

複数種別資源同時割当てを前提とした次世代ネットワークふくそう制御方式

畠山 賢一^{*1}, 栗林 伸一^{*2}

Congestion control method for joint multiple resource allocation in all-IP networks

Kenichi HATAKEYAMA^{*1} and Shin-ichi KURIBAYASHI^{*2},

ABSTRACT : *In all-IP networks where all types of services are integrated using IP technology, it is highly likely that traffic congestion in one service causes congestion in other services, or that abnormal traffic in one service degrades the quality of other services.*

This paper proposes the way how to implement the flexible reallocation of both computing and network resource between services, in order to reduce the chance of congestion in an all-IP network, assuming that both computing and network resources are simultaneously allocated to each service. This paper also proposes issues to be resolved to implement the smart request restriction scheme, which restricts or make the size of allocated resource small for those request that require a large amount of resource of the congested type.

Keywords : congestion control, resource reallocation, joint resource allocation, all-IP network

(Received August 19, 2008)

1. まえがき

メール, Webアクセス, ブログ, 動画配信だけでなく, 固定電話, 携帯電話, テレビ, センサ情報など様々なサービスを提供するオールIPネットワーク⁽¹⁾では, インターネット上で問題となっているウイルス, 不正アクセス, DDoS攻撃などが今までそのような問題が発生していなかったサービスにも拡大する懸念がある。また, 今まで特定サービスでしか発生していなかったふくそうが他サービスのふくそうを誘発したり, 特定サービスの異常トラヒック(スパムなども含む)が他のサービスの品質劣化を引き起こすという可能性も高くなる。さらに, オールIPネットワークは, 従来のネットワークのようにネットワーク資源(帯域)だけでなくコンピューティング資源(計算能力)も‘同時に’提供し, かつサービス品質の大きく異なる複数のサービスを統合するという特徴を持つ。オールIPネットワークが本格導入ならびに社会インフラ化する前に, これらに対応する対策を確立しておくことは大変重要である。

そのため, 筆者らは既存ネットワークのふくそう制御

の調査・分析に基づくオールIPネットワークふくそう制御の基本的な考え方を明らかにした。さらに, その基本的な考え方を実現する対策として, 混雑している資源種別に着目してサービス間で予め割り当てられた資源を融通しふくそうの発生を回避する方式(資源融通方式), ふくそう資源種別を多く必要とする要求をまず規制することによりサービス性と資源効率を高める方式(特定要求種別規制方式), の可能性を提案した^{(2), (3)}。

本論文は, 資源融通方式の具体的な実現法ならびに特定要求種別規制方式の実現上の課題を明らかにすることを目的とする。資源種別としては, コンピューティング資源(以後, 計算能力)とネットワーク資源(以後, 帯域)の2つの資源種別を想定する。また, 各要求に対して計算能力と帯域を同時に割当てるものとする。

2章では, 前提とするふくそう制御ステップを説明し, 資源融通方式と特定要求種別規制方式の位置付けを示す。3章では, 計算能力と帯域の両方の資源種別を融通対象とした資源融通を前提し, 資源融通の可否と融通量の決定手順, 融通の実施・解除手順, を明らかにする。また, C言語を用いたシミュレーション評価により, 資源融通効果を定量的に明らかにする。4章では, 特定要求種別規制方式の概要と課題を述べる。5章はまとめである。

*1: 工学研究科物理情報工学専攻修士学生

*2: 理工学部情報科学科教授 (kuribayashi@st.seikei.ac.jp)

2. 前提とするふくそう制御ステップ

通常、資源量は想定要求量に基づいて準備される。しかし、想定を大幅に超えた要求が発生する可能性があり、その場合には通常サービス規制という対策がとられる。複数資源種別を扱うオールIPネットワークでは、特定の資源種別や資源だけが込み合う状況も想定される。従来の考え方では特定の資源種別だけがふくそうしても、サービス品質を維持するために全ての要求種別を対象に規制が行われ、サービス性ならびに資源使用効率が大幅に低下する可能性がある。

その対策として、特定の資源種別だけが組み合っても全てのサービス要求を直ぐに規制するのではなく、組み合っている資源種別の空き資源を一時的にサービス間で融通することによりふくそうの回避を図るのが資源融通方式の狙いである。それでも混雑が緩和されずふくそうが発生した場合には、ふくそう資源種別を多く要求するサービス要求(ふくそうを助長する要求)だけをまず規制しサービス性および資源効率を高めるのが特定要求種別規制方式の狙いである。

これらを踏まえたオールIPネットワークのふくそう制御ステップを図1に示す。ステップ1が資源融通、ステップ2が特定要求種別規制に対応する。なお、資源融通はリアルタイムに実施するのではなく、数分以上の周期で実施することを想定する。また、ステップ0は平常時に最適資源割当て^{(4), (5)}に対応する。

