

資源要求とトラフィック計測結果の両方にもとづくアドミッション制御法と その On/Off モデルのもとでのダイナミクス

金田 泰

日立製作所 中央研究所

〒185-8601 東京都国分寺市東恋ヶ窪 1-280

E-mail: Yasusi.Kanada.yq@hitachi.com

あらまし アプリケーションによる資源要求と NetFlow によってえられる DiffServ クラスごとのトラフィック計測結果の両方にもとづくアドミッション制御法を開発した。この方法を使用したポリシーサーバのプロトタイプを開発し、シミュレートされた音声トラフィックを使用して実験した。その結果、この方法を使用すればひどい計測負荷で帯域使用率をたかめることができ、呼損を減少させられることがわかった。On/off モデルを使用したシミュレーションの結果、興味ぶかいが害をなすようなダイナミクスが観察された。すなわち、このアドミッション制御法においてはパラメタの値によって発振や 100~150 分間にわたる変化、帯域のオーバーシュートなどの現象がおこることがわかった。これらの現象をおさえ、アドミッション制御を適切に動作させられるおおよそのパラメタ値の範囲をもとめることができた。

キーワード アドミッション制御, トラフィック計測, QoS保証, DiffServ, ネットワーク・ダイナミクス。

A Method of Admission Control Based on Both Resource Requests and Traffic Measurement and Its Dynamics Under An On/Off Model

Yasusi Kanada

Central Research Laboratory, Hitachi, Ltd.

Higashi-Koigakubo 1-280, Kokubunji, Tokyo 185-8601, Japan

E-mail: Yasusi.Kanada.yq@hitachi.com

Abstract A method of admission control based on both resource requests by applications and traffic measurement results per DiffServ class using NetFlow was developed. Several experiments using a newly-developed policy-server prototype using this method and simulated voice traffic showed that bandwidth usage can be improved and call blocking ratio was decreased with low measurement load by using this method. Interesting but possibly harmful dynamics were observed by simulations using simulated traffic that follows an on/off model; i.e., this admission control method may cause oscillation or long-term development that lasts for 100 to 150 minutes, and it may also cause bandwidth overshooting. The range of parameters, with which such phenomena can be properly suppressed and the admission control correctly works, was experimentally obtained.

Keywords Admission Control, Traffic Measurement, QoS Guarantee, DiffServ, Network Dynamics.

1. はじめに

ITU-T, 3GPP などの標準化組織において標準化されてきた NGN (Next Generation Networks) には資源制御とアドミッション制御のための機能 RACF (Resource Admission Control Function) がふくまれている。ここでアドミッション制御 [Geo 08] は、これから通信をおこなう 2 者間でその通信のために指定された帯域が確保できるかどうかを判断することをいうが、これは回線交換網や ATM において使用されてきた古典的な技術 [Lew 98] [Hab 00] だが、IP 網におけるアドミッション制御に関しては課題がおおい。そのなかには、アドミッション制御の基準となる使用帯域として事前要求した値を使用するかトラフィック計測値を使用するかという問題がある。

従来 ATM などで使用されてきたアドミッション制御法は基本的に要求 (事前に申告) した帯域に依存している。この方法の問題点は、おおくのアプリケーションは要求よりはるかにすくない帯域しか使用せず、ネットワークが有効に利用されなくなることである。

これに対して、トラフィック計測にもとづくアドミッション制御法 (Measurement-Based Admission Control, MBAC) が多数、提案されている (たとえば Nam ら [Nam 08])。この方法によって、いわゆる

オーバーブッキング (要求値の総和が最大帯域をこえる帯域わりあて) をおこなうことによって、むだにわりあてられる帯域をへらすことができる。しかしこの方法では、予測がはずれてネットワークの容量をこえるトラフィックがながれこみ、輻輳をおこす可能性がある。

オーバーブッキングをおこなうかぎりは予測誤差が生じることを完全にふせぐことはできない。しかし、要求値と計測値の両方を使用して予測をおこなうことによって、より予測の精度をたかめることができるのかんがえられる。これらをかみあわせた方法が Georgoulas ら [Geo 08] によって提案されている。Georgoulas らはさらにトラフィック量の平均値だけでなく分散すなわちジッターも考慮してアドミッション制御をおこなっている。しかし、この種の研究はかぎられている。この論文ではこれらをかみあわせる方法を提案し、そのふるまいを On/Off モデルを使用したシミュレーションによってしめす。

2. 要求と計測の両方にもとづくアドミッション制御法

この節においては、アプリケーションの要求値とトラフィック計測結果の両方にもとづくアドミッション制御法 (以下 RMBAC とよぶ) をしめし、それがふくむパラメタの数値例やパラメタを変化させてえられるさまざまなアドミッション制御法について議論する。

2.1 方法

音声、ビデオなどの通信をおこなう際には、NGN のわくぐみにしたがってセッション開始時に資源を予約し、セッション終了時にそれを解放する。通常は SDP/SIP によって使用する資源を宣言するが、アプリケーションがより詳細な QoS 条件を指定する必要があるれば RSVP のような資源予約用のプロトコルを使用して宣言する。

ポリシーサーバ (RACF) を使用して資源予約時にアドミッション制御をおこなう、すなわちフロー単位で要求した資源がわりあてられるかどうかを判定し、承諾または拒否する。この判定においては特定のネットワーク・パス (すなわちコア網の特定の入口ルータから特定の出口ルータまでのパスごと)、または特定のサブネットから特定のサブネットへのパスなどに関して、DiffServ のトラフィック・クラスごとにあらかじめ上限帯域を設定し、それをこえないようにアドミッション制御をおこなう。このようなパスを通過する特定のクラスのトラフィックをマクロフローとよぶ。

