

自己組織化メカニズムによる分散型 QoS ルーティング方式

中 沢 実[†] 服 部 進 実[†]

近年のインターネットは、動画・音声など、さまざまなメディアの普及により、複数の QoS 要求を満たす経路計算が必要となる。しかし、ネットワーク規模の拡大と要求者側アプリケーションから頻繁に発生する各種 QoS 要求により、送信ルータは経路計算による負荷集中を発生する可能性がある。そこで、本論文では受信ルータに QoS 制御情報を含むルーティングに関する LRU (Least-Recently-Used) 方式のキャッシュを持たせ、経路計算によって生成された QoS 要求を満たす経路情報を保存しておくことで、要求者からの QoS 要求に対しての経路計算量を削減するメカニズムを提案する。さらに、ネットワークシミュレーションを行い、その有効性を明らかにする。

A Distributed QoS Routing by the Self Organized Mechanism

MINORU NAKAZAWA[†] and SHIMMI HATTORI[†]

The route calculation satisfying multiple QoS requirements will be necessary in the next generation Internet due to the appearance of multimedia application such as video, voice, etc. However, the sending router can be congested by load concentration of many applications with various QoS requirements. In this paper, we propose a new QoS control mechanism in which the LRU system on the side of the receiving router has a cache of the QoS routing information. This mechanism can reduce the calculation time for the QoS requirements of the user. Performance evaluation via simulations has proven the effectiveness of the proposed QoS routing algorithm.

1. はじめに

近年、インターネットでは、動画・音声など、さまざまなメディアの普及にとともに、ネットワーク効率化のみならず、ネットワーク利用者から要求される QoS (Quality of Service) を考慮した経路制御が必要となってきている¹⁾。

End-to-End 間では、サービス提供をするプロバイダ (以後、Source 側と呼ぶ) と、サービスを受け QoS 要求をするアプリケーションを持つクライアント (以後、Destination 側と呼ぶ) が存在する。その End-to-End 間における、ネットワーク上の QoS 全体を考慮するような経路制御を行うためには、Source 側でネットワーク全体の経路を管理するような Source Routing 方式が有効である²⁾。

この方式によるヒューリスティック経路選択アルゴリズムとして、Fallback アルゴリズムがある。当該アルゴリズムは、End-to-End 間の経路で十分に予約可

能なネットワーク資源を検索できる。つまり、フローの複数の要求サービス品質 (帯域幅、遅延、誤り率) を保証できる経路を探索する⁶⁾。

しかし、当該アルゴリズムでは完全な QoS を保証する経路を探索することはできるが、ネットワーク資源の変動環境で許容しうる範囲での経路探索が行われていない。そのような経路探索ができるアルゴリズムとして QoSFinder⁷⁾がある。

QoSFinder は、利用者の要求品質を保証し、ネットワーク資源を有効利用するが、これらの制御はすべて Source 側で行う必要がある。また、サービスアプリケーションから頻繁に発生する QoS ルーティング要求や、要求者が求める連続メディアが Source 側に集中しやすいことから、Source 側の CPU に対しての負荷集中を起こす原因となる⁸⁾。

本論文では、QoS 要求を満たす経路計算を行う QoS-Finder によって生成された経路情報を Destination 側ルータへ LRU 方式のキャッシュを持たせ、この中に保存しておくことで、経路の再計算を削減するメカニズムを提案し評価する。

さらに経路情報をキャッシュすることにより、以下の効果が得られた。

[†] 金沢工業大学工学部情報工学科

Faculty of Engineering, Department of Information and Computer Engineering, Kanazawa Institute of Technology

- LRU(Least-Recently-Used)方式のキャッシュにより、使用頻度の高い QoS ルーティング情報ほど検索順位が高く、使用頻度が少ないものは順次消される。上記機能を用いて、QoSFinder によって生成された経路情報を破棄せずにキャッシュに保存することで、利用者の QoS 要求に対して経路の再計算を削減できること、また経路の使用頻度により経路荷重を自己組織的に生成することで高速かつ効率的な QoS ルーティング制御が可能となった。
- 通信網では、ユーザ個人の視点から考えるとアクセスするネットワーク空間(経路空間 × QoS 空間)は、ユーザ個人の職業、つきあいの範囲、あるいは趣味・嗜好によるアクセスなどに見られるように、ネットワーク全体の規模に比べれば限定された空間になる。したがって、ユーザ個人のアクセス空間の偏りが経路の使用頻度として履歴に残せれば効率的なアクセスが可能となる。言いかえると、特定のサーバなり Destination 側のアプリケーションにアクセスする頻度が高いほど、キャッシュの検索時間が速く、ヒット率が高くなる(ネットワークアクセス空間の削減)。

本論文では、2章で計算負荷を削減するための分散型 QoS Routing 方式について述べる。3章では、本方式のシミュレーションモデルについて述べる。4章では、その結果を評価しその有効性を示す。

2. 分散型 QoS ルーティング方式

本方式では、Source 側ルータに QoS 制御機構を、Destination 側ルータに分散型 QoS ルーティング機能を持たせている。QoS 経路制御のアルゴリズムとして、QoSFinder を使用した。QoSFinder は QoS を満たした経路を探索する。しかし、要求側アプリケーションから頻繁に発生する QoS 要求によって QoSFinder における計算負荷が増大する。本章では分散型 QoS ルーティング方式を提案するにあたり前提となる条件とその方式について述べる。

2.1 前提条件

サービス提供するサーバと QoS 要求するアプリケーションの End-to-End 間でネットワーク上の QoS 全体を考慮するような経路制御をするためには Source 側でネットワーク全体の経路を管理する必要がある。そのため、本方式では経路決定方式として Source Routing 方式を採用した。また、経路制御プロトコルとして、リンク状態型アルゴリズムを採用する。これは、各リンクの状態が帯域などのネットワーク資源が動的

に変化するため、各ルータは隣接ルータへリンクの情報を定期的に交換する必要があることによる。そして、個々のルータは送られてくるリンクの情報をもとにネットワークポロジを知り最短経路を求めるものである。各ルータのネットワーク QoS の状況把握を考慮すると、リンク状態やネットワーク資源とその変動を各ルータに反映するまでの時間を短くする必要がある。