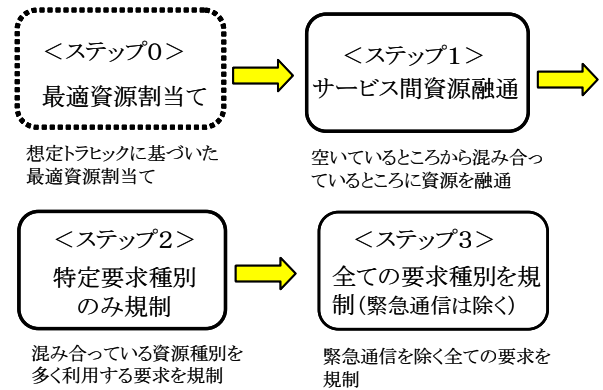
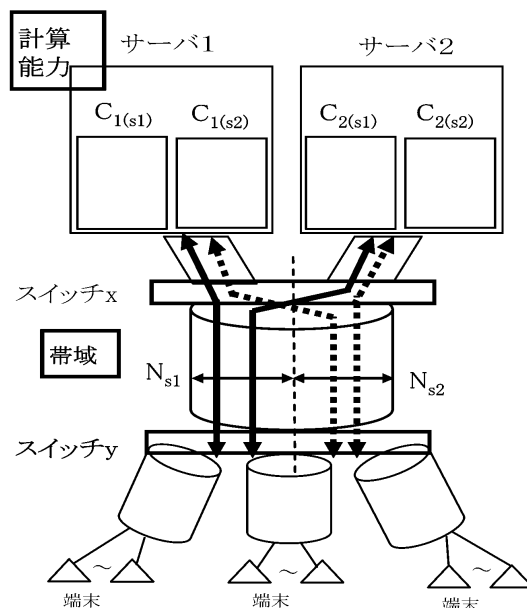


図1 前提とするふくそう制御ステップ

3. 資源融通方式の実現法

3.1 検討モデル

オールIPネットワークの最も基本的なモデルとして図2を考える。図2では、サービス1とサービス2の2つのサービス種別を想定する。スイッチxとスイッチy間のネットワーク資源を事前のトラフィック予想に基づきサービス1用とサービス2用に予め割当てる。サーバ1, 2はそれぞれ物理的に離れたところに存在し、どちらもサービス1とサービス2の処理を行う。サーバ1とサーバ2を合わせた総クロック数を正規化したものをCmaxとし、それを事前のトラフィック予想に基づきサービス1とサービス2に分けて割当てる。その際、サービス毎に割り当てる計算能力がサーバ1とサーバ2で均等になるようにする。



- ・スイッチxとスイッチyの帯域をサービス1とサービス2で共用
- ・サーバ1とサーバ2の計算能力をサービス1とサービス2で共用(各サービスはサーバ1とサーバ2の両方を利用可能)
- ・資源融通量の決定ならびに実際の帯域融通はスイッチyで実施。計算能力の実際の融通はサーバで実施。

N_{s1}: サービス1向けに割り当てられた帯域
N_{s2}: サービス2向けに割り当てられた帯域
C_{1(s1)}: サーバ1のサービス1向けに割り当てられた計算能力
C_{2(s1)}: サーバ2のサービス1向けに割り当てられた計算能力
C_{1(s2)}: サーバ1のサービス2向けに割り当てられた計算能力
C_{2(s2)}: サーバ2のサービス2向けに割り当てられた計算能力
C_{1max}: サーバ1の総計算能力 (=C_{1(s1)} + C_{1(s2)} = C_{1(s1)0} + C_{1(s2)0})
C_{2max}: サーバ2の総計算能力 (=C_{2(s1)} + C_{2(s2)} = C_{2(s1)0} + C_{2(s2)0})
Cmax: 総計算能力 (=C_{1max} + C_{2max})
Nmax: 総帯域 (=N_{s1} + N_{s2} = N_{s10} + N_{s20})

→: サービス1の情報の流れ
...→: サービス2の情報の流れ

図2 基本評価モデル

3.2 資源融通イメージ

図3は、サービス1に想定より大きな計算能力を必要とする要求が多数到着し計算能力が混み合っているが帯域には余裕があり、かつサービス2は計算能力も帯域も余裕がある状況を想定した資源融通の例を示す。この例では、サービス2向けに予め割り当てられた計算能力や帯域の一部をサービス1に一時的に融通する。これにより、サービス1の要求処理数を増加させ、ネットワーク全体の要求棄却率を削減できる。このように、資源融通は資源に余裕のあるサービスと資源に余裕のないサービスが存在して初めて有効となる。なお、サービス間の計算能力融通は、サーバ毎に個別に考えるのではなく、サーバ1とサーバ2の総和で考えるものとする。例えば、

サービス1からサービス2へ計算能力 Pro を融通する場合、 $Pro/2$ ずつ各サーバで融通する。

ところで、図2のモデルでは資源融通の可否ならびに最適融通量の決定はスイッチ y で実施する（集中管理サーバで実施する方式は制御に時間がかかるため）。スイッチ y は各資源の利用状況を“一定周期で”把握する必要がある。ネットワーク資源の利用状況はスイッチ y 自身が把握できるが、計算能力の利用状況はサーバ1ならびにサーバ2からの通知により把握する。その処理シーケンスを図4に示す。なお、MPLS網⁽⁶⁾であれば、スイッチ y とサーバ間の情報のやりとりはLDPメッセージを拡張したメッセージ（図5）を用いて実施することが考えられる。