ここで提案する方法の特徴は、資源要求とトラフィック計測の両方にもとづいてこの判定をおこなうことである。すなわち、この方法においては要求された資源量を考慮するが、それを常時確保するかわりにオーバーブッキングをみとめる。トラフィックがあふれないようにするため、ポリシーサーバが NetFlow プロトコルによってネットワーク・ノードからクラスごとのトラフィック量を受信して使用する。

アドミッション制御をおこなうネットワーク上の場所に関してもいくつか選択枝があるが、ここではコア・ネットワーク入口のエッジノードで制御する。トラフィック計測はエッジノードの機能を使用してえるが、そのための方法としては NetFlow を使用する。

ポリシーサーバは一定時間ごとに各エッジノードから NetFlow によって DiffServ のクラスごとにトラフィック計測結果を収集する。ポリシーサーバの計算もこの時間間隔によって離散化する。時刻 $t + 1$ (つぎの期間) におけるマクロフロー f の使用帯域 $Bu(f, t)$ は、現在の時刻 t までの収集結果をもとに、つぎの式によって予測する。

$$Bu(f, t) = \gamma(1 + \beta) Ba(f, t) + (1 - \gamma) Br(f, t)$$

ここで $Ba(f, t)$ はトラフィック計測によってえられた帯域、 $Br(f, t)$ は要求帯域の和である。

マクロフロー f にわりあてる上限帯域 (または最大帯域) を $Bm(f)$ と定義する。 $Bm(f)$ と $Bu(f, t)$ との差が新規のフローにわりあて可能な帯域であり、それがあらたな要求帯域以下なら、わりあて可能である。NetFlow からはクラス単位で集約された情報をえるが、この値をつぎの式によって指数移動平均をもとめて平滑化する。

$$Ba(f, t) = \alpha ba(f, t) + (1 - \alpha) Ba(f, t-1), Ba(f, 0) = Bm(f)$$

ここで $ba(f, t)$ が NetFlow からえられた値である。このアドミッション制御法においては α, β, γ という 3 個のパラメタを調整して使用する必要がある。これらのパラメタの意味はつぎのとおりである。

- 計測値平滑化係数 α : トラフィック計測値の移動平均のなめらかさを制御するパラメタ ($0 \leq \alpha \leq 1$).

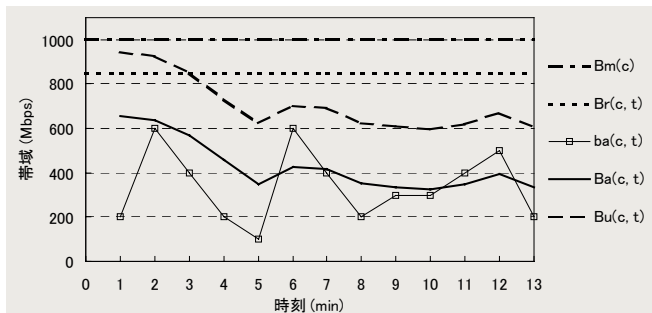


図 2.1 トラフィック量の測定値と推定値 ($\alpha = 0.3, \beta = 0.5, \gamma = 0.7$)

- 帯域余裕率 β : トラフィック計測値 (移動平均) に対するアドミッション制御の限界値のマーヅンを制御するパラメタ ($0 \leq \beta \leq 1$).
- 計測値寄与率 γ : アドミッション制御の限界値に関する測定値の寄与率 ($1 - \gamma$ が宣言値の寄与率) ($0 \leq \gamma \leq 1$).

2.2 数値例

前節の方法の数値例を図 2.1 にしめす。この時間範囲ではあらたなわりあて要求はないとしている。ここでは $\alpha = 0.3, \beta = 0.5, \gamma = 0.7$ とした。この図で移動平均をとれば平均帯域 ($ba(f, t)$) が平滑化されていることがよみとれる。新規トラフィックにわりあて可能な帯域は予測帯域 $Bu(f, t)$, 上限帯域 $Bm(f)$ 間の領域である。

2.3 さまざまなアドミッション制御

上記の RMBAC 法を使用すれば、パラメタ α, β, γ の値をかえることによって、つぎのようなさまざまなアドミッション制御をおこなうことができる。いずれの方法がよいか、あるいは他のパラメタ値をとるのがよいかは、トラフィックの性質に依存するとかんがえられる。

- 要求帯域の総和 $Br(f, t)$ だけによる CBR 型制御

$\alpha = 0$ とすれば、計測結果を計算にいれず、わりあて可能な最大帯域 $Bm(f)$ と要求帯域の総和 $Br(f, t)$ との差だけが新規のトラフィックにわりあてられるようにできる。図 2.1 でいえば、表示された時刻のあいだでつねに $Bm(f)$ (1000 Mbps) と $Bu(f, t)$ (850 Mbps) との差の帯域がわりあて可能である。この方法は新規の c のトラフィックを CBR (Constant Bit Rate) としてあつかう。

- 直前の計測値 $ba(f, t)$ だけによる無記憶的制御

$\alpha = 1$ とすれば、時刻 t の計測結果が最新であるときに、ふるい計測値 $ba(f, tt)$ ($tt < t$) を計算にいれず (移動平均をとらず)、直前の計測値 $ba(f, t)$ だけであらたなトラフィックにわりあて可能な帯域をもとめる。図 2.1 におけるパラメタのうち α だけを 1 にしたときの例を図 2.2 にしめす。

