

# リッチメディアコミュニケーションにおけるリソースの傾斜配分による品質制御方式の提案

金友 大 中島 一彰 大芝 崇 田淵 仁浩

NEC サービスプラットフォーム研究所

本稿では、複数のアプリケーションを同時に利用するリッチメディアコミュニケーションにおいて、ユーザの要求に基づいて限られたリソースを傾斜配分することでアプリケーションの品質を動的に制御し、ユーザの感じる品質を高めることができる品質制御方式を提案する。企業内におけるイントラネットと汎用 PC を利用してのリッチメディアコミュニケーションでは、性能のばらつきや同時に利用されるアプリケーションの負荷により、コミュニケーションのためのアプリケーションが端末やネットワークのリソースを十分に利用することは困難である。提案方式では、ユーザの品質向上要求に対し、ユーザにとって重要性の低いアプリケーションの品質を抑制することで、品質向上にその時点での余剰リソース量より多くのリソースを利用可能とする。また実際の品質向上時には、ユーザがその時点で主に利用するアプリケーションを対象に、必要とするリソース量に対し効率よく品質が向上できるアプリケーションから順にリソースを割当てる。これによりリソースの有効活用とユーザ要望を満足させることによる体感品質の向上を実現する。

## QoE Control of Rich Media Communication by Intensive Resource Allocation

Dai Kanetomo, Kazuaki Nakajima, Takashi Oshiba and Masahiro Tabuchi

We propose a QoE (Quality of Experience) control system for allocating limited resources and changing QoS of rich media communication application dynamically based on user's request. It is hard to allocate enough resources to rich media communication applications when we communicate by multipurpose office PCs on intranet. When it is requested to improve QoS of rich media communication applications by user, our proposed system acquires more resources than current resource limitation for improving QoS by degrading quality of unimportant applications from user's viewpoint. Moreover, it allocates resources intensively to the primary applications based on resource/quality improving ratio, and then it can improve QoE of communication.

### 1. はじめに

近年、企業における意思決定の迅速化への期待から、音声通信・映像通信・資料共有など、リッチメディアによるコミュニケーション手段を複数組み合わせる利用するユニファイドコミュニケーション (UC) [1]が注目されている。UC によるリアルタイムコミュニケーションでは、IP 電話のような音声通話だけでなく、相手の表情を確認できる映像通信や、同じ資料を使って説明箇所を共有ポインタで指し示せる資料共有、相互に画面上で操作することで共同での編集作業を行なうことができる画面共有、などのアプリケーション (AP) を複数組み合わせることで同時に利用することができるため、意思決定の迅速化が期待できる。

しかし、企業内での UC 利用においては、ユーザが全ての AP を常に高い品質で利用できるとは限らない。なぜなら、UC の端末にユーザが日常業務において様々なアプリケーションを実行している PC を利用し、ネットワークに Web やメールなどのデータも流れるイントラネットを利用するため、端末・ネットワーク

の能力のばらつきと様々な AP との同時利用によるリソースの共用により、UC の AP が利用できる端末やネットワークのリソースが限られるためである。

本稿では、複数の AP を同時に利用するコミュニケーションにおいて、AP でのリソース利用を適切に制御することで、利用可能なリソースが限られる場合でもユーザの感じる品質 (体感品質) を高めることができる品質制御方式を提案する。本提案でのリソース利用の制御においては、高すぎる品質状態の設定によるリソース枯渇を防ぐとともに、同じリソース量でより高い品質レベルを実現するため、各 AP におけるリソース利用量に対する品質向上の割合を意識した品質向上方法の選択を行なう。またユーザが考える AP の優先度とその変化に応じてリソースを臨機応変に傾斜配分することで、優先度の高い AP の品質をより高め、体感品質を向上させる。

### 2. 品質制御の要件と課題

#### 2.1 品質制御の要件

体感品質を高めるために個々の AP の品質を向上さ

せる際には、その時点で利用可能なリソース量の範囲で実現可能な品質レベルに設定しなければならない。なぜなら、リソース量を考えることなく AP において無闇に高品質な状態を実行しようとする、特に端末においてリソースが枯渇してデータの送受信の失敗や再送などが発生し、コミュニケーションに必要なリアルタイム性を損ない、逆に体感品質を劣化させるためである。また、複数の AP を対象とした制御では、ユーザが優先的に利用する AP の品質を特に向上させることで、体感品質を高めることができる。このとき優先的に利用する AP はコミュニケーション中に変化するため、それに合わせて臨機応変に品質を変更しなければならない。例えば、UC による営業会議では、商品を映しての会話では映像通信を、計画の説明では図や表を表示するために資料共有を主に利用するため、優先度の変化に伴うユーザの品質変更要求に合わせて映像通信・資料共有のそれぞれの品質を特に向上させることが必要である。