提案方式は帯域保証型ネットワークや RSVP (Resource ReSerVation Protocol)⁹⁾などの帯域予約プロトコルを前提としている。RSVP では、各 RSVP クライアントから中継ルータに対し、予約すべき帯域を指定した Resv メッセージを定期的に送信する。中継ルータは、Resv メッセージを基に各クライアントへの経路に帯域幅を予約し、Resv メッセージを集約して上流の Resv 中継ルータに送信する。Source ルータに Resv メッセージが到達すると中継経路の帯域が予約される。予約すべき帯域を各ルータの上位 RSVP クライアントに通知すればよい。

2.2 提案方式

キャッシュシステムの構造

図 1 に本提案に基づく分散型 QoS ルーティング方式について、キャッシュシステムを中心とした全体の構成を示す。

Destination 側ルータには、利用者の QoS 要求を満たす経路を探索するキャッシュシステムがある。システムは、利用者の QoS 要求を満たす経路を探索する QoS Search Module、経路がネットワーク QoS を満たしているか判断する QoS Check Module から構成されている。

処理の流れ

計算負荷のかかる QoSFinder において、要求者の QoS ルーティングアクセス空間の特性を学習し、計算負荷を削減する分散型 QoS ルーティング方式の手続きを図 2 に示す。図では Source ルータが行う経路制御を Destination ルータが制御することで経路制御の分散化を行う。すなわち Destination ルータでは QoS 制御情報を含むルーティング情報を LRU 方式でキャッシュ化することで分散型 QoS ルーティングを実現している。

図 2 に処理の流れを示す。キャッシュへ経路情報が保存されていくプロセスは以下のとおりである。

- Destination 側のアプリケーションからの QoS 要求により、Destination ルータ内のキャッシュに保存されている経路情報を探索する。
- キャッシュの経路情報にヒットしない場合(図 3)

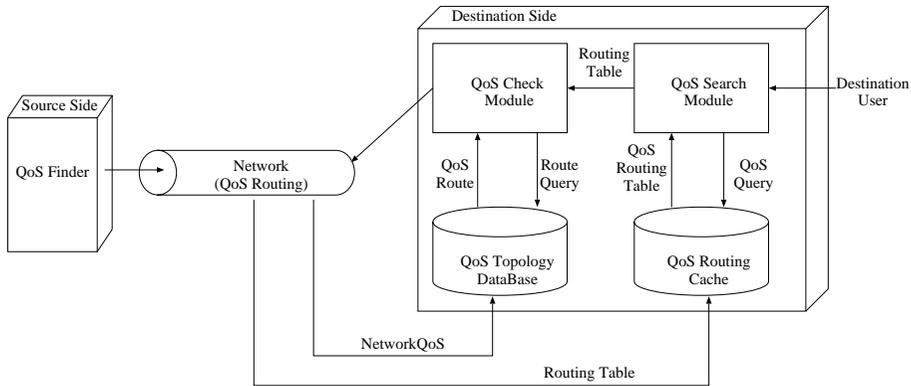


図 1 Destination 側のキャッシュシステムの構造
Fig. 1 Structure of cache system on destination.

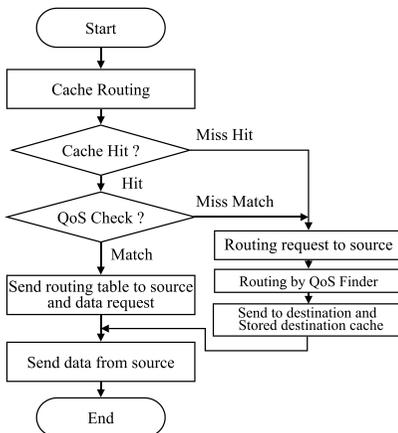


図 2 分散型 QoS ルーティングのフローチャート
Fig. 2 Flowchart of distributed QoS routing.

Source ルータへ経路計算の依頼をする。QoS-Finder で取得した経路情報を Destination ルータへ送信し、さらに Source ルータからデータ転送する。

- キャッシュにヒットした場合(図 4) Destination 内での探索によって得られた経路情報がネットワーク QoS に適合するか否か調べる。つまり、経路情報がネットワーク QoS を満たしているものであるか、QoS を含む Topology DataBase によって経路の要求値が利用者の要求を満たしているかを調べる。これはネットワーク QoS が変化し続けているために、キャッシュ内の経路情報が適合できなくなる可能性があるためである。
- ネットワーク QoS に適合しない場合はキャッシュにヒットしない場合の処理をする。つまり、Source ルータの QoSFinder へ経路計算を依頼する。

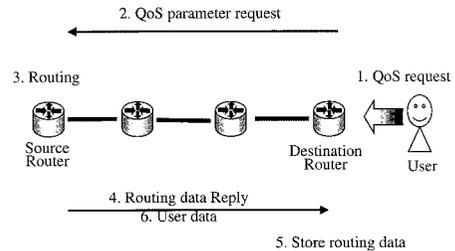


図 3 キャッシュヒットミスの場合のプロトコル
Fig. 3 Protocol for cache hit miss.

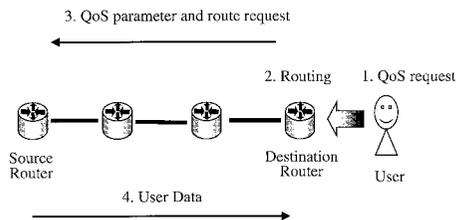


図 4 キャッシュヒットの場合のプロトコル
Fig. 4 Protocol for cache hit.

