

修士論文

ダイナミックキューを利用したステートレスIMS 発信規制手法

2012年8月16日

指導教員 浅見 徹 教授

東京大学大学院情報理工学系研究科
電子情報学専攻 106451

朱 瑋

■ 内容梗概

地震などの災害時には膨大なトラフィックが IP Multimedia Subsystem (IMS) ネットワークに入り輻輳が生じやすく、これを避けるため、通信会社は発信を規制することで重要通信の確保を行っている。しかし、IMS には音声や Short Message Service(SMS)、パケット通信など様々なトラフィックが含まれるため、複数のサービスそれぞれのスループットを最大限に保つことは難しい。本論文では、設計値を大幅に超えるトラフィックが IMS ネットワークに流入した場合にも設計上の最大交換性能を維持し、同時に、セッションサービス (session services) より通信オーバーヘッドの少ない非セッションサービス (non-session services) を自動的に優先させ、可能な限り多くの人々に通信サービスを提供できるシステムの実現を目標にしている。提案する自動発信規制手法では、スケラビリティを持たせるため、セッションの状態に依存しないステートレスなキュー制御を導入し、それぞれのセッションでの SIP メッセージの種類 (タイプ)、時系列上に現れる順序、および再送回数だけを考慮した優先制御となっている。また、最適キュー長も求めた。シミュレーションの結果は、2011 年東北大震災のような平時の 60 倍以上のトラフィック増という極端な輻輳下でも各マルチメディアサービスの最大スループットが保たれることを示している。

目次

第 1 章 序論	1
1.1 本研究の背景	2
1.1.1 IP マルチメディアサブシステム	2
1.1.2 災害時の過負荷障害について	2
1.1.3 IMS の過負荷回避手段	3
1.1.4 本研究の目的と目標	4
1.2 本論文の構成	5
第 2 章 IP Multimedia Subsystem(IMS) の概要	6
2.1 はじめに	7
2.2 IP マルチメディアサブシステム	7
2.2.1 コンセプト	7
2.2.2 IMS 開発の歴史	7
2.2.3 IMS の利点	7
2.3 IMS のアーキテクチャ	8
2.4 Session Initiation Protocol	10
2.5 IMS の提供するサービス	10
2.6 おわりに	11
第 3 章 IMS における過負荷回避と関連研究	12
3.1 はじめに	13
3.2 IMS におけるトラヒックの分類	13
3.3 IMS サービスにおける SIP メソッド	14
3.4 サービスにおける SIP シーケンス	15
3.4.1 音声通話における SIP シーケンス	15
3.4.2 ページャインスタントメッセージングにおける SIP シーケンス	17
3.5 IMS の過負荷回避に関する関連研究	17
3.5.1 メディアプレーンにおける過負荷回避	17
3.5.2 シグナリングプレーンの過負荷回避	18
3.6 おわりに	19
第 4 章 ステートレス発信規制のためのダイナミックキュー構造の設計	20
4.1 はじめに	21
4.2 2つのキューを使った SIP シグナリングの優先制御手法	21
4.3 2クラスのプライオリティキューイング構造の欠点	21
4.4 ステートレスなシグナリング優先制御を実現するためのダイナミックキューイング機構の提案	22

4.4.1	キューの構成	22
4.4.2	メッセージタイプによる優先制御	23
4.4.3	再送メッセージに対する優先制御	26
4.4.4	ダイナミックキューイングの手続き	27
4.5	おわりに	28
第 5 章	ダイナミックキューの NS2 シミュレータへの実装	29
5.1	はじめに	30
5.2	IMS コアネットワーク上でのダイナミックキューイング実装箇所	30
5.3	必要なプログラム修正箇所	31
5.4	NS2 シミュレータ上での実装方法	31
5.4.1	UDP ベースの SIP 実装	31
5.4.2	サービスモジュールの追加	35
5.4.3	ダイナミックキューイングモジュール	36
5.5	おわりに	38
第 6 章	評価	39
6.1	はじめに	40
6.2	ネットワークモデル	40
6.3	3つのキューイング構造	40
6.3.1	シングルキューイング構造	41
6.3.2	2クラスキューイング構造	41
6.3.3	ダイナミックキューイング構造	42
6.4	シミュレーションのシナリオ	42
6.4.1	トラヒックの負荷	46
6.5	スループット特性の評価	46
6.6	最適キュー長	52
6.7	ダイナミックキューイングの置換対象メッセージの検索方法について	53
6.8	考察	54
6.9	おわりに	55
第 7 章	結論	56
7.1	本研究の背景と目標	57
7.2	本研究の提案手法	57
7.3	本研究の主たる成果	58
	謝辞	59
	参考文献	60

目次

1.1	IP マルチメディアサブシステム [1]	2
1.2	東日本大震災における障害の観測事例	3
1.3	電話とメールサービスのエリア別サービス状況 [8]	3
2.1	IMS アーキテクチャの階層	9
3.1	IMS のセッション時のシグナリングとメディアトラヒック	13
3.2	音声通話における SIP メッセージのシーケンス	16
3.3	PMIM の場合の SIP シグナリングシーケンス	17
3.4	2 状態キュー制御によるスループット特性 [31]	19
4.1	既存研究のキューイング構造 [4]	22
4.2	2 クラスキューイングによるスループット特性 [4]	23
4.3	キューの非効率な利用	24
4.4	ダイナミックキューイング:キューの動的割り当て	24
4.5	セッション解放の失敗	25
4.6	ダイナミックキューイング:メッセージタイプによる優先制御	26
4.7	ダイナミックキューイング:再送メッセージの置換	27
4.8	ダイナミックキューイングのアルゴリズム	28
5.1	ダイナミックキューの導入箇所	30
5.2	SIP ヘッダの Struct	32
5.3	SIP メッセージタイプの定義	33
5.4	SIP メッセージの優先順位の定義	33
5.5	SIP メッセージタイプと優先順位をセットする	33
5.6	‘INVITE’ の再送処理のプログラム	34
5.7	‘non-INVITE’ の再送処理のプログラム	35
5.8	ダイナミックキューのヘッダファイル	36
5.9	ダイナミックキューのプログラム	37
6.1	過負荷時のネットワークの動作を説明するモデル [4]	41
6.2	ノードの定義をする NS2 の Tcl スクリプト	42
6.3	リンクの定義をする NS2 の Tcl スクリプト	43
6.4	ns2 ノード間の接続	44
6.5	複数の受信側のメディアサービスを定義する NS2 Tcl スクリプト	44
6.6	複数の送信側のメディアサービスを定義する NS2 Tcl スクリプト	45
6.7	音声と PMIM が 0.2 対 0.8 の時のスループット	47

6.8	音声と PMIM が 0.5 対 0.5 の時のスループット	48
6.9	音声と PMIM が 0.8 対 0.2 の時のスループット	48
6.10	音声と SMIM が 0.2 対 0.8 の時のスループット	49
6.11	音声と SMIM が 0.5 対 0.5 の時のスループット	49
6.12	音声と SMIM が 0.8 対 0.2 の時のスループット	50
6.13	発呼数 (Offered load) に対する完了呼数比率 (音声と PMIM が 0.5 対 0.5 の時) . .	51
6.14	実完了呼数の理想完了呼数に対する比率 (音声と PMIM が 0.5 対 0.5 の時)	51
6.15	スループットのキュー長への依存性	53
6.16	ダイナミックキューにおける 2 つの置換対象サーチ方法のスループット	54

■ 目 次

3.1	IMS の SIP 要求メソッド例	15
4.1	SIP メッセージの優先割り当て	25
6.1	負荷 60 でのスループット比較	52

第1章

序論

1.1 本研究の背景

1.1.1 IP マルチメディアサブシステム

IP マルチメディアサブシステム (IMS) は、音声とビデオなどのマルチメディアサービスやアプリケーションを、IP (Internet Protocol) をベースとしたパケット通信ネットワーク上で、柔軟に提供するために開発され、次世代ネットワーク (Next Generation Network/NGN) の中核的な仕組みである。ユーザが何処にいるに関わらず、IMS 対応のデバイスを使用できれば、Web サーフィンやオンラインゲームや音声通話など様々なマルチメディアサービスを楽しむことができる。元々は、第3世代パートナーシッププロジェクト (3GPP) が、2つの最も成功している通信パラダイムである携帯電話とインターネット技術の間のギャップを埋めるため企画し、第3世代パートナーシッププロジェクト2 (3GPP2) と共に標準化したものがIMSである。

図 1.1 に IMS のアプリケーション例を示す。このシナリオでは、ビデオ・ショップで面白そうな DVD を見つけたジェフが、マリーのケータイに VoIP (Voice over IP) で電話をかけ、買うかどうか相談するという設定である。途中で、ジェフは DVD のパッケージについて IC タグ (RFID) をケータイで読み取って、プロモーション・ビデオを保存しているストリーミング・サーバの URL (Uniform Resource Locator) をマリーに送り、2人で同じプロモーション・ビデオを見ながら、この DVD を買って帰るかどうかを相談することができる [1]。

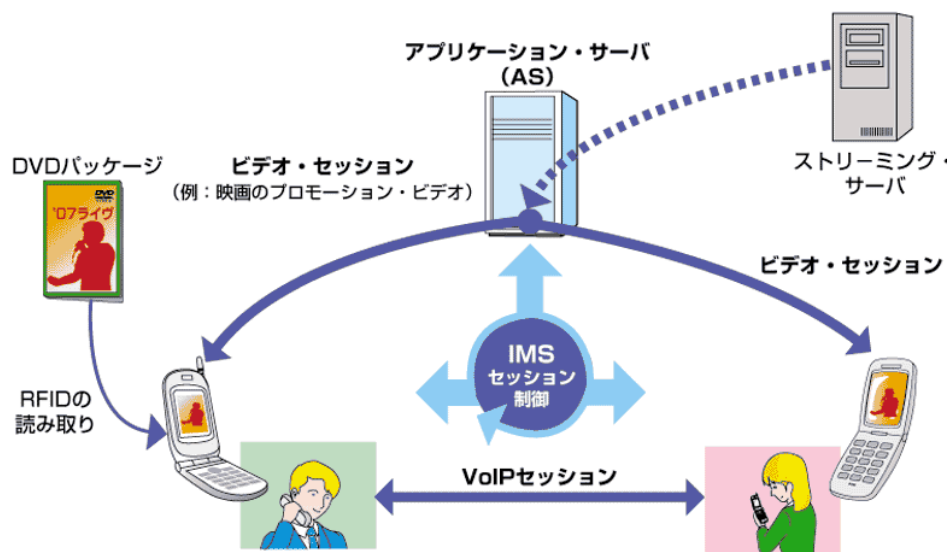


図 1.1: IP マルチメディアサブシステム [1]

1.1.2 災害時の過負荷障害について

突然の自然災害や地震などでトラフィックが爆発的に増加したり、IMS インフラの中の一部のコンポーネントの損傷したりして、IMS ネットワークが通常のサービスを提供できなくなるときがある。図 1.2 に、2011 年 3 月の東日本大震災時の状況を示す。被災地では、インフラの破壊が、非被災地では通信ネットワークに深刻な輻輳という二つの障害が観測された。

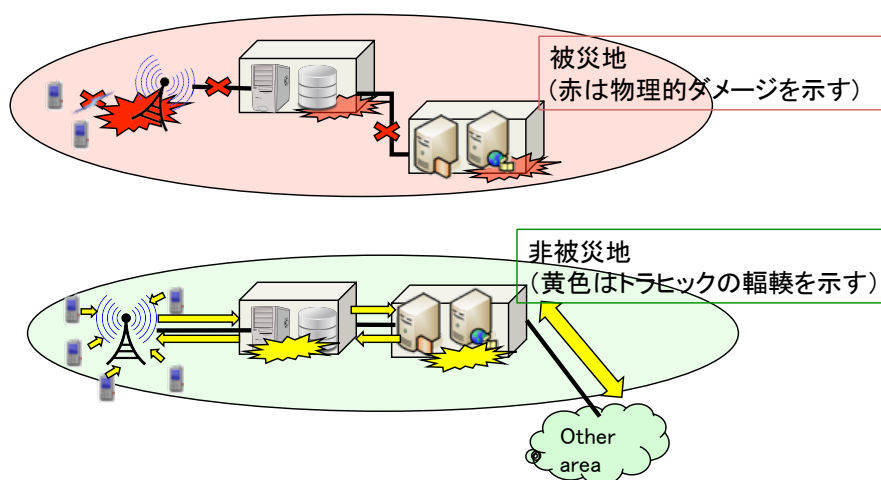


図 1.2: 東日本大震災における障害の観測事例

	Disaster area	No-disaster area
Call service	✕	95%~70% Rejected
Mail service	✕	80 minutes to deliver 90% of messages to the destination

図 1.3: 電話とメールサービスのエリア別サービス状況 [8]

図 1.3 は、地震後の通信サービスの状態を示す。ここで「✕」はサービス不能を意味する。まず、被災地のユーザは、携帯電話のすべてのサービスが提供できず、完全に外部から遮断された。非被災地の状況は、統計によると、約 70% から 95% までのコールが発信規制された [8]。一方、メールサービスは音声通話よりも優れていたが、それでも、送信者から受信者に届くまで長時間を要し、十分なサービス品質を提供できていたとは言い難い。被災地の場合は、インフラが破壊、非被災地の場合は、トラヒックの爆発的増加による障害が主因である。NTT ドコモの発表によれば、トラヒック量が通常よりも 50 から 60 倍と推定されている [8]。

1.1.3 IMS の過負荷回避手段

災害時にユーザに可能な限り通信サービスを提供するためには、IMS の過負荷回避を分析する必要がある。IMS はシングナリングプレーンとメディアプレーンからなり、それぞれに対して様々な研究がある。

従来は、メディアプレーンの研究事例が多い。映像信号や音声信号を伝送するため、Real-time Transport Protocol(RTP) が使われているが、James らは、パケット損失、遅延、音声コーデ、パケット損失補償アルゴリズムが VoIP の音声品質に与える影響を評価している [9]。また、Xu らは、複数の VoIP クラスに対してクラス別に通信サービスを保証する方法について研究している [10]。一方、東日本大震災では、観測された携帯電話ネットワークの音声トラヒック増が通常の 50 から 60 倍であったのに対し、パケットトラヒックは通常の約 2 倍ほどで、ほとんど発信規制の必要が

なかったことが報告されている [8]. 従って, 災害時にはシグナリングプレーンのトラヒック制御がメディアプレーントラヒックの制御よりも重要である.

IMS はシグナリングプレーンで, セッションの初期化や制御に Session Initiation Protocol(SIP)を採用している. IMS の発信規制は, SIP シグナリングメッセージの規制であり, 一般的にはパフォーマンスの低下につながる. SIP プロトコルは, メッセージ転送の信頼性を確保するために再送機構を用いている. Internet Engineering Task Force(IETF) によって標準化された RFC3261 [3]によると, 一定時間内に応答メッセージを受信できなかった場合, 以前送った SIP メッセージを再送する. このような制御は, 過負荷状態のネットワークでは, さらなる性能の低下や深刻な輻輳につながる可能性がある. しかし, IMS では発信規制手法は標準化されておらず, 事業者のノウハウに任されている. このため, 過負荷時の IMS ネットワークでの発信規制に関しては, いかなる手法が有効か不明な点が多い.

SIP シグナリングのトラヒック制御にも, これまで様々な研究がある. キューイング構造の観点から, 2つのキューを使った発信規制の自動化手法に関する Ohta の提案があり [4], また, Amooee と Falahati [5] は IMS のセッションセットアップ時に複数の P-CSCF ノードでプライオリティキューを用いた場合の効果を調べているが, 本論文で扱うような全セッション期間中での優先制御や一台ごとの P-CSCF のスループット向上に関しては未検討である. 他に, トランスポートプロトコルの差異が SIP シグナリングに与える影響を分析した例 [6] や, 再送が爆発的に増加する問題の解決を目指した検討 [7] などがある.

1.1.4 本研究の目的と目標

しかし, 以下の3問題に関しては, 十分な検討がされてきた訳ではない. 第1に, IMS は音声だけでなく Pager Mode Instant Messaging (*PMIM call*), Session Mode Instant Messaging (*SMIM call*) など多彩なマルチメディアサービスを提供している. このため, 発信規制は音声通話だけでなく, これら各種のサービスを考慮して実施する必要がある. 第2に, 緊急時の発信規制により各サービスの品質は低下する. 2011年の東日本大震災の際には, 携帯電話網における音声トラヒックが通常よりも50から60倍も増加したことが報告されている [8]. トラヒックが設計容量を大幅に超えたときにもシステムの最大スループットを前提にサービスの品質を論じたいが, 一般にこのような場合, システムの最大スループットを維持することは困難である. 第3に, SMIM と PMIM は IM の標準技術として規定されているが, これらは処理するために必要な Proxy Call Session Control Function(P-CSCF) の負荷処理能力が異なる. 輻輳が甚だしい時に多くの人々にサービスするには, セッションを張らなければデータを送れない SMIM よりも PMIM の方が望ましいことが予想できる. 以上から, 極端な輻輳時に適切な品質のマルチメディアサービスを提供するための方法を検討する必要がある.

本論文では, 設計値を大幅に超えるトラヒックが IMS ネットワークに流入した場合にも設計上の最大交換性能を維持し, 同時に, セッションサービス (session services) より通信オーバーヘッドの少ない非セッションサービス (non-session services) を自動的に優先させ, 可能な限り多くの人々に通信サービスを提供できるシステムの実現を目標にしている. 提案する自動発信規制手法では, スケーラビリティを持たせるため, セッションの状態に依存しないステートレスなキュー制御を導入し, それぞれのセッションでの SIP メッセージの種類 (タイプ), 時系列上に現れる順序, および再送回数だけを考慮した優先制御となっている. また, 最適キュー長も求めた. シミュレーションの結果は, 2011年東北大震災のような極端な輻輳下でも各マルチメディアサービスの最大スループットが保たれることを示している.