以上より、本稿で提案する品質制御方式の要件を以下の二点とする。

- (1) **利用可能なリソース量に見合った品質レベルに設定すること**： 限られたリソースを無駄なく利用して品質を向上させることと、リソース不足を起ささないことを両立させる品質レベルに設定する。
- (2) **ユーザの要求に対応した動的な品質の向上が実現できること**： コミュニケーション中の AP の優先度の変化に基づいたユーザの品質要求要求に対応して、AP の品質を向上させる。品質向上の割合が大きいほど、体感品質もより高まる。

## 2.2 従来技術の課題

従来の品質制御方式として、IP 網における分散マルチメディアシステムにおいて複数の異なる AP を同時に利用する場合に、受信端末で検知した品質劣化を送信元にフィードバックし、予め決められたポリシーに従って複数の AP を適切に制御することで、その時点で実行可能な品質レベルを実現する「QMS」[2]がある。QMS は要件(1)を満たすものの、ポリシーは管理者によって予め決定されたものが常に適用されるため、コミュニケーション中のユーザの要求に対応して制御方法を変化させることはできず、要件(2)を満たさない。

別の品質制御方式として、マルチメディア通信システムにおいて、ユーザによる AP の品質レベル指定に対し、端末上のブローカ機能が自端末のリソース状態を確認後、相手端末でリソースを管理するブローカと交渉して指定された品質レベルが実現可能かどうか判

断し、実現可能なときのみ指定された品質レベルへの変更を行なう「QoS Broker」[3]がある。QoS Broker では、ユーザの要求に応じて動的に品質レベルが決定される場合もあるため要件(2)を一部満たしているが、端末やネットワークのリソースが不足する場合にユーザ要求による品質レベル変更が実現されないため、完全に満たしてはいない。また、指定された品質レベル変更が実現できない場合に、実現できないことのみを応答し、品質レベルの決定や品質調整はまったく行なわないため、要件(1)を満たさない。

## 3. 提案する品質制御方式

2章で述べた要件(1)(2)を同時に満たす品質制御方式について本章で説明する。

### 3.1 制御モデルと概要

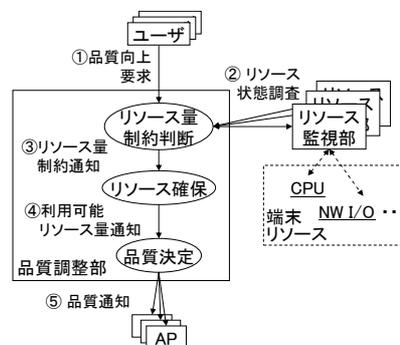


図1 制御モデル

図1に本方式の制御モデルを示す。品質調整部内の楕円は振る舞いを示している。本制御モデルは、複数のユーザが複数の AP を同時に利用するコミュニケーション実行中に、ユーザからの品質に対する要求に基づいて、端末のリソース状態に見合った品質を実現するものである。

本図を使って品質制御の概要を説明する。最初にユーザから AP の品質向上要求が行なわれる（図1 ①）。要求を受けた品質調整部はその時点で利用可能な端末のリソース量を知るために、各端末のリソース監視部に対し利用可能なリソース量を通知するよう要求する。リソース監視部が利用可能なリソース量を通知する（図1 ②）と、それを受信した品質調整部は、全ての通知から現状でのリソース量に関する制約を判定する。その結果（図1 ③）とユーザからの要求を踏まえて、重要でない AP の品質抑制も実施し、品質向上に利用可能なリソース量を確保する。確保されたリソース量に基づき（図1 ④）要求に応じて品質を向上させた状態を決定し、AP に対し新たな品質レベルを通知する

