分散システムにおけるアプリケーションQoS管理手法

加藤由花，箱崎勝也†

本稿では，分散システムにおけるアプリケーションQoS (Quality of Service) に着目し，ポリシーに従ったシステム運用を可能とする，アプリケーションQoS管理システムを提案する．本システムは，アプリケーションレベルの制御ルールと，ネットワークレベルの制御ルールの組み合わせによって，複数サービス間のQoSネゴシエーションを実現する．本稿ではこれを，シミュレーション実験によって確認する．

Application QoS Management for distributed Computing Systems

Yuka KATO and Katsuya HAKOZAKI†

This paper focuses on application QoS (Quality of Service) in distributed computing systems and proposes the Application QoS Management System. By using this system, we can manage computer systems according to the QoS management policies. In this system, QoS negotiation among many applications is done by combination of the control rule on an application level and that on a network level. We verify the negotiation by simulated experiments.

1. はじめに

CORBA (Common Object Request Broker Architecture)[1]等の分散オブジェクト技術の発達により，コンピュータネットワーク上には多くの分散アプリケーションが導入されるようになってきた．特にビジネス分野では，戦略的に情報システムが利用され，システムの一時的な障害が企業に大きな損失を与えるようになってきた．一方，社内LAN上には様々なサービスが共存しており，基幹業務サービスの性能が，他の優先度の低いサービスの影響で劣化してしまうケースも起きている．そのため，サービス毎に管理ポリシーを設定し，このポリシーを満足するように情報システムを運用する，という考え方が一般化しつつある．そこでは，各サービスが，利用者とシステム間であらかじめ合意されたサービス品質 (Quality of Service: QoS) を満足しているかが常に監視され，このサービス品質に基づくシステム管理が行われる．

ここで監視対象となるサービス品質は，各サービスにおいて利用者に対して提供されるサービス品質，つまりアプリケーションのQoSである (以降，アプリケーションQoSと記す)．このアプリケーションQoSは，利用者毎，サービス毎，さらに利用環境毎に異なる評価基準を持つ．従って，分散システムの効率的な運用には，サービスへの要求条件，利用環境，利用者の選好への要求を考慮した，情報システムアーキテクチャ，およびシステム運用手法が必要となる．

近年，このアプリケーションQoSに着目し，QoS制御を実現するためのシステムアーキテクチャ，制御手法等の研究が盛んに行われている．IETFではRSVP[2]やDiffServ[3]の標準化が行われているし，Active Network[4]に関する研究やポリシーサーバの製品化も行われている．これらの技術ではサービス毎にルールの記述が必要であり，管理ポリシーを事前に決定しておく必要がある．しかし，どのように管理ポリシーを決定したら良いか，また複数サービス間のQoSネゴシエーシ

† 電気通信大学大学院 情報システム学研究科
Graduate School of Information Systems,
University of Electro-Communications
yuka@hako.is.uec.ac.jp

ョンをどのように実現するか，という問題を解決する必要がある．

一方，エンドーエンドのアプリケーションQoSを制御対象とする研究としては，QoS ブローカー[5]や QoS を考慮したやわらかいマルチメディアシステム[6]などがある．これらの研究では，システムによるアプリケーション QoS のマッピング方法を研究対象としているため，複雑なモデル化が行われる．しかし，システム構成が複雑化し需要予測の困難な実システムでは，より単純な制御方式が望ましい．

これらの観点から我々は，分散システムの適応的運用に着目し，測定ベースの簡易な制御を実現する，アプリケーション QoS 管理システム (application QoS Management System: **QMS**) の研究を進めている[7]．**QMS** における QoS 制御は，アプリケーション QoS を基に決定されるアプリケーションレベルの制御ルールと，サービスのトラヒック特性を基に決定されるネットワークレベルの制御ルールの組み合わせによって実現される．その結果，複数サービス間の QoS ネゴシエーションが実現され，システム全体として管理ポリシーに従った運用が可能となる．

以下，2 章でアプリケーション QoS 管理システムについて，3 章で QoS 管理ポリシー設定方法について述べる．4 章では，本システムによって複数サービス間の QoS ネゴシエーションが実現され，システム全体で管理ポリシーに従ったシステム運用が可能になることを，シミュレーション実験によって確認する．5 章で本稿をまとめる．