1.2 本論文の構成

本論文は、以下の各章によって構成される。

第1章 序論

第2章 IP Multimedia Subsystem(IMS) の概要

第3章 IMS における過負荷回避と関連研究

第4章 ステートレス発信規制のためのダイナミックキュー構造の設計

第5章 ダイナミックキューの NS2 シミュレータへの実装

第6章 評価

第7章 結論

まず2章でIMS出現の由来とその概念を説明し、現在使われている標準アーキテクチャを説明する。また、メディアセッションを制御するためにIMSが採用しているセッション初期化プロトコル(SIP)について述べる。さらに、IMSが提供するメディアサービスをいくつか概説する。

3章では、IMSにおける過負荷回避法について述べる。トラヒックのカテゴリをシグナリングプレーンとメディアプレーンに分けて分類した後、マルチメディアサブスに於けるSIPの役割を示し、過負荷回避に関する既存研究をまとめる。

4章では、IMSのシグナリング制御のためのステートレス発信規制手法が依拠するダイナミックキューイング構造の設計について検討する。まず第1に、既存のSIPシグナリングの優先制御手法の設計例と問題点を説明する。次に、これをもとに、改善方法を提案する。既存手法の代表例である2クラスのプライオリティキューイング構造を比較することにより、ダイナミックキューイング構造の特長を示す。

5章では、NS2で作成したダイナミックキューイングの実装の詳細および、ダイナミックキューイングをサポートするために必要な変更について述べる。最後に、NS2シミュレータの変更モジュールを全て示すことにする。

6章では提案したダイナミックキューイングによる自動発信規制手法を評価するための実験内容とその結果について示す。

7章では本研究の成果を結論としてまとめる。

■ 第2章

IP Multimedia Subsystem(IMS)の概要

2.1 はじめに

本章では、現在使われている IP マルチメディアサブシステムについて概観する。最初に、IMS 出現の由来とその概念を説明し、現在よく使われているアーキテクチャを説明する。また、メディアセッションを制御するために IMS によって採用されているセッション初期化プロトコル (SIP) を述べ、さらに、IMS によって提供された複数のメディアサービスについてまとめることにする。

2.2 IP マルチメディアサブシステム

2.2.1 コンセプト

インターネットは、小規模なネットワークとして開始したが、現在では世界規模のネットワークに発展した。一方、世界の総人口 70 億人内で 2011 年末には携帯電話加入者数は 54 億 4,270 万加入が見込まれている [11]。これらの近年最も成功している通信パラダイムである携帯電話とインターネット技術の間のギャップを埋めているため、第 3 世代 (3G) ネットワークで IMS が開発された。IMS は 3G アーキテクチャの重要な要素であり、ユーザが何処にいるにかかわらず、IMS 対応のデバイスを使用しているに限り、プレゼンス、音声、Short Message Service(SMS) およびマルチプレイヤーゲームなどを含めた種々のサービスを利用できる。

2.2.2 IMS 開発の歴史

IMS は、当初は GSM ネットワークと無線技術の進化をサポートするため、その標準化作業の一環として、第三世代パートナーシッププロジェクト (3GPP) によって定義された。

これに続いて、第三世代パートナーシッププロジェクト 2 (3GPP2) も IMS を標準化した。3GPP2 仕様の初期リリースでは、主に 3GPP のリリース 5 に準拠している。二つの組織によって定義された 2 つの IMS ネットワークはかなり似ているが、全く同じではない。3GPP2 は、固有の問題に対する仕様を追加している。それにもかかわらず、両方の組織の目的は、IMS アプリケーションが異なったネットワークインフラストラクチャを跨っても一貫して動作するようにすることにある。

2.2.3 IMS の利点

既存の携帯電話ネットワークインフラストラクチャ上で IMS を採用する利点は、次の 4 つの側面から示すことができる [16]。

- **IMS は、新しいマルチメディアサービスを展開する際、市場投入までの時間を削減する共通プラットフォームを提供できる:** 今日の通信ネットワークにおける最大の課題の 1 つは、新しいサービスを作成する際の長いコストのかかるプロセスを改善することにある。IMS インフラストラクチャは、標準化されたプラットフォームと再利用可能なコンポーネントを提供することにより、この問題を解決できる。さらに、任意のプロバイダは、IMS で標準化されたインタフェースを実装すれば、サービスを提供することができる。マルチベンダーサービスがオープンマーケットになり、各サービスプロバイダは新しいサービスを展開する際、最も効果的な方法を選択することができるようになる。

- **IMS は、マルチメディアサービスのサービス品質 (QoS) を制御できる:** 3G セルラーネットワークが提供するベストエフォートサービスでは、ネットワークは必要な帯域幅を確保するために最善を尽くすが、同じ品質レベルである保証はない。サービス品質 (QoS) メカニズムは、いわゆる”ベストエフォート”の代わりに送信中にネットワーク帯域を保証をするために開発された。IMS には、IP ネットワーク上でサービス品質制御を有効化し、伝送品質の向上と保証を実現する QoS メカニズムがある。
- **IMS により、オペレータが適切なマルチメディアセッションに課金することができる:** IMS の利点は、ユーザが使うサービスの情報を提供してくれるため、オペレータはサービスの種類に基づいてユーザに課金する方法を決定することができることにある。例えば、セッション持続時間や転送されたバイト数によりユーザに課金できる。また、それらを組み合わせた新しい課金を実行することもできる。
- **IMS では、すべてのサービスがユーザの位置にかかわらず利用できる:** 携帯電話技術において、典型的な、しかし厄介な問題は、ユーザが他の国にローミングしたときに一部のサービスが利用できないことである。この問題を解決するために、IMS では、インターネット技術とプロトコルを使用し、ユーザが国を越えて移動してもホームネットワークと同じに、すべてのサービスを実行することができるようにしている。

2.3 IMS のアーキテクチャ

IMS は、携帯電話網だけでなく、IP ベースのインフラストラクチャ上で音声やビデオやデータのコンバージェンスを目指した次世代ネットワーク (NGN) アーキテクチャの標準にも採用されている。

図 2.1 は 3GPP で標準化された IMS アーキテクチャの概要を示す。

この簡略化された IMS のアーキテクチャは、3つの主要な層から構成されている。

1. サービス/アプリケーション層: この層で、実際のアプリケーションサービスが実行される。音声メール、音声応答などの伝統的な音声サービスや IMS アーキテクチャ上に開発される新しいサービスもこの層に含まれる。第二世代 (2G) ネットワークのように垂直統合されたアーキテクチャと異なり、HSS、ルーティング、セッション制御などのネットワーク機能を全て組み込んで構成する必要はない。この抽象化があるため、IMS では新しいサービスを迅速に導入することができる。
2. コアネットワーク層: コアネットワーク層は、認証、ルーティング、およびトランスポート層とサービス層の間の IMS トラヒックの制御をする。この層でのトラヒックのほとんどは VoIP 技術に関連付けられることの多いセッション開始プロトコル (SIP) に基づいている。サービスに SIP メッセージを、ホームネットワークや他のサービスプロバイダの IMS ネットワーク上のサービスに適切にルーティングすることに加えて、コアネットワーク層は、サービス層と他のサービスとのインタフェース機能を提供する。
3. トランスポート層: トランスポート層は、IMS アーキテクチャから、実際のアクセスネットワーク (固定回線、パケット交換無線チャンネルなど) を抽象化している。本質的には、この層はアクセスネットワークとその上の IP ネットワークからなる。これは、IP の初期化プロビジョニング (DHCP 経由で IP アドレスやデフォルトゲートウェイを割り当てる) や、デ

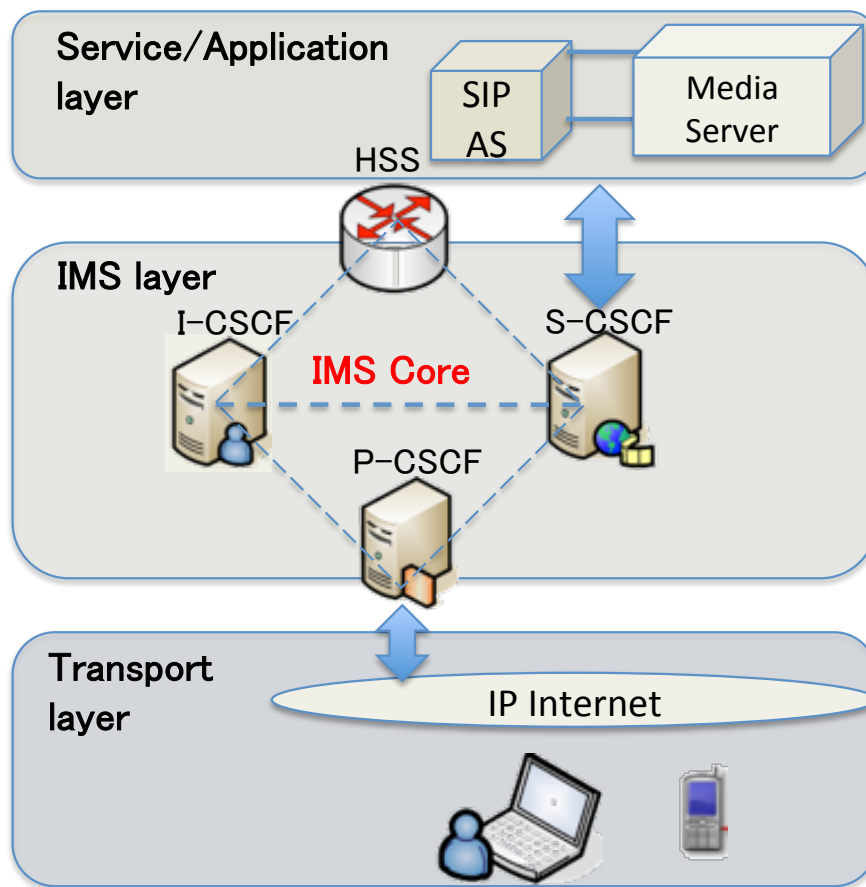


図 2.1: IMS アーキテクチャの階層

バイスを上位層に登録すること等を実行する。トランスポート層について重要なことは、その上のコントロールとサービス層は全て IP ベースであるのに対して、アクセスネットワークそのものは IP ベースである必要はないことである。

コアネットワーク層は、IMS コアに含まれているノードから成る。これらのノードは以下のものがある：

ホーム加入者サーバ (HSS) ホーム加入者サーバ (HSS) は、マスターユーザデータベースである。加入者ユーザプロファイルのような加入者の情報を持っていて、ユーザの認証と認可を実施し、加入者の位置情報や IP 情報を提供できる。これは GSM の Home Location Register (HLR) および Authentication Centre(AUC) に似ている [20]。

呼/セッション制御 (CSCF) Call Session Control Function (CSCF) は IMS の中で SIP シグナリングの処理を実行するノードである。

1. Proxy-CSCF(P-CSCF) は IMS 端末が最初に接続する SIP プロキシである。
2. Interrogating-CSCF(I-CSCF) は管理ドメインの端に位置する SIP 機能の 1 つで、SIP のレジストレーション時にユーザに S-CSCF をアサインし、そこへシグナリングの中継をすることができる。その IP アドレスはそのドメインの Domain Name System(DNS) によって公開されるので (DNS レコードとしては NAPTR や SRV 形式を使用)、リモートサーバが探し出すことができ、ドメイン間の SIP パケットの転送点として使われる。I-CSCF は HSS に対して DIAMETER の Cx インタフェースでユーザ位置情報を要求し (Dx インタフェースは I-CSCF から SLF に対して使い、HSS を特定する)、SIP 要求を割り当てられた S-CSCF にルーティングする。
3. Serving-CSCF(S-CSCF) は信号層の中心的ノードである。S-CSCF は SIP サーバであり、セッション制御を実行し、常にホームネットワークに位置する。DIAMETER の Cx および Dx インタフェースを使い、HSS にユーザプロファイルのダウンロード/アップロードをする。つまり、S-CSCF にはユーザに関する情報を保持するデータベースはなく、必要な情報は全て HSS からロードする。

2.4 Session Initiation Protocol

Session Initiation Protocol(セッションイニシエーションプロトコル, SIP, セッション確立プロトコル) とは、インターネットと IMS の統合のために、インターネットエンジニアリングタスクフォース (IETF) によって IMS のセッション制御プロトコルとして選択され、2 つ以上のクライアント間でセッションを確立するための IETF 標準の通信プロトコルである [18]。SIP は IP 電話などのセッションの開始、変更、終了などの操作をおこなう汎用のセッション制御プロトコルとして開発された。SIP のおもな用途は、電話、テレビ会議やインスタントメッセージング、複数プレイヤーのオンラインゲームなどである。

2.5 IMS の提供するサービス

前述したように、IMS の主な目標はマルチメディアサービスを提供することである。IMS の第 1 の目標は、できるだけ早くサービスを開発およびデプロイできることである。この特長があるた

め、多くの新しい、革新的なサービスは、IMS インフラストラクチャを使用している。したがって、IMS サービスのすべてをリストアップすることは不可能であるが、IMS の中で最も一般的なサービスは、音声通話、プッシュツートーク、インスタントメッセージング、プレゼンス、会議などである。

2.6 おわりに

本章では、現在流布している IP マルチメディアサブシステムの由来とアーキテクチャを説明した。また、メディアセッションを制御するために採用されている SIP に関してその役割を説明した。さらに、IMS が提供するメディアサービスの代表例をいくつか概説した。次章では、IMS における過負荷回避と関連研究について検討する。

第3章

IMSにおける過負荷回避と関連研究

3.1 はじめに

本章では、IMSにおける過負荷回避法について述べる。まず、トラヒックのカテゴリをシグナリングプレーンとメディアプレーンに分けて分類する。そして、マルチメディアサブサービスにおける SIP の役割を示し、過負荷回避に関する研究をまとめる。

3.2 IMSにおけるトラヒックの分類

一つのセッション中には、シグナリングプレーンとメディアプレーン双方でトラヒックが生成される。それぞれのトラヒックを示すため、図 3.1 に1つのセッションの確立時に生ずるトラヒックを示した。

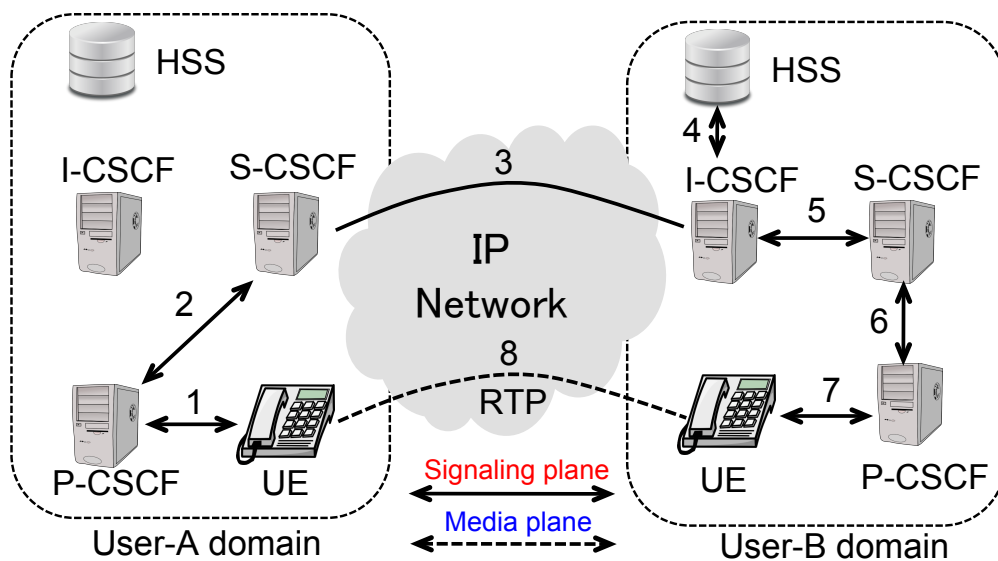


図 3.1: IMS のセッション時のシグナリングとメディアトラヒック

セッション確立は、8つのステップから成る。ステップ1から7までは、SIP メッセージがシグナリングプレーンで転送される。ステップ8における音声データはメディアプレーンで運ばれる。

1. シグナリングプレーン:

Step 1: ドメイン A のユーザ A はリクエストを P-CSCF(A) に送る。P-CSCF(A) は、ユーザと IMS コアネットワークの最初のコンタクトポイントである。

Step 2: この SIP メッセージは、P-CSCF(A) からユーザ A のホームネットワークの Serving-CSCF(S-CSCF(A)) へ転送される。S-CSCF(A) は、レジストレーション手順がユーザに割り当てる。

Step 3: SIP パケットに含まれるドメインアドレスをチェックすることによって、S-CSCF(A) は、ユーザ B がユーザと同じドメインに属していないことを知る。そして、S-CSCF(A) はドメイン間の SIP パケットの転送点として使われる Interrogating-CSCF(I-CSCF(B)) へ SIP パケットを転送する。

Step 4: I-CSCF(B) は HSS に、ドメイン B のユーザ B にアサインされている S-CSCF(B) の IP アドレスを要求する。このとき、DIAMETER の Cx インタフェースでユーザ B 位置情報を要求する。ここで、Dx インタフェースは I-CSCF から SLF に対して使い、HSS を特定する。

Step 5: I-CSCF(B) は SIP 要求を割り当てられた S-CSCF(B) にルーティングする。

Step 6: リクエストを受信した後、S-CSCF(B) は保持される情報からユーザ B に割り当てられた P-CSCF(B) を割り出す。これは、ユーザ B のレジストレーション時に決まっている。S-CSCF(B) はリクエストをこの P-CSCF(B) に転送する。

Step 7: P-CSCF(B) はユーザ B の代わりにリクエストを受信し、最終宛先のユーザ B にリクエストを転送する。

2. メディアプレーン:

Step 8: シグナリングプレーンでのセッションのネゴシエーションが正常に完了した後、ユーザ A とユーザ B が参加するリアルタイム通信で生成されるメディアトラヒックはメディアプレーンで運ばれる。

最も重要なことは、シグナリングとメディアの完全な分離である。シグナリングトラヒックはコアネットワークの CSCF 間で転送されるが、メディアトラヒックは、別経路で直接ユーザ A からユーザ B、ユーザ B からユーザ A へエンドツーエンドに送信される。

3.3 IMS サービスにおける SIP メソッド

SIP シグナリングトラヒックは IMS のトラヒックの中で大きな割合を占めている。したがって、IMS における過負荷回避の問題を論じるには、SIP シグナリングトラヒックに注意しなければならない。

