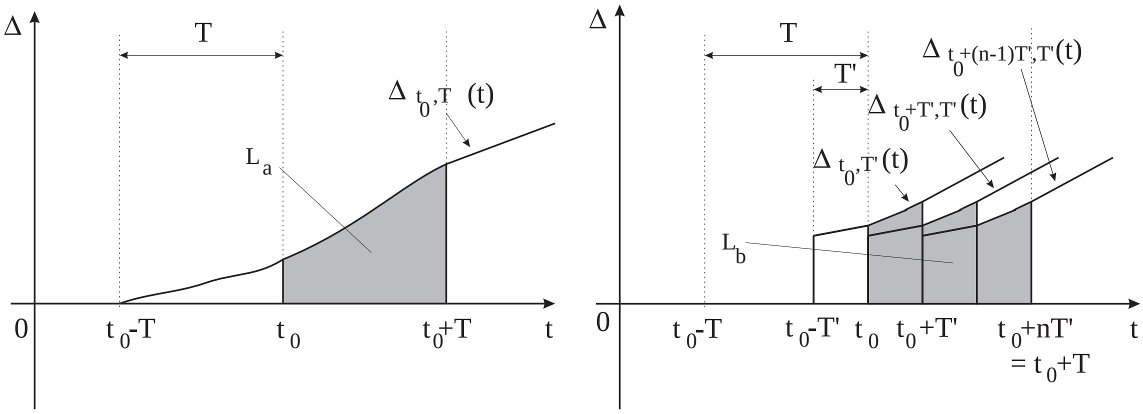
本研究で対象とするネットワーク環境では，状況変化に対処するためにこの価格探索を繰り返し行う必要がある．この変化がある程度大きくなると，毎回収束を長時間待って状況がまったく変ってから割当てを行うというのは非現実的である．むしろ, たとえ割当てに多少の無駄が生じても，ある一定以上の品質の割当てが得られればそこで計算をうちきって次の割当て計算にうつっていく方が良い場合もあるだろう．例えば FreeWalk 市場では, 利用者の動きに追随するために, ミリ秒単位の割り当てが望まれる. また RSVP 市場では, クライアント数が大きくなった場合に, 均衡解の計算を高速化する必要が生じる.

本章では，計算の精度向上にかける時間と, 市場におけるデマンドの変化への追従という時間的トレードオフについて議論する．

5.1 トレードオフの分析

ここでの関心事は，(a) 収束まで待って割当てを決定する場合と，(b) 早期に計算をうちきって均衡価格でない価格を採用する場合とのトレードオフである．以下，単純化のために均衡解への収束に要する時間を T 一定として議論を進めるものとする．

市場モデルにおいて，均衡価格でない価格を採用して割当てを採用した場合，各 market においては需給は均衡せず，したがって超過需要の量は非零となる．(a) の場合にも，そこで求まった価格はあくまでも計算開始



(a) 均衡解を収束するまで求めた場合　　(b) 計算を打ち切る場合

図 5: デマンドの変化による損失

時刻 $t_0 - T$ のものであり，収束時刻 t_0 における真の需要供給に対しては均衡価格ではない．不均衡状態の割当てをある期間続けて採用した場合のシステム全体での損失を超過需要量の大きさと時間との積として捉えるものとすれば[2]，時刻 t_0 において求まった割当てが採用される期間 $[t_0, t_0+T]$ での total loss L_a は，時間 T をかけて時刻 t_0 で求まった価格を時刻 t において用いることによる超過需要量 $\Delta_{t_0,T}(t)$ の当該期間内での積分として以下のように表すことができる．

$$L_a = \int_{t_0}^{t_0+T} |\Delta_{t_0,T}(t)| dt \tag{10}$$

この様子を図 5(a) に示す．図中，横軸は時刻，縦軸は超過需要量である．曲線は時刻 t_0 において求まる価格を採用したときのそれぞれの時刻 t における超過需要量を示す．このとき $L_{t_0,T}$ は曲線と横軸によって囲まれた陰の部分の面積に等しい．計算開始時刻 $t_0 - T$ においては $\Delta_{t_0,T}(t) = 0$ であり，したがってもし環境に全く変化がなければ $L_a = 0$ となることはこの図より明白である．