- 測定値から計算したトラフィック量 $Ba(f, t)$ だけによる制御

$\gamma = 1$ とすれば、時刻 t の計測結果が最新のときに、要求帯域の総和 $Br(f, t)$ を計算にいれず計測結果だけから計算したトラフィック量にもとづいてアドミッション制御をおこなう。図 2.1 におけるパラメタのうち γ だけを 1 にしたときの例を図 2.3 にしめす。

- 要求帯域と最近の計測値すべてを加味した制御

このモデルにおける一般的ななばあいの制御である。 $0 < \alpha < 1, 0 < \gamma < 1$ のときは、要求帯域の総和 $Br(f, t)$ と計測値 $ba(f, tt)$ ($tt \leq t$, 時刻 t の計測結果が最新) のすべてがアドミッション制御のために使用される。ただし、指数移動平均をとっているために、ふるい計測値の影響は時間とともに急速に減少する。

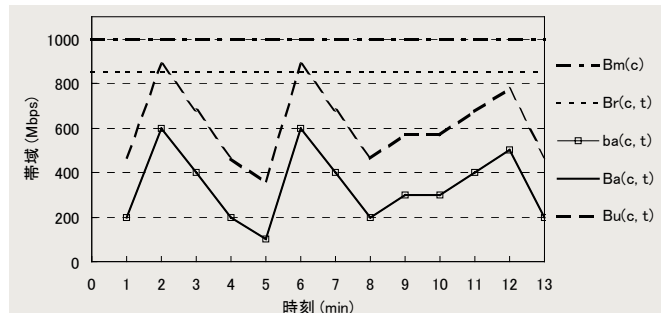


図 2.2 $\alpha = 1$ のときのトラフィック量推定値 ($\beta = 0.5, \gamma = 0.7$)

3. プロトタイプの概要

RMBAC をポリシーサーバのプロトタイプ [Kan 07] にくみこん

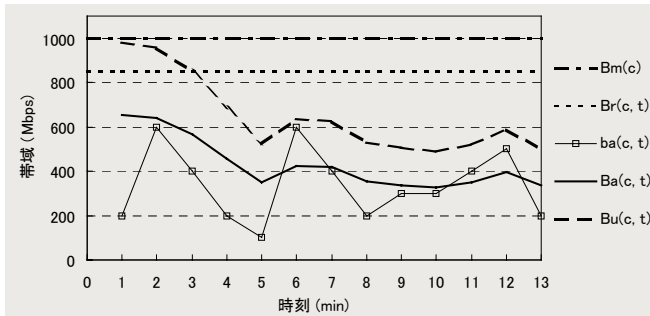


図 2.3 $\gamma = 1$ のときのトラフィック量推定値 ($\alpha = 0.3, \beta = 0.5$)

だ。その概要を3項目にわけて記述する。

3.1 トラフィック計測機構

Cisco や Alaxala のルータにおいては、NetFlow によって計測結果を送信する間隔を1分よりこまかい単位では指定できない。しかし、QoS 保証のための計測間隔はもっとせまいほうがよいし、Diff-Serv を使用するとき、それはルータにとっておおきなオーバーヘッドにはならないとかがえられる。なぜなら、DiffServ においてはトラフィックは比較的少数のクラスが単位となり、基本的にはクラス単位で計測すればよいからである。だが、当面は既存のルータがもつ NetFlow の機能をそのまま使用するため、1分間隔で情報を取得する。すなわち、1分ごとにマクロ・フローごとの集約情報をえる。

NetFlow による出力形式としてさまざまな形式が選択できるが、プロトタイプにおいては、現在のところ、コア・ネットワーク入口エッジルータにおいてクラスごとの計測値をえるのに適した 32 バイトの Destination-Prefix-ToS Aggregation レコード (表 3.1 参照) だけを受信する。設定コマンドにおいてこの形式を指定し、最短宛先マスクビット数 (mask-destination-minimum) として 32 bit を指定すれば、ToS バイトの内容 (DSCP) と出力インタフェース、宛先 AS (Autonomous System) がひとしいフローはすべてひとつに集約される。

この方法を使用すれば、たとえば DiffServ クラスが 5 種類、宛先が 1000 種類ならば各ルータから 1 分ごとに出力されるレコード数は 5000 である。1500 バイトの UDP パケットに 25 レコードずつ収容すれば、1 分に 200 パケットつまり 40 kbps のストリームが生成される。このストリームはルータにとっても、それを受信するポリシーサーバにとっても、おおきなオーバーヘッドなしに処理可能である。

3.2 アドミッション制御機構

RMBAC の手順の概要をしめす (図 3.1 参照)。

- **トラフィック計測と使用帯域予測:** ポリシーサーバは一定時間ごとに各エッジノードから NetFlow によってトラフィック計測結果を収集し、現在のわりあて可能帯域 $Ba(f, t)$ を 2 節の式によって予測する。NetFlow からはクラス単位すなわち ToS (DSCP) によって集約された情報をえて、その値を平滑化する。
- **資源要求 (セッション情報) の処理:** ポリシーサーバは SIP による