IMS で提供されるさまざまなサービスは、要求時にそれぞれの SIP メソッドを採用しているので、すべての IMS サービスを含めることは不可能であり、ここでは5つの共通サービスだけを説明する。表 3.1 はそれぞれのサービスに割り当てた各 SIP メッセージを示す。ここで、○はその SIP メソッドを使うことを意味し、×は使わないことを示す。ページ制限のため、正常またはエラーに関する自明の応答は、すべて省略されている。

表 3.1 から二つの結論を得ることができる。

1. IMS のサービスは、セッションサービスと非セッションサービスに分割することができる。音声通話やプッシュトゥートークやセッションモードのインスタントメッセージングなどのセッションサービスはセッションを確立する必要がある。セッション制御はセッションを初期化する‘INVITE’メソッドを用いる。一方、非セッションサービスは、セッションの確立を必要としない。
2. 第二に、トラヒックの観点から、IMS のサービスは、シグナリングのみと、シグナリングとメディアプレーン両方が必要なものとに分類することができる。たとえば、ページャモード・インスタントメッセージングとプレゼンスサービスはシグナリングトラヒックしか発生しない、他の3つのサービスでは、シグナリングとメディアトラヒックの両方を必要とする。

表 3.1: IMS の SIP 要求メソッド例

Method	Signaling			Signaling & Media	
	No-session services		Session services		
	Presence	Pager Mode IM	Session Mode IM	Voice Call	Push to Talk
INVITE	×	×	○	○	○
ACK	×	×	○	○	○
BYE	×	×	○	○	○
CANCEL	×	×	○	○	○
PRACK	×	×	○	○	○
SUBSCRIBE	○	×	×	×	×
NOTIFY	○	×	×	×	○
PUBLISH	○	×	×	×	×
REFER	×	×	×	×	○
MESSAGE	×	○	×	×	×
UPDATE	×	×	○	○	○

3.4 サービスにおける SIP シーケンス

シグナリングプレーン上の SIP シグナリングトラヒックを制御する為には、各サービスで使用する SIP メソッドを理解するとともに、一つのリクエスト中に交換される詳細な SIP シーケンスを理解することも必要である。

3.4.1 音声通話における SIP シーケンス

IMS ネットワークでの SIP の役割を示すため、図 3.2 に音声通話における SIP メッセージのシーケンスを示す [19]。

一つの音声通話中には、シグナリングプレーンとメディアプレーン双方でトラヒックが生成される。メディアプレーンは音声データを運び、7つの SIP メッセージがシグナリングプレーンで転送される。

(1) まず、通信セッションを初期化するために、ユーザ A からユーザ B へ ‘INVITE’ メッセージが送信される。以前説明したように、この SIP メッセージは P-CSCF(A) から P-CSCF(B) へ転送される。

(2) ユーザ B がリソース確保を開始すると、暫定応答 ‘100Trying’ が経路に沿ってユーザ A まで返される。

(3) 次にもう一つの暫定メッセージ ‘180Ringing’ がユーザ A に返される。これで、ユーザ B の電話機が鳴り、ユーザ B が出るのを待つ状態になっていることがユーザ A に分かる。

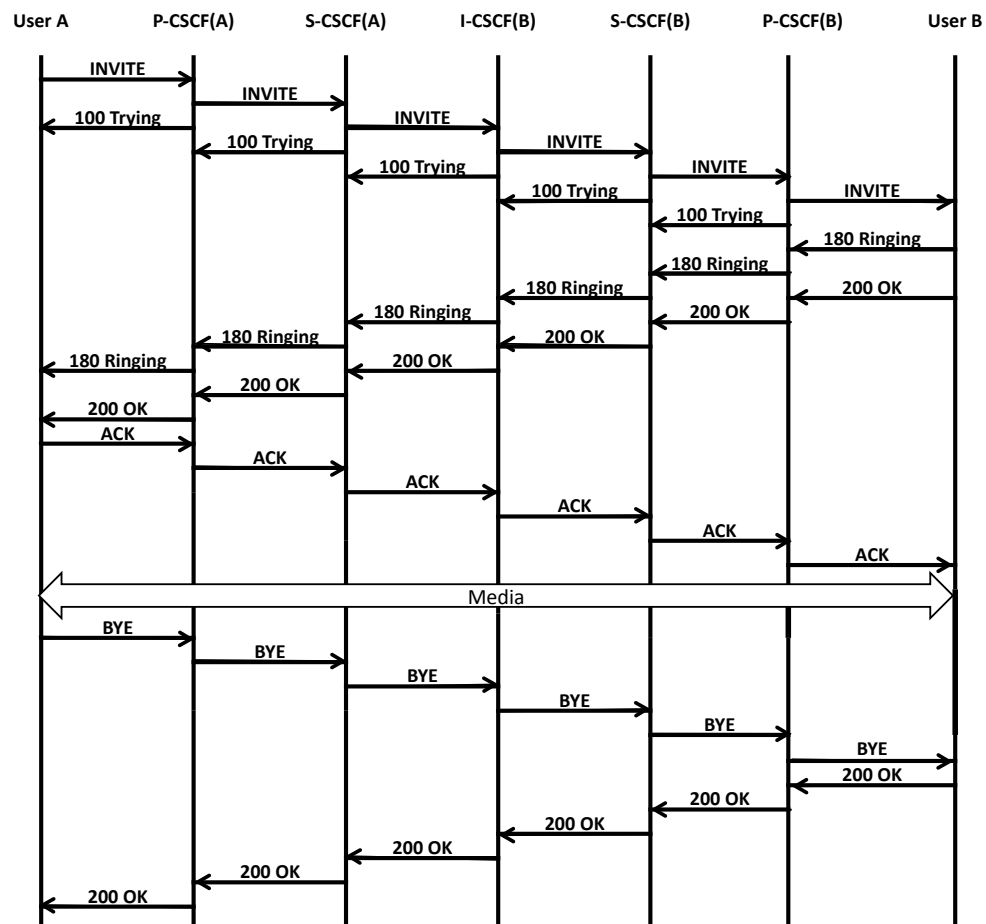


図 3.2: 音声通話における SIP メッセージのシーケンス

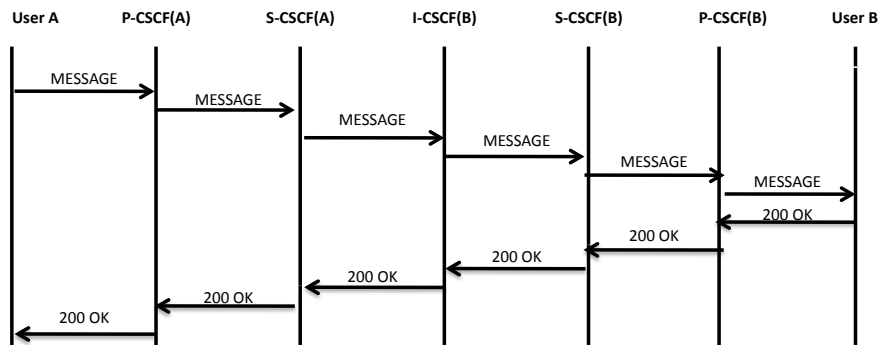


図 3.3: PMIM の場合の SIP シグナリングシーケンス

(4) リソース確保が完了すると、ユーザ B からユーザ A にセッション応答 ‘200OK’ が通知される。

(5) セッションの初期化は、この SIP メッセージの ‘ACK’ がユーザ A からユーザ B へ到達したとき完了する。

(6) また、セッションを終了するには、ユーザ A がユーザ B へ ‘BYE’ メッセージを送ればよい。

(7) ユーザ A へメッセージ ‘200OK’ が返されたとき、すべてのリソースが解放されセッションは終了する。

3.4.2 ページャインスタントメッセージングにおける SIP シーケンス

図 3.3 に、ページャインスタントメッセージング (以下 PMIM と呼ぶ) の SIP シーケンスを示す。音声通話と比べると、主に二つの違いがあることがわかる。

1. 音声通話はシグナリングとメディア両方のプレーンが必要であるが、PMIM の場合はシグナリングプレーンだけが必要である。
2. 2 つだけの SIP メッセージで完了できるので、PMIM はオーバーヘッドが少ないという利点がある。

3.5 IMS の過負荷回避に関する関連研究

3.5.1 メディアプレーンにおける過負荷回避

IMS はシグナリングプレーンとメディアプレーンからなり、従来、メディアプレーンの研究事例が数多く実施されてきた。映像信号や音声信号を伝送するため、Real-time Transport Protocol(RTP)が使われているが、James らは、パケット損失、遅延、音声コーデ、パケット損失補償アルゴリズムが VoIP の音声品質に与える影響を評価している [9]。また、Xu らは、複数の VoIP クラスに対してマルチレベルの通信サービスを保証する方法について研究している [10]。

しかし、東日本大震災で観測された携帯電話ネットワークの音声トラヒック増が通常の50から60倍であったのに対し、パケットトラヒックは通常の約2倍ほどで、ほとんど発信規制の必要がなかったことが報告されている [8]。従って、災害時にはシグナリングプレーンのトラヒック制御がメディアプレーントラヒックの制御よりも重要である。

3.5.2 シグナリングプレーンの過負荷回避

シグナリングプレーンの過負荷保護は、SIP サーバがユーザから受け取るすべての SIP メッセージを処理するには、計算リソースが不足しているときに起こる過負荷（オーバーロード）に注目している。この過負荷結果は、単にサーバのパフォーマンスが低下するだけでなく、提供するサービス全体がダウンすることにもつながることがある。

オーバーロードを引き起こす理由に基づいて、これらの研究は、2つのカテゴリにさらに分割することができる。

再送ストーム

この研究では、ロードを増加させ、SIP シグナリングネットワークの処理性能低下をもたらす再送 SIP メッセージに関して検討する。

過負荷制御の観点から、文献 [31] では、ローカルなオーバーロード制御のためのアルゴリズム手法を提案している。アルゴリズムは詳細には次のように設計されている。

1. サーバには2つの状態がある:(1) 過負荷と (2) 低負荷状態。サーバ側の受信メッセージキューの長さが最高しきい値 h を超えたときに、サーバは低負荷状態から過負荷状態へ遷移する。一方、サーバ側の受信メッセージキューの長さが最低しきい値 l を超えたときに、サーバは過負荷状態から低負荷状態へ戻る。
2. 低負荷状態では、すべてのメッセージが処理される。
3. 過負荷状態では、すべての着信した 'INVITE' メッセージは、503（サービス利用不可）応答で拒否される。

図 3.4 に示すように、このアルゴリズムはメッセージの再送を回避できる。

発信規制

ここでの研究は、非常に多くのユーザすべてが同時に発呼するときに発生するオーバーロードを扱っている。たとえば、自然災害が発生すると、緊急通話が劇的に増加し、過負荷が発生する。年末年始のような特別なイベントやテレビ等の番組が誘導したイベントへの申し込み受付によってもオーバーロードが発生する。

本研究の提案は、この分野に絞っているので、関連する研究手法は次の章で詳しく説明する。

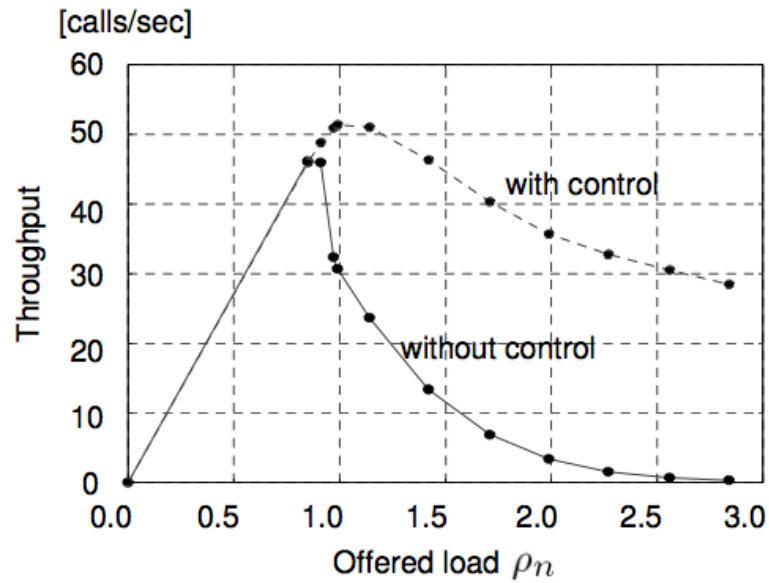


図 3.4: 2 状態キュー制御によるスループット特性 [31]

3.6 おわりに

本章では、IMS におけるトラヒックの分類、IMS サービスにおける SIP メソートを概説した。また、過負荷回避に関する既存研究ををまとめ、さらにシグナリングプレーン上のマルチメディアサブスにおける SIP の役割を示した。次章から、提案するステートレス発信規制について詳述する。

第4章

ステートレス発信規制のための ダイナミックキュー構造の 設計

4.1 はじめに

本章では、IMS のシグナリング制御のためのステートレス発信規制手法が依拠するダイナミックキューイング構造の設計について検討する。まず第1に、既存の SIP シグナリングの優先制御手法の設計例と問題点を説明する。次に、これをもとに、改善方法を提案する。最後に、既存手法の代表例である2クラスのプライオリティキューイング構造を比較することにより、ダイナミックキューイング構造の特徴と利点を示す。

4.2 2つのキューを使った SIP シグナリングの優先制御手法

SIP シグナリングのトラヒック制御にも、これまで様々な研究がある。キューイング構造の観点から、2つのキューを使った発信規制の自動化手法に関する Ohta の提案があり [4]、また、Amooee と Falahati [5] らは、IMS の各ノードを複数のサーバーで構成して待ち時間を短くし、それにより呼がタイムアウトになって捨てられる確率を小さくできることを示した。同時に、有線分に呼を用いて、待つ時間が短くなるにより、放棄呼の確率が小さくなる。また、プライオリティキューがシステム性能に与える影響を待ち行列理論を使って検討している。しかし、本論文は扱うような全セッション期間中での優先制御や一台ごとの P-CSCF のスループット向上に関しては未検討である。他に、トランスポートプロトコルの差異が SIP シグナリングに与える影響を分析した例 [6] や、再送が爆発的に増加する問題の解決を目指した検討 [7] などがある。

図 4.1 に、2つのキューを使った発信規制の自動化手法の設計を示す。

図 4.1(a) は、すべての SIP メッセージが同じキューを共有し、First In, First Out (FIFO) のスタイルで処理される伝統的なシングルキューイングを示している。

図 4.1(b) は、音声の発信規制を行う Ohta の2つのキューを使った発信規制の自動化手法（以下2クラスキューイングと呼ぶ）が用いていた構造 [4] である。

この手法は二つの特徴がある。

1. この手法では、SIP メッセージを ‘INVITE’ と ‘non-INVITE’ に分類し、
2. この分類された SIP メッセージそれぞれは特定のサイズのキューを持つ2つのキューに格納され、‘non-INVITE’ キュー内の SIP メッセージは高い優先度で処理され、‘INVITE’ メッセージは低い優先度で処理される。

4.3 2クラスのプライオリティキューイング構造の欠点

2クラスのプライオリティキューを使った SIP シグナリングの優先制御手法は、図 4.2 に示すように、音声通話を効果的に発信規制できるが、2つの問題がある。

1. 第1は、図 4.2 から分かるように、設計容量の数十倍に達するトラヒックのような過負荷時の交換性能がよくない。
2. 第2に、音声だけしか処理できず、インスタントメッセージングなどマルチメディアサービスに対応できていない。

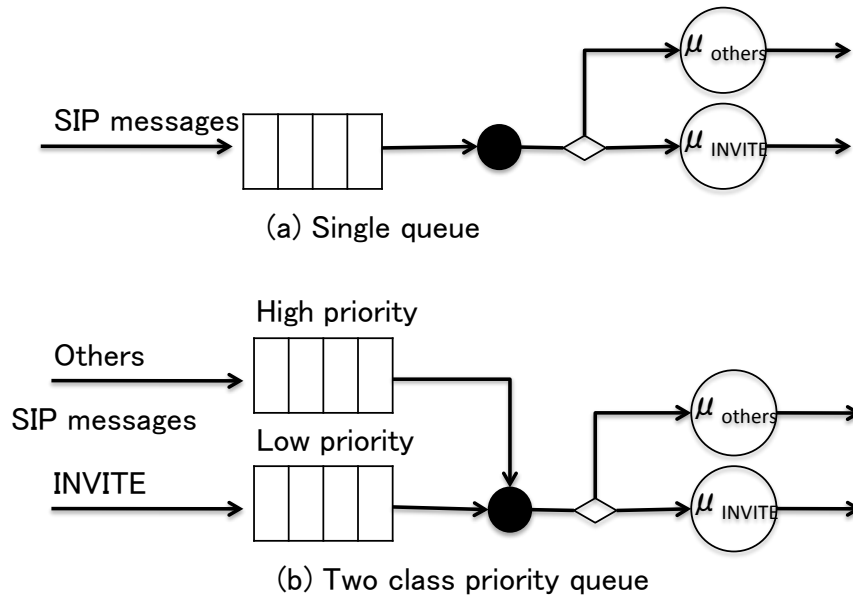


図 4.1: 既存研究のキューイング構造 [4]

本論文はこれらの問題に対応して、新たなキューイング構造を提案し、以下の3点を目指す。

1. 第1は、音声サービスとPMIMやSMIMを組み合わせた複数のサービストラヒックに対する発信規制法の実現である。
2. 第2に、極端な輻輳時にもシステムの最大性能を維持する形で全てのサービスが提供できることを目標にする。
3. 第3に、輻輳時にPMIMのような軽負荷のサービスの品質が相対的に良くなるような制御を目標にする。

4.4 ステートレスなシグナリング優先制御を実現するためのダイナミックキューイング機構の提案

本節では2クラスのプライオリティキューイング構造の問題点を、キューの構成、メッセージタイプによる優先制御と再送メッセージに対する優先制御という三つの観点から検討し、ダイナミックキューイング機構と呼ぶ改善方法を提案する。そして、2クラスのプライオリティキューイング構造を比較することにより、ダイナミックキューイング構造の特徴と利点を示す。

4.4.1 キューの構成

輻輳時にキューの使用率を最大限にするためには、キューがパケットで満たされているいなければならない。しかし既存手法 [4] では、優先処理を管理するキューのサイズが固定されているため、‘INVITE’のキューがあふれても、‘non-INVITE’のキューに空きがある可能性があり、キュー利用