2. アプリケーション QoS 管理システム

2.1 システム概要

QMS の設計コンセプトは以下の 3 点である．

1) アプリケーション QoS を制御対象とする．

2) 実システムへの適用を目指し，複雑なモデル化を伴わない簡易な制御を実現する．

3) サービス間の QoS ネゴシエーションを実現し，管理対象システム全体として，管理ポリシーに従った運用を可能にする．

QMS は 3 種類のモジュールから構成される．クライアントマシン上に実装され，サービス品質を測定，監視する *Notificator*，管理ノード上に実装され，管理ポリシーに従って制御方法を決定する *Manager*，制御実施マシン上に実装され，実際に制御を行う *Controller* の 3 種類である．

2.2 システムアーキテクチャ

QMS のシステムアーキテクチャを**図 1** に示す．

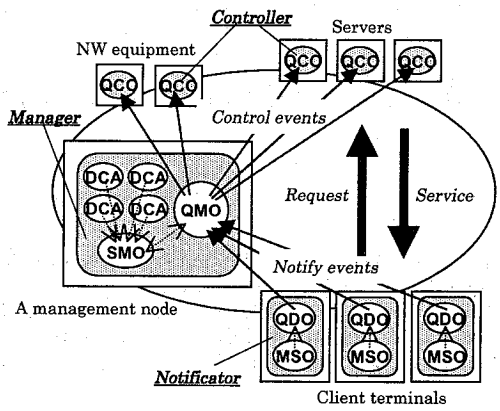

図 1 システムアーキテクチャ

各モジュールは以下のオブジェクトから構成され，ネットワーク上に分散配置されるこれらオブジェクト間の通信によって，QoS 制御を実現する．オブジェクト間の通信手順を**図 2** に示す．

***Manager* モジュール**：

QMO (QoS Management Object)：管理ポリシーを保持し，制御方法を決定する．

DCA (Data Collection Object)：制御方法を決定するため，各種性能データを収集する．

SMO (State Management Object)：収集された性能データを，システム状態として保持する．

***Notificator* モジュール**：

MSO (QoS MeaSurement Object)：アプリケーション QoS を測定する．

QDO (QoS Detection Object)：管理ポリシーに基づき，QoS 劣化を検知する．

***Controller* モジュール**：

QCO (QoS Control Object)：*Manager* モジュールで決定された処理を実施する．

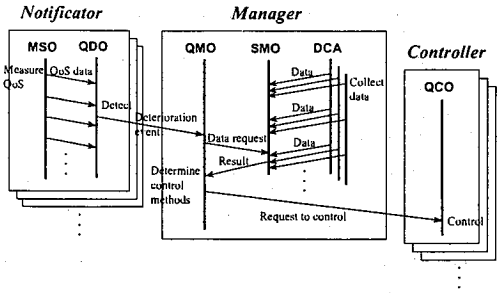

図 2 オブジェクト間の通信

3. QoS 管理ポリシーの設定

3.1 基本方針

QMS における QoS 管理ポリシーの設定は，以下の 2 段階で行われる．

1) ネットワークレベルの制御の優先順位を決定．
2) アプリケーションレベルの制御方法を決定．

複数サービス間の QoS ネゴシエーションを実現するため，ネットワークレベルの制御の優先順位を決定し，アプリケーションレベルの制御によってアプリケーション QoS の維持を目指す．ネットワークレベルの制御順位は，各サービスのトラヒック特性から，デフォルトの優先順位をシステムが自動的に決定する．

3.2 制御対象

QMS における QoS 制御は，測定ベースの適応的制御であるため，システム運用中に準リアルタイムに変更可能な要素が制御対象となる．サーバやクライアントマシンの CPU，メモリ，ディスクアクセス等の変更は考慮しない．

（1）ネットワークレベルの制御対象

インターネットにおけるエンド－エンドサービス品質として規定される，スループット，IP データグラムの遅延，遅延ゆらぎ，損失の 4 種類のデータ [8] を制御対象とする．各データに対する制御方法は，RSVP を利用した帯域予約，ルータにおけるキュー管理 (CBQ)，トラヒックシェーピング (トークンバケット方式)，DiffServ を利用したパケット廃棄優先度の設定とする．

（2）アプリケーションレベルの制御対象

アプリケーションレベルの制御対象は，アプリケーション QoS に従って決定される．QMS では，サービスレベルの変更 (映像送出レートの変更など)，ロードバランシング (接続先サーバの変更など)，エラー訂正 (冗長パケットによるエラー検出など)，提供サービスの変更 (映像を静止画に変更するなど) の 4 種類の制御を行う．