次に (b) の場合を考える．損失に関する考え方は上と同様であるが，計算を打ち切っているために，その価格に対してはそれぞれの計算開始時刻においても超過需要量は非零であることに注意する必要がある．時間 $T' = T/N$ (N は正整数) で計算をうち切るものとしたときの期間 $[t_0, t_0+T]$ における総損失は図 5(b) の陰の部分の面積に相当し，以下のように表すことができる．

$$L_b = \sum_{k=1}^{n} \int_{t_0+(k-1)T'}^{t_0+kT'} |\Delta_{t_0+(k-1)T',T'}(t)| dt \tag{11}$$

[2]帯域幅の市場で考えれば，これはその期間に送れるはずなのに送らなかった，あるいは送らなくてもよいのに送ってしまったデータの量と解釈できる．

環境の変化が大きいとき，図 5 の曲線の傾きは一般的に急になると考えられ，それに従って L_a も大きくなる．一方，(b) の場合にもその影響は避けられないが，再計算のたびに真の均衡解とのずれが補正されるために，L_b の増加は比較的小さくて済む．このことから，環境の変化が小さい場合には時間をかけて収束計算を行う方がよいが，変化が急激になるに従って L_a と L_b との大小が逆転し，計算を途中で打ち切って環境に追従していく方が有利となることがわかる．

もう一つの論点として，解の実行可能性に関する議論がある [7]．一般に均衡解を求める計算手続きには, 計算途中の解が実行可能なもの [3] とそうでないものがある. 本論文で用いているワルラスの模索過程は後者に属し, 計算を途中で打ち切ると実行可能な解は得られない. しかしながら, 正確な意味での実行可能な解が得られないことがそのまま, 計算途中の解が利用可能でないことを意味するわけではない. インターネットのようにそもそもベストエフォートで帯域割当が行なわれているようなネットワークでは, 理論的に実行可能な解でなくても, その解の示す指針にしたがって帯域を利用することが実際的である. また, RSVP による割当のような場合でも, 一般に RSVP サービス全体に確保された帯域の割り当て量にある程度の変動は許容されるものである.

5.2 議論

上述の解析結果が成立するためには, 計算の過程において超過需要量が単調に減少して, 図 5中の陰の部分が低くなる必要がある. この変化をシミュレーションにより測った.

図 6 はワルラスの調整過程を計算方法として用いた場合の, 現在の帯域幅に対する計算途中における超過需

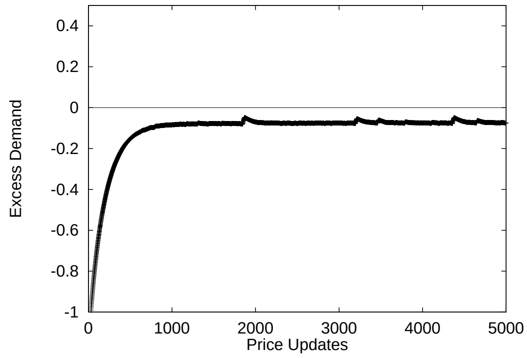


図 6: 価格計算の過程における超過需要量の変化

要変化を示したものである． クライアント数は 5，生産者や消費者のもつ各種パラメータは乱数により生成した．横軸は価格の更新回数，縦軸は超過需要量の全供給量に対する割合を示す．

この図に示されるように，超過需要量の絶対値は計算の進行とともに減少する． 誤差が全体の 10% 以下になるのに 1000 回の価格修正で済んでいるが，それ以降は需要量の収束は鈍化しているのが見てとれる．一般的にこのような収束型の計算では最後の少量の誤差の修正に大半の時間が割かれる傾向があり，このことからもある程度の精度が得られた上での計算打ち切りの有効性が支持される．

6　おわりに

本稿では、利用者の効用に基づくネットワーク資源の割当てという立場から得られたアプリケーション QoS 市場モデルについて、実際のネットワークへ適用していく場合の問題点について述べた．