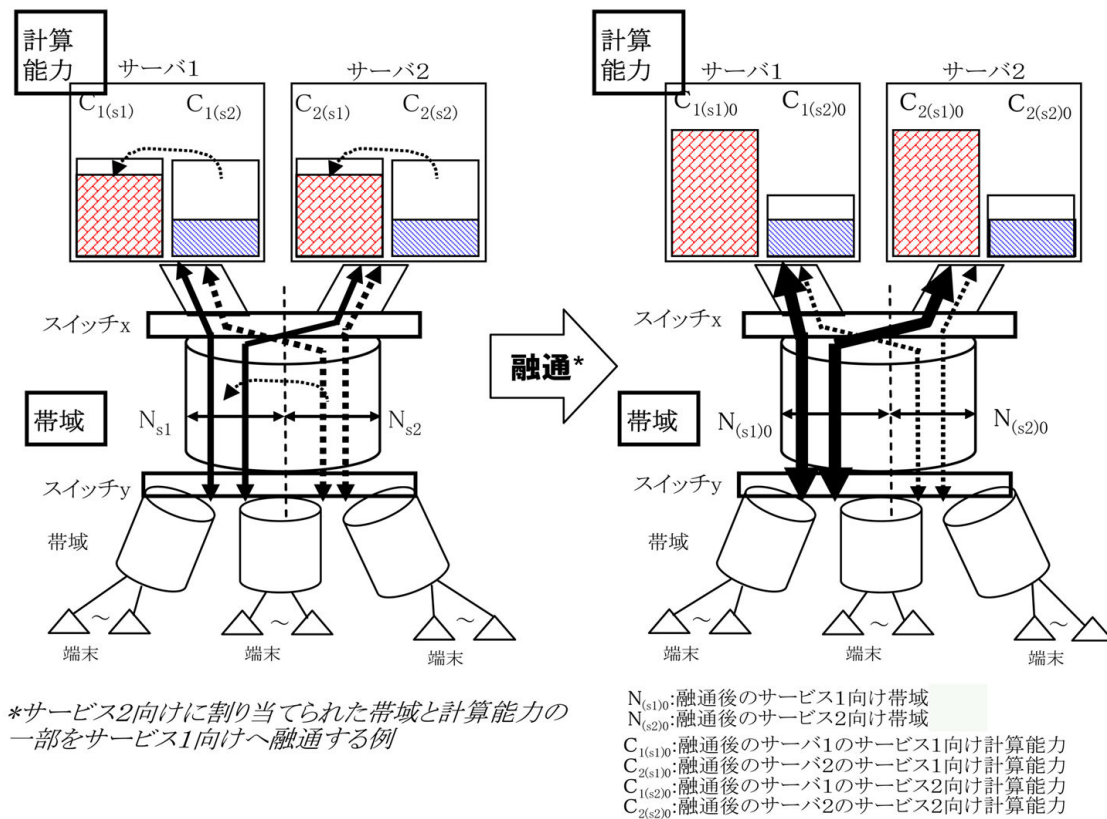


図3 資源融通の例

3.3 資源融通の可否と融通量の決定手順

資源融通は資源に余裕のあるサービスと資源に余裕のないサービスが存在して初めて有効となるため、想定される融通形態を整理する必要がある。計算能力と帯域の両方を融通対象とした場合の融通形態は図6に示す4つに分類できる。但し、形態3ならびに形態4では、計算

能力と帯域を同時に融通するのではなく、どちらか一方の資源種別の融通だけを行う。詳細は3.4節を参照のこと。

融通形態ごとに融通の決定方法が異なるため、以下では融通形態ごとに融通可否ならびに融通量の決定手順を説明する。融通量は、‘融通可能な最大資源量’とする。

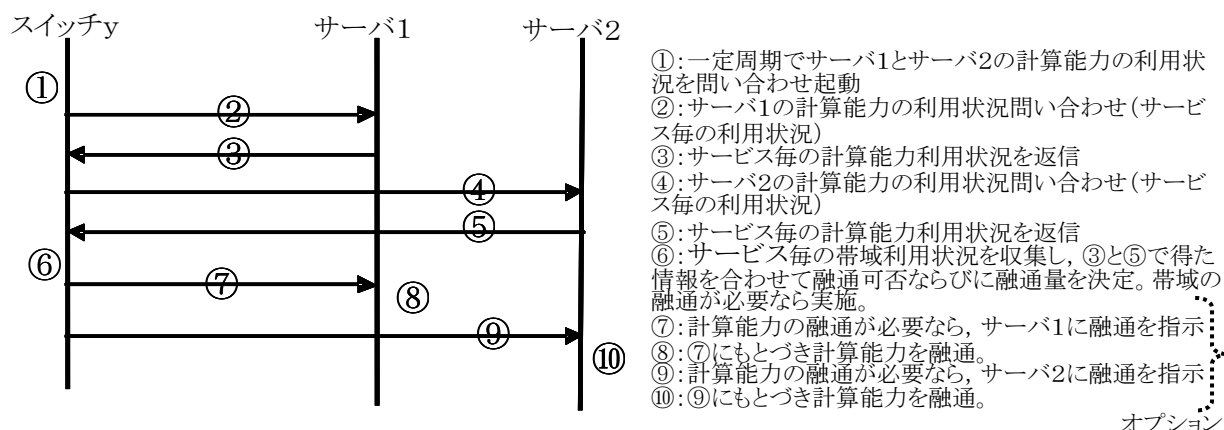


図4 各資源の利用状況把握とそれに基づく資源融通決定&指示シーケンス例

なお、記号 α , β , γ_1 , γ_2 , δ , ε などは図6を参照のこと。

3. 3. 1 形態1

<前提>

・X: γ_2 を全てサービス1向けに融通し(サービス1向けの残帯域は $\gamma_1 + \gamma_2$ になる), それを全て使いきるために必要なサービス1向け計算能力量

$$X = \{(C_{1(s1)} + C_{2(s1)}) / (N_{s1} - \gamma_1)\} \cdot (\gamma_1 + \gamma_2) \quad (1)$$