- ネットワーク QoS に適合した場合は、経路情報を Source ルータへ転送し、データ転送の依頼を行うとともに、QoS 要求を満たす該経路情報をキャッシュに格納する。

QoS Search Module

経路の Destination 側のルータのキャッシュ内に保存されている経路情報とその検索方法であるキャッシュ制御の構造を図 5 に示す。キャッシュ内の経路情報は、LRU 方式によって取り出されている。すなわち、その内部の検索は、最新の QoS によって計算された経路を対象に、ネットワーク QoS 条件を満たす経路として優先的に使用される。当該メカニズムにより、毎回 QoS ルーティングのための計算をすべて Source 側

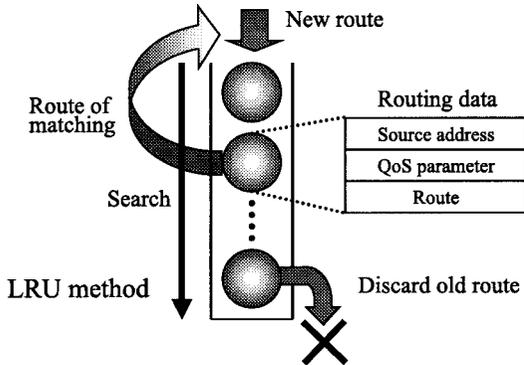


図5 キャッシュ制御の構造
Fig. 5 Structure of cache control.

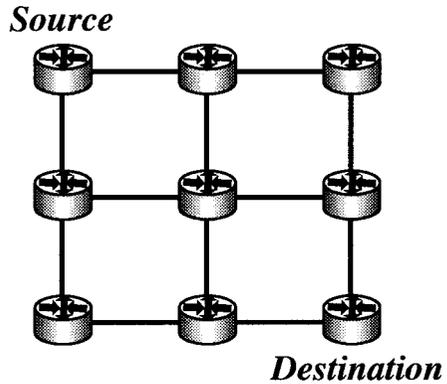


図6 ネットワークトポロジー
Fig. 6 Network topology.

で行う QoSFinder のみに依存する方式に比較し、経路選択の速度が向上する。

式 (1) によって、QoS 要求を満たす経路の検索を行い、ヒットした場合、新しいルートとしてキャッシュの先頭へ格納する。このことにより、使用頻度の高い QoS ルーティング情報ほど検索順位が高くなる。つまり、古くなって、使用されなくなった情報はキャッシュサイズを超えることにより削除されていく。

$$R_t \leq C_t \wedge R_d \geq C_d \wedge R_e \geq C_e \quad (1)$$

R: 利用者の QoS 要求

C: キャッシュに保存している QoS

(*t*: 帯域, *d*: 遅延, *e*: 損失率)

これによって、キャッシュ内部の情報は使用頻度による荷重を持たせた経路が自己組織的に設定できる。ヒットしない場合、QoSFinder により計算された新しい経路情報が格納される(付録参照)。

キャッシュの要素となる経路情報は、Source 側のアドレス、利用者の QoS 要求値、Source 側までの経路が含まれている。

QoS Check Module

各ルータは、ネットワーク QoS 値やトポロジーなどの変化するネットワーク全体の状態を定期的に把握している。キャッシュ内部でヒットした QoS 要求を満たす経路の候補は、ネットワークの変化に適應できていない可能性がある。そこで、ネットワーク QoS や、候補となる経路の QoS 値は、付録に示す式 (5) によって、算出され、式 (2) によって経路の候補がネットワーク QoS 値を満たしていれば、キャッシュヒットによる経路決定となる。

$$C_t \leq N_t \wedge C_d \geq N_d \wedge C_e \geq N_e \quad (2)$$

N: ネットワーク QoS

C: キャッシュでヒットした経路の QoS

実際の経路計算では、最適な経路を探索するために、

すべての起こりうる経路の QoS 値を算出しなければならないが、本方式では、候補となりうる QoS 経路の検索選択処理に置き換えることができる。

3. シミュレーションモデル

本方式の評価をネットワークシミュレータ OPNETversion5.1 を用いて行う。対象とするシミュレーションモデルとして、図 6 に示すような、格子状のネットワークでノード数 3×3 の場合を用いて Destination 側アプリケーションからの QoS 要求と、ネットワーク QoS 変化に対する適應性を評価した。比較的小規模のネットワーク構造によるシミュレーションを行う理由は、分散ルーティングの有効性を明確に評価することにある。すなわち、大規模ネットワークへの現実的な適用性については、領域ごとに適性サイズに分割されたネットワーク単位を想定し、前述のキャッシュメカニズムを多段形式で配置する QoS ルーティングキャッシュ方式が考えられる。本論文では、前述のネットワーク規模でシミュレーション評価を行うが、ノード数を拡大した場合の応答時間の増加特性やヒット率を把握する。ただし、多段方式のメカニズムや評価については、稿を改める。

具体的には、キャッシュ内の QoS ルーティング情報がほぼ定期的特性を示し、ヒット率の飽和傾向が観測される試行回数として、1000 回の Destination 側の要求回数(表 2)を選択しリアルタイムにネットワークの QoS 値が変化する条件で、シミュレーションを行った。

表 1 は、各リンクにおけるネットワーク QoS 値の設定値である。Throughput の確率分布は、指数分布とし最小値・最大値・平均値・分散値の項目を持つ。一方、Delay と Error Rate の確率分布は一様分布と

表 1 リンクあたりのネットワーク QoS 設定値
Table 1 Network QoS parameter per link.

	Throughput	Delay	ErrorRate
最小値	0 bps	1 ms	1×10^{-6}
最大値	476.1 kbps	10 ms	3×10^{-6}
平均値	160.2 kbps	-	-
分散値	3159 bps	-	-

表 2 利用者の QoS 要求範囲
Table 2 QoS request range of user.

Throughput	Delay	ErrorRate
1 ~ 100 kbps	50 ~ 55 ms	$9 \sim 11 \times 10^{-6}$

し、最小値・最大値の項目を持つ。

Throughput の確率分布を表 1 の値とした理由は、Throughput 自身がネットワークリンクの QoS の変化を表しており、コネクションリクエスト数に対し平均値と分散値を有する正規分布に従って、現在使用している帯域が変化するためである。

Delay と Error Rate の確率分布は、ネットワークリンクの固定値分を除く変動分については、ノードの QoS 値に従うものであり、ノードの QoS 値は一度決定されると、利用率などによる変化は比較的小さく、ネットワークリンクの Throughput のようにコネクションリクエストごとに変化することがないため、表 1 のような小さな範囲でランダムな値を与えた。