表 3.1 NetFlow の Destination-Prefix-ToS Aggregation レコード

収集項目	説明
Flows	集約したフロー数
Packets	集約したフローに含まれる総パケット数
Bytes	集約したフローに含まれるパケットの総バイト数
First time stamp	フロー開始パケット受信時の時刻 [秒]
Last time stamp	フロー最終パケット受信時の時刻 [秒]
Destination prefix	宛先 IPv4 アドレスのプレフィクス
Destination mask bits	宛先 IPv4 アドレスのプレフィクス・マスクビット数
ToS	Type of Service
Destination AS	宛先または宛先側隣接ピアの AS 番号
Output interface	出力インタフェースの SNMP interface index

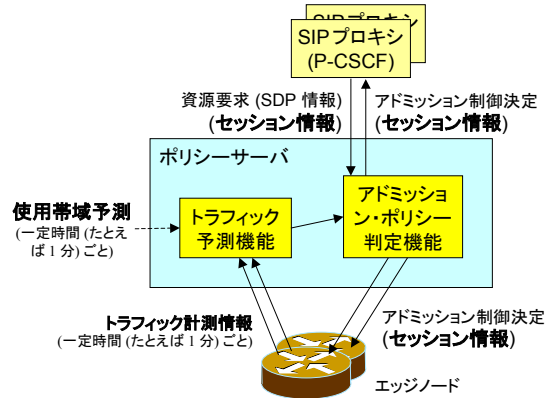


図 3.1 アドミッション制御機能と周辺の機能との関係

資源要求が SIP プロキシ経由で通知されると、そのたびにその要求内容と直前のトラフィック予測の結果とを使用してアドミッション制御をおこなう。すなわち、余裕帯域が要求された帯域をうまわわってればその要求を許可し、そうでなければ拒否する。SIP プロキシに応答ががえされる。また、必要であればエッジノードに制御コマンドが送信される。

図 3.2 には、図 3.1 でしめた各機能のネットワーク内での配置をしめす。図 3.2 にしめすようにこのプロトタイプにおいては各エッジノードに SIP プロキシが 1 対 1 に対応することを前提としている。したがって、各 SIP メッセージは入口エッジノードに対応する SIP プロキシと出口エッジノードに対応する SIP プロキシをそれぞれ 1 回、かならず通過する。ポリシーサーバはこれらの SIP プロキシからそれぞれ RRTP (3.3 節参照) によるメッセージを受信する。そのため要求されたセッションのコア網への入口と出口とを把握できる。

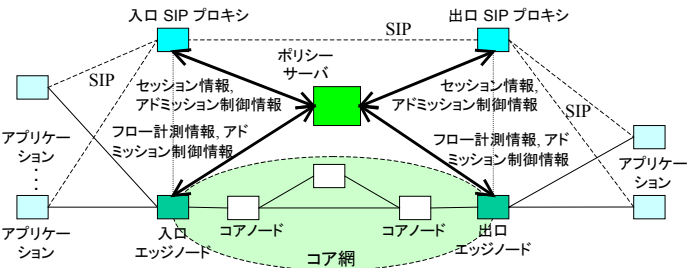


図 3.2 各機能のネットワーク内での位置

3.3 PSシーケンサ

図 3.2 におけるポリシーサーバ以外の部分の機能をシミュレートするためのプログラム「PS (ポリシーサーバ) シーケンサ」を開発した。PS シーケンサは、くみこまれたトラフィック・モデルにしたがってつぎの情報を生成してポリシーサーバに送信する機能をもつ (図 3.3 参照)。

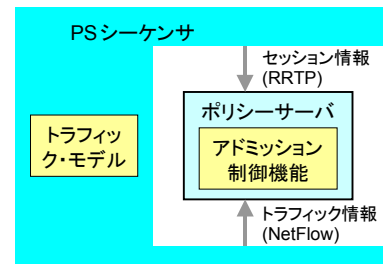


図 3.3 PSシーケンサ

- **セッション情報:** 本来はアプリケーションが送信した SIP メッセージを SIP プロキシが参照して、セッション情報をポリシーサーバに通知する。PS シーケンサは、自律的に仮想的なセッション情報を生成し、RRTP (Resource Request Transport Protocol) という独自プロトコルによってサーバと通信する¹。

¹ 標準プロトコルを使用するとすれば、3GPP 等において Diameter にもとづ

- **NetFlow 情報:** ポリシーサーバはトラフィック計測結果を NetFlow プロトコルによって受信する. このプロトタイプでは NetFlow の Destination-Prefix-ToS Aggregation レコードを使用する.

なお, ノードが出力する NetFlow 情報は設定によってさまざまな形式, さまざまなタイミングで受信可能である. しかし, PS シーケンサはそれを 1 分 (これは仮想的な時間であり, シミュレーションにおける実時間はもっと短縮できる) に 1 回だけ送信する.

4. アドミッション制御法の評価

PS シーケンサを使用してポリシーサーバ・プロトタイプにおけるトラフィック計測法とアドミッション制御法の評価をおこなった. その方法と評価結果についてのべる.

4.1 評価項目と評価法

アドミッション制御法に関する評価項目はつぎの 2 つとする.

- **呼損率:** セッション開始要求のうち, どれだけのわりあいのものが拒否されたかをあらわす. 呼損率はひくいほうがよい.
- **帯域使用率:** わりあて帯域のうちどれだけを有効に使用することができたかをあらわす. 帯域使用率は NetFlow によって受信した値からもとめられたトラフィック量にもとづいて計算する. 通常は最大値が 1 だが, わりあて帯域をこえて使用すると 1 をこえる.

3 節でしめたアドミッション制御の評価パラメータを変化させながら, また以下でのべるトラフィック条件を変化させながら, これらの評価項目の値を計測し評価する.