一般均衡理論にもとづいた市場指向のアルゴリズムでは，消費者，生産者，競売人といった各計算要素を完全に分散して実装することが可能であるが，計算 / 通信コストや実装の複雑さの観点からはこれをそのまま採用することは得策ではない．また，これらにおいては均衡を求めるために，繰り返し計算を収束するまで行う必要があるが，本研究で扱うような状況が動的に変化して中で繰り返し割当て決定を行っていく場合には，完全な均衡を求めるのは常に有益とは限らない．

本研究では問題となっているアプリケーション QoS 市場とその適用対象であるネットワークアプリケーションについて，規模としては小さく要求の変化の激しい FreeWalk 環境と，逆に規模は大きいが要求変化の緩やかな RSVP 環境を想定し，その中での空間的，時間的トレードオフについてその分析および実験による確認を行った．その結果，以下の結論を得た．

1. 空間的トレードオフにおいて，クライアント数の小さい FreeWalk 環境では競売人を集中化するメリットが上まわり，クライアント数の大きい RSVP 環境では逆に分散化にメリットがある．

2. 時間的トレードオフにおいて，実行時に QoS デマンドが激しく変化する FreeWalk や，デマンドの変化は緩やかであるがクライアント数の多い RSVP 環境では，共に収束計算を最後まで行なうことは実際的でなく，適度に打ち切る必要がある．

上記のトレードオフの分析を通じ，市場モデルにもとづく割当て手法を実装するにおいては，その計算結果である割当て品質の分析だけではなく，適用対象による実装形態や計算打ち切りのタイミングの選択が重要となることを示すことができた．

本稿では上述の空間的 / 時間的トレードオフについてその存在と発生する要件とを，2 種の典型的な応用を想定して議論した．現在，本システムは実装中であり，実際のサービスを通じた問題点については別途報告する．

参考文献

[1] N. R. Bogan, *Economic Allocation of Computation Time with Computation Markets*, MIT Laboratory for Computer Science Technical Report 633, August 1994.

[2] S. H. Clearwater (ed.), *Market-Based Control: A Paradigm for Distributed Resource Allocation*, World Scientific, 1996.

[3] J.F. Kurose, R. Simha, "A Microeconomic Approach to Optimal Resource Allocation in Distributed Computer Systems," *IEEE Trans. on Computers*, Vol. 38, No. 5, pp. 705–717, 1989.

[4] K. Kuwabara, T. Ishida, Y. Nishibe, and T. Suda, "An equilibratory market-based approach for distributed resource allocation and its application to communication network control," in (Clearwater96).

[5] M. S. Miller, D. Krieger, N. Hardy, C. Hibbert, and E. D. Tribble, "An Automated Auction in ATM Network Bandwidth," in (Clearwater96).

[6] H. Nakanishi, C. Yoshida, T. Nishimura, and T. Ishida, "FreeWalk: Supporting Casual Meeting in a Network," *Proc. of CSCW'96*, pp. 308–314, 1996.

[7] T. Sandholm and F. Ygge, "On the Gains and Losses of Speculation in Equilibrium Markets," 15th International Joint Conference on Artificial Intelligence (IJCAI-97), pp. 632–638, 1997.

[8] J. B. Shoven and J. Whalley, *Applying General Equilibrium*, Cambridge University Press, 1992.

[9] C. A. Waldspurger, T. Hogg, B. A. Huberman, J. O. Kephart and W. S. Stornetta, "Spawn: A Distributed Computational Economy," *IEEE Transactions on Software Engineering*, Vol. 18, pp. 103–117, 1992.

[10] M. P. Wellman, "A Market-Oriented Programming Environment and Its Application to Distributed Multicommodity Flow Problems," *Journal of Artificial Intelligence Research*, Vol. 1, pp. 1–22, 1993.

[11] H. Yamaki, M. P. Wellman, and T. Ishida, "A Market-Based Approach for Allocating QoS to Multimedia Applications," *ICMAS-96*, pp. 385 – 392, 1996.

[12] L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala, "RSVP: A New Resource ReSerVation Protocol," *IEEE Network*, Vol. 7, No. 5, pp. 8 – 18, 1993.