・Y: サービス1向け計算能力 ($\alpha + \beta$) を全て使いきるのに必要なサービス1向け帯域量

$$Y = \{(\gamma_1 + \gamma_2) / X\} \cdot (\alpha + \beta) \quad (2)$$

<融通量の算出手順>

if $\{(\alpha + \beta) \geq X\}$

サービス2向け残り帯域を全て ($= \gamma_2$) サービス1向けへ融通。計算能力はサービス2向け残り計算能力の中からXだけサービス1向けへ融通(図7に決定イメージを示す)。

else

if $(Y \leq \gamma_1)$

帯域の融通なし。計算能力はサービス2向け残り計算能力を全て ($= \alpha + \beta$) をサービス1向けへ融通。

else

サービス2向け残り帯域の中から $(Y - \gamma_1)$ をサービス1向けへ融通。計算能力はサービス2向け残り計算能力を全て ($= \alpha + \beta$) をサービス1向けへ融通。

3. 3. 2 形態2

<前提>

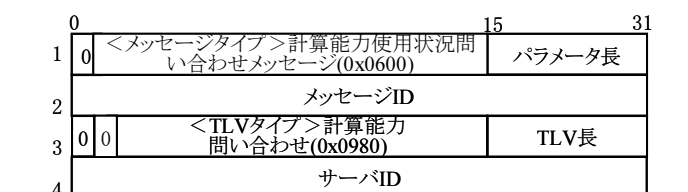
・X: γ_1 を全てサービス2向けに融通し(サービス2向けの残帯域は $\gamma_1 + \gamma_2$ になる), それを全て使いきるために必要なサービス2向け計算能力量

$$X = \{(C_{1(s2)} + C_{2(s2)}) / (N_{s2} - \gamma_2)\} \cdot (\gamma_1 + \gamma_2) \quad (3)$$

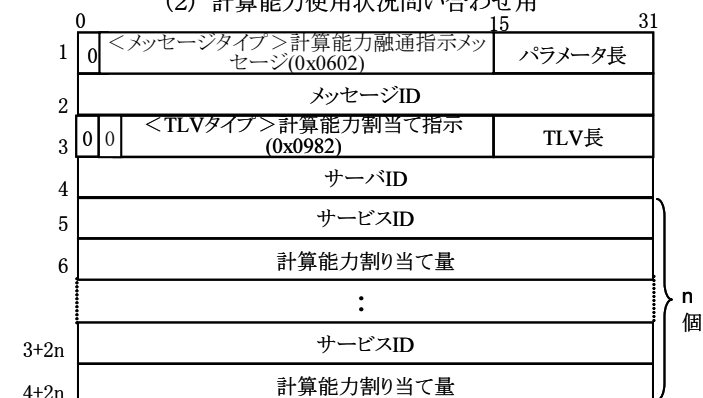
・Y: サービス2向け計算能力 ($\delta + \varepsilon$) を全て使いきるのに必要なサービス2向け帯域量