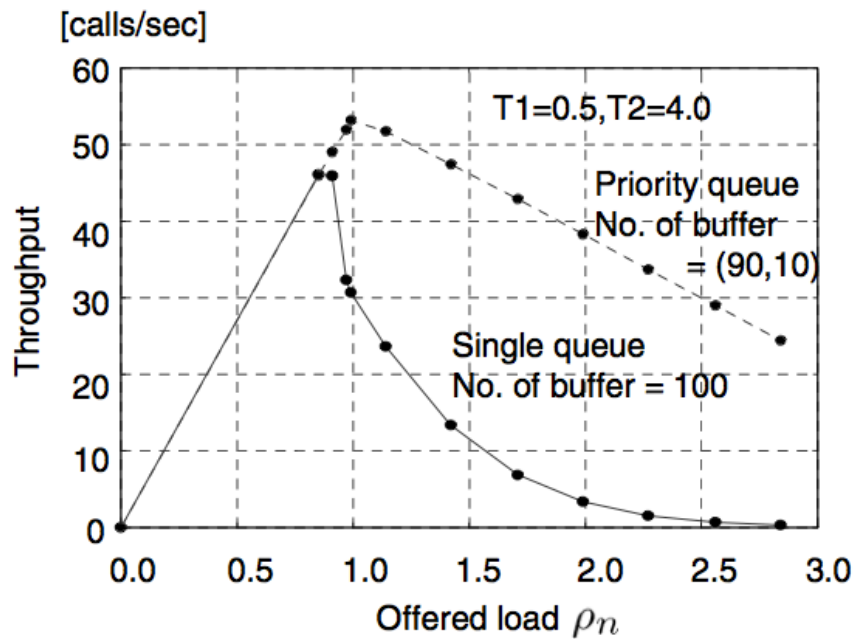


図 4.2: 2 クラスキューイングによるスループット特性 [4]

率を最大にすることはできない。

図 4.3 に、2 クラスのプライオリティキューイングでは、キュー利用率を最大にすることはできない例を示す。この例では、SIP メッセージ ‘TRYING’ が来たとき、‘INVITE’ キューには余裕があるが、‘non-INVITE’ キューが一杯になっているので、‘TRYING’ は破棄されてしまう。

本研究では、すべての SIP メッセージが同じキューを共有する動的なキューの割り当て（以下ダイナミックキューイングと呼ぶ）を提案する。P-CSCF の容量以下の負荷の場合、単一の待ち行列を FIFO で処理するモデルとなり、キューの長さが最大キュー長より短ければ、SIP メッセージはキューの最後尾に付加される。

図 4.4 に、ダイナミックキューイングの場合に関して、図 4.3 と同じ状況に於ける例を示す。ダイナミックキューイングでは、SIP メッセージ ‘TRYING’ が来たとき、現在のキューがいっぱいになっていないに限り、キューに格納できる。このため、キュー利用率を最大にすることができる。

4.4.2 メッセージタイプによる優先制御

既存手法 [4] では、SIP メッセージを 2 つの優先レベルに分類していたが、以下の観点からさらに改善できる余地がある。1 つの音声のセッションを終了するには少なくとも 7 つの SIP メッセージが必要であり、まず、それぞれのメッセージはセッションの期間中に異なる役割をしている。

たとえば、‘100Trying’ または ‘RINGING’ のような SIP メッセージは、リソース予約のための暫定的な応答としての役割を持っている。一方、実際にセッションを開始するのは、‘200OK’ と

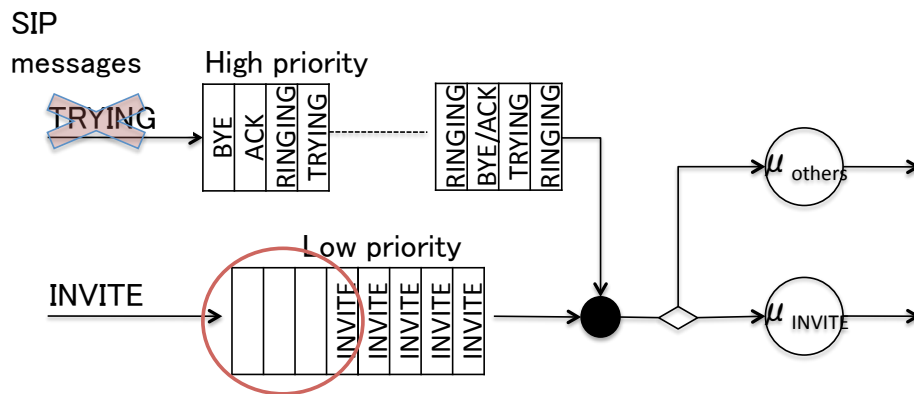


図 4.3: キューの非効率な利用

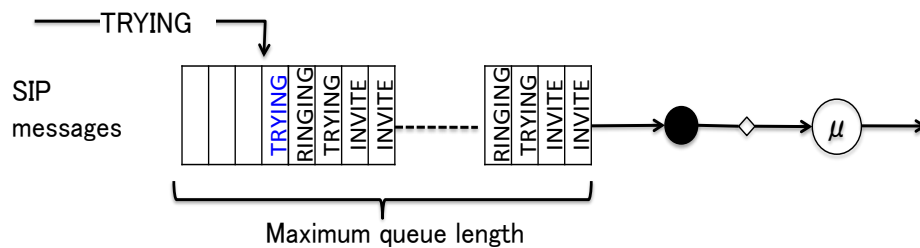


図 4.4: ダイナミックキューイング:キューの動的割り当て

‘ACK’のSIPメッセージが正常に送信された後である。又、‘BYE’や‘BYE/200OK’のようなSIPメッセージは、セッションを終了させ占有リソースを解放する役割を持っている。

つまり、1つのセッション中で、時系列上に後に現れるSIPメッセージの方が、セッションの初期段階に現れるSIPメッセージより、セッション確立に関して重要な役割を果たしている。しかし、[4]の提案ではこれを考慮していなかった。

図4.5に、セッション解放の失敗例を示す。音声通話セッションで送信された最後のSIPメッセージ‘BYE/ACK’が来たとき、‘non-INVITE’キューがいっぱいになっているので、‘TRYING’は破棄されている。この例では、‘non-INVITE’キューに、リソース予約のための暫定的な応答としてのSIPメッセージ‘RINGING’がある。‘RINGING’と‘BYE/ACK’を比較すると、後者は既存のセッションを終了させ占有リソースを解放する役割を持っているので、優先させるべきである。しかし、[4]の提案では破棄されてしまう。

また、サービスに対して、優先度の考慮も必要である。音声通話の場合は、送信されるSIPメッセージは7個であるが、PMIMの場合は、送信されるSIPメッセージは2個だけで、2つ目のSIPメッセージの‘IM/200OK’が正しく受信された場合だけ、IMは成功したと見なされる。このため、PMIMを他のサービスより優先させるためには他のSIPメッセージよりも‘IM/200OK’を優先させればよい。

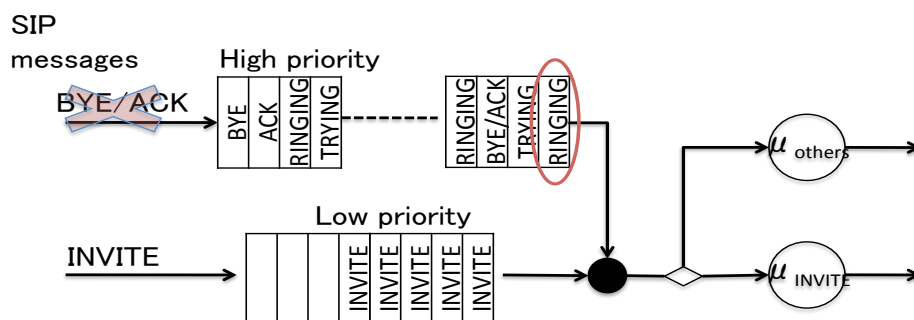


図 4.5: セッション解放の失敗

表 4.1: SIP メッセージの優先割り当て

Method	PMIM	SMIM	Voice	Prio.TP	Prio.D
INVITE	×	○	○	1	1
100Trying	×	×	○	2	2
180Ringing	×	×	○	2	3
200OK	×	○	○	2	4
ACK/200OK	×	○	○	2	5
BYE	×	○	○	2	6
BYE/ACK	×	○	○	2	7
MESSAGE	○	×	×	1	1
IM/200OK	○	×	×	2	8

表 4.1 に、各 SIP メッセージへ割り当てた優先順位を示す。Prio.TP は 2 クラスキューインクの優先度、Prio.D はダイナミックキューイングの優先度とする。ここで、表中の数字の大きいメッセージほど優先され、○、× は PMIM 等のサービスが用いるメッセージか否かを示している。

図 4.6 にメッセージタイプによるメッセージの優先制御の手法の 1 例を示す。この例では、キューは満たされているが、優先順位が 7 の 'BYE/ACK' メッセージが到着している。このとき、ダイナミックキューイングは次のように操作する。

1. キューの先頭から探す。最初のメッセージは同じ優先順位を持っている 'BYE/ACK' メッセージなので、次のメッセージを移る。
2. 到着した 'BYE/ACK' メッセージの優先順位は次の 'TRYING' メッセージより高いため、'TRYING' を置き換える。

ここで、キューの先頭に近いメッセージほどタイムアウトで既に無効になっている可能性が高いと考え、キューの最初から探索することになっている。また、キューの管理コストを考え、単純にメッセージの置換を行うことにした。

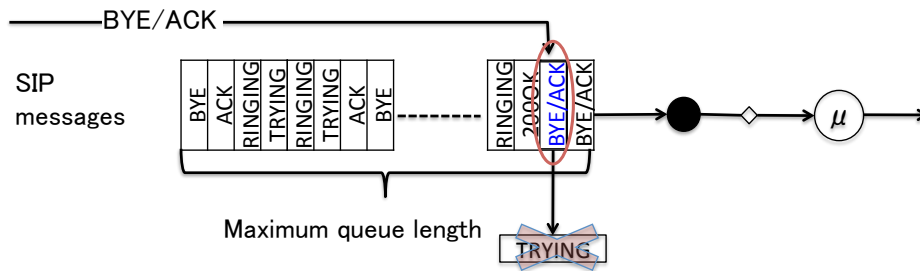


図 4.6: ダイナミックキューイング:メッセージタイプによる優先制御

メッセージタイプによる優先制御によって、完了する可能性が最も高い既存のセッションの継続性を最大限に確保できる。

4.4.3 再送メッセージに対する優先制御

輻輳時には、SIP メッセージが再送されるため、さらにスループットが低下する。最適なパフォーマンスを維持するためには、再送 SIP パケットの優先順位について検討する必要がある。

SIP メッセージの再送機構

SIP プロトコルは、TCP または UDP をトランスポート・プロトコルに用いることができる。一般に、UDP 上に実現される場合が多く、UDP は信頼性が不十分なので、SIP には再送機構がある。RFC3261 [3] によると、‘INVITE’ メッセージの再送機構は：

1. 再送信タイマー ($T1$) は 500 ミリ秒を基本値として持つ。つまり、‘INVITE’ メッセージが転送されて $T1$ 経過後 ‘100Trying’, ‘180Ringing’, 又は ‘200OK’ メッセージが受信できない場合、‘INVITE’ メッセージを再送する。
2. 再送信タイマーを 2 倍にする手続きになっている。
3. 以下同様であり、上記の応答を受信できるか、または 6 回目の再送後 $64 \times T1$ 秒のタイマーが経過するまで、再送処理が継続される。

したがって、最初に送信した ‘INVITE’ パケットの応答がない場合、RFC3261 [3] に基づいて、最初のメッセージ送信時刻を起点とすると、0.5 秒後、1.5 秒後、3.5 秒後、7.5 秒後、15.5 秒後と 31.5 秒後の計 6 回の再送が行われる。

一方、‘non-INVITE’ の場合は：

1. 再送信タイマー ($T1$) は 500 ミリ秒を基本値として持つ。
2. $T1$ からスタートし、タイムアウトごとに 2 倍にするが、4 秒を最大タイムアウト値に持つタイマー $T2$ で再送制御される。

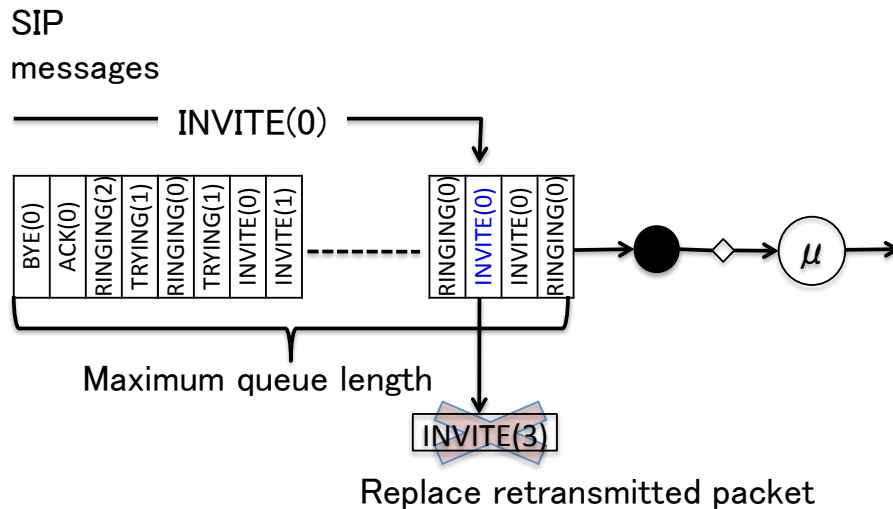


図 4.7: ダイナミックキューイング:再送メッセージの置換

3. 以下同様であり、上記の応答を受信できるか、最初の送信後 10 回再送するか、または、対応する応答を受信したとき再送を中止する。

したがって、‘non-INVITE’ パケットの場合、パケットの再送は、最初のメッセージ送信時刻を起点とすると、0.5 秒後、1.5 秒後、3.5 秒後、7.5 秒後、11.5 秒後、15.5 秒後、19.5 秒後、23.5 秒後、27.5 秒後と 31.5 秒後の計 10 回実施される。

再送回数に基づいた置換手続き

上記の SIP プロトコルの再送動作を考慮し、再送回数の多いメッセージの優先度を下げることにした。その根拠は以下のように説明することができる。例えば、3 回再送済みの ‘INVITE’ メッセージを新着の ‘INVITE’ メッセージと置換する場合、可能な最大再送回数はそれぞれ、3 と 6 になる。潜在的な再送回数は、常に新着メッセージより、再送されたメッセージの方が少なく、最終的に完了する確率が低い。

図 4.7 に、一例を示す。キューが満杯の時、キューの先頭から探索し、最初に見つけた ‘INVITE’ メッセージはすでに 3 回再送されたことがあり、受信 ‘INVITE’ メッセージより優先度が低いので、置換される。

4.4.4 ダイナミックキューイングの手続き

音声通話とインスタントメッセージング等のマルチメディア通信環境を想定し、上記の事情を考慮し図 4.8 に示す制御手法を提案する。

ここで、 p は SIP メッセージ、 $priority(p)$ は p の優先度、 $retrans(p)$ は p の再送回数とする。

1. キューに空きがあれば、着信したパケット p は最後尾にエンキューされる。

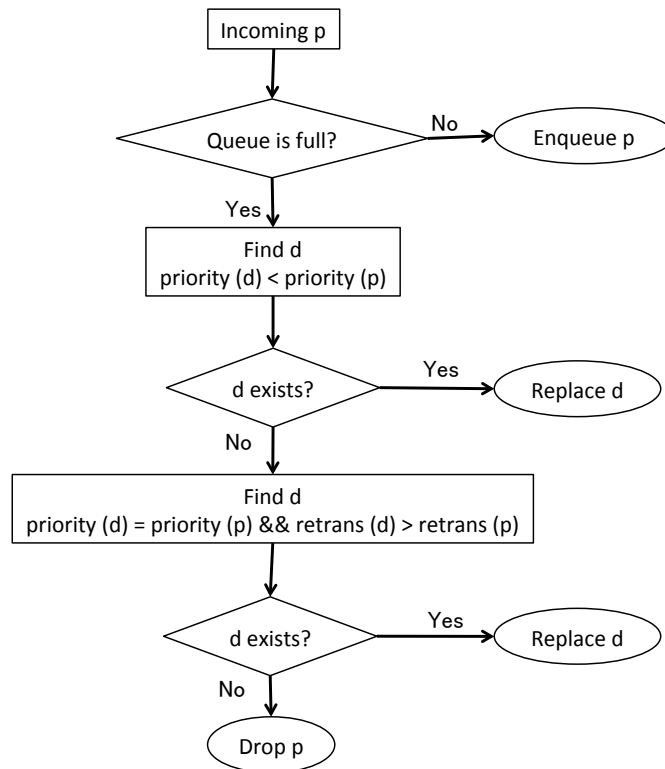


図 4.8: ダイナミックキューイングのアルゴリズム

2. キューが満杯の時は、より優先順位の低い既存のメッセージ d をキューの先頭から探す。
3. メッセージ d が検出された場合は、メッセージ d を p で置換する。
4. もし、メッセージ d が検出されない場合は、より大きな再送回数を持つ同じタイプのメッセージ d をキューの先頭から探す。
5. メッセージ d が検出された場合は、メッセージ d を p で置換する。
6. 適切な p が見つからない場合はメッセージ p を破棄する。

4.5 おわりに

本章では、IMS のシグナリング制御のための発信規制手法について検討した。既存の 2 クラスのプライオリティキューイングの優先制御手法の設計例と問題点を説明しつつ、ダイナミックキューイング構造と呼ぶ改善方法を提案した。次章から、提案したダイナミックキューの NS2 シミュレータへの実装について述べる。