トラフィック条件は以下のとおりである (図 4.1 参照). PS シーケンサによってセッション開始要求の到着間隔の平均値が固定されたポアソン到着する (つまり独立な) 80 kbps (ヘッダをのぞくと 64 kbps) の音声フローをシミュレートした. すなわち, SIP プロキシ/ポリシーサーバへの要求到着間隔を指数分布の乱数によってもとめた¹. 到着間隔の平均値は 0.5 ~ 1 秒程度とした. 各フローの持続時間 (保留時間) も平均 120 秒の指数分布とした. これらの要求に対応する各音声フローは, 要求した 80 kbps のうち実際には平均で 60% (48 kbps) を使用する. つまり, 無音区間が 40% 程度存在し, 静寂検出 (silence detection) にもとづいて動作することを仮定する.

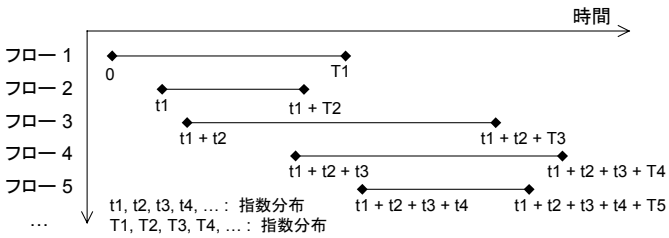


図 4.1 PS シーケンサが仮想的に生成するフロー

このような無音区間をふくむトラフィックを生成するため, つぎのような on/off モデルにもとづくトラフィック生成法によってフローごとに計算をおこなった (各フローはセッション開始要求が許可されるとただちに開始されると仮定した). ひとの音声はトークスパート (talkspurt) とよばれる音声が続く区間と無音区間 (gap) とで構成される. これを on と off という 2 状態が交代するマルコフ過程 (マルコフ連鎖) またはかくれマルコフ過程によってモデル化するのが on/off モデルである. これを 6 状態のかくれマルコフ過程によって近似する Brady [Bra 65] のモデルが有名である. しかし, もっとも

¹ プロトコルが標準化されている.

¹ それぞれの呼が独立に発生すれば到着はポアソン分布し, 到着間隔は指数分布になる. しかし, すくなくともインターネットのトラフィックどうしは統計的に独立とはいえず, 指数分布からはずれることが知られている.

単純な on/off モデルは図 4.2 のような 2 状態のマルコフ過程であり, ここではそれを使用した. 図 4.2 には評価で使用した遷移確率もあわせて記述した. このモデルではトークスパートの平均持続回数が 50 (持続時間 1 s), 無音区間のそれが 33.3 (持続時間 667 ms) となり, いずれも指数分布にしたがう². 実際には時間スケールを 1/3 にして (到着間隔を 1/3 にして) シミュレートしている.

1 種類のトラフィックだけを使用したのは, このアドミッション制御法においてはクラスごとに独立に帯域が管理されるため, 他のトラフィックを混合しても音声トラフィックには影響しにくいからである. とくに, 音声トラフィックは通常, ノードにおいて優先制御されるので, さらに他のトラフィックの影響をうけにくい.

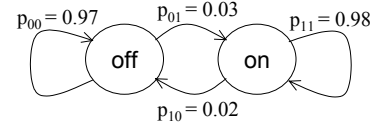


図 4.2 2 状態のマルコフ過程によって表現される単純な on/off モデル

4.2 評価結果

まず提案しているアドミッション制御法を従来の方法と比較し, その後, 提案方法の性質を分析する.

4.2.1 計測値を考慮しない方法との比較

この節においては, RMBAC を電話の制御などにおいて使用されてきた実使用帯域を考慮しないアドミッション制御法と比較する. 以下の計測は, 各パラメータ・セットについて 2 回ずつ, 乱数の種をかえて実施した. 図 4.3 はフローの到着間隔が 0.79 秒であり, $\alpha = 1.0$, $\beta = 0.2$, $\gamma = 0.8$ のときの呼損率と帯域使用率の例である. NetFlow 情報は PS シーケンサから 1 分おきにえられるが, そのタイミングで呼損率と帯域使用率を計算した.

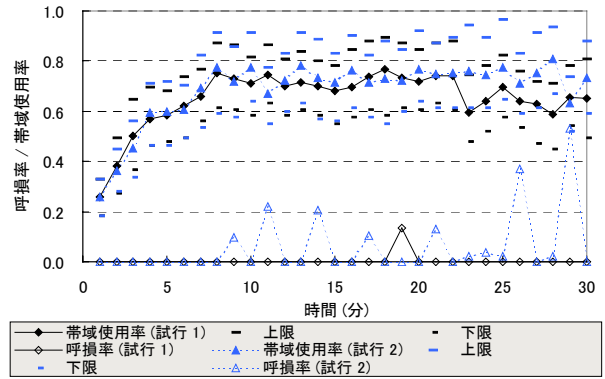


図 4.3 $\alpha = 1.0$, $\beta = 0.2$, $\gamma = 0.8$ のときの呼損率と帯域使用率

図 4.4 は基準のパラメータのうち γ だけを 0 に変化させたときの呼損率と帯域使用率をしめす. γ の値が 0 だということは, 計測値を計算にいれていないことを意味する. すなわち, 従来方式と同様にセッション開始時の宣言値だけで使用帯域をみつもっている. そのため, 帯域使用率は 0.6 程度に低下している. これは音声フローが平均で 60% の帯域を使用していることに対応している. その結果として呼損率は前記と同様の平均値において 0.18 となる.