表 2 は、利用者のアプリケーションから要求する QoS 値の設定値であり、それぞれ表に示す範囲内で一樣分布をとっている。利用者の QoS 要求範囲を上記のように設定したのは、たとえば JPEG 圧縮形式を考慮した場合に、1 画像サイズはおおよそ 1 ~ 100 k 程度と考えられるからである。滑らかな画像を送るために、1 秒間に約 20 コマの画像を送ることを想定した場合に許される転送遅延は平均で 50 ms と算出される。損失率については、通信路の誤り率が一般に約 10^{-6} オーダーとすると、9 ノードのネットワークにおける最大損失率より許容範囲を定めた場合にこのような要求範囲として設定される。

4. シミュレーションによる評価

提案方式の有効性を示すため、下記の評価を行った。

- 本提案方式を導入した場合の応答時間（プロトコルメッセージ転送時間+経路探索時間）による性能評価
- ネットワーク利用者のアクセスする空間を削減したことによるキャッシュヒット率の向上度評価

処理時間の比較

本提案方式を導入した場合の計算時間とプロトコルメッセージ転送時間を合わせた 1 要求あたりの平均応答時間による性能評価を行った。

後述の評価データに示すように、各種 QoS パラメータに対するキャッシュヒット率 α は 0.8 を境に飽和傾向を示し定常値に達し始めている。そこで、キャッシュシステムが有効に働いているヒット率 α を 0.8 とした場合の応答時間に関して、本提案方式を採用した場合と QoSFinder のみによる場合と比較評価した。

図 3、図 4 のプロトコルにおけるメッセージ転送時間については、キャッシュシステムによる経路制御メッセージ転送時間 T_{cpk} と QoSFinder での経路制御メッセージ転送時間 T_{fpk} が全体のメッセージ転送時間である。

経路探索時間は、キャッシュ内の経路探索処理時間 T_c と QoSFinder での経路探索処理時間 T_f である。

キャッシュシステムを導入した場合の 1 要求あたりの平均応答時間 T_{cres} は、キャッシュ内の経路探索処理時間 T_c と、キャッシュシステムによる経路制御メッセージ転送時間 T_{cpk} が、キャッシュでの 1 応答時間となり、QoSFinder での経路探索処理時間 T_f と経路制御メッセージ転送時間 T_{fpk} が、QoSFinder での 1 応答時間で表すことができる。キャッシュシステムを導入した場合の 1 要求あたりの平均応答時間は、次式 (3) で表される。

$$T_{cres} = \alpha(T_c + T_{cpk}) + (1 - \alpha)(T_c + T_f + T_{fpk}) \quad (3)$$

キャッシュシステムを導入せず QoSFinder のみの場合の 1 要求あたりの平均応答時間 T_{fres} は、次式 (4) で表される。

$$T_{fres} = T_f + T_{fpk} \quad (4)$$

キャッシュシステムの方がヒット率 0.8 では、導入していないときに比べ平均応答時間が少ないことが図 7 から明確である。よって、本システム導入によりヒット率が高い値の場合は、処理時間が短縮され、ルータにかかる処理負荷が削減される。また、ネットワークのノード数に対して経路が増加するため、経路計算の処理時間は、指数関数的に増加している。

このため、大規模ネットワークへの適用を考えた場合、QoSFinder のみの QoS ルーティングでは、応答時間が大きくなり実用的でない。また、キャッシュシステムを用いた分散 QoS ルーティングにおいても、応答時間は改善されるが、そのままの適用では、大規模ネットワークでは限界を生じる。しかし、キャッシュシステムを用いた場合には、領域単位にネットワーク

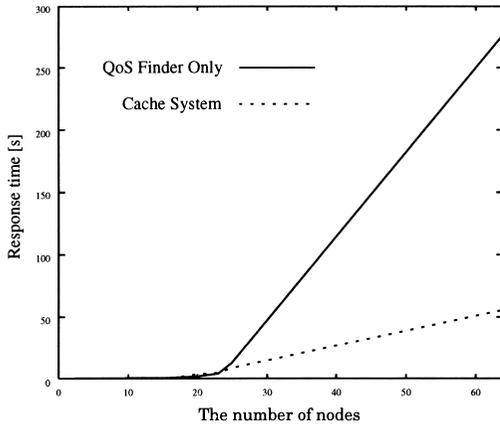


図 7 1 要求あたりの平均応答時間

Fig. 7 Average response time per request.

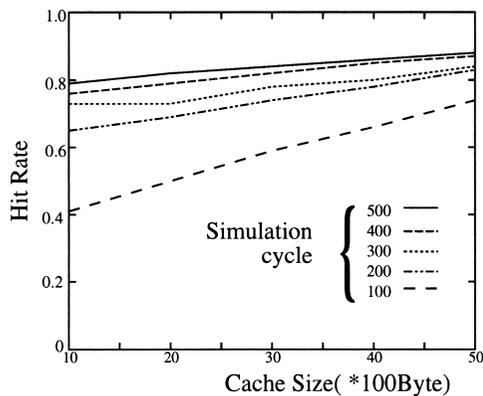


図 8 キャッシュサイズとヒット率

Fig. 8 Hit rate and cache size.

を階層化し分割した多段キャッシュシステムとすることにより応答時間や計算時間などの改善が可能である。
有効なキャッシュサイズ

次にキャッシュシステムの導入において、有効となるキャッシュサイズについて、ヒット率との関係を検証した。

図 8 は、ネットワークポロジとしてルータを 3×3 個とした場合に QoS 要求の条件値を前述と同じ条件で、試行回数 1000 回のうち、それぞれのキャッシュサイズにおけるヒット率が定常状態に到達する 500 回目まで 100 回ごとのヒット率を調べたものである。

図 8 より、試行回数が多くなるほどキャッシュヒット率は高くなるが、試行回数 400 回から 500 回においてヒット率の向上度は、キャッシュサイズを 10 個においては 2.1% であり、キャッシュサイズを 50 個に増加させても 0.4% であるため、ヒット率の向上傾向はほぼ飽和している。

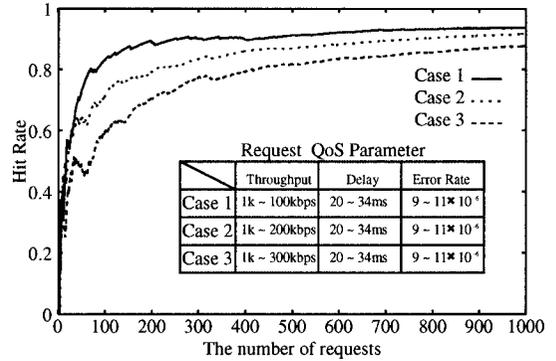


図 9 QoS 要求 (Throughput) のヒット率

Fig. 9 Hit rate for throughput parameter.