第5章

ダイナミックキューのNS2シミュレータへの実装

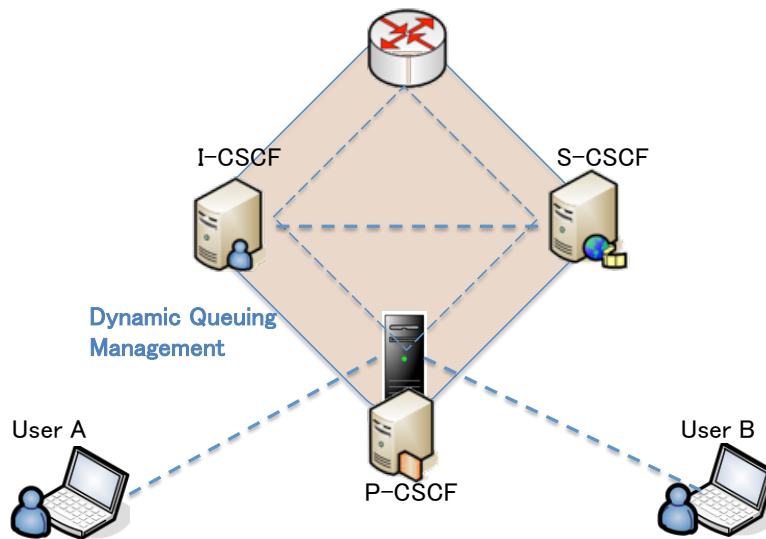


図 5.1: ダイナミックキューの導入個所

5.1 はじめに

本章では、IMS のシグナリングプレーンでのステートレス発信規制のために第 4 章で設計したダイナミックキューイングの実装の詳細を説明する。最初に、提案手法の NS2 上での実装場所を示し、ダイナミックキューイングをサポートするために必要な変更について述べる。最後に、NS2 シミュレータの変更モジュールを全て示すことにする。

5.2 IMS コアネットワーク上でのダイナミックキューイング実装個所

第 4 章では、IMS のマルチメディアサービスの自動発信規制のソリューションとしてダイナミックキューイングを提案した。しかし、実装面で最初の考慮事項は実装場所である。提案手法を観察すると、二つの特徴を持つことが分かる。

1. この提案の利点は、軽負荷かつ人の介入なしで自動的に発信規制を達成できることである。提案手法はステートレスなシグナリングの優先順位に基づいているので、特定の P-CSCF にインストールする必要はない。IMS コアコンポーネントのフロントエンドに置かれた特定のロードバランサに実装して着信 SIP メッセージを処理すればよい。
2. 第 2 に、このダイナミックキューは従来の IMS コアネットワークに費用対効果を考えてインストールしていく必要がある。実装上のコストと得られるパフォーマンス間のトレードオフを考慮すると、特定のサーバーまたは SIP パケットを処理するロードバランサとしてインストールすることもできる。

以上の点から考慮し、本研究は図 5.1 に示すように、Proxy-CSCF の前を実装の場所として、提案する。

5.3 必要なプログラム修正箇所

次に、必要な修正点を検討した。

提案手法を実現するには、今の RFC3261 [3] に含まれていないいくつかの情報が必要である。要約すると、次の3つがある。

1. まず、SIP メッセージの種類に基づいて優先レベルを設計する。各到着パケット p がどのように処理されるかは、その SIP メッセージタイプに付随する優先度によって異なる。したがって、提案したダイナミックキューイング構造を構築し、利用するためには、SIP パケットのヘッダ内の SIP メッセージの種類の情報を取得する必要がある。
2. 第二に、RFC3261 [3] によると、再送回数を示すための特定のフィールドは SIP パケットヘッダに定義されていない。提案手法を実現するためには、SIP のヘッダにこの情報を追加する必要がある。ユーザ端末は再送を扱うので、ダイナミックなキューイングをサポートするためには、ユーザ端末のソフトウェアを修正しなければならない。
3. 第三に、サービスが異なっても、1つの要求を完了するには、同じ SIP メッセージの種類を用いることが分かる。例えば、‘200OK’ は、音声通話セッションの確立期間中にユーザ B から応答する。また、ユーザ B からの正常な応答として、ページャモードインスタントメッセージングでも送信される。同様に、一つのリクエストシーケンス中セッションモードインスタントメッセージングと音声コールセッションは、少なくとも5つの同じ SIP メッセージを共有している。あるサービスに優先順位を付与するためには、単に SIP メッセージのタイプを考えるだけでなく、サービスの種類も考慮しなければならない。

実際に IMS コアネットワークへ提案手法を実現するには、上記の検討を踏まえた実装が必要である。本研究では過負荷状態の時、提案したダイナミックキューイング構造がどのようなふるまうかを観察する為に、NS2 シミュレータを採用した。そのような設計を、NS2 シミュレータ上で、具体的に実装する方法を次に示す。

5.4 NS2 シミュレータ上での実装方法

上記のようにいくつかの課題を解決する必要があるが、この論文では、シミュレーションをするために、Michele Luca Fasciana が開発した ‘SIP Module for Network Simulator 2’ [33] を用いた。このプログラムは、SIP サーバや SIP クライアントの SIP プロトコルのステートマシンを実装しているので、改修対象として最適と判断した。

5.4.1 UDP ベースの SIP 実装

SIP プロトコルは TCP, UDP, および SCTP に基づくことができるが、ここでは、現実的にはほとんどのケースが UDP なので、UDP 上で SIP 動作させることにした。

```

struct hdr_SIP {
    // type of SIP message
    int type;
    // SIP session sequence number
    int sipseqno;
    //who send?
    int from;
    //who receive?
    int to;

    // priority of current packet
    int priority;
    // retransmitted times of current packet
    int retrans;

    // Packet header access functions
    static int offset_;
    inline static int& offset() { return offset_; }
    inline static hdr_SIP* access(const Packet* p) {
        return (hdr_SIP*) p->access(offset_);
    }
};
// SIP end

```

図 5.2: SIP ヘッダの Struct

SIP パケットヘッダ

Michele Luca Fasciana [33] は, SIP パケットのヘッダを *hdr_SIPstruct* で定義した. しかし, この SIP パケットのヘッダには, 優先レベル, または再送回数などの必要な情報が含まれていない. 図 5.2 に, SIP パケットのヘッダの構造体を示す.

ダイナミックなキューイング構造を実現するために, 本研究は二つの変数, *priority* と *retrans*, を追加した. ヘッダ内のすべてのフィールドの説明は次のとおりである.

1. SIP メッセージの種類は *type* に格納する.
2. 現在のコールセッションのシーケンス番号は *sipseqno* に格納する. この値は新しいコールセッション毎に増加される.
3. 送信側のノード番号は *from* に格納する.
4. 受信側のノード番号は *to* に格納する.
5. 優先レベルは *priority* に格納する.
6. 再送回数は *retrans* に格納する.

SIP メッセージタイプと優先順位

‘SIP Module for Network Simulator 2’ [33] は, 音声呼の処理だけしか実装していなかったため, ページャモードインスタントメッセージングに対応するために, 2つの SIP メッセージ, ‘MESSAGE’

```

#define INVITE 0
#define TRYING 1
#define RINGING 2
#define OK 3
#define ACK 4
#define BYE 5
#define BYEACK 6
#define MESSAGE 7
#define IMOK 8

```

図 5.3: SIP メッセージタイプの定義

```

#define P_INVITE 0
#define P_TRYING 1
#define P_RINGING 2
#define P_OK 3
#define P_ACK 4
#define P_BYE 5
#define P_BYEACK 6
#define P_MESSAGE 0
#define P_IMOK 7

```

図 5.4: SIP メッセージの優先順位の定義

と 'IMOK' を追加した.

図 5.3 は SIP メッセージタイプの定義を示す.

また, 提案した SIP メッセージの優先度を実現するために, SIP メッセージタイプに対して図 5.4 に示すような複数のプライオリティレベルを定義した.

そして, 実際に SIP メッセージを送信する時, 呼び出した関数の中で, SIP メッセージの種類を変数 *type* に, SIP メッセージの優先度レベルを変数 *priority* にセットする.

図 5.5 は 'BYE/ACK' メッセージを送信する例を示す.

```

void SIP_UDP_Traffic::send_ack_bye_msg(int seqToSend, int from, int to, int retrans)
{
    hdr_SIP sh_buf_send;
    sh_buf_send.type = BYEACK;
    sh_buf_send.sipseqno = seqToSend;
    sh_buf_send.from = from;
    sh_buf_send.to = to;
    sh_buf_send.priority = P_BYEACK;
    sh_buf_send.retrans = retrans;

    agent-> sendmsg (PACKET_SIZE_NON_INVITE, (char*) &sh_buf_send);
}

```

図 5.5: SIP メッセージタイプと優先順位をセットする

```

switch (retrans++) {
    case 0:
        resched(1.0);
        this->interval=1.0;
        break;
    case 1:
        resched(2.0);
        this->interval=2.0;
        break;
    case 2:
        resched(4.0);
        this->interval=4.0;
        break;
    case 3:
        resched(8.0);
        this->interval=8.0;
        break;
    case 4:
        resched(16.0);
        this->interval=16.0;
        break;
    case 5:
        resched(0.5);
        this->interval=MAX_INVITE_INTERVAL;
        break;
    default:
        if (this->status()==TIMER_PENDING) {
            this->cancel();
        }
        this->next_sip = EXPIRE;
        return;
}

t->send_invite_msg_(this->seq, this->from, this->to, this->retrans);

```

図 5.6: ‘INVITE’ の再送処理のプログラム

SIP メッセージの再送

また, RFC3261 [3] に基づいて, ‘SIP Module for Network Simulator 2’ [33] に SIP の標準的再送機構を実装した.

‘SIP Module for Network Simulator 2’ [33] の再送機構は, 標準の RFC3261 [3] に沿っていない. このため, できるだけ標準に近いとシミュレーションを行うために, ‘INVITE’ と ‘non-INVITE’ メッセージに対して, SIP の標準的再送機構を実装した.

図 5.6 は ‘INVITE’ メッセージの再送機構を実現する関数に修正したコード部分を示す.

1. 最初の再送のためのデフォルトのタイムアウトは 0.5 秒である.
2. その後, 毎回再送するたびに, タイムアウト間隔が倍になる. そして同時に, 再送回数が増加し, その値が SIP パケットヘッダ内の *retrans* に格納される.
3. 合計期間が 32 秒を超えると, 再送信が中止される. 現在の音声コールセッションのステータスは ‘EXPIRE’ としてマークされる.

```

switch (retrans++) {
    case 0:
        resched(1.0);
        this->interval=1.0;
        break;
    case 1:
        resched(2.0);
        this->interval=2.0;
        break;
    case 2:
        resched(4.0);
        this->interval=4.0;
        break;
    case 3:
    case 4:
    case 5:
    case 6:
    case 7:
    case 8:
        resched(4.0);
        break;
    case 9:
        resched(0.5);
        break;
    default:
        if (this->status()==TIMER_PENDING) {
            this->cancel();
        }
        this->next_sip = EXPIRE;
        return;
}
t_>send_200ok_msg_(this->seq, this->to, this->from, this->retrans);

```

図 5.7: ‘non-INVITE’ の再送処理のプログラム

一方、全ての‘non-INVITE’メッセージは、同じ再送機構を共有している。図 5.7 は‘200OK’メッセージの再送機構を実現する関数に修正したコード部分を示す。

1. 最初の再送のためのデフォルトのタイムアウトは 0.5 秒である。
2. 間隔が 4 秒に達するまで、毎回再送するたびに、タイムアウト間隔が倍になる。その後、再送間隔は 4 秒に維持される。再送回数が増加し、その値が SIP パケットヘッダ内の *retrans* に格納される。
3. 合計期間が 32 秒を超えると、再送信が中止される。現在の音声コールセッションのステータスは‘EXPIRE’としてマークされる。

5.4.2 サービスモジュールの追加

元の‘SIP Module for Network Simulator 2’ [33] は、音声通話サービスしか提供していない。しかし、本研究は複数のメディアサービスを前提に発信規制手法を提案するので、音声通話サービスだけでなく、インスタントメッセージングサービスも含まれるべきである。そして、ここでは音声通話サービスを参照し、他の 2 つのサービスモジュールも追加開発した。

```

#ifndef ns_Dynamic_h
#define ns_Dynamic_h

#include <string.h>
#include "queue.h"
#include "config.h"
#include "udp_sip.h"

class Dynamic : public Queue {
public:
    Dynamic() {
        q_ = new PacketQueue;
        pq_ = q_;
        bind_bool("drop_front_", &drop_front_);
        bind_bool("summarystats_", &summarystats);
        bind_bool("queue_in_bytes_", &qib_); // boolean: q in bytes?
        bind("mean_pktsize_", &mean_pktsize_);
        bind("queuelength_otcl", &queuelength);
    }
    ~Dynamic() {
        delete q_;
    }
protected:
    void reset();
    int command(int argc, const char*const* argv);
    void enqueue(Packet*);
    Packet* deque();
    void shrink_queue(); // To shrink queue and drop excessive packets.
    PacketQueue *q_; /* underlying FIFO queue */
    int drop_front_; /* drop-from-front (rather than from tail) */
    int summarystats;
    void print_summarystats();
    int qib_; /* bool: queue measured in bytes? */
    int mean_pktsize_; /* configured mean packet size in bytes */
    int queuelength; /*create new variable for set buffersize*/
};
#endif

```

図 5.8: ダイナミックキューのヘッダファイル

5.4.3 ダイナミックキューイングモジュール

UDP モジュール上で SIP を実装し、新たなアプリケーションサービスを追加作成した後、次に、ダイナミックキューイング構造の実装に移った。

既存の *Droptail* キューモジュールを参考することによって、本研究はダイナミックキューイングをプログラムした。また、ダイナミックキューイングのキュータイプとして、*Dynamic* という名前を定義する。

図 5.8 は *Dynamic* のヘッダファイルを示す。

Droptail キューとの主な違いは、エンキューの処理である。エンキューの部分のソースコードを図 5.9 で示す。ここでは、パケット *p* が来たときの処理を説明する。

1. `hdr_SIP::access(p)` を呼び出すによって、ヘッダ情報を得る。
2. `q->len<=queuelength` を判断して、現在のキューがいっぱいになっていない場合、パケッ

```

void Dynamic::enqueue(Packet* p)
{
    if (summarystats) {  
        //get head of packet  
        hdr_SIP *sip_udp_h=hdr_SIP::access(p);  
        if (q_len_>0){  
            if (q_->len_ <= queuelength) {  
                q_->enqueue(p); //put into the end of queue  
            } else{  
                while(rd_cur){  
                    hdr_SIP *tmp=hdr_SIP::access(rd_cur);  
                    if (tmp->priority < sip_udp_h->priority)  
                    {  
                        rd_lp_drop = rd_cur;  
                        rd_lp_pre = rd_pre;  
                        break; //jump out of the loop when the packet to be replaced is found  
                    } else if (rd_rt_drop == NULL && ( (tmp->priority == sip_udp_h->priority) && (sip_udp_h->retrans < tmp->retrans) )){  
                        rd_rt_drop = rd_cur;  
                        rd_rt_pre = rd_pre;  
                    }  
                    rd_pre = rd_cur;  
                    rd_cur = rd_cur->next_;  
                }  
                if (rd_lp_drop !=NULL) {  
                    rd_drop = rd_lp_drop;  
                    rd_pre = rd_lp_pre;  
                } else if (rd_rt_drop != NULL){  
                    rd_drop = rd_rt_drop;  
                    rd_pre = rd_rt_pre;  
                } else {  
                    drop(p);  
                    return;  
                }  
                if(rd_str==rd_end) {  
                    rd_drop->next_=0;  
                    drop(rd_drop);  
                }  
            }  
        }  
    }
}

```

図 5.9: ダイナミックキューのプログラム

ト p は、エンキューされる。

3. もし、現在のキューがいっぱいになっている場合、while ループに入り、キュー q の先頭から置換できる候補パケット d を探す。
4. $tmp \rightarrow priority$ と $sip_udp_h \rightarrow priority$ を比べて、現在のパケット tmp の優先度が p より低い場合、置換対象パケットとして選び、ループを終了する。
5. もし、現在のパケット tmp の優先度が p と同じ場合、 $tmp \rightarrow retrans$ と $sip_udp_h \rightarrow retrans$ を比べる。現在のパケット tmp の再送回数が p より多い場合、置換対象パケットの候補としてマークして、次のパケットへ移る。
6. while ループを終わった後、 d が見つかったかどうかを確認する。適切な d が見つかった場合はパケット d を p で置換し、見つからない場合はパケット p を破棄する。

5.5 おわりに

本章では、第4章で設計したダイナミックキューイングの実装の詳細を説明した。提案手法の NS2 上での実装場所と、ダイナミックキューイングをサポートするために必要な変更点と NS2 シミュレータの変更モジュールについて述べた。次章で、実験環境と実験結果について述べる。

■ 第6章

評価

6.1 はじめに

本章では提案したダイナミックキューイングという自動発信規制手法を評価する。まず、シミュレーションのネットワークモデルを紹介し、性能が観察された3つのキューイング構造の特徴を比較してまとめる。そして、設計したシミュレーションシナリオを説明し、実験環境と実験結果について述べ、結果に対する考察をまとめる。

6.2 ネットワークモデル

本論文は、過負荷トラヒックに対して、IMS サービスのスループットを維持するための自動発信規制を実現する手法を提案している。

シミュレーションを行うためには、まず、ネットワークモデルを決定する必要がある。提案するダイナミックキューイングの比較対象として、シミュレーションのネットワークモデルに関しては Ohta の論文に基づいた [4]。

図 6.1 はオーバーロード時のネットワークモデルを示す。予期しない地震などの自然災害、または祝日やスポーツ大会などのイベントによって、非常に過負荷なトラヒックが発生したというシナリオである。

1. 全部で $(n + 1)$ のドメインがあるとする。
2. 各ドメインには、一つの SIP プロキシサーバー (Proxy-CSCF) とルータが装備されている。
3. 最初の n 個のドメインは、それぞれ m 個 SIP ユーザーエージェント (UA) ヘサービスを提供している。
4. ボトルネックドメインとして BD を表示されている最後のドメインには $n \times m$ 個の SIP ユーザーエージェント (UA) ヘサービスを提供している。
5. 最初の n 個のドメインのルータが直接 BD のルータと接続されているとする。
6. 最初の n 個のドメインに属するそれぞれのユーザは、 BD の特定のユーザと連絡しようとしているものとする。
7. その結果として、 BD の SIP プロキシサーバー (Proxy-CSCF) で過負荷が発生した。