(図1 ⑤)。通知された AP が品質を変更することで、ユーザの要求に基づく品質制御が完了する。

2章で挙げた品質制御の要件に対し、要件(1)の実現のため、リソース監視部からのリソース量通知に基づいて、コミュニケーション中の全端末に共通したリソースの制約条件を明らかにした後に、それに基づいて品質決定を行なっている。また要件(2)の実現のため、ユーザからの品質向上要求を受け付けるとともに、単純に利用可能なリソースだけを使うのではなく、ユーザ要求に従って重要でない AP の品質を下げることで、ユーザ要求を高いレベルで実現するためのリソース確保を行なっている。

### 3.2 リソースと品質の関係と管理

本節では品質レベル決定に利用する、リソースと品質の関係と、品質調整部における管理モデルについて述べる。

#### 3.2.1 リソースと品質の関係

コミュニケーションで利用される AP は各々品質を決定するパラメータを持ち、それが複数存在する場合もある。またパラメータは各々複数の値を取り得る。映像通信 AP におけるパラメータの例を表1に示す。

表1 映像通信におけるパラメータ

パラメータ	値	単位
ビットレート	512K, 1M, 2M	bps
フレームレート	15, 30	fps
色	8, 16, 32	bit

本提案方式では、これらのパラメータの値が一意に決まると、それに対応した品質の AP の状態（品質状態）が一つ決まるとする。つまり表1の例では18通りの品質状態が存在する。そしてある一つの品質状態が決定すると、その実現に必要なリソース量も一意に決まる。例えば品質状態実現に必要なリソースの中から、CPU 処理量とネットワーク帯域を取り上げると、表2のような関係が定義される。ここではCPU処理量を、ある標準のCPUを仮定した場合のCPU使用率として表現している。

表2 パラメータとリソースの関係

パラメータ			必要リソース	
ビットレート	フレームレート	色	CPU処理量	NW帯域
512k	15	8	5	512k
512k	15	16	10	1.5M
1M	30	16	20	2M
2M	30	32	40	2.5M

AP によっては操作や扱うコンテンツにより、同じ品質状態によっても必要なリソース量が変動する場合がある。ここで「品質状態の実現に必要なリソース量」としているのは、その品質状態を常に維持することが可能なリソース量という意味であり、変動が大きい場合には、常にその必要量をカバーするために、最大値を採用する。

#### 3.2.2 管理モデル

品質調整部では、各 AP について、品質状態とパラメータ、リソースの関係を次のように管理する。

- パラメータ
  - ・ 一つの AP の一つのパラメータについて、設定可能な値から最大  $2n+1$  個を選択し ( $P_m, \dots, P_0, \dots, P_n$ ) として管理する。
  - ・ AP において標準的な品質状態で選択されるパラメータ値を  $P_0$  とし、 $P_x$  において、 $x$  が大きいほど品質が高くなる。
  - ・ パラメータとして設定可能な値が  $2n+1$  個未満の場合は、 $P_0$  の前後に連続する  $P_x$  として管理する。
- 品質状態とリソース
  - ・ 品質状態はパラメータ値の組合せであるため、 $m$  種類のパラメータに対し ( $P_{x_1}, \dots, P_{x_m}$ ) として表現可能で、最大  $(n+1)^m$  個の品質状態が定義される。
  - ・ 品質制御の指標として利用するリソースの集合を  $S=\{1, \dots, s\}$  とする場合、個々の品質状態について各リソースの必要量の組合せ ( $x_1, \dots, x_s$ ) が決まる。
  - ・ 品質調整部では、( $P_{x_1}, \dots, P_{x_m}$ ) と ( $x_1, \dots, x_s$ ) を紐付けて管理する。これらは予め計測された値が設定される。

例えば表1・表2で、品質制御の指標として、CPU処理量とネットワーク帯域を利用する場合、ビットレート1Mbps、フレームレート30fps、色数16bitを標準的な品質状態で選択されるパラメータ値とすると、品質状態とリソースの関係として  $(0, 0, 0) \rightarrow (10\%, 1.5Mbps)$  が品質調整部により管理される。