さらに、キャッシュサイズにおいても、キャッシュ数が 40 個から 50 個で安定している。このことから、本ネットワークポロジにおける有効キャッシュサイズを 50 個として、以後のシミュレーションを行い、さらにキャッシュサイズとネットワーク規模の関係についても検証した。

アクセス空間の削減

キャッシュに保存される情報の特性に着目して、ヒット率とアクセス空間との関係を検証するシミュレーションを行った。

利用者個人の QoS を含むアクセス空間は (要求 QoS 空間 \times 経路空間 \times リソース空間) の膨大なネットワーク空間となるが、前述のように個人の視点からは、利用者のアクセス空間は、利用者の属性 (職業、付き合いの範囲、興味のある対象など) に依存し、有限時間内では、特定の範囲であることが多い。たとえば、コンテンツに関する WWW アクセスを高速化するためのキャッシュサーバなどは、アクセス経路や要求品質を特定のものに限定することができる。

そこで、QoS 値パラメータの変化に対してヒット率を比較評価するため、利用者の QoS アクセス要求空間に制限を付加した場合と QoS アクセス空間を拡大した場合のヒット率の変化を定量的に把握する。

図 9 は、利用者の QoS アクセス要求空間である帯域において制限を付加した場合 (1 ~ 100 kbps) から、QoS アクセス空間を拡大した場合 (1 ~ 300 kbps) を、それぞれ、100 kbps, 200 kbps, 300 kbps の 3 つの帯域について、QoS 要求値に対するキャッシュのヒット率を測定した結果である。縦軸はヒット率、横軸は試行回数である。試行した QoS アクセス空間のすべてにおいて、試行回数を重ねるごとにヒット率の上昇は見られている。500 回目の試行での QoS 要求での帯域 100 kbps におけるヒット率は 91% に対し、帯域

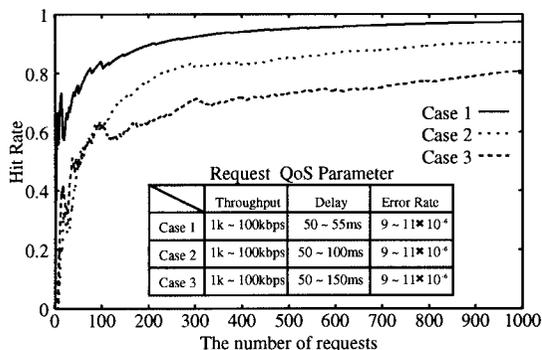


図 10 QoS 要求 (Delay) のヒット率
Fig. 10 Hit rate for delay parameter.

300 kbps におけるヒット率は 81% となる。言い換えると変動帯域幅を 1/3 に圧縮することにより、10% のヒット率の向上が図られる。

また、図 10 は、遅延における利用者の QoS アクセス要求空間において制約の付加に関する評価である。帯域 (図 9) と同じように、利用者の QoS アクセス要求空間である遅延において制限を付加した場合 (50 ~ 55 ms) から、QoS アクセス空間を拡大した場合 (50 ~ 150 ms) を、それぞれ、5 ms, 50 ms, 100 ms の 3 つの変動遅延幅について、QoS 要求値に対するキャッシュのヒット率を測定した結果である。図 9 と同様に、試行した QoS アクセス空間すべてにおいて、試行回数を重ねるごとにヒット率の上昇が見られている。500 回目の試行での QoS 要求での遅延 50 ~ 55 ms におけるヒット率は 96% であり、遅延 50 ~ 150 ms におけるヒット率は 72% となる。すなわち、変動遅延幅を 1/10 に圧縮することが可能であれば、ヒット率を 24% 向上できることを示している。

このように遅延についても、帯域同様に QoS アクセス空間の制限を付加した方がヒット率が良くなっている。

ただし、シミュレーション条件において、表 1 に示すリンクあたりのネットワーク QoS 設定値において、帯域については指数分布、遅延については一様分布をとっているため、変動遅延幅についてのヒット率にばらつきが生じている。

帯域、遅延にかかわらず、利用者の QoS アクセス要求空間を限定することにより、キャッシュ内部によく利用する経路情報ほど蓄積される学習効果が生じ、ヒット率は、この点でも向上傾向を示す。

次に特定のサーバにアクセスを限定した場合について、ヒット率を測定した結果を図 11 に示す。本シミュレーションにおいて利用されるサーバの数の状況

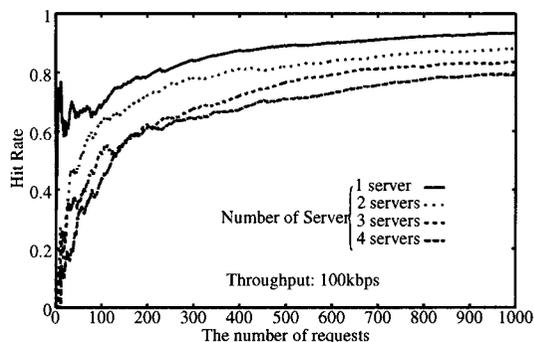


図 11 サーバへのアクセスのヒット率
Fig. 11 Hit rate for different number of servers.