この測定条件はほぼ 125 (= 10 M / 80 k) の電話回線があるときの呼損の発生に対応している. 電話の呼損率は Erlang B 式 [Aki 00] によってモデル化されている. すなわち, 呼損率を B , 呼量量を a , 回線数を n とすると, つぎのような関係がなりたつ.

$$B = (a^n / n!) / (1 + a / 1! + a^2 / 2! + \dots + a^n / n!)$$

² 実際のトークスパートや無音区間の持続時間の分布は指数分布からはずれることが知られている.

上記の測定条件をこの式にあてはめると、呼損率は 0.2 となり、ほぼ上記の測定結果と一致する。

このモデルにおいては各音声フローは長時間の平均では 60% の帯域を使用するはずだが、時間依存性があるために使用帯域は変動する。しかし、 $\gamma = 0.8$ のときに呼損の発生をおさえつつ帯域を有効利用することができ、 $\gamma = 0$ のときと明白な差がある (図 4.3 における平均呼損率は 0.04, 図 4.4 では 0.18 である)。

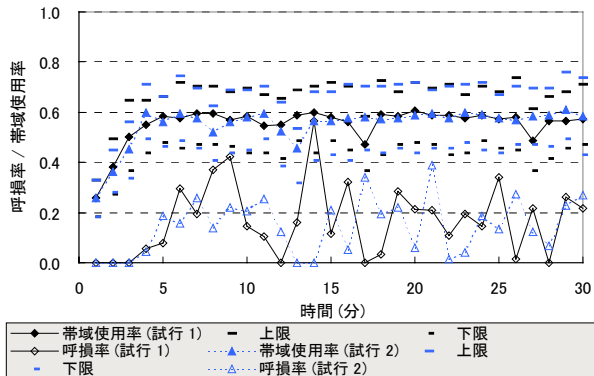


図 4.4 $\alpha = 1.0, \beta = 0.2, \gamma = 0$ のときの呼損率と帯域使用率

これらの図には帯域使用率の上限値と下限値を表示しているが、これらの意味はつぎのとおりである。帯域使用率は 1 分の計測間隔のあいだにも変動するが、このモデルでは 20 ms (音声パケットの想定送信間隔) ごとに on/off モデルから出力されるパケット数をカウントして、20 ms ごとの帯域使用率を計算している。その値の上限、下限を平均値とともに表示している。上限値は 1 をこえることもあるが、このシミュレーションにおいてはこえていない。

α, β, γ の値についてはつぎのようなことがいえる。

- α の値は 1.0 としている。定常的なトラフィックにおいては、これまでの実験によれば、この値を 0 ~ 1 の範囲で変化させても、この測定条件においては最初の 10 分程度をのぞけば大差ない。
- β の値は 0.2 としている。これは、計測値に対して 20% の余裕をみているということである。帯域使用率が 0.8 ちかくにとどまっているのはそのためである。
- γ の値は 0.8 である。これは、セッション開始時に宣言された帯域幅より測定値のほうをかなり重視していることを意味している。測定開始時 (時刻 0) にはトラフィックは存在しない。そのため、ほぼ定常な状態になるまでに 10 分ほどを要している。

もしあらかじめ各フローが常時 60% の帯域を使用することがわかれば、計測結果を利用しなくても帯域使用率をたかめられる。しかし、通常は各フローが使用する帯域幅はわからないし使用帯域は変動するので、このような方法はとれない。計測をおこなうことによって初めて、図 4.3 のような帯域の有効利用が可能になる。

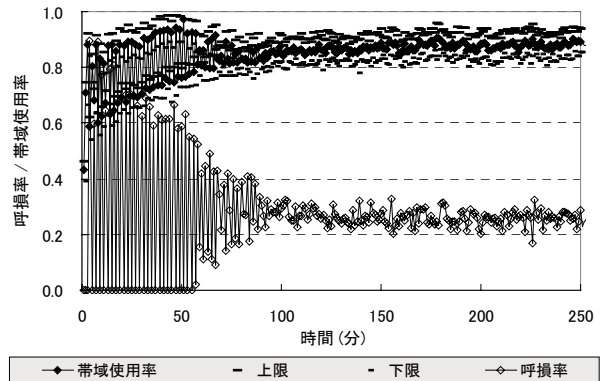
4.2.2 帯域使用率の振動

この項以降はパラメタの値をかえて測定をくりかえすことによって、提案しているアドミッション制御法の性質を分析する。この項においては帯域使用率が振動 (共振) する現象についてのべる。

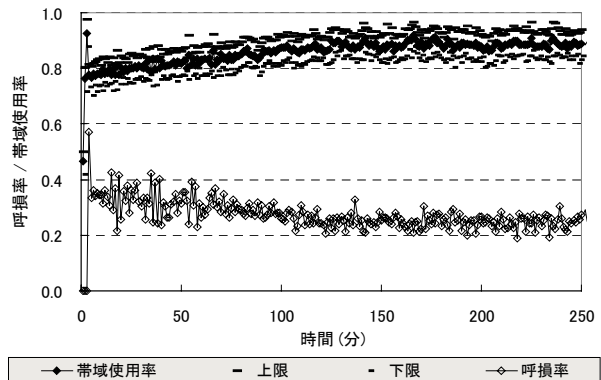
図 4.5 には、帯域幅の上限を 100 Mbps とし α の値を変化させたときの振動現象の変化をしめす。すなわち、図 4.5 (a) には $\alpha = 1.0, \beta = 0.2, \gamma = 0.8$ としたときの呼損率と帯域使用率の変化をしめし、図 4.5 (b) には α の値だけを 0.5 に変化させたときのそれらの変化をしめす。測定は 1 回だけおこなった。 $\alpha = 1.0$ のときは振動が収束するまでに 100 分ほど要したが、 $\alpha = 0.5$ のときは 10 分程度で収束した。帯域幅の上限をちいさくすると、 $\alpha = 1.0$ のときの収束時間はさらにのびる。これは、振動の発生によって定常状態の変化への