### 3.3 品質向上要求

本節より品質制御の具体的な処理手順について、順を追って説明する。

ユーザからの品質向上要求においては、「品質向上対象」と「品質抑制許容対象」の二つを組み合わせで指定する。

#### 3.3.1 品質向上対象の指定方法

品質向上対象は次の 1,2 およびその組合せによって

指定される。

1. AP を指定
2. AP とそのパラメータを指定

実際の利用場面として、1 は「これから映像をメインに使うから、とにかく映像通信の品質を上げたい」、2 は「細かいところが見たいから映像通信の解像度を上げたい」といった例が挙げられる。また、これらを組み合わせて「映像通信は解像度と音声通信の品質を上げたい」という要求も可能である。

品質向上において具体的な品質状態や品質の上げ幅などを指定する場合、リソースの状態によっては指定の品質状態を実現できないだけでなく、リソース不足を引き起こし、逆に品質劣化を招いてしまう。そこで本提案方式では品質向上について、その対象のみを指定する。

### 3.3.2 品質抑制許容対象の指定方法

本提案方式では、品質向上要求時に品質を抑制しても構わない AP を指定する。これは、要求された品質向上を高いレベルで実現することと、品質向上のためのリソースが確保できない場合に、重要でない AP の品質を抑制することで利用可能なリソースを捻出するという二つの目的がある。

品質抑制を許容する対象として、次の二通りが考えられる。

1. 品質向上対象以外の全ての AP
2. 具体的に指定された AP

本制御方式では品質向上を主であるため、品質制御対象が具体的に指定されない場合は1とし、品質向上対象以外のすべての AP やパラメータを品質抑制の対象とする。例えば、映像通信と資料共有を組み合わせた UC において、品質向上対象として「映像通信のフレームレート」が指定された場合の品質抑制許容対象は、ビットレートや色など（表1参照）映像通信のフレームレート以外のパラメータと資料共有の全パラメータとなる。2 の場合は、具体的に品質向上対象と品質抑制許容対象を指定するため、現在の品質状態を維持して欲しい AP を間接的に指定することにもなる。品質向上との組合せが利用される具体的なコミュニケーション中の状況として「今までメインで使っていた資料共有はとりあえずもう使わないので、そちらの品質を下げて、次に映像を見るので映像通信の品質を上げたい」といった例が挙げられる。

### 3.4 リソース量制約判定

品質向上のための制御に伴って、端末でリソース不足が発生して、逆に品質が大きく劣化することを防ぐため、制御前にその時点で利用可能なリソース量を確

認し、制約条件を判定する。

まず品質調整部は、各端末のリソース状態を知るために、リソース監視部に対し、品質制御の指標となるリソース  $S$  全てについて、その時点での利用可能なリソース量を通知するように要求する。各リソース監視部は自分が受け持つ端末のリソース状態を常に計測しており、通知要求に対して、その前の一定時間の計測値から利用可能なリソース量を算出して通知する。例えば CPU 使用率のように、端末性能の差異が値に影響を与えるものについては、品質調整部で仮定している性能と実際の端末の性能の違いを、予め与えられた係数によって補正した値を通知する。

全端末について利用可能なリソース量の通知を受けた品質調整部は、リソース種別毎に最低の通知内容を選択し、端末における現状のリソース制約条件を判定する。この値の組合せを  $R_{idle} = (r_{idle,1}, \dots, r_{idle,s})$  とする。例えば CPU 処理量とネットワーク帯域について、端末 A が使用率換算で 10% と 5Mbps、端末 B が 15% と 2Mbps であれば、 $R_{idle} = (10\%, 2Mbps)$  が品質制御のための制約条件となる。

### 3.5 リソース確保

リソース確保には次の二つのステップがある。

#### 3.5.1 品質抑制によるリソースの確保

品質状態を決定する前に、品質抑制により品質向上に利用できるリソース量を増やす。具体的には 3.3.2 で述べた形式で指定された品質抑制許容対象のパラメータについて、現在のパラメータ値  $P_x$  が  $x \geq 0$  であるもの全てについて、パラメータ値を標準的な品質状態である  $P_0$  とする。パラメータ値を  $P_x$  から  $P_0$  に変更することで、必要リソース量が減少するため、その差分が品質向上に利用可能となる。この差分を  $R_{diff} = (x_{diff,1}, \dots, x_{diff,s})$  とする場合、3.4 で把握された  $R_{idle}$  と  $R_{diff}$  の各値の合計が実際に品質向上に利用可能なリソース量となり、これを  $R_{com}$  とする。