は、インターネット放送や、インターネット会議などの、特に使用目的が映像メディアや音声メディアに関するものや、企業内ユーザなどのユーザがその目的に合わせて使用するサーバを特定するケースが想定される。たとえば、企業内におけるコンテンツを格納する WWW サーバは、そのアクセスを高速化するために、キャッシュサーバを階層化 (企業内の本社や支店などの複数拠点における各階層でのキャッシュサーバ) したり、分散化 (1 カ所にキャッシュサーバを複数台設置) したりする場合が考えられる⁵⁾。この場合、設置されるキャッシュサーバは、過度に分散させると処理能力が落ち、WWW サーバに直接アクセスする場合より遅くなることから通常数台のオーダーとなる。

そこでサーバの数を特定した場合 (1 server) と、分散化した場合 (4 servers) では、前者の場合が後者の場合よりヒット率が良いことから、WWW サーバに関する高速アクセスの観点からキャッシュサーバの分散数の制限要因となる可能性がある。

具体的には、試行回数 500 回目におけるサーバ数 1 のときのヒット率は 91% であり、サーバ数 4 のときのヒット率は 70% となり、Destination アクセス対象範囲が 1/4 に限定されるとヒット率は 21% 向上する。

以上述べたようにキャッシュヒット率は、ネットワーク QoS 値・利用者の QoS 要求範囲・利用者のアクセスするサーバ数・ネットワーク規模など多くの点に依存する。

ここで、ネットワーク QoS 設定値・利用者の QoS 要求範囲について、表 1、表 2 に設定し、ネットワークノード数を、 4×4 から 7×7 まで拡大した場合のヒット率の変化を示す。図 12 は、利用者がアクセスするサーバ数を特定した場合 (1 server) であり、図 13 は、分散化した場合でサーバの数を 4 つにした場合 (4 servers) のそれぞれのヒット率を示している。そ

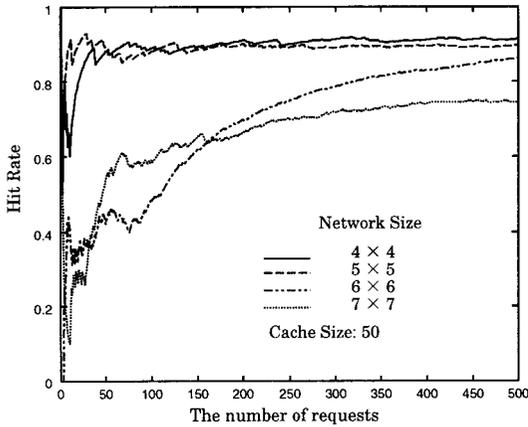


図 12 ネットワーク規模とヒット率 (1 server)
Fig. 12 Hit rate and network size (1 server).

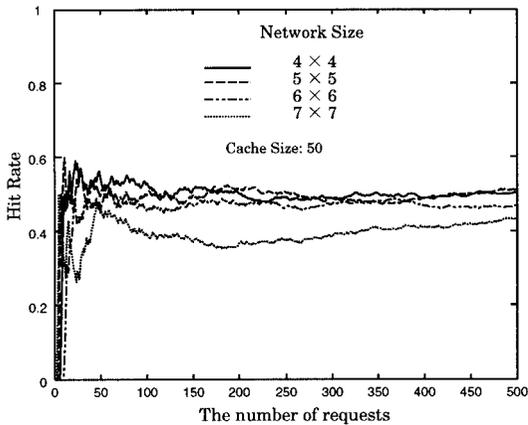


図 13 ネットワーク規模とヒット率 (4 servers)
Fig. 13 Hit rate and network size (4 servers).

それぞれネットワーク規模の変化に対して、サーバ数 1 の場合は、 7×7 のネットワーク規模ではヒット率は低下しているが、それより小さい規模のネットワークでは、ヒット率は 80% 以上となっている。一方、サーバ数 4 つではネットワーク規模が大きいほどヒット率は下がってはいるが、ネットワーク規模による差は図 12 より小さく、ネットワーク規模の変化に対しヒット率は 45% から 50% の間に分布している。

これらの結果より、キャッシュサイズ 50 個の場合、ネットワーク規模の拡大の影響よりも利用者のアクセス空間を限定できる場合の方が、より効果的な分散 QoS ルーティング制御が可能になるといえる。

以上のシミュレーション結果から、Destination 側でキャッシュを管理することで、利用者の特性に合わせた経路情報が保存される。頻りに利用する経路は保存され続け、利用されない経路は削除される。その結

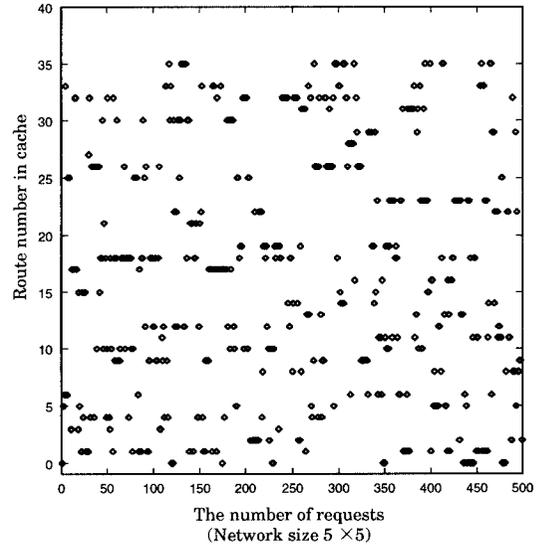


図 14 キャッシュされた経路の使用履歴 (QoS 優先)
Fig. 14 A snapshot of cached route (QoS prioritized).

果、Source 側の経路計算が削減され、負荷の分散化が可能となった。

ネットワーク規模に対する改善策

QoSFinder は付録に示したように、ネットワークの対するユーザ要求に応える最適品質を求めることを目的としていることから、ネットワークポロジや Source 側から Destination 側への経由ホップ数などよりも QoS 条件を第 1 条件として QoS 経路制御を行うアルゴリズムである。リアルタイムにネットワークのリソース空間や QoS 条件が変動する条件下ではつねに最適解を求めるアルゴリズムが必ずしも効率が良いとはいえない。

特にネットワーク規模が拡大させた場合には、経路空間 \times QoS 空間は $O(N^2)$ で増大する。そこで QoSFinder のアルゴリズムを経由ホップ数最小の経路を最優先に選択するアルゴリズムに変更することで、QoS を含めた準最適解を求めることの効果を検証する。

図 14 は、通常の QoSFinder を用いた場合の LRU キャッシュ内の Request 回数に対する QoS ルーティング情報の使用履歴をグラフ化したものであるが、図の Request 回数の範囲内では、LRU キャッシュ内で自己組織化マップの存在は確認できない。一方、図 15 の経由ホップ数を最優先にさせたアルゴリズムでは、使用頻度の高い QoS ルーティング情報が自己組織化されているのが確認できる。