(1) 計算能力使用状況応答、報告用



(2) 計算能力使用状況問い合わせ用



(3) 計算能力割当て指示用

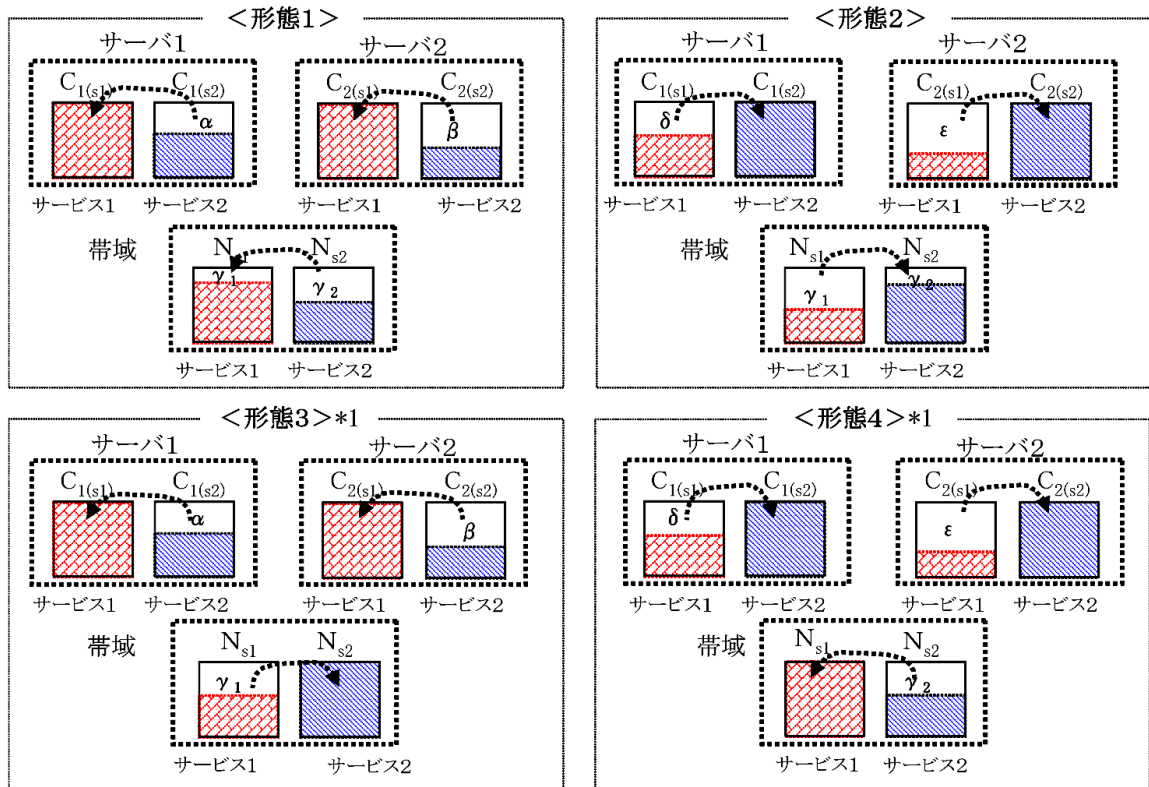
図5 計算能力ふくそう時のLDPメッセージの例

$$Y = \{(\gamma_1 + \gamma_2) / X\} \cdot (\delta + \varepsilon) \quad (4)$$

<融通量の決定手順>

if $\{(\delta + \varepsilon) \geq X\}$

サービス1向け残り帯域を全て ($= \gamma_1$) サービス2向けへ融通。計算能力はサービス1向け残り計算能力の中



α:サーバ1のサービス2向けに残された計算能力, β:サーバ2のサービス2向けに残された計算能力
δ:サーバ1のサービス1向けに残された計算能力, ε:サーバ2のサービス1向けに残された計算能力
γ₁: サービス1向けに残された帯域 γ₂: サービス2向けに残された帯域

*1: 帯域または計算能力の一方の資源種別のみ融通(同時に両方の資源種別を融通するケースはない)

図6 想定される資源融通形態

からXだけサービス2向けへ融通。

else

if ($Y \leq \gamma_2$)

帯域の融通なし。計算能力はサービス1向け残り計算能力を全て(= $\delta + \epsilon$)をサービス2向けへ融通。

else

サービス1向け残り帯域の中から($Y - \gamma_2$)をサービス2向けへ融通。計算能力はサービス1向け残り計算能力を全て(= $\delta + \epsilon$)をサービス2向けへ融通。

3. 3. 3 形態3

<前提>

・V: γ_1 を全てサービス2向けに融通し, それを使い切るために必要なサービス2向け計算能力量

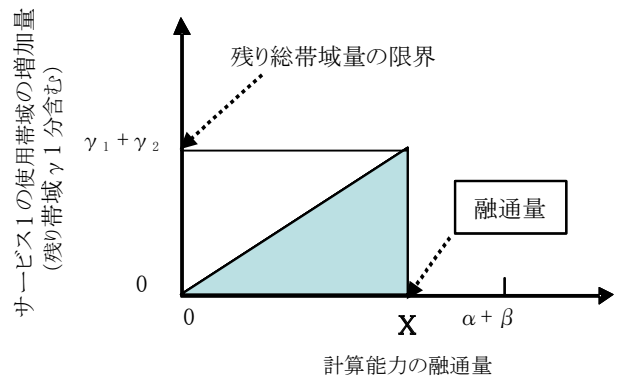
$$V = \{ \{ (C_{1(s2)} - \alpha) + (C_{2(s2)} - \beta) \} / N_{s2} \} \cdot \gamma_1 \quad (5)$$

・ γ_1' : サービス1からサービス2向けに融通する帯域量

$$\gamma_1' = (\alpha + \beta) \cdot N_{s2} / \{ (C_{1(s2)} - \alpha) + (C_{2(s2)} - \beta) \} \quad (6)$$

・W: ($\alpha + \beta$)を全てサービス1向けに融通し, それを使い切るために必要なサービス1向け帯域量

$$W = \{ (N_{s1} - \gamma_1) / \{ C_{1(s1)} + C_{2(s1)} \} \} \cdot (\alpha + \beta) \quad (7)$$



* 図6の資源融通形態1 (サービス2からサービス1へ資源融通) で, 残り帯域量の限界で融通量が決定する場合

図7 資源融通量と使用帯域増加量の関係例

・W': サービス2からサービス1向けに融通する計算能力量

$$W' = \{ \{ C_{1(s1)} + C_{2(s1)} \} / (N_{s1} - \gamma_1) \} \cdot \gamma_1 \quad (8)$$

<融通可否判断ならびに最適融通量の算出>

以下の2つのケース (ケース1, ケース2) でZが最も大きくなるケースが最適と考えて融通する。つまり, 最も多くの帯域を使用するという事は, 最も多くのサービスを処理できることに対応するからである (公平性