#### 3.5.2 最低限必要なリソース量の算出

次に、要求された品質向上に最低限必要なリソース量を算出する。品質制御の指標として利用するリソースが複数ある場合は、それぞれについて算出する。

品質向上対象として「AP を指定」の場合、指定された AP の複数のパラメータのうち、一つのパラメータについて現在のパラメータ値  $P_x$  から  $P_{x+1}$  に変更する場合のリソース量の差分を算出する。リソースの有効利用の観点から最適なパラメータを選択するために、異なるリソースを一つの尺度で扱うための品質向上コスト  $C$  を導入する。指標として利用するリソースの集合  $S$  に対し、あるパラメータの変更による品質向上コ

スト C を、利用可能なリソース量に対する影響として次のように定式化する。

$$C = \sum_{j=1}^s \alpha_j (r_{x+1,j} - r_{x,j}) / x_{con,j}$$

$r_{x,j}$ : パラメータ値  $P_x$  におけるリソース  $j$  の必要量  
 $x_{con,j}$ : 品質向上に利用可能なリソース  $j$  の量  
 $\alpha_j$ : リソース毎に定められた品質に対する影響度 (重み)

「AP を指定」におけるパラメータの選択では、各パラメータについて  $C$  を算出し、 $C$  が最小値を取るパラメータにおける現在のパラメータ値  $P_x$  から  $P_{x+1}$  に変更する場合の各リソース量の差分  $R_{min} = (x_{min,1}, \dots, x_{min,s}) = (r_{x+1,1} - r_{x,1}, \dots, (r_{x+1,s} - r_{x,s}))$  が、品質向上に最低限必要なリソース量となる。

「AP とそのパラメータを指定」の場合には、指定されたパラメータについて同様に  $R_{min}$  を算出する。組合せによる指定の場合には、個々の対象指定について算出される  $R_{min}$  の各値の合計を、最終的な  $R_{min}$  とする。

### 3.5.3 品質向上の可能性判断

$R_{con}$ ,  $R_{min}$  に基づいて、要求された品質向上が可能かどうかを判断する。要求された品質向上を可能にするためには、全てのリソース  $j$  について  $x_{con,j} > x_{min,j}$  でなければならない。

不足するリソースがあるために品質向上が不可能な場合、品質向上が最低限可能になるリソースが利用可能になるまで、品質抑制許容対象のパラメータについて、さらに品質を下げる。このとき品質抑制が少ないほうが UC 全体の品質を維持することになるため、効率的にリソース確保できるパラメータを選択しなければならない。そこで、次に定式化する品質抑制効果  $E$  をパラメータ選択の指標として利用する。

$$E = \sum \alpha_j (r_{x,j} - r_{x-1,j}) / (x_{min,j} - x_{con,j}) \quad |\forall j, (x_{min,j} < x_{con,j})$$

$r_{x,j}$ : パラメータ値  $P_x$  におけるリソース  $j$  の必要量  
 $x_{con,j}$ : 品質向上に利用可能なリソース  $j$  の量  
 $x_{min,j}$ : 品質向上に最低限必要なリソース  $j$  の量  
 $\alpha_j$ : リソース毎に定められた品質に対する影響度 (重み)

パラメータの値を変化させた場合の影響度を測るという点では  $C$  と類似だが、次の二点が異なる。

- ・ 分母はこの品質抑制で確保すべきリソース量
- ・ 総和の対象はリソース全てではなく、現在不足しているリソースのみ

コミュニケーションにおいて大きく品質を損なったとユーザに感じさせることなく品質抑制を行なうため、品質抑制対象がある一つのパラメータに偏るのではな

く、満遍なく品質を下げるのが望ましい。そのため、 $E$  を使った品質抑制対象のパラメータの選択は、次のステップで行なう。

1. 全ての品質抑制許容対象パラメータについて  $E$  を算出。
2.  $E$  が大きなパラメータ程、値を現在の  $P_x$  から  $P_{x-1}$  に変更した場合により多くのリソースが利用可能になるため、 $E$  が最大となるパラメータ  $q$  を選択。
3.  $q$  の値を現在の  $P_x$  から  $P_{x-1}$  とした場合の必要リソース量の差分と  $R_{con}$  の各値の合計を新たな  $R_{con}$  とする。
4.  $R_{con}$ ,  $R_{min}$  の各値を比較し、品質向上が可能かどうか判断する。品質向上が可能であれば、パラメータ選択は終了。
5. 4 で品質向上が不可能な場合、次に大きな  $E$  になるパラメータ  $q$  選択し、ステップ 3 に戻る
6. 選択すべき  $E$  が無い場合、ステップ 1 に戻る。