これは通常の QoSFinder では利用者の QoS 要求に対し、その時点での最適な QoS 値を優先して経路を選択するアルゴリズムであることから、QoS 空間

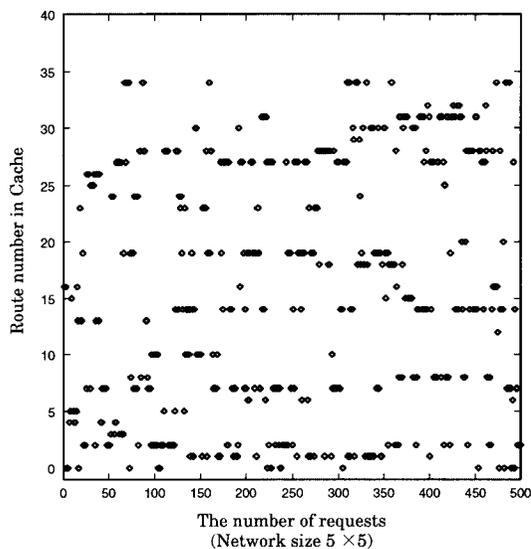


図 15 キャッシュされた経路の使用履歴 (ホップ数優先)

Fig. 15 A snapshot of cached route (hop count prioritized).

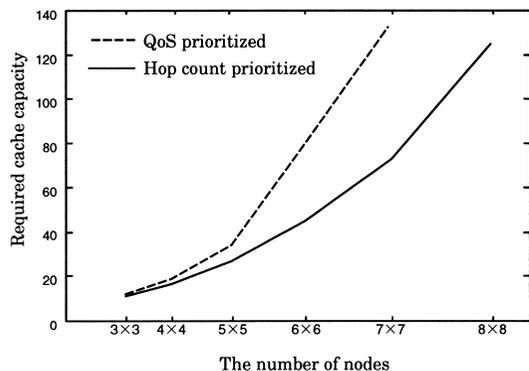


図 16 ネットワーク規模に対するキャッシュ容量

Fig. 16 Required cache capacity versus network size.

でのキャッシュ内自己組織化が優先される。したがって、図 14 の縦軸に対応する経路空間では自己組織化が Request 回数 500 回の範囲では形成されにくい。一方、経由ホップ数を最優先にした改良方式では、QoS 空間よりも経由ホップ数最小の経路空間が優先的に選択されるので、図 15 のように LRU キャッシュ内で使用頻度の高い経路がいくつか表れている。

また、ネットワーク規模の拡大に対し、LRU キャッシュサイズの増加状況を示したのが図 16 である。この図より明らかにホップ数を優先させる効率重視のアルゴリズムを採用することにより、LRU キャッシュサイズの増大を抑制できることが分かる。たとえば、ネットワークノード数 7×7 では、改良方式 (ホップ数優先) では、キャッシュ容量を約 50% に抑えること

ができる。

5. むすび

ユーザの要求品質を保証し、ネットワーク資源を有効利用を考慮する経路探索を行う QoSFinder は、Source 側で行う必要がある。さらに、QoS ルーティングは、特定のサーバへのアクセス経路空間、Destination ルータのアプリケーションからの QoS 空間、ネットワーク規模によるリソース空間のそれぞれを考慮した経路制御が必要となり、NP-Hard 問題となる。これにより、Source ルータは経路計算による負荷集中を発生する可能性がある。

本論文では、計算負荷のかかる QoSFinder において、要求者の QoS ルーティングアクセス空間の経路使用履歴を保存し、計算負荷を削減する分散型 QoS ルーティング方式を提案した。すなわち、Source ルータが行う経路制御を Destination ルータが制御することで経路制御の分散化を行い、Destination ルータでは QoS 制御情報を含むルーティング情報を LRU 方式によりキャッシュ化する。QoSFinder によって生成された経路情報を破棄せずキャッシュに保存することで、ネットワーク規模が 3×3 のとき、80% 以上のキャッシュヒット率が得られ、利用者の QoS 要求に対して経路の再計算を削減することが可能となった。

さらに、分散型 QoS ルーティング方式の有効性を示すために、Source ルータで QoSFinder による経路計算を行い、その経路情報を保存する LRU 方式によるキャッシュを Destination ルータへ持たせ、 3×3 から 7×7 のノード規模の格子型ネットワークでシミュレーションを行った。その結果、有限容量のキャッシュにより、古い情報または、あまり使われない情報は順次消されることから効率的な QoS ルーティングの実現が行えた。また、ネットワーク規模の拡大の影響よりも利用者のアクセス空間 (経路空間 \times QoS 空間 \times リソース空間) を限定できる場合の方が、より効果的な分散ルーティング制御が可能であることを確認した。

次にネットワーク規模に対するキャッシュ容量を抑制するための手段として、Source ルータでの経路計算方法であり最適解を求める QoSFinder を、経由ホップ数最小の経路を最優先に選択する方式に変更することで、準最適解を求めることの効果を検証した。その結果、経由ホップ数を最優先にさせたルーティング方式では、使用頻度の高い QoS ルーティング情報がキャッシュ内に自己組織化されていることを確認した。そこで、両方式 (QoSFinder, ホップ優先) のネットワーク規模に対する必要キャッシュサイズのシミュレーシ

ンを行ったところ、ホップ数を優先させる効率重視の方式が、ネットワークノード数 7×7 では、QoSFinder に比べて、キャッシュ容量を約 50% に抑えることができた。

本論文の結果として、従来の QoSFinder などの Source Routing 方式において、発生する NP-Hard 問題に対して、ルーティング情報の分散化を行うことで、経路計算に費やす時間を縮小する。結果としてルーティングに関する応答時間が向上するとともに、経路計算の集中化を防ぐことになる。さらに、ネットワーク上で QoS 経路獲得のためのプロトコルによるネットワーク負荷も削減される。

今後は、複数ユーザによるマルチキャスト型 QoS 経路制御に対するキャッシュの構築や、広域ネットワークにおける QoS 経路制御に対する多段のキャッシュ方式の検討が必要である。すなわち、1 キャッシュあたりのネットワーク規模の検討を行ったうえで、多段キャッシュシステムにおける制御メカニズム、応答時間やヒット率への影響を評価する必要がある。さらに、LRU 方式以外のキャッシュ方式との比較・評価を行う必要がある。