の対応がおくれることを意味しているとかがえられる。

図 4.5 においては、これ以前の図における 30 分間の変化からはよみとれない長期にわたる変化があるのがわかる。つまり、 $\alpha = 1.0$ のときも $\alpha = 0.5$ のときも 100 ~ 150 分間にわたって呼損率は漸減し、帯域使用率は漸増している。これは最初の 100 ~ 150 分間のあいだは帯域使用率がややひくくおさえられていることを意味する。できればこのような長期の変動はなくしたほうがよいであろう。



(a) $\alpha = 1.0$ のとき



(b) $\alpha = 0.5$ のとき

図 4.5 呼損率と帯域使用率の振動収束の比較

計測結果のフィードバック (帰還) にもとづくアドミッション制御によってこのような振動的なふるまいがおこることは Xu ら [Xu 04] も報告している。正帰還が防止できれば発振は防止できるとかがえられるが、このような非線形系の解析は容易でない。Xu らは Web サーバにおけるアドミッション制御の振動的なふるまいをおさえるためにアドミッション制御を確率的におこなう方法を提案しているが、この方法では振動を完全におさえることはできない。

上記のような振動現象は Erlang B 式における呼損率が 30% 以上のときに顕著だが、呼損率が 20% 程度のときは観察しにくい。しかし、実験においてはトラフィックに時間依存性がなければ振動の発生がみとめられた。すなわち、頻度はひくいが呼損が発生し、その直後に帯域使用率が大幅に低下するのが観察された。

4.2.3 帯域使用率のオーバーシュートの発生

この項では帯域使用率のオーバーシュートの発生についてのべる。図 4.6 はフローの到着間隔を 0.6 秒としたときの呼損率と帯域使用率であり、パラメタは $\alpha = 0.3, \beta = 0.2, \gamma = 0.8$ としている。試行 B においては開始後 5 分のときに帯域使用率がオーバーシュートしている。このときにわりあて帯域をこえる可能性があるため、このような現象をおさえる方法をくふうする必要がある。なお、 β の値だけを変化させても同様の現象がみられることを確認している。

パラメタをかえて実験すると、 α の値がちいさいときはすどいオーバーシュートがおこりやすかった。したがって、オーバー

シュートと振動とのあいだにトレードオフがあるとかがえられる¹。

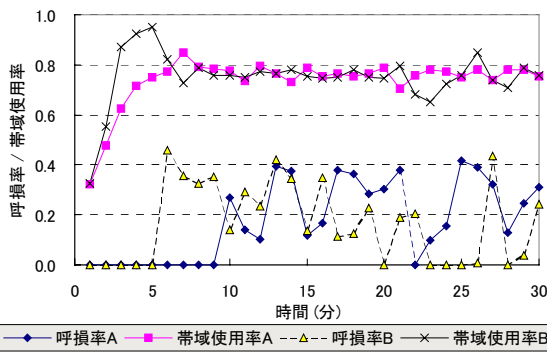


図 4.6 $\alpha = 0.3, \beta = 0.2, \gamma = 0.8$ のときの呼損率と帯域使用率

4.2.4 帯域余裕率の最適値

帯域余裕率 β の値を 0 にちかづけると、帯域使用率が向上する一方でパケット損失や遅延が発生しやすくなる。そこで、 α の値を 0.5、 γ の値を 0.8 に固定して β の値を変化させ、シミュレーションによって帯域使用率と損失率 (より正確には帯域超過率、つまりわりあて帯域をこえたバイト数を通過可能なバイト数でわったもの) とをもとめた。この測定のためには呼損が定常的に発生する状態にする必要があるため、Erlang B 式における呼損が 50% となるような要求到着頻度を選択した。各パラメータ・セットについて 150 分間の測定をおこない、最初の 20 分間をのぞいて平均値をもとめた。

結果を図 4.7 にしめす。ここではわりあて帯域も 10 Mbps (実線) から 100 Mbps (1 点鎖線) まで変化させて測定している。これらのわりあて帯域は電話回線換算でそれぞれ 125 本、375 本、1250 本に相当する。帯域使用率はほぼ常に $1 - \beta$ の値よりひくく、 $\beta = 0$ のときでも 0.92 以下である (1 をこえるときもふくめて平均値を計算)。