### 3.6 品質決定

リソース確保により利用可能となったリソース量を効率的に利用可能なパラメータを選択し、可能な限り高い品質状態を決定する。

最初に  $R_{min}$  の算出時に選択したパラメータについて、品質を向上させるものとする。このために必要なリソース量  $R_{min}$  と  $R_{con}$  の各値の差分を新たな  $R_{con}$  とする。

以降の効率的なパラメータの選択のために、3.5.2 で定式化した品質向上コスト  $C$  を利用する。パラメータ選択のステップを次に示す。

1. すべての品質向上対象パラメータについて、 $C$  を算出し、 $C$  が最大となるパラメータ  $q$  を選択
2.  $q$  の値を現在の  $P_x$  から  $P_{x+1}$  とした場合の必要リソース量の差分を算出し、各値における  $R_{con}$  との差分を新たな  $R_{con}$  とする。
3.  $R_{con}$  において全ての値が正であれば、選択した  $q$  における品質向上は可能となる。負の値が一つでもある場合には選択を終了する。

品質向上におけるパラメータ選択ではステップ 3 の次に、再度 1 から繰り返す方法と、1 で算出した  $C$  がなくなるまで順に選択していく方法の二つが考えられる。前者は指定された範囲で満遍なく品質を上げる方法であり、後者は品質向上効率を優先する方法で、ある特定の AP やパラメータのみを集中的に品質向上させる可能性がある。この二つの方法についてはユーザによる要求の中で指定させ、それに従うものとする。

### 3.7 AP に対する品質の指示

3.5.1, 3.5.3 で決定した品質抑制のためのパラメータ

値の変更と、3.6 で決定した品質向上のためのパラメータ値の変更を、それぞれ対象となった AP に対し通知する。通知を受けた各 AP は、それによってパラメータ値を変更することで、意図したとおりの品質向上状態が実現される。

## 4. 評価と考察

### 4.1 評価のためのシミュレーション

本提案方式と既存技術である QoS Broker[3]を以下のようなシミュレーションにより比較する。

#### 4.1.1 前提条件

シミュレーションの前提条件は以下の通り

- AP1・AP2 の 2 種類のコミュニケーション用 AP を同時に利用する場面を想定する。
- 品質制御の指標とするパラメータは CPU 処理能力を用いる。5 段階 ( $n=2$ ) のパラメータ値があるものとする。

表 3 にパラメータ値  $P_x$  と必要なリソース量を示した。ここではある CPU 処理能力を使用率で表現している。性能の低い端末を想定しているため、CPU 使用率を高めの値としている。

表 3 AP 毎のパラメータ値

パラメータ値 $P_x$	AP1(%)	AP2(%)
$P_2$	60	65
$P_1$	50	50
$P_0$	40	35
$P_{-1}$	30	20
$P_{-2}$	25	15

#### 4.1.2 試行内容

次の通りの品質制御シミュレーションを行った。

- AP1・AP2 とともに品質制御前は標準の品質状態とし、AP2 を優先的に利用するために AP2 の品質の向上させる制御を行う。
- 端末における現状の余剰リソース量を乱数で与える。標準状態で既に  $40+35=75\%$  の CPU 使用率であるため、 $0 < r_{idle} < 25$  とする。
- 既存制御方式におけるユーザの要求する品質レベルも乱数で与える。本シミュレーションでは、CPU 使用率の値で品質レベルを指定することとする。現状および最高品質は認識しているものとし、36 ~ 65 が品質レベルとして要求される。
- 提案制御方式では「AP2」という AP 指定で品質向上を要求する。

以上を 10000 回試行した結果を次節以降に示す。

## 4.2 シミュレーション結果と考察

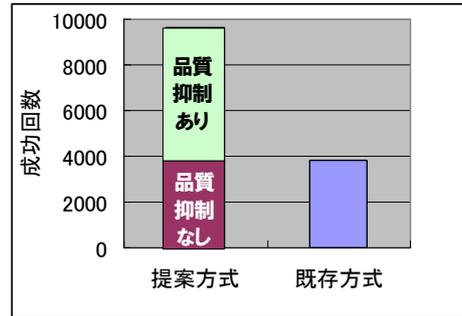


図 2 品質向上成功回数 (n=10000)