3.3 サービスの分類

ネットワークレベルの制御は，サービスのトラヒック特性に応じて，デフォルトの優先順位が決定される．ここで，マルチメディアサービスにおけるトラヒック特性は，トラヒックの種類と応答時間に対する要求によって分類される [8]．この分類を**表 1**，**表 2** に示す．

表 1 トラヒックの種類による分類

トラヒックの種類	特徴
ブロック型 (B)	大きなサイズの情報が低い頻度で発生 (静止画など)
ストリーム型 (S)	中サイズの情報が周期的に発生 (映像など)
トランザクション型 (T)	小サイズの情報が不規則に発生 (コマンドなど)

表 2 応答時間への要求による分類

応答時間への要求	サービス例
リアルタイム (RT)	数100ms以下が必要 (電話など)
準リアルタイム (QRT)	数秒以下が望ましい (WWW，情報検索など)
非リアルタイム (NRT)	100秒以上も許容 (電子メールなど)

3.4 管理ポリシーの設定方法

各トラヒックタイプに対するデフォルトの優先順位を**表 3**～**表 6** に示す．

表 3 帯域予約の優先順位

帯域保証順位	タイプ	帯域予約の有無
0	S-RT	○
1	B-RT	○
2	その他	×

表 4 キュー管理の優先順位

送出優先順位	タイプ
0	RT
1	QRT
2	NRT

表 5　シェーピングクラスの分類

クラス分け	タイプ	シェーピングの有無
0	S-RT	○
1	B-RT	○
2	その他	×

表 6　パケット廃棄の優先順位

廃棄優先順位	タイプ	サービスクラス
0	S-RT	なし
1	B-RT	なし
2	B-QRT	AF
3	B-NRT	AF
4	その他	EF

3.5 実装例

QoS 管理ポリシー設定機能を Java アプリケーションで実装した．実装例を**図 3**，**図 4** に示す．

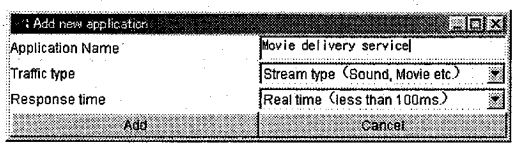

図 3　新規サービス追加画面

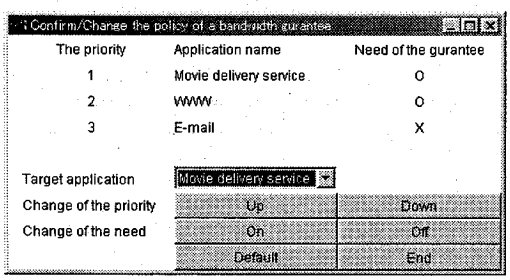

図 4　優先順位変更画面

新しいサービスを追加する場合，新規サービス追加画面において，サービス名，トラヒックの種類，要求される応答時間を設定する．各制御の優先順位は，4 種類の優先度設定画面において確認可能であり，必要に応じて順位の変更を行う．

3.6 管理ポリシーの設定方法

新規サービス追加画面においてサービスのトラヒック特性を設定すると，設定結果に基づいてネットワーク機器へ管理ポリシーが設定される．

帯域保証：QMO から要求帯域を取得し，クライアントプログラムに設定する．

キュー管理：QMO から TCP/UDP ポートを取得し，ポートとクラスの対応表，クラスと送出優先順位の対応表をルータに設定する．

シェーピング：QMO からバースト帯域を取得し，ポートと帯域の対応表を出口ノードに設定する．

パケット廃棄：QMO から TCP/UDP ポートを取得し，ポートとクラスの対応表，クラスと廃棄順位の対応表をサーバ側の入口ノードに設定する．

4. シミュレーション実験

4.1 実験条件

1 台のマシン上に擬似ネットワークを構築し，擬似トラヒックを発生させることによって，各サービスの動作を模擬した．ネットワーク構成を**図 5** に示す．

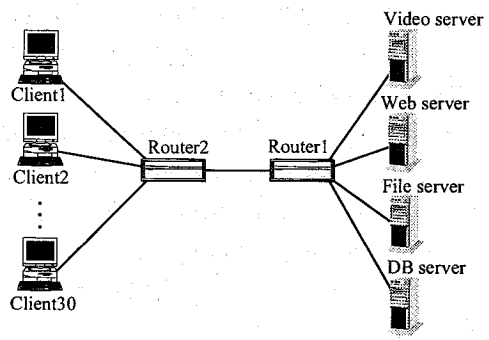

図 5　ネットワーク構成

実験の対象となるサービスとそのトラヒック形態，実装マシンを**表 7** に示す．各サービスの送出レートは一定値，映像配信のサービス時間は平均 600 秒のランダム値，各サービスの送信間隔は表に示す平均を持つランダム値とした．