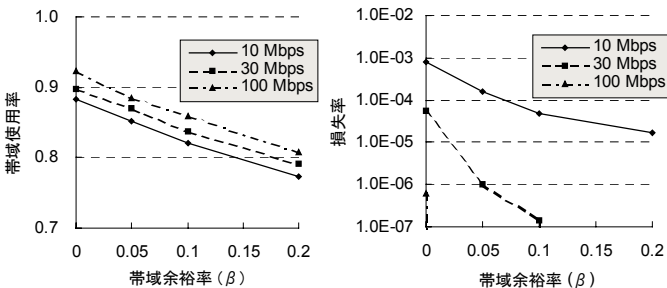


図 4.7 $\alpha = 0.15, \gamma = 0.8$, わりあて帯域 10 ~ 100 Mbps のときの帯域使用率とパケット損失率

損失率は 20 ms ごとにもとめたトラフィック量がわりあて帯域をこえたとき、超過分がすべて廃棄されるとして計算している。しかし、実際には優先制御されたトラフィックにおいてはただちに廃棄されることはない。また、帯域制御されたトラフィックにおいても遅延するだけで廃棄されないこともある。わりあて帯域が 10 Mbps のときは、要求される損失率が 10^{-3} ならばつねにみたされるが、 10^{-4} 以下の損失率が要求されるときは $\beta \geq 0.1$ でなければならない。なお、損失率が 10^{-5} 以下のときは誤差がおおきくて測定値は信頼できない(とくに、グラフに表示していない部分は測定値が 0 である)。

5. まとめ

RMBAC によって、トラフィック計測結果を利用して帯域使用率

をたかめ、その結果、呼損を減少させられることがわかった。使用帯域を考慮しない従来のアドミッション制御法において呼損率が 0.18 となるばいりに呼損率を 0.04 にまで減少させられた。

実用化するためには α, β, γ の値を最適化する必要があるが、無音区間が 40% あるシミュレートされた音声トラフィックを使用した実験において、これらについてつぎのようなことがわかった。

- α の最適値はさらに要検討: 平滑化定数 α の値を調整することによって、場合によっては振動現象をおさえたりオーバーシュートをおさえたりできることがわかった。 α の最適値は 0.5 から 1 のあいだにあるとかがえられるが、さらに検討する必要がある。
- β の最適値は 0.1 ~ 0.2 程度: 帯域余裕率 β の値は 0 にちかひほど帯域を有効に使用できる。しかし、帯域使用率が急激に増加してパケット損失が生じるのをさけるためには、対象とするフロー数が 100 程度のときは通常は 0.1 ~ 0.2 程度にするのがよいとかがえられるが、要求されるパケット損失率や遅延、フロー集約の程度、バースト性などに依存する。詳細はここでは省略する。トラフィック・モデルが複雑なときは β の最適値をもとめるのは容易でないとかがえられる。
- γ の最適値は 0.6 ~ 0.8 程度: 計測値寄与率 γ の値が 0 のときは測定結果を使用せず、それを使用するときより性能は低下する。 γ が 1 のときは発振がおこりうる。発振すると間歇的に呼損率が 1 にちかひ値をとるので、防止する必要がある。実験の結果から γ の値は、振動現象がおさえられる 0.6 ~ 0.8 程度にするのがよいとかがえられるが、最適値をもとめるにはさらに実験や理論的な解析が必要である。

ただし、 α だけでなく β や γ の値もトラフィックの性質によって変化する可能性があるため、非定常なトラフィック・モデルや動画など、音声以外のトラフィックを使用した実験をおこなう必要がある。

また、パラメータをかえながら実験していくうちに、前記の振動以外にも興味ぶかいダイナミクスすなわち帯域使用率のオーバーシュートが観察された。この現象はさらに調査する必要がある。

今後 RMBAC がもつ振動やオーバーシュートの問題を解決する必要があるが、システムを複雑化するとかえって予期しない現象が発生する可能性もあるため、単純さを維持するべきであろう。

参考文献

[Aki 00] 秋川 春夫, 川島 幸之助, “情報通信トラフィック — 基礎と応用 [改訂版]”, 電気通信協会, 2000-10.

[Bra 65] Brady, P. T., “A Technique for Investigating On-off Patterns of Speech”, *Bell Syst. Tech. J.*, Vol. 44, pp. 1–22, January 1965.

[Geo 08] Georgoulas, S., Trimintzios, P., Pavlou, G., and Ho, K., “An Integrated Bandwidth Allocation and Admission Control Framework for the Support of Heterogeneous Real-time Traffic in Class-based IP Networks”, *Computer Communications*, Vol. 31, pp. 129–152, 2008.

[Hab 00] Habibi, D. and Mao, G., “A Hybrid ATM Connection Admission Control Scheme Based on On-Line Measurements and User Traffic Descriptors”, *IEEE International Conference on Networks (ICON 2000)*, pp. 248–254, 2000.

[Kan 07] 金田 泰, “パスにそったシグナリングにもとづく端点間 QoS 保証法の開発と評価”, 電子情報通信学会 コミュニケーションオリティ研究会, 2007 年 7 月.

[Lew 98] Lewis, J. T., Russell, R., Toomey, F., McGurk, B., Crosby, S., and Leslie, I. M., “Practical Connection Admission Control for ATM Networks Based on On-line Measurements”, *Computer Communications*, Vol. 21, pp. 1585–1596, 1998.

[Nam 08] Seung Yeob Nam, Sunggon Kim, and Dan Keun Sung, “Measurement-Based Admission Control at Edge Routers”, *IEEE/ACM Transactions on Networking*, Vol. 16, No. 2, pp. 410–423, April 2008.

[Xu 04] Xu, Zhengdao, and Bochmann, G. v., “A Probabilistic Approach for Admission Control to Web Servers”, *2004 Int'l Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS '04)*, pp. 787–794, 2004.

¹ 線形システムであれば、共振をおさえるために抵抗をくわえるとオーバーシュート(過渡現象)もおさえられるが、アドミッション制御のシステムは非線形システムなので、ことなるふるまいをするのであろう。しかし、振動やオーバーシュートが発生する点においては線形システムと似ている。