図 2 に示した品質向上成功回数においては、本提案方式による品質向上の成功回数が既存技術方式の約 2.5 倍となった。既存制御方式の品質向上失敗は利用可能なリソース量を超える品質レベルをユーザが要求したためなのに対し、本提案方式の品質向上失敗は余剰リソースが非常に少なく、品質向上の余地がなかったためであり、本提案方式が 2 章で述べた品質制御方式の要件(1)を満たしていると言える。

また本制御方式による品質向上回数の内訳を見ると、AP1 の品質を下げることによるリソース確保を行うことで初めて AP2 の品質向上が可能になった回数が全体の 6 割を占める。この結果は、重要でない AP の品質を下げて品質向上のためのリソースを確保する本提案方式の効果を示している。

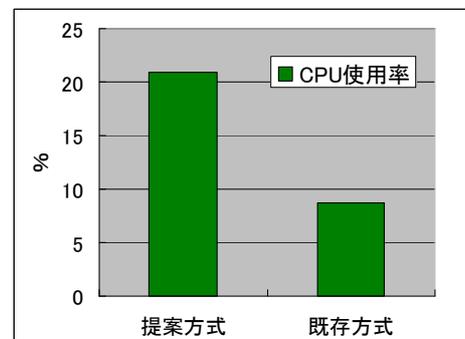


図 3 品質向上時に追加で割り当てられたリソース量の平均

AP2 の品質向上が成功した場合に、AP2 に対し品質向上のために追加で割り当てられたリソース量の平均を図 3 に示した。既存方式と比較して提案方式では 2.4 倍のリソースを AP2 に追加割当することができた。この割当により、品質も同様の割合で向上できたと考えられ、品質制御方式の要件(2)で挙げられている「ユーザの要求に対応した品質の向上」を既存方式に比べて、

より高いレベルで実現できている。

最後にユーザの体感する品質について考察する。本制御方式により、コミュニケーション中に必要に応じて AP1, AP2 をそれぞれ最高品質まで向上させられる。AP の品質切替ができない場合、 $60+65=125\%$ であるから、本シミュレーションで仮定した PC の 1.25 倍以上の CPU 性能の PC でなければ、同時に両 AP を最高品質で利用できない。本制御方式ではその 1.25 倍の CPU 性能の端末と見かけ上同等の品質を実現できているとすると、本制御方式の下、両 AP をそれぞれ最高品質で利用する際に実際に消費するリソース量は、75% (AP1:60%, AP2:15%) ~ 90% (AP1:25%, AP2:65%) なので、約 1.4~1.7 倍の CPU 性能の PC と同等の体感品質を実現していると考えられる。

## 5. おわりに

本稿では、複数の AP を同時に利用するコミュニケーションにおける品質制御方式について提案した。本提案方式では、リソース利用効率を考慮した品質向上方法の選択と、ユーザの考える AP の優先度変更に応じた動的品質向上により、限られたリソースでも、より高い体感品質を実現することが可能となった。

今後の課題として、現在は事前に与える必要がある品質状態における必要なリソース量と端末性能の差異を解消する係数の自動取得が挙げられる。例えば AP の状態とリソース状態を計測することで、リソース量を取得することや、その比較により係数を算出するなどである。もう一つの課題として自律的な制御の実現が挙げられる。現在ユーザが明示的に指示している優先度の変更を、AP の状態から推測することや、リソース状態の変動に追従して、最適な状態を維持することの実現が必要である。

## 参考文献

- 1) Gartner, Inc., "Magic Quadrant for Unified Communications, 2006", June 2006
- 2) 加藤由花、佐々木徹、箱崎勝也：分散マルチメディアシステムにおけるアプリケーション QoS 管理手法の提案とその実装、情報処理学会論文誌, Vol. 42, No. 12, pp.2941-2951
- 3) Klala Nahrstedt, Jonathan M. Smith: The QoS Broker, IEEE Multimedia, Vol 2, No 1 (Spring 1995), pp 53-67