表 7　サービスの種類

AP名	タイプ	レート (一定)	送信間隔 (平均)	実装マシン
映像配信	S-RT	10M	100s.	1～20
WWW	B-QRT	1M	100s.	21～30
FTP	B-QRT	30M	200s.	21～30
DBアクセス	T-QRT	10k	50s.	21～30

4.2 QoS 管理ポリシーの設定

QoS 管理ポリシー設定機能を利用し，4 種類のサービスの管理ポリシーを決定する．今回の実験では，3 種類の管理ポリシーを用意した．

ポリシー1：DBアクセスの可用性，応答時間を最優先とする．
ポリシー2：2台のFTPクライアントの可用性，応答時間を最優先とする．
ポリシー3：7台の映像クライアントの遅延防止，映像品質を最優先とする．

次に，各ポリシーにおけるネットワークレベルの制御の優先順位，およびアプリケーションレベルの制御方法を決定する．今回の実験では，帯域保証，およびシェーピングは行わなかった．ポリシー毎のキュー管理の優先順位，およびパケット廃棄の優先順位を**表 9**，**表 10** に示す．

表 9　キュー管理の優先順位

AP名	送出順位 (ポリシー1)	送信順位 (ポリシー2)	送出順位 (ポリシー3)
DBアクセス	0	1	1
FTP (非優先)	1	3	2
FTP (優先)	-	0	-
WWW	2	2	3
映像配信 (非優先)	3	4	4
映像配信 (優先)	-	-	0

表 10　パケット廃棄の優先順位

AP名	廃棄順位 (ポリシー1)	廃棄順位 (ポリシー2)	廃棄順位 (ポリシー3)
映像配信 (非優先)	0	0	0
映像配信 (優先)	-	-	4
WWW	1	2	1
FTP (非優先)	2	1	2
FTP (優先)	-	4	-
DBアクセス	3	3	3

アプリケーションレベルの制御方法は全ポリシー共通で，映像配信サービスにおいて，映像品質の劣化を検知した時に送出レートを1Mbps下げ，劣化が回復した時に次周期の送出レートを10Mbpsに達するまで1Mbpsずつ上げていく．実環境では，映像劣化はユーザ（またはクライアントマシン）によって検出されるが，今回の実験では，パケット損失の検出を映像劣化とみなし制御を行った．

4.3 実験結果

各ポリシーに対し3種類のシナリオを用意した．制御を全く行わないシナリオ1，ネットワークレベルの制御のみ行うシナリオ2，ネットワークレベルの制御とアプリケーションレベルの制御を組み合わせて行うシナリオ3の3種類である．シナリオ毎に，シミュレーション時間で1800秒の間，各サービスが要求した帯域，および実際にクライアントが受信した帯域を測定した．実験結果を**表 11** に示す．表中の値はデータ損失率(受信帯域/要求帯域)を，カッコ内の値は要求帯域（単位はMbps）を表す．

制御を行わない場合（シナリオ1），どのポリシーのどのサービスにおいても高い割合でデータ損失が発生している．この場合，ネットワーク機器ではサービスの種類を特定することができないので，ネットワークの品質劣化が始まると，どのサービスも同じ割合で劣化する．特に，データ損失によってサービス提供自体が不可能になるDBアクセスやFTPではアプリケーションQoSの劣化が顕著になる．つまり，制御を全く行わないと，多くの帯域を必要とするサービス（この場合，映像配信サービス）が他のサービスを圧迫し，システム全体としてアプリケーションQoSが劣化することがわかる．

ネットワークレベルの制御を行う場合（シナリオ2），映像配信以外のサービスではほとんどデータ損失は発生していない．しかし，ネットワークレベルの制御における優先度が低い映像配信サービスでは，多くのデータ損失が発生している．特に広帯域を必要とする映像配信サービスを優先する管理ポリシー3においては，非優先の映像配信サービスのQoS劣化が著しい．このように，ネットワークレベルの制御を行うと，優先度の高いサービスの品質はある程度維持できるが，広帯域を必要とするサービスを優先するなど，ふくそう状態が引き起こされるポリシーが設定された場合，システム全体でポリシーを満たした制御を行うことはできない．特に，優先度の低いサービスのQoSが著しく劣化する傾向にあることがわかる．