は考慮していない)。また、以下に当てはまらない場合には融通は実施しない。

<ケース 1>

if $\{(\alpha + \beta) \geq V\}$

サービス 1 向け残り帯域 γ_1 を全てサービス 2 向けへ融通。そのときの総使用帯域量 Z は

$$Z = N_{s1} + N_{s2} \quad (9)$$

else

サービス 1 向け残り帯域 γ_1 の中から γ_1' だけサービス 2 向けへ融通。そのときの総使用帯域量 Z は

$$Z = \{(N_{s1} - \gamma_1) + \gamma_1'\} + N_{s2} \quad (10)$$

<ケース 2>

if $\{\gamma_1 \geq W\}$

サービス 2 向け残り計算能力 $(\alpha + \beta)$ を全てサービス 1 向けへ融通。そのときの総使用帯域量 Z は

$$Z = \{(N_{s1} - \gamma_1) + W\} + N_{s2} \quad (11)$$

else

サービス 2 向け残り計算能力 $(\alpha + \beta)$ の中から W' だけサービス 1 向けへ融通。そのときの総使用帯域量 Z は

$$Z = N_{s1} + N_{s2} \quad (12)$$

3. 3. 4 形態 4

<前提>

・ V : γ_2 を全てサービス 1 向けに全て融通し、それを使い切るために必要なサービス 1 向け計算能力量

$$V = \{(C_{1(s1)} - \delta) + (C_{2(s1)} - \varepsilon)\} / N_{s1} \cdot \gamma_2 \quad (13)$$

・ γ_2' : サービス 2 からサービス 1 向けに融通する帯域量

$$\gamma_2' = (\delta + \varepsilon) \cdot N_{s1} / \{(C_{1(s1)} - \delta) + (C_{2(s1)} - \varepsilon)\} \quad (14)$$

・ W : $(\delta + \varepsilon)$ をサービス 2 向けに全て融通し、それを使い切るために必要なサービス 2 向け帯域

$$W = \{(N_{s2} - \gamma_2) / \{C_{1(s2)} + C_{2(s2)}\}\} \cdot (\delta + \varepsilon) \quad (15)$$

・ W' : サービス 2 からサービス 1 向けに融通する計算能力量

$$W' = \{(C_{1(s2)} + C_{2(s2)}) / (N_{s2} - \gamma_2)\} \cdot \gamma_2 \quad (16)$$

・ Z : サービス 1 ならびにサービス 2 で使用する帯域量の総和 ($N_{s1} + N_{s2}$ が最大)

<融通量の決定手順>

形態 3 と同様に、以下の 2 つのケース (ケース 1, ケース 2) で Z が最も大きくなるケースが最適と考えて融通する。また、以下に当てはまらない場合には融通は実施しない。

<ケース 1>

if $\{(\delta + \varepsilon) \geq V\}$

サービス 2 向け残り帯域 γ_2 を全てサービス 1 向けへ融通。そのときの総使用帯域量 Z は

$$Z = N_{s1} + N_{s2} \quad (17)$$

else

サービス 2 向け残り帯域 γ_2 の中から γ_2' だけサービス 1 向けへ融通。そのときの総使用帯域量 Z は

$$Z = \{(N_{s2} - \gamma_2) + \gamma_2'\} + N_{s1} \quad (18)$$

<ケース 2>

if $\{\gamma_2 \geq W\}$

サービス 1 向け残り計算能力 $(\delta + \varepsilon)$ を全てサービス 2 向けへ融通。そのときの総使用帯域量 Z_3 は

$$Z = \{(N_{s2} - \gamma_2) + W\} + N_{s1} \quad (19)$$

else

サービス 1 向け残り計算能力 $(\delta + \varepsilon)$ の中から W' だけ サービス 2 向けへ融通。そのときの総使用帯域量 Z_4 は

$$Z = N_{s1} + N_{s2} \quad (20)$$

3. 4 融通の解除・変更手順

3. 3 節に従い実施した資源融通は、資源利用状況を測定する周期ごとに以下の手順で見直す。

- 1) 融通元サービスで資源不足による要求棄却が発生している場合には (融通できない状態)、融通先サービスの状況に係わらず融通していた資源を融通元サービスに戻す。
- 2) 融通先サービスで資源不足による要求棄却が発生しかつ融通元サービスの資源にさらに余裕がある場合には (融通量が不足している状態)、3. 3 節で提案した手順に従い融通量を再計算し、さらに融通量を増やす。
- 3) 融通元サービスならびに融通先サービスのいずれでも資源不足による要求棄却が発生していない場合には (融通量が多すぎる状態)、3. 3 節で提案した手順に従い融通量を再計算し、融通量を必要な分まで削減する。

3. 5 シミュレーション評価

3. 5. 1 条件

- 1) C 言語を用いたコンピュータシミュレーションにより評価を行う。
- 2) 図 2 の検討モデルを前提とする。
- 3) 要求毎に必要なとなる計算能力と帯域の大きさの平均値をそれぞれ C , N で表す。計算能力と帯域の大きさはガウス分布に従うものとする。
- 4) サービス 1 とサービス 2 の要求発生間隔はそれぞれ平均 r_1 , r_2 の指数分布に従う。サービス 1 とサービス 2 の資源保留時間をそれぞれ H_1 (一定), H_2 と