6.3 3つのキューイング構造

本研究は3つのキューイング構造に基づいてシミュレーションを行った。それぞれ、シングルキューイング、2クラスキューイングとダイナミックキューイング構造と呼ぶ。

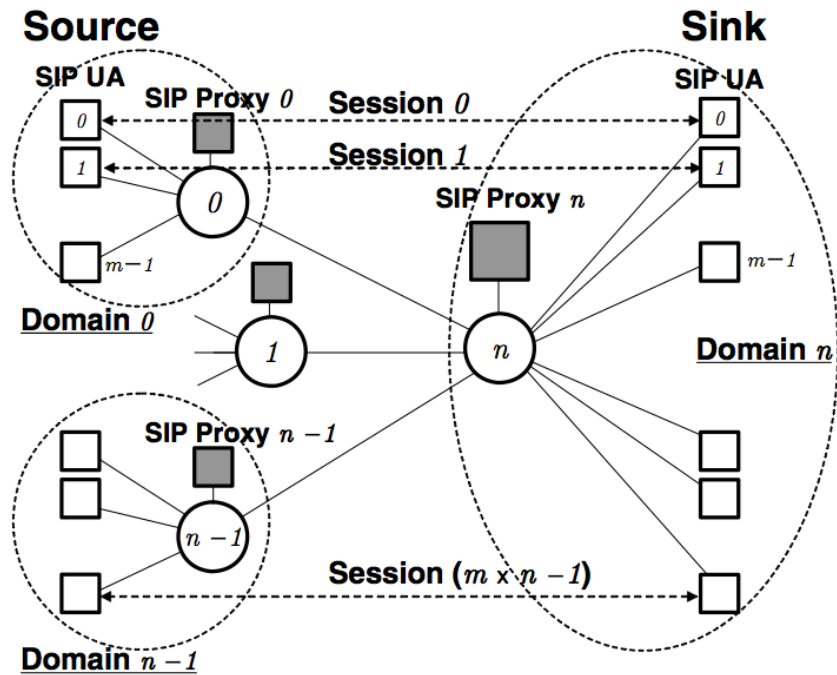


図 6.1: 過負荷時のネットワークの動作を説明するモデル [4]

6.3.1 シングルキューイング構造

シングルキューイング構造は最も一般的なものであり、次のような特徴がある。

1. その名前が示すように、すべてのパケットを1つの待ち行列で処理する。
2. パケットは First In, First Out (FIFO) のスタイルで処理される。
3. キューが満杯の時は、着信パケットを自動的に破棄する。

シミュレーションを行う時、シングルキューイングの場合は、DropTail を用いている。

6.3.2 2クラスキューイング構造

2クラスキューイングは音声の発信規制を行うために、Ohta [4] が提案した、2つのキューを使った発信規制の自動化手法である。その詳細は4章で説明した。提案するダイナミックキューイングの比較対象として、性能を評価する。要約すると、主な特徴は以下である。

1. その名前が示すように、特定の長さを持つ2つのキューがある。
2. SIP メッセージを 'INVITE' と 'non-INVITE' に分類し。この分類された SIP メッセージそれぞれは特定のサイズのキューを持つ2つのキューに格納する。
3. 'non-INVITE' キュー内の SIP メッセージは高い優先度で処理される。'non-INVITE' キューが空であるの場合にのみ、'INVITE' キューが処理される。

```

#node of SIP proxy server
#R1,R2,R3 receiver for voice call, PMIM, SMIM
set R1 [$ns node]
set R2 [$ns node]
set R3 [$ns node]
#S1, S2, S3 sender for voice call and PMIM and SMIM
set S1 [$ns node]
set S2 [$ns node]
set S3 [$ns node]
set P_0 [$ns node]
set P_1 [$ns node]
set P_2 [$ns node]

```

図 6.2: ノードの定義をする NS2 の Tcl スクリプト

4. パケットは First In, First Out (FIFO) のスタイルで処理される.
5. キューが満杯の時は, 着信パケットを自動的に破棄する.

2 クラスキューイングは既存の ns2 キュータイプではないので, 提案するダイナミックキューイングの比較対象としてシミュレーションを実験するために, Ohta の論文に基づき [4], 2 クラスキューイング方式を実装した.

6.3.3 ダイナミックキューイング構造

ダイナミックキューイングは過負荷トラヒックに対して, IMS サービスのスループットを維持するための自動発信規制手法として提案した. その設計と実装の詳細については, 第4章と5章で説明した. ここにその主な概要を示す.

1. すべての SIP メッセージが同じキューを共有する. キューは原則 FIFO で動作する.
2. パケットは, 各 SIP メッセージの種類に応じて優先順位付けされる.
3. 再送回数の多いパケットの優先度を下げることにする.
4. キューが満杯の時は, 第4章で説明したアルゴリズム手続きに基づいて, 置換を行う.
5. より優先順位の低い既存のパケットをキューの先頭から探す.
6. 適切な既存のパケットが見つからない場合は, 着信したパケットを破棄する.

シミュレーションを実験するために, 説明したアルゴリズム手続きに基づいて, ダイナミックキューイング方式を実装した.

6.4 シミュレーションのシナリオ

ネットワークモデルおよびキューイング構造を決定した後, 過負荷シナリオに基づいて, NS2 シミュレーションのための Tcl スクリプトを作成した.

図 6.2 にノードの定義をする NS2 の Tcl スクリプトを示す.

```

#link transmission delay
$ns duplex-link $S1 $P_0 100Mb 0ms Dynamic
$ns duplex-link $S2 $P_0 100Mb 0ms Dynamic
$ns duplex-link $S3 $P_0 100Mb 0ms Dynamic
#link for middle P_0 -> P_1 very fast
$ns simplex-link $P_0 $P_1 100Gb 0ms Dynamic
#link processing delay
$ns simplex-link $P_1 $P_2 2.25Mb 0ms Dynamic
#set link for middle R1 -> P_2 very fast
$ns simplex-link $P_2 $R1 100Gb 0ms Dynamic
$ns simplex-link $R1 $P_1 100Gb 0ms Dynamic
$ns simplex-link $P_2 $R2 100Gb 0ms Dynamic
$ns simplex-link $R2 $P_1 100Gb 0ms Dynamic
$ns simplex-link $P_2 $R3 100Gb 0ms Dynamic
$ns simplex-link $R3 $P_1 100Gb 0ms Dynamic
$ns simplex-link $P_2 $P_0 100Gb 0ms Dynamic

```

図 6.3: リンクの定義をする NS2 の Tcl スクリプト

1. $R1$, $R2$ と $R3$ は音声, PMIM と SMIM コールの受信側のノードとして定義される. またそれぞれは, UDP ベースの SIP エージェントと接続される.
2. $S1$, $S2$ と $S3$ は音声, PMIM と SMIM コールの送信側のノードとして定義される. またそれぞれは, UDP ベースの SIP エージェントと接続される.
3. P_0 , P_1 と P_2 はパケットを転送するために設計された中間ノードである.

受信側のエージェントすべての音声とインスタントメッセージングの要求への応答を行う. 転送や処理の遅延のパラメータは [4] に準じた. 図 6.3 はリンクの定義する NS2 Tcl スクリプトを示す.

1. 転送の遅延は $S1$, $S2$, $S3$ と P_0 間のリンク上に生成される.
2. 処理の遅延は P_1 と P_2 間のリンク上に生成される.
3. 他のリンクの帯域幅は伝送遅延と処理遅延と比べて, 非常に大きいので, 他の遅延時間は無視することができる.

ノードとリンクを定義した後, 図 6.4 に示すように, UDP ベースの SIP エージェントをそれぞれのノードに接続する

本提案は, IMS 内の複数のメディアサービスに基づいているため, トラヒックの生成には, 音声とセッションモードインスタントメッセージングとページャモードインスタントメッセージングに対応した 3 つの送信サービスを設けた.

図 6.5 は複数の受信側のメディアサービスを定義する NS2 Tcl スクリプトを示す. 受信側のエージェントすべての音声とインスタントメッセージングの要求への応答を行う.

1. sip_1 , sip_2 と sip_3 は音声, PMIM と SMIM トラヒックを受信するアプリケーションである. それぞれは, 以前に定義された SIP エージェントに接続される.
2. 上記トポロジによると, すべてのパケットが P_1 と P_2 間を通過するので, それらの 2 つのノード間のリンクが, 全体のシナリオではボトルネックとなる.

```
#R1, R2, R3 receiver proxy
set udp_1 [new Agent/UDP/UDPSIP]
$ns attach-agent $R1 $udp_1
set udp_2 [new Agent/UDP/UDPSIP]
$ns attach-agent $R2 $udp_2
set udp_3 [new Agent/UDP/UDPSIP]
$ns attach-agent $R3 $udp_3
#S1, S2, S3 sender proxy
set udp_4 [new Agent/UDP/UDPSIP]
$ns attach-agent $S1 $udp_4
$ns connect $udp_1 $udp_4
set udp_5 [new Agent/UDP/UDPSIP]
$ns attach-agent $S2 $udp_5
$ns connect $udp_2 $udp_5
set udp_6 [new Agent/UDP/UDPSIP]
$ns attach-agent $S3 $udp_6
$ns connect $udp_3 $udp_6
```

図 6.4: ns2 ノード間の接続

```
#Application for Voice call Receiver
set sip_1 [new Application/Traffic/SipVoice]
$sip_1 set from_ 1
$sip_1 set to_ 0
$sip_1 attach-agent $udp_1

#Application for PMIM Receiver
set sip_2 [new Application/Traffic/SipPMIM]
$sip_2 set from_ 2
$sip_2 set to_ 0
$sip_2 attach-agent $udp_2

#Application for SMIM Receiver
set sip_3 [new Application/Traffic/SipSMIM]
$sip_3 set from_ 3
$sip_3 set to_ 0
$sip_3 attach-agent $udp_3
```

図 6.5: 複数の受信側のメディアサービスを定義する NS2 Tcl スクリプト

```
#Application for Voice call Sender
set sip_4 [new Application/Traffic/SipVoice]
$sip_4 set packetSize_ $v_size
$sip_4 set burst_time_ 0
$sip_4 set idle_time_ $v_interarrivo
$sip_4 set from_ 4
$sip_4 set to_ 1
$sip_4 attach-agent $udp_4

#Application for PMIM Sender
set sip_5 [new Application/Traffic/SipPMIM]
$sip_5 set packetSize_ $pmim_size
$sip_5 set burst_time_ 0
$sip_5 set idle_time_ $pmim_interarrivo
$sip_5 set from_ 5
$sip_5 set to_ 2
$sip_5 attach-agent $udp_5

#Application for SMIM Sender
set sip_6 [new Application/Traffic/SipSMIM]
$sip_6 set packetSize_ $smim_size
$sip_6 set burst_time_ 0
$sip_6 set idle_time_ $smim_interarrivo
$sip_6 set from_ 6
$sip_6 set to_ 3
$sip_6 attach-agent $udp_6
```

図 6.6: 複数の送信側のメディアサービスを定義する NS2 Tcl スクリプト

図 6.6 は複数の送信側のメディアサービスを定義する NS2 Tcl スクリプトを示す。送信側のエージェントすべての音声とインスタントメッセージングの要求を行う。

1. *sip_4*, *sip_5*, *sip_6* は音声, PMIM と SMIM トラヒックを送信するアプリケーションである。それぞれは, 以前に定義された SIP エージェントに接続される。
2. トラヒックの生成やパケットサイズのパラメータは, 変数 *burst_time_*, *idle_time_* と *packetSize_* によって定義される。

6.4.1 トラヒックの負荷

3つのキューイング構造の性能を評価するために, ボトルネックの SIP プロキシサーバー (Proxy-CSCF) に対して掛かるトラヒックの負荷を分析する必要がある。

受信側での音声とインスタントメッセージングのトラヒックの分析方法は, 関連研究 [4] と同様である。音声通話の SIP メッセージの平均到着率を λ_{voice} , インスタントメッセージングの SIP メッセージの平均到着率を λ_{IM} とする。すべてのリクエストに対して, 受信側 P-CSCF 側の処理速度を μ_{sip} とする。つまり, 一つの SIP メッセージは $\frac{1}{\mu_{sip}}$ 秒で処理できるとする。このとき, 負荷の総和 ρ_{total} , すなわち使用率は次式で表すことができる。

$$\rho_{total} = \frac{\lambda_{voice} + \lambda_{IM}}{\mu_{sip}} \quad (6.1)$$

また, 音声トラヒックやインスタントメッセージングトラヒックの総トラヒックに対する比率を α と $\beta = 1 - \alpha$ と想定すると, 音声トラヒックは $\alpha \cdot \rho_{total}$, インスタントメッセージングのトラヒックは $(1 - \alpha) \cdot \rho_{total}$ で表すことができる。音声, セッションモードインスタントメッセージングおよび PMIM の中で使われている SIP メッセージ数は, それぞれ 7, 5 および 2 個であるため, それぞれのスループットは, 以下のように表わすことができる。

$$M_{voice} = \frac{\alpha \cdot \mu_{sip}}{7} \quad (6.2)$$

$$M_{SMIM} = \frac{(1 - \alpha) \cdot \mu_{sip}}{5} \quad (6.3)$$

$$M_{PMIM} = \frac{(1 - \alpha) \cdot \mu_{sip}}{2} \quad (6.4)$$

6.5 スループット特性の評価

4.8 のアルゴリズムの性能を評価するために, これまで述べた 3つのキューイング方式に基づき, NS2 [12] で音声と二種類のインスタントメッセージングサービスが共存する場合に関してシミュレーションを行った。ここで, ダイナミックキューイングに関しては表に示した優先順位を用い, 2 クラスキューイングに関しては, 優先順位 1 の 'INVITE' と 'MESSAGE' が同じキューを共有するように変えて実験を行った。

スループット評価には, 2つのトラヒックパターンを使ってシミュレーションした。第 1 のトラヒックパターンは音声と PMIM が混在している場合であり, もう一つは音声と SMIM が混在した

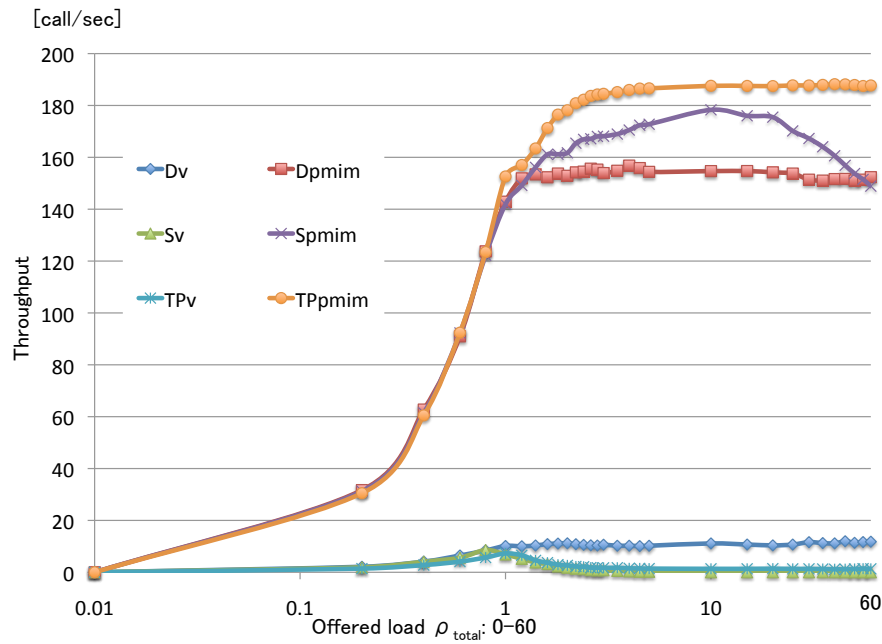


図 6.7: 音声と PMIM が 0.2 対 0.8 の時のスループット

場合である。2つのパターンは（音声，IM）の比率に関して，3つの組み合わせ (0.2,0.8)，(0.5,0.5) と (0.8,0.2) を選んで評価した。

ここで， $\frac{1}{\mu_{sip}}$ の値は，[4] と同じで 2.6 ミリ秒とする。負荷は，0 から 60 の範囲とする。この負荷の範囲は東日本大震災時のトラヒックが通常よりも 60 倍大きかった [8] ことから選んでいる。比較対象のキューの構造は 3 種類である。第 1 に，シングルキューイングで，これを比較の基準とした。第 2 は，2 クラスキューイング [4] である。第 3 が本研究の提案手法である。

なお，シミュレーションは 10 回行い，その平均値をプロットしている。

図 6.7 から 6.12 までに，3 種類のキューイングによる各トラヒック比率に対するスループットを示す。Dv, Sv および TPv はそれぞれ，提案方式，シングル，2 クラスキューイングによる音声通話のスループットを表す。Dsmim, Ssmim と TPsmim はそれぞれ，提案方式，シングルと 2 クラスキューイングによる SMIM のスループットを表す。Dpmim, Spmim と TPpmim はそれぞれ，提案方式，シングルと 2 クラスキューイングによる PMIM のスループットを表す。

シングルキューイング:

図 6.10, 6.11, 6.12 に示した Sv と Ssmim の結果では，音声と SMIM のスループットは，負荷が 0.6 を越えると急速に減少し，負荷が 10 になるとほぼ 0 に低下する。一方，図 6.7, 6.8, 6.9 に示すように，負荷が増加するにつれて，PMIM のスループットは増加した。この理由は，PMIM は 2 つだけの SIP メッセージから成るため，呼の完了確率が高く，リクエストの数が増えると完了する PMIM が増えると考えられる。