一方，ネットワークレベルの制御とアプリケーションレベルの制御を組み合わせて行う場合(シナリオ3)，アプリケーションレベルで映像送

表 11　シミュレーション実験において観測されたデータ損失率

AP名	ポリシー1			ポリシー2			ポリシー3		
	シナリオ1	シナリオ2	シナリオ3	シナリオ1	シナリオ2	シナリオ3	シナリオ1	シナリオ2	シナリオ3
映像配信 (非優先)	0.53	0.52	0.89	0.52	0.52	0.89	0.52	0.26	0.73
	(331390)	(328460)	(191050)	(335360)	(331310)	(192191)	(215410)	(210450)	(78542)
映像配信 (優先)	-	-	-	-	-	-	0.54	1.0	1.0
	-	-	-	-	-	-	(116090)	(115510)	114360
WWW	0.53	1.0	1.0	0.50	1.0	1.0	0.51	0.91	0.92
	(359)	(360)	(352)	(357)	(357)	(346)	(367)	(362)	(352)
FTP (非優先)	0.53	1.0	1.0	0.41	1.0	1.0	0.52	0.94	0.95
	(5340)	(4950)	(5370)	(4440)	(4920)	(4140)	(5100)	(5130)	(5070)
FTP (優先)	-	-	-	0.51	1.0	1.0	-	-	-
	-	-	-	(1200)	(1050)	(1140)	-	-	-
DBアクセス	0.53	1.0	1.0	0.50	1.0	1.0	0.46	1.0	1.0
	(7.18)	(7.16)	(6.97)	(6.84)	(7.36)	(7.28)	(6.90)	(7.07)	(7.31)

出帯域を減少させることによって，映像配信サービスのデータ損失が大きく減少している．データ損失による映像品質の劣化より，送出帯域の減少による劣化の方が少ないと考えられるため，この制御によって特に優先度の低いサービスの QoS 向上が期待できる．これは 3 種類のポリシー全てに対して言えることで，システム全体でポリシーに従った運用が実現することがわかった．

5. まとめ

本稿では，分散システムの適応的運用に着目し，測定ベースの簡易な制御を実現する，アプリケーション QoS 管理システム (**QMS**) の提案を行った．**QMS** における QoS 制御は，アプリケーション QoS を基に決定されるアプリケーションレベルの制御ルールと，サービスのトラヒック特性を基に決定されるネットワークレベルの制御ルールの組み合わせによって実現される．その結果，複数サービス間の QoS ネゴシエーションが実現され，システム全体として管理ポリシーに従ったシステム運用が可能となる．

本稿ではさらに，3 種類の QoS 管理ポリシーに対して，**QMS** によって管理ポリシーに従ったシステム運用が可能になることを，シミュレーション実験結果として示した．特にふくそう時や，相対的に優先度の低いサービスのアプリケーション QoS が向上することを確認した．

今後，より広域なシステム，一般的なマルチメディアサービスでの **QMS** 適用実験，および様々なシステム条件下でのシミュレーション実験なシステム条件下でのシミュレーション実験を行う予定である．

参考文献

1) Object Management Group, "The Common Object Request Broker: Architecture and Specification", OMG, 1998.
2) RFC2205, "Resource reSerVation Protocol (RSVP)", 1997.
3) RFC2475, "Architecture for Differentiated Services (DiffServ)", 1998.
4) J.M. Smith, et al. "Activating Networks: A Progress Report", IEEE Computer, Vol.32, No.4, pp.32-41, 1999.
5) K. Nahrstedt, J.M. Smith, "The QoS broker", IEEE MultiMedia, Vol.2, No.1, pp.53-67, 1995.
6) 橋本, 野村, 柴田, 白鳥, "QoS 保証を考慮したやわらかいマルチメディアシステム", 情処学会論文誌, Vol.40, No.1, pp.113-123, 1999.
7) 加藤, 金井, 中西, 箱崎, "分散システムにおけるアプリケーション管理フレームワークの提案", 情報処理学会分散システム／インターネット運用技術シンポジウム, Vol.99, No.3, pp.81-86, 1999.
8) 間瀬 編著, "マルチメディアネットワークとコミュニケーション品質", (社)電子情報通信学会編, コロナ社, 1998.