参 考 文 献

- 1) 佐藤 純, 橋本浩二, 柴田義孝: 連続メディア転送における動的レート制御について, マルチメディア通信と分散処理, pp.85-90 (Dec. 1997).
- 2) 後藤幸功, 太田昌孝, 荒木啓二郎: インターネット資源予約のための QoS 経路制御方式の提案, マルチメディア通信と分散処理ワークショップ, pp.65-69 (Dec. 1996).
- 3) 名部正彦, 馬場健一, 村田正幸, 宮原秀夫: インターネット・アクセスネットワーク設計のための WWW トラヒックの分析とモデル化, 電子情報通信学会論文誌 (B-I), Vol.J80-B-I, No.6, pp.428-437 (1997).
- 4) 名部正彦, 村田正幸, 宮原秀夫: キャッシングを考慮した WWW トラヒック特性の分析とモデル化, 電子情報通信学会論文誌 (B-I), Vol.J80-B-I, No.5, pp.325-334 (1998).
- 5) イントラネット渋滞の即効薬, 日経コミュニケーション, 1998.10.5, pp.70-82 (1998).
- 6) Pornavalai, C., Chakraborty, G. and Shiratori, N.: QoS Routing Algorithm for Multimedia Communication, Technical Report of IEICE., MVE97-11, pp.15-22, (Apr. 1997).
- 7) Vogel, R., Herrtwich, R.G., Kalfa, W., Wittig, H. and Wolf, L.C.: QoS-Based Routing of Multimedia Streams in Computer Networks, *IEEE J. Select. Areas Commun.*, Vol 14, No.7, pp.1235-1243 (1996).
- 8) 谷岡秀昭, 木下和彦, 滝根哲也, 村上考三: Fallback+: 複数の QoS 要求を満たす経路選択アルゴリズム, 信学技報交換システム研究会, pp.31-36 (May 1998).
- 9) Braden, R., Zhang, L., Berson, L., Herzog, S. and Jamin, S.: Resource ReSerVation Protocol (RSVP) - Version 1 Functional Specification (Sep. 1997).

付 録

QoSFinder

本方式では、Source 側ルータに QoS 経路制御を、Destination 側ルータに分散型 QoS ルーティング機能を持たせている。QoS 経路制御のアルゴリズムとして、QoSFinder を使用した。QoSFinder は、QoS を満たした経路を探索する。End-to-End 間のあるパスにおける QoS 値は次の式 (5) で表現できる。

$$\begin{aligned} T_t &= \min \{t_i | (1 \leq i \leq n)\} \\ T_d &= \sum_{i=1}^n d_i \\ T_e &= 1 - \prod_{i=1}^n (1 - e_i) \end{aligned} \quad (5)$$

T_t : パスの最小帯域
 T_d : パスの遅延
 T_e : パスの誤り率
 n : パスのノード数

各 (t_i, d_i, e_i) は、それぞれ、帯域、遅延、損失率のリンクあたりのネットワーク QoS 値であり、式 (5) により求めた QoS 値をルートに対する要求値 (D_t, D_d, D_e) と以下の条件のルートを探す。

$$D_t \leq T_t \wedge D_d \geq T_d \wedge D_e \geq T_e \quad (6)$$

この条件に適合するルートが見つからない場合、デフォルトルートを用いて Destination ルータへ、条件を満たすルートがないことを知らせる。

この条件に適合するパスが 2 つ以上の場合には要求に適したルートの選択が必要である。その選択として、各パスのすべての QoS 値を正規化し比較する。この際の正規化した値を Availability (利用率) という。

$$A_t = \frac{D_t}{T_t} \quad A_d = \frac{1/D_d}{1/T_d} \quad A_e = \frac{1/D_e}{1/T_e} \quad (7)$$

path1 の利用率を (A_{t1}, A_{d1}, A_{e1}) , path2 の利用率を (A_{t2}, A_{d2}, A_{e2}) とした場合、以下の比較によってルートが決定する。

$$A_{t1} > A_{t2} \wedge A_{d1} > A_{d2} \wedge A_{e1} > A_{e2} \Rightarrow \text{path 1}$$

$$A_{t1} < A_{t2} \wedge A_{d1} < A_{d2} \wedge A_{e1} < A_{e2} \Rightarrow \text{path 2}$$

利用率の数値が全条件において大きなパスがルートと

して選ばれる．この条件に適合するパスがさらに 2 つ以上の場合は，すべての利用率を比較し，最小の利用率を含む QoS 条件を見つけ出す．また，これまでの検索方法で複数のパスが存在する場合，パスの使用状況を比較する．

$$U_b = \frac{b_{use}}{b} \quad (8)$$

U_b : 帯域の利用率

b_{use} : 使用帯域

b : ルータの帯域

式 (8) において各パスごとに計算し，最小のパスをルートとする．これは帯域が混雑しているパスを回避することが目的である．この検索方法で複数のパスが存在する場合は，各パスで最小のホップ数を持つパスをルートとする．この方法でも困難な場合は，経路表から初めに検索したパスをルートとする．以上のアルゴリズムにより，QoS 要求をより満たすルートを検索することができる．

(平成 11 年 4 月 30 日受付)

(平成 11 年 12 月 2 日採録)



中沢 実 (正会員)

1991 年金沢工業大学工学部情報工学科卒業．1993 年同大学院工学研究科情報工学専攻修士課程修了．同年富士通研究所に入社．1996 年金沢工業大学助手．1999 年同講師．工学博士．自律分散システム，ネットワークアーキテクチャの実装方式に関する研究に従事．1993 年電気通信普及財団テレコムコロンブス賞受賞．1998 年電子情報通信学会北陸支部優秀学生賞受賞．電子情報通信学会会員．



服部 進実 (正会員)

1964 年東北大学工学部電子工学科卒業．同年より富士通にて情報通信システムの研究開発部門に所属．1989 年金沢工業大学情報工学科教授．1995 年同大学人間・情報・経営工学系長．電子情報通信学会通信グループ副委員長．本学会北陸支部長等を歴任．工学博士．情報通信システム，分散処理システムの研究に従事．電子情報通信学会，人工知能学会，IEEE 各会員．