する（一定）。

5) 要求の発生パターンは $\{C=a, N=b\}$ で表わす。

これは、 C と N がそれぞれ a, b の大きさの要求が繰り返し発生することを示す。

6) 以下で定義される要求棄却率を評価で使用する。

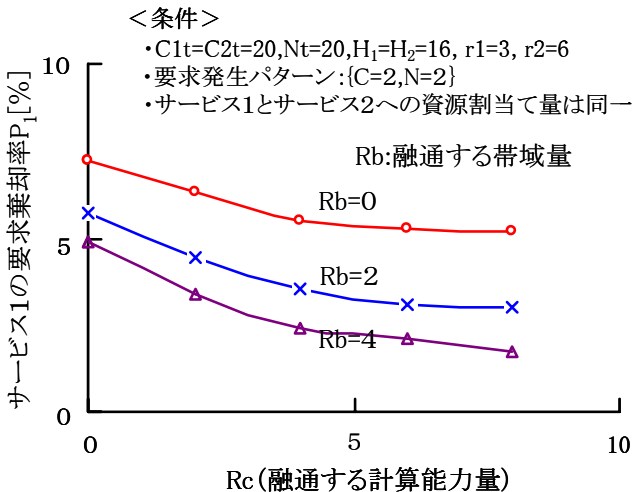
要求棄却率[%]=

「棄却した要求数/発生した要求の総数」 $\times 100$ (21)

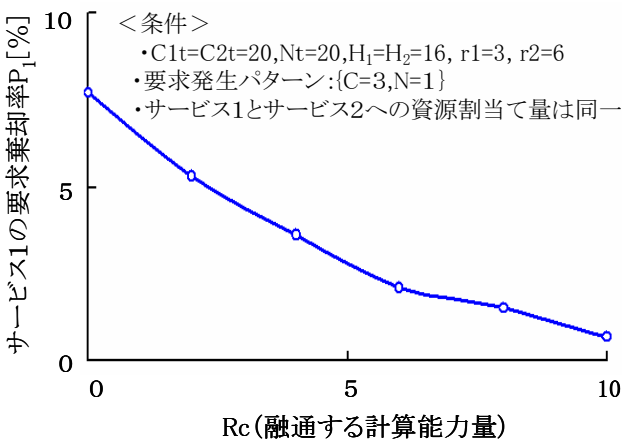
3. 5. 2 シミュレーション結果と評価

3. 3節で説明した融通形態1（サービス2の帯域と計算能力の一部をサービス1に融通）における資源融通量とサービス1の要求棄却率 P_1 の評価結果を図8(1)に示す。また、融通形態3（サービス2の計算能力をサービス1向けに融通）における資源融通量とサービス1の要求棄却率 P_1 の評価結果を図8(2)に示す。

この例では、融通する資源量が多くなるにつれ融通先サービスの要求棄却率が最大で半分以上に減少することがわかる。つまり、サービス間の資源融通により、資源融通しない従来方式に比べ処理するサービス数が増加し、ふくそう回避に効果が期待できる。



(1) 融通形態1の融通効果



(2) 融通形態3(計算能力の融通)の融通効果

図8 資源融通効果の評価結果例

4. 特定要求種別規制方式の概要と課題

4. 1 規制方式の概要

規制方式として、以下の2方式を検討する。

方式1：ふくそう資源種別を多く必要とする要求をだけを規制^{(2), (3)}。例えば、計算能力だけが混み合っている場合には、従来のように全ての要求を規制するのではなく、混み合っている計算能力（これが特定資源）を多く利用する要求だけを規制し、計算能力を多く利用しない要求を処理する(図9)。これにより、全体の要求処理数を増加させることができる。

方式2：ふくそう資源種別を多く必要とする要求の要求資源量そのものを小さくする。例えば、計算能力だけが混み合っている場合には、従来のように全ての要求を規制するのではなく、混み合っている計算能力（これが特定資源）を多く利用する要求の資源量を小さくし($1/n$)、資源保留時間を長くして(n 倍)、全体の要求処理数を増加させる(図10)。ただし、計算能力と帯域は同時割り当てを前提としているため、計算能力に合わせて帯域を小さくする($1/n$)。

なお、この方式は、要求資源量を小さくするとサービス自体が成り立たないサービスには適用できない（例えば、高品質を必要とする動画配信など）。

4. 2 規制方式の課題

4. 2. 1 方式1の課題

- 1) 特定要求種別の規制基準（資源量がどの大きさ以上でどの位規制するか）
- 2) 規制タイミング、規制解除タイミング
- 3) 規制実施場所（要求受付処理を行う集中管理サーバで実施か）と制御シーケンス

4. 2. 2 方式2の課題

- 1) 要求時にユーザが申告する項目（サービス仕様）
- 2) 要求資源量を小さくするタイミング
- 3) 要求資源量をどこまで小さくするか（ふくそう度合いによっても異なる）
- 4) 要求資源量をもとに戻すタイミング
- 5) 制御実施場所（帯域はスイッチ y 、計算能力はサーバにおいて要求ごとに実施）と制御シーケンス