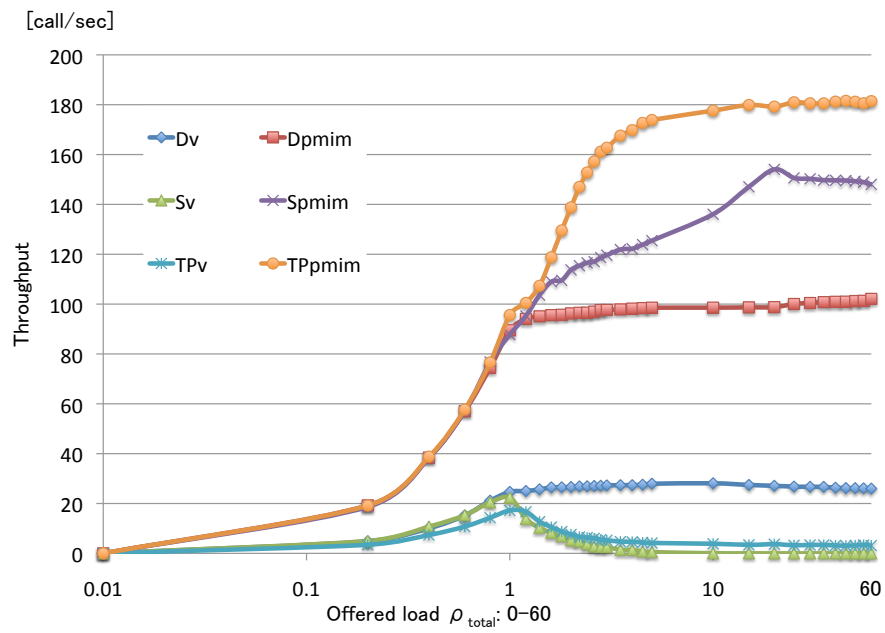


図 6.8: 音声と PMIM が 0.5 対 0.5 の時のスループット

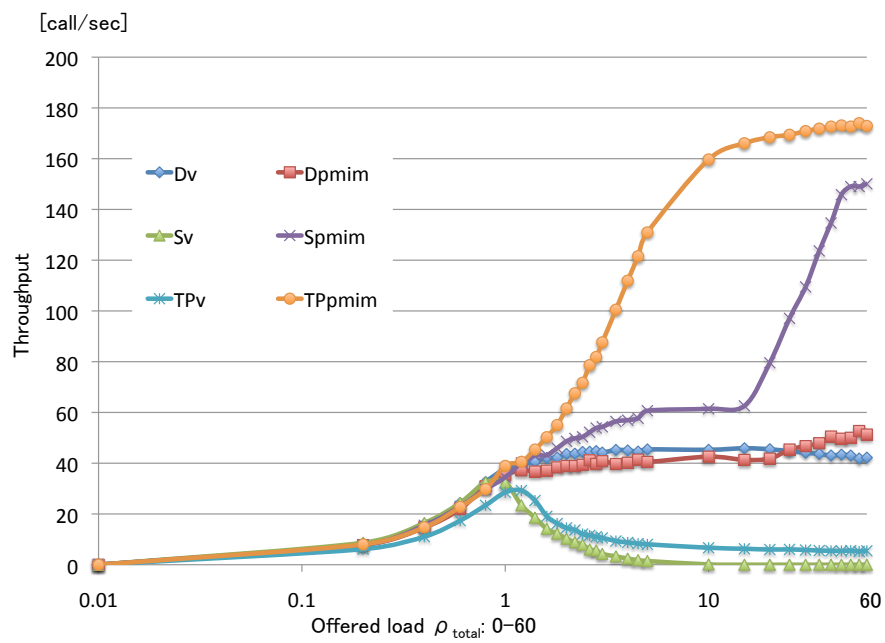


図 6.9: 音声と PMIM が 0.8 対 0.2 の時のスループット

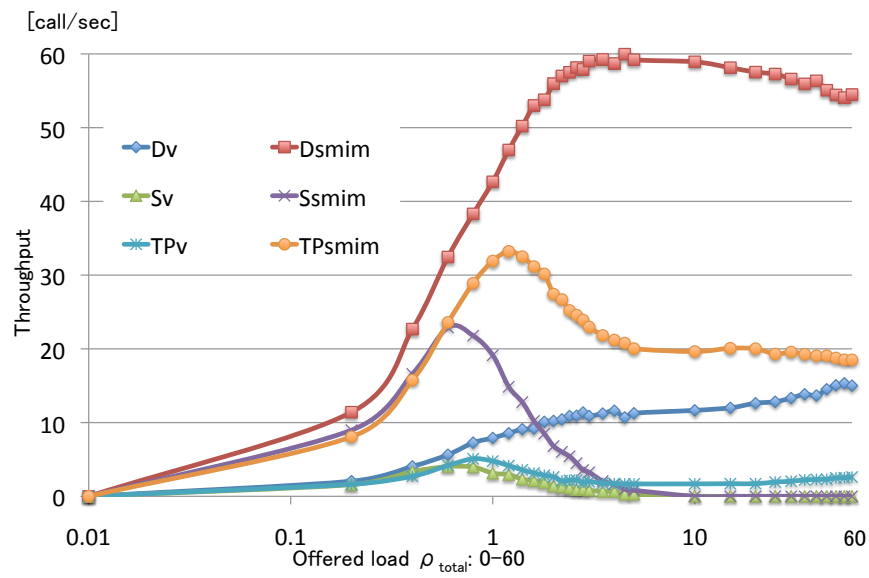


図 6.10: 音声と SMIM が 0.2 対 0.8 の時のスループット

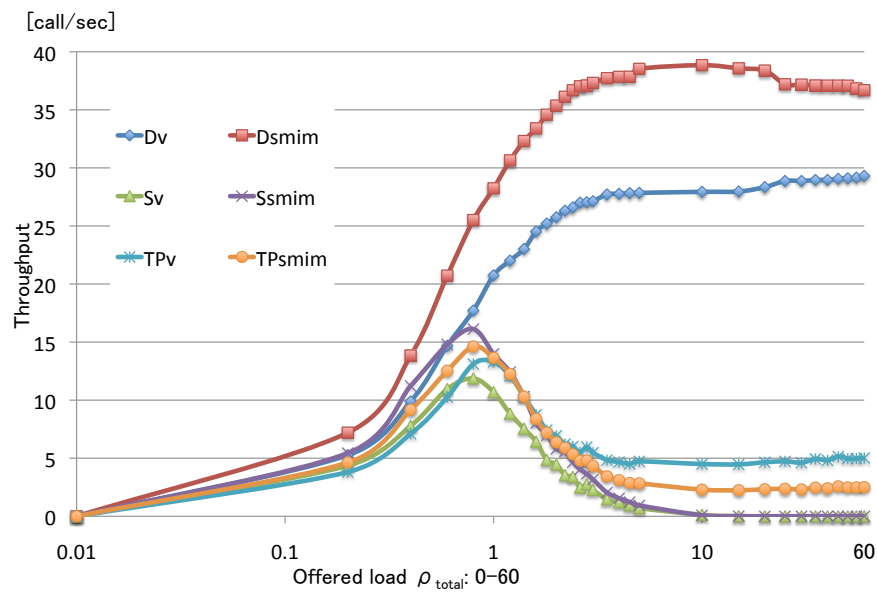


図 6.11: 音声と SMIM が 0.5 対 0.5 の時のスループット

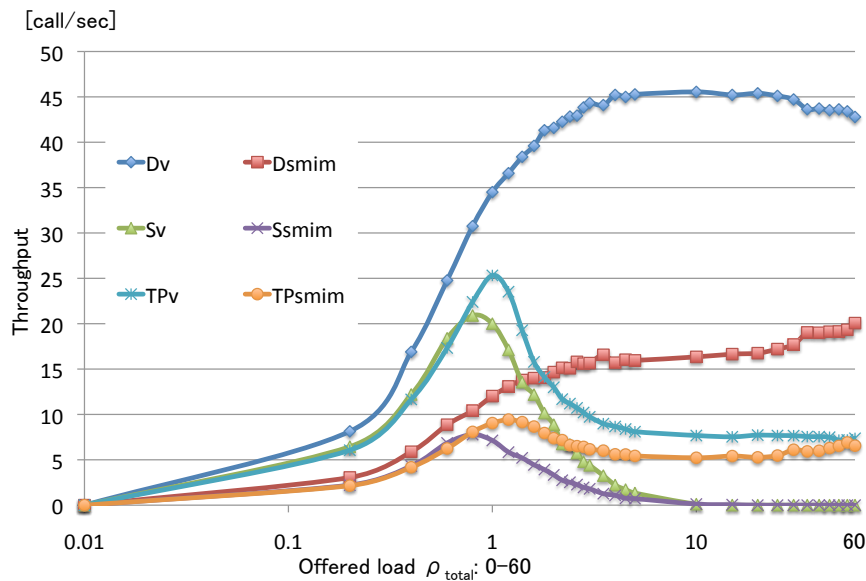


図 6.12: 音声と SMIM が 0.8 対 0.2 の時のスループット

2 クラスキューイング:

図 6.10, 6.11, 6.12 に示した TPv と TPsmim の結果では、シングルキューイングと比較すると、負荷が 1.0 近傍で音声呼の性能が良くなっている。また、負荷が大きくなっても、スループットは低下するが、一定規模の性能は維持できている。一方、図 6.7, 6.8, 6.9 の TPv と TPsmim が示す結果から、音声呼の性能は似ているが、PMIM のスループットは負荷の増加に合わせて増加し、負荷 10 以上でも増加傾向が見える。これは、表 4.1 で同じ優先順位を持つ ‘MESSAGE’ と ‘INVITE’ を同じキューに入れたが、PMIM は 2 つだけの SIP メッセージから成るため、シングルキューイングで説明したように、過負荷時は完了する確率が PMIM の方が高いからである。

ダイナミックキューイング: 図 6.10, 6.11, 6.12 に示す音声呼と SMIM の共存時に関しては、過負荷時も Dv, Dsmim はほぼ一定、もしくは増加傾向を維持している。また、図 6.7, 6.8, 6.9 に示す音声呼と PMIM の共存時に関しても、高負荷時に、それぞれの最大スループットを維持していることが分かる。負荷が 1.0 を超えた場合の Dpmim の結果からみると、PMIM のスループットは既存の 2 つのキューイングよりも小さくなっているが、音声呼のスループットは良好であり、音声サービスと資源をバランスよく配分できていることが分かった。これは、‘MESSAGE’ メッセージと ‘INVITE’ メッセージが同じ優先順位であるため、呼の開始は同一条件であること、一旦開始された呼は後のメッセージほど優先順位が高いため、音声呼も IM と比して完了確率が下がりにくいことからだと考えられる。ここで、‘IM/200OK’ メッセージに割り当てる優先順位を最大にし、音声呼より、IM の完了を優先させた効果も出ていることが分かる。

音声呼と PMIM 呼が 0.5 対 0.5 の時、発呼数 (INVITE または MESSAGE) に対する完了呼数 (Throughput) の割合を 3 種類のキューイング方式に対して求め、図 6.13 に示す。TPv と Sv は負荷が 1 以上になると急速に減少し、総発呼数の内で無駄になるものが増えることを示している。それ以外の減少速度は似ていて、図 6.8 の Throughput がほぼフラットな特性を負荷が大きい

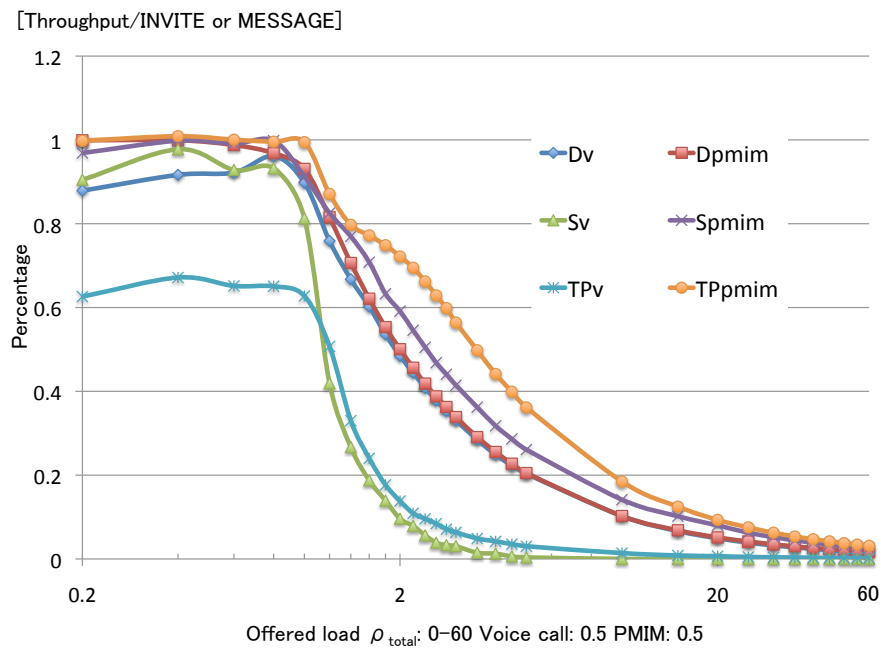


図 6.13: 発呼数 (Offered load) に対する完了呼数比率 (音声と PMIM が 0.5 対 0.5 の時)

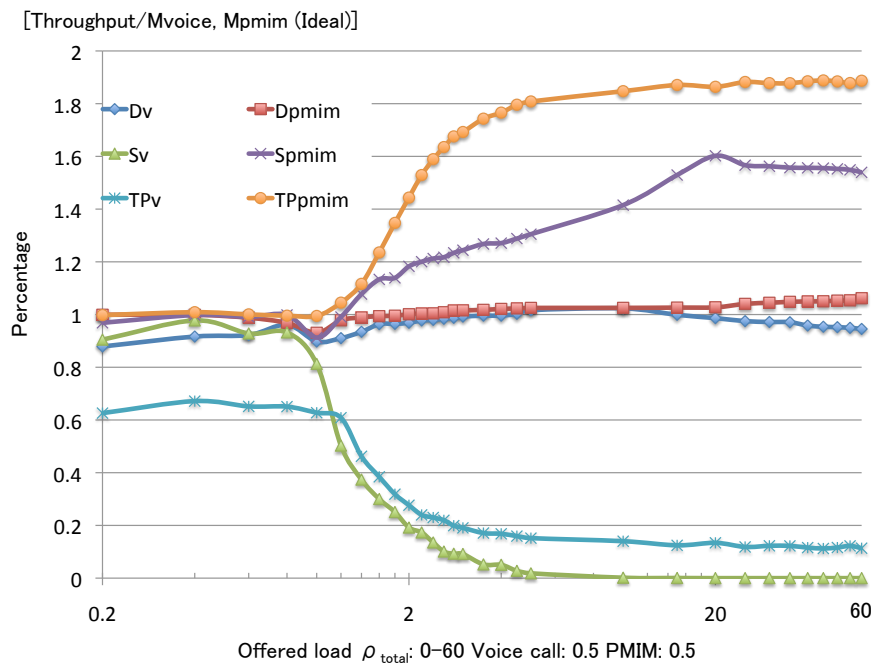


図 6.14: 実完了呼数の理想完了呼数に対する比率 (音声と PMIM が 0.5 対 0.5 の時)

表 6.1: 負荷 60 でのスループット比較

Traffic ratio	Service	Ideal	Dynamic	Single	Two-priority
0.2	Voice	10.98	11.78	0	1.33
0.8	PMIM	153.84	152.34	148.9	187.7
0.5	Voice	27.47	25.97	0	3.12
0.5	PMIM	96.15	102.1	148.02	181.43
0.8	Voice	43.95	41.84	0	5.5
0.2	PMIM	38.46	45.4	149.01	172.89
0.2	Voice	10.98	15.01	0	2.61
0.8	SMIM	61.53	54.48	0	18.51
0.5	Voice	27.47	29.31	0	5.02
0.5	SMIM	38.46	36.68	0	2.51
0.8	Voice	43.95	42.78	0	7.36
0.2	SMIM	15.38	19.03	0	6.55

時に見せていることに対応している。

また、それぞれキューイングの完了呼数 (Throughput) の理想値 (M_{voice} , M_{pmim}) に対する比率を図 6.14 に示す。ここで、ダイナミックキューイングは、負荷が 1 から 60 倍まで増加しても、それぞれのスループットはほぼ理想値に近いパフォーマンスを確保できていることが分かった。これに対して、他の 2 つの キューイングでは PMIM にリソースを割り過ぎ、音声呼のスループットが低下してしまっていることが分かる。

特に、負荷が 60 倍になった時点での、音声とインスタントメッセージングのスループットを音声と SMIM または PMIM の 3 つの組み合わせのトラフィック比率に対して計算したものを、表 6.1 に示す。ここで、Dynamic, Single, Two-priority はそれぞれ、ダイナミック、シングルおよび 2 クラスキューイングを示す。また、Ideal は M_{voice} , M_{PMIM} , M_{SMIM} から求めた理想値である。ダイナミックキューイングは、3 つのトラフィック比率全てに対して、ほぼ理想値に近いパフォーマンスを確保できることが分かった。負荷が設計時の 60 倍のような極端な場合でも、推定の最大 P-CSCF 容量に近い音声、PMIM, SMIM 呼のスループットを確保することができる。以上から、ダイナミックキューイングは音声のスループットを全く損なうことなく、小さなオーバーヘッドを持つ PMIM サービスを優先することができ、既存手法である 2 クラスキューイングやシングルキューイングより優れていることが分かる。

6.6 最適キュー長

ダイナミックキューイングのキュー長は、長いと確率的に安定した性能を期待できるが、メッセージの送受信時のタイムアウト値、および P-CSCF の処理速度を考慮し、キュー内にメッセージが滞留する時間がタイムアウト値を越えないように設計する必要がある。通常時の 60 倍の高負荷時に、10 から 10,000,000 までの異なるキュー長に対してシミュレーションを行った結果を図 6.15

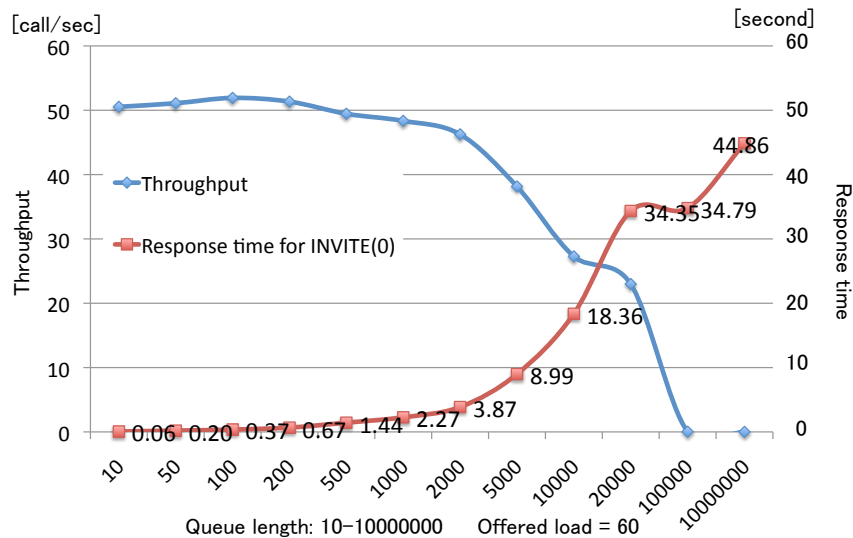


図 6.15: スループットのキュー長への依存性

に示す。

キュー長に対し、スループットと共に、最初の‘INVITE’と、それに対応する応答メッセージとの間の時間をプロットした。キュー長が2,000を越えると急速にスループットが低下するとともに、応答時間も増加し、キュー長が100,000以上になるとスループットはゼロになる。最適キュー長は、次のように求めることができる。キュー長 Q を持つキューが満杯になったとき、最後に到着したメッセージが‘INVITE’メッセージの応答だとする。これが処理されるまでの待ち時間 T は $T = \frac{Q}{\mu_{sip}}$ で表すことができる。ここで、 $T \leq 0.5$ であるから、キュー長 Q の最大値は $Q = T \times \mu_{sip} = \frac{0.5}{2.6 \times 10^{-3}} = 192.3$ であることが分かる。図 6.15 で、キュー長が200以上になると最初の‘INVITE’とそれに対応する応答メッセージ間の時間差が増え、スループットも低下し出すことが分かるが、これは最適キュー長192を越えてしまったからである。100から200までのキュー長に対しては、平均応答時間は0.5秒未満であり、かつ最大スループットを確保している。

6.7 ダイナミックキューイングの置換対象メッセージの検索方法について

もう一つ検討すべき課題は、ダイナミックキューイングで置換対象メッセージを検索する際の検索手順である。第4章で説明したアルゴリズムでは、より優先順位の低い既存パケット d はキューの先頭から探して置換される。着信パケット p で既存のパケット d を置き換えると、 p は自身より先に受信したパケット (p の次からキューの最後まで) より先に処理されることになる。これはFIFOキューの原則の合わない。原則に合わせるには、キューの最後尾からサーチして置換する必要があるが、キューの先頭からのサーチの方が、有効である。この節では、シミュレーションによ

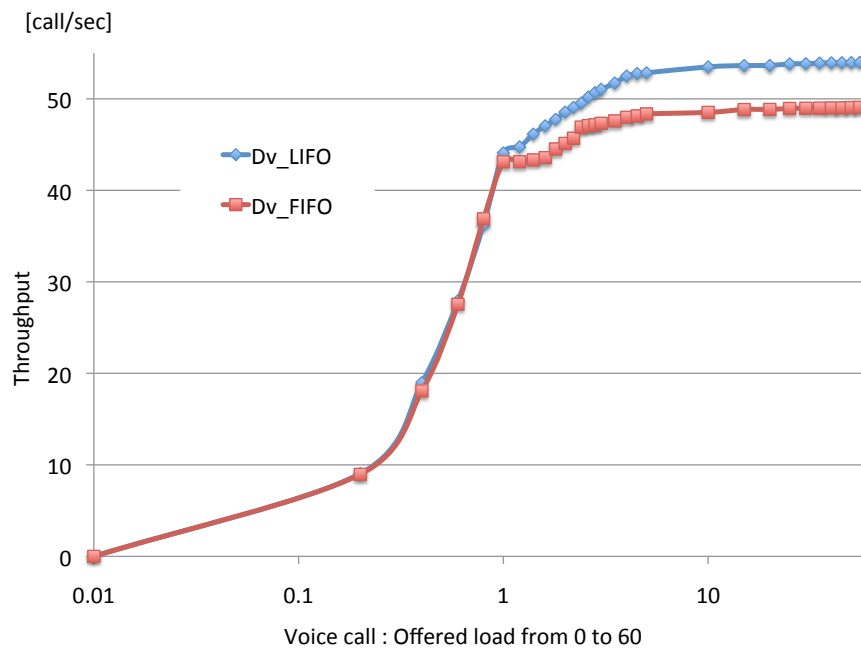


図 6.16: ダイナミックキューにおける2つの置換対象サーチ方法のスループット

りこれを裏付けることにする。

どちらの方法が有効か確認するために、両方の置換手順で音声通話のシミュレーションを行ってみた。結果を図 6.16 に示す。Dv_LIFO は、キューの先頭からサーチして置換した時の、音声通話のスループットである。また、Dv_FIFO は、キューの最後尾からサーチして置換したときの音声通話のスループットである。明らかに、Dv_LIFO のスループットの方が Dv_FIFO より高い。

図 6.16 は、キューの最後尾より、キューの先頭からサーチして置換するの方のスループットが良いという結論が得られる。キューの先頭に近いパケットほど、待ち時間が長いため、タイムアウトで既に無効になっているパケットの可能性が高い。これを新着のパケットで置換すると、新着パケットはより短時間の待ち時間で処理されるため、タイムアウトで無効パケットになる確率が下がる。先頭からサーチする方が、最後尾からサーチするよりキューの先頭に近いパケットの置換をする頻度が高くなるので、サーチの順序として FIFO の原則には反するが、キューの先頭からサーチした方が有効となる。

6.8 考察

シミュレーションの結果、提案手法が IMS のマルチメディアサービスの自動発信規制のための手法として適していることが明らかになった。この提案の利点は、軽負荷かつ人の介入なしで自動的に発信規制を達成できることである。しかし、実装面では、解決すべき問題が3つある。まず、RFC3261 [3] には再送回数を示すための特定のフィールドは SIP パケットヘッダに定義されていない。この提案手法を実現するには、‘MESSAGE’や‘INVITE’などの SIP のヘッダにこの再送回数を追加することを標準化する必要がある。また、再送制御はユーザ端末により行われるので提案手

法を適用するためにはユーザ端末のソフトウェアの修正は不可欠である。第2に、このダイナミックキューは従来のIMS コアネットワークに費用対効果を考えてインストールしていく必要がある。提案手法はステートレスなシグナリングの優先順位に基づいているので、特定のP-CSCFにインストールする必要はない。IMS コアコンポーネントのフロントエンドに置かれた特定のロードバランサに実装して着信 SIP メッセージを処理すればよい。実装上のコストと得られるパフォーマンス間のトレードオフを考慮すると、ダイナミックキューイングはIMS における複数のサービストラヒックの自動発信規制の観点で、ほぼ理想的な性能を持っていることが分かる。

6.9 おわりに

本章では提案したダイナミックキューイングという自動発信規制手法を評価した。シミュレーションのネットワークモデルを紹介し、3つのキューイング構造の特徴を比較してまとめた。さらに、設計したシミュレーションシナリオを説明した。最後、実験環境と実験結果について述べ、結果に対する考察をまとめた。

第7章

結論

7.1 本研究の背景と目標

突然の自然災害や大きなイベントが開かれる時に、膨大なトラヒックがIP Multimedia Subsystem (IMS) ネットワークに流れ込み、深刻な輻輳が発生する場合がある。これを避けるため、過去数十年間、通信会社は発信規制という呼制御を用いて重要通信の確保を行ってきた。発信規制という呼制御は、通信事業者が輻輳を防ぐために、発信数を制限することである。IP マルチメディアサブシステム (IMS) [17] は、汎用的インターネットプロトコルを使用してエンドユーザに音声通話などの様々なマルチメディアサービスを提供することを目標に、第3世代パートナーシッププロジェクト (3GPP) によって標準化され、セッションの初期化や制御に Session Initiation Protocol (SIP) を採用している。このため、IMS の発信規制は、SIP シグナリングメッセージの規制となる。

しかし、以下の2問題に関しては、十分な検討がされてきた訳ではない。第1に、IMS は音声だけでなくインスタントメッセージングなど多彩なマルチメディアサービスを提供している。このため、発信規制は音声通話だけでなく、これら各種のサービスを考慮して実施する必要がある。第2に、緊急時の発信規制により各サービスの品質は低下する。2011年の東日本大震災の際には、携帯電話網における音声トラヒックが通常より50から60倍も増加したことが報告されている[8]。トラヒックが設計容量を大幅に超えたときにもシステムの最大スループットを前提にサービスの品質を論じたいが、一般にこのような場合、システムの最大スループットを維持することは困難である。

本論文では、2つの目的のためにダイナミックキューイングを提案した。第1は、音声とPMIMあるいはSMIMが混在する場合に、総トラヒックに対して自動発信規制を実施する手法を求めることにある。第2は、高負荷時にも音声サービスを損なうことなく、PMIMの交換性能を最大限上げることにある。

7.2 本研究の提案手法

既存のシングル、および2クラスキューイングでは、過負荷時に音声やSMIMのスループットが低下してしまう欠点があった。本提案は、この問題を克服するために、三つの観点からダイナミックキューイング構造を提案した。

1. 各SIPメッセージは、メッセージの種類に応じて優先順位付けされる。優先順位は、セッション中に現れるメッセージ順序に基づいて設計される。セッションのより後に転送されるメッセージにより高い優先順位を割り振る。一方、ページャモードインスタントメッセージングなど非セッションベースのサービスを、セッションベースのサービスより優先される。
2. SIPメッセージの再送回数に基づいて優先順位付けされる。再送されたメッセージの方が新着メッセージより残りの可能な再送回数が少なく、受信成功率が低くなるので、優先度を下げることにする。
3. 輻輳が発生した時、前に説明した優先順位に基づいて、キューの先頭から既存のメッセージを交換する。

7.3 本研究の主たる成果

NS2 シミュレーション結果によって、三つの結論が得る。第1は、本提案が大規模災害などの高負荷状態で有効であることである。第2は、最適キュー長はキュー内にメッセージが滞留する時間がタイムアウト値を越えないように設計する必要があることである。第3は、キューの先頭から置換するによって、置換対象のサーチをキューの先頭から行うことが有効であるということである。

今後の課題として、IMS プレゼンス、Push To Talk などの他のサービスも含めて提供できるように、対象サービスを拡張する必要がある [15]。また、離散状態マルコフ連鎖モデルに基づいて、さらにダイナミックキューイングを分析する必要もある。最後に、災害時に人々の安否確認を保証するためには、警察や救急車など緊急の呼び出し優先順位を検討すべきと考える。これらの検討に基づいて、より洗練されたレベルへ我々の提案を拡張し、向上させたい。

■ 謝辞

本研究および本論文の作成にあたり，ご多忙を極めるなか日本語に翻訳してくださり，終始熱心なご指導を頂いた浅見徹教授に心から感謝いたします。

さらに研究生活全般において，丁寧に指導して下さいました川原圭博講師に心からお礼を申し上げます。

また，共同研究者である博士課程卒業生の Abdul Hamid Ahmad Kamil, 研究内容に関して親身に相談に乗って下さった博士課程の宋さん，そして，日々の研究室生活でお世話になりました浅見研究室の秘書の方々ならびに先輩，後輩のみなさまに感謝いたします。

最後に，留学する機会を与えてくださり，苦楽をともにし，ありとあらゆる場面で私を温かく見守り続けてくれた両親に深く深く感謝いたします。

2012 年 8 月 13 日

参考文献

- [1] IMS application, <http://wbb.forum.impressrd.jp/feature/20060720/165>
- [2] IMS Services, Mr Miikka Poikselka, Aki Niemi, Hisham Khartabil, Georg Mayer, “ The IMS: IP Multimedia Concepts and Services, 2nd Edition ” , p5
- [3] J.Rosenberg, et al, “SIP: Session Initiation Protocol, ” RFC3261, <http://www.ietf.org/rfc/rfc3261.txt>, June. 2002.
- [4] M. Ohta, “Overload Protection in a SIP Signaling Network,” IEEE International Conference on Internet Surveillance and Protection, 2006.
- [5] A.M.Amoee, A.Falahati, “Overcoming Overload in IMS by Employment of Multiserver Nodes and Priority Queues,” IEEE Signal Processing Systems, 2009.
- [6] Masataka Ohta, “Effects of Interaction between Transport and Application Layers on SIP Signaling Performance,” NEW2AN, 2008.
- [7] Jie Yang, et al, “An Optimized Algorithm for Overload Control of SIP signaling Network, ” IEEE Wireless Communications, Networking and Mobile Computing, 2009.
- [8] http://www.soumu.go.jp/main_content/000117676.pdf.
- [9] J.H.James, et al, “Implementing VoIP: A Voice Transmission Performance Progress Report, ” IEEE Communication Magazine, vol.42, 6, pp.36-41, June. 2004.
- [10] Y. Xu, et al, “An Investigation of Multilevel Service Provision for Voice over IP Under Catastrophic Congestion, ” IEEE Communication Magazine, vol.42, 6, pp.94-100, June. 2004.
- [11] <http://www.yanoict.com/yzreport/210>
- [12] VINT Project, “Network simulator ns-2, ” <http://www.isi.edu/nsnam/ns/>
- [13] SIP module, <http://www.tti.unipa.it/fasciana/materiale.htm>.
- [14] W. Zhu, A. K. A. Hamid, M. Mizutani, M. Isomura, Y. Kawahara and T. Asami, “Possible Overload Protection of IMS Network at Emergent Situation,” Proc. of the 2011 IEICE Society Conference, BS-6-47, Sept. 2011.
- [15] W. Zhu, A. K. A. Hamid, M. Mizutani, M. Isomura, Y. Kawahara and T. Asami, “A Damage Analysis and Possible Services of IMS Network in Disaster Situations,” IEICE Tech. Report, vol.111, no.245, pp.57-62, Oct. 2011.

- [16] Benefits, <http://www.ibm.com/developerworks/webservices/library/ws-soa-ipmultisub1/>
- [17] IMS, http://en.wikipedia.org/wiki/IP_Multimedia_Subsystem
- [18] SIP, http://ja.wikipedia.org/wiki/Session_Initiation_Protocol
- [19] Gonzalo Camarillo, Miguel-Angel, “ The 3G IP Multimedia Subsystem (IMS): Merging the Internet and the Cellular Worlds ”, John Wiley & Sons, 2006, ISBN 0-470-01818-6.
- [20] IMS architecture, http://en.wikipedia.org/wiki/IP_Multimedia_Subsystem
- [21] J.H.James, B. Chen and L. Garrison, “ Implementing VoIP: A Voice Transmission Performance Progress Report ”, IEEE Communication Magazine, vol.42, 6, pp.36-41, June(2004)
- [22] Y. Xu, M. Westhead and F. Baker, “ An Investigation of Multilevel Service Provision for Voice over IP Under Catastrophic Congestion ”, IEEE Communication Magazine, vol.42, 6, pp.94-100, June(2004)
- [23] Masataka Ohta, “ Overload Protection in a SIP Signaling Network ”, IEEE 2006
- [24] Chitra Balakrishna, Khalid Al-Begain, “ Towards a User-Centric and Quality-Aware Multimedia Service Delivery Implementation on IP Multimedia Subsystem ”, IEEE NGMAST, 2007
- [25] Zoric, S., Barakovic, J., Hodzic, H., “ QoS architecture in IP multimedia subsystem of UMTS ”, IEEE International Symposium ELMAR-2008
- [26] Gouya, A., Crespi, N., Bertin, E., “ Service Invocation Issues within the IP Multimedia Subsystem ”, IEEE Networking and Services, 2007
- [27] Patnaik, D., Deshpande, A., Poonawalla, A., Clark, R. ; Hunter, M.T., Takaya, N., Koike, A., “ A Framework for Converged Video Services in the IP Multimedia Subsystem ”, Internet Multimedia Services Architecture and Applications (IMSAA), 2009 IEEE International Conference
- [28] Yang Hong, Changcheng Huang, James Yan, “ Analysis of SIP Retransmission Probability Using a Markov-Modulated Poisson Process Model ”, Network Operations and Management Symposium (NOMS), 2010 IEEE
- [29] Volker Hilt, Indra Widjaja, “ Controlling Overload in Networks of SIP Servers ”, International Conference on Network Protocols, 2008 IEEE
- [30] Masataka Ohta, “ Effects of Interaction between Transport and Application Layers on SIP Signaling Performance ”, NEW2AN '08
- [31] Masataka Ohta, “ Overload Control in a SIP Signaling Network ”, PROCEEDINGS OF WORLD ACADEMY OF SCIENCE, ENGINEERING AND TECHNOLOGY VOLUME 12 MARCH 2006

-
- [32] Jie Yang, Fei Huang, ShuZhi Gou, “ An Optimized Algorithm for Overload Control of SIP signaling Network ”, 2009 IEEE
 - [33] SIP module, <http://www.tti.unipa.it/fasciana/materiale.htm>.

外部発表論文

学術雑誌等

- [1] W. Zhu, 宋 泰永, 川原 圭博, 浅見 徹, “ステートレスな IMS 発信規制手法の設計と評価,” 情報処理学会論文誌, Mar. 2013 (投稿中) .

国際会議

- [2] W. Zhu, A. K. A. Hamid, M. Mizutani, M. Isomura, Y. Kawahara, T. Asami and Y. Murata, “Automatic Originator Regulation of IMS Multiple Traffic by Stateless Signaling Prioritization,” IEEE Global Communications Conference 2012, Dec. 2012.
- [3] A. K. A. Hamid, W. Zhu, Y. Kawahara and T. Asami, “Design and Implementation of an AR-assisted Tool for Basic Home Network Management,” Proc. IEEE SAINT 2011, pp. 208-213, Munich, Germany, July. 2011.

国内研究会

- [4] W. Zhu, A. K. A. Hamid, M. Mizutani, M. Isomura, Y. Kawahara and T. Asami, “A Damage Analysis and Possible Services of IMS Network in Disaster Situations,” IEICE Tech. Report, Vol. 111, No. 245, IN2011-89, pp. 57-62, Oct. 2011.
- [5] W. Zhu, A. K. A. Hamid, M. Mizutani, M. Isomura, Y. Kawahara and T. Asami, “Automatic Originator Regulation of IMS Voice Traffic by Stateless Signaling Prioritization,” IEICE Tech. Report, Vol. 111, No. 469, IN2011-195, pp. 347-352, Mar. 2012.

全国大会

- [6] W. Zhu, A. K. A. Hamid, M. Mizutani, M