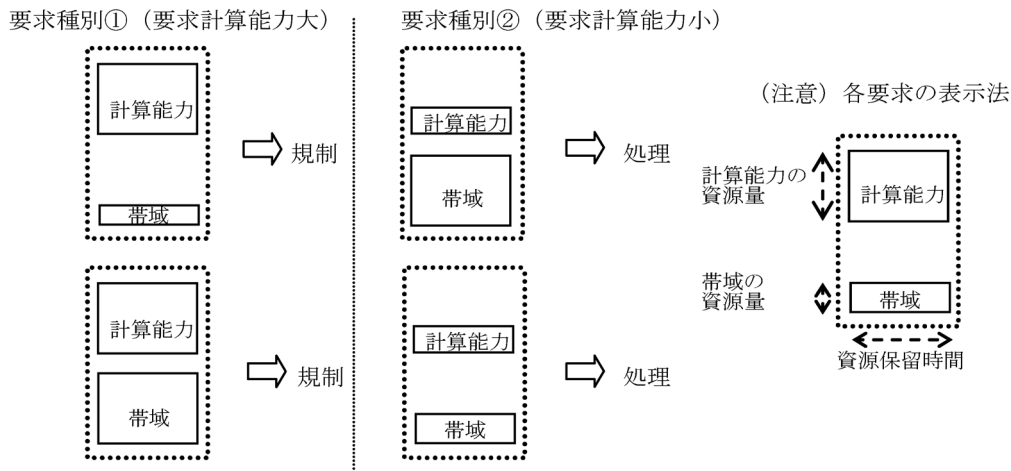


図9 特別要求種別規制の方式1のイメージ

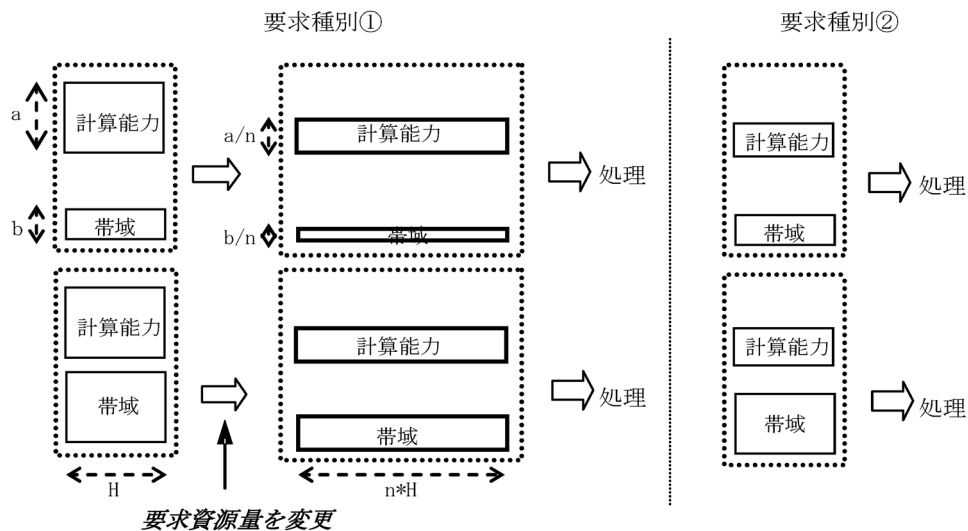


図10 特別要求種別規制の方式2のイメージ

5. むすび

本論文は、オールIPネットワークのふくそう制御方式として資源融通方式と特定要求種別規制方式を取り上げ、その実現法または実現上の課題を明らかにした。ふくそう発生回避対策と位置付けられる資源融通方式については、4つの資源融通形態ごとに資源融通の可否と融通量の決定手順を明らかにした。また、シミュレーション評価により資源融通を実施しない従来方式に比べサービス要求処理数を大幅に増加させ、ふくそうを回避する効果が期待できることを明らかにした。特定要求種別規制方式については、ふくそう資源種別を多く必要とする要求をだけを規制する方法とふくそう資源種別を多く必要とする要求の要求資源量そのものを小さくする方法の2つの考え方を提案し、それらを実現するために必要な

課題を整理した。

今後は、本論文で明らかにした課題の具体的な実現法、ならびに有効性を明らかにしていく予定である。

参考文献

- (1) N.Morita and H.Imanaka, "Introduction to the Functional Architecture of NGN," Proc. IEICE, vol.E90-B, no.5, pp.1022-1031, May 2007.
- (2) 畠山, 栗林, "オールIPネットワークにおけるふくそう制御方式の提案", 成蹊大学 理工学研究報告 Vol.44, No.2, pp.83-88 (2007.9).
- (3) K.Hatakeyama and S.Kuribayashi, "Proposed congestion control method for all-IP networks including NGN", ICACT2008, 06C-02 (2008.2)

- (4) S.Tsumura and S.Kuribayashi, “Simultaneous allocation of multiple resources for computer communications networks”, APCC2006, 2F-4 (2006.8)
- (5) S.Tsumura and S.Kuribayashi, “Delayed resource allocation method for a joint multiple resource management system”, APCC2007, TPM2-3 (2007.10)
- (6) E.Rosen, A.Viswanathan and R.Callon, “Multiprotocol label switching architecture”, IETF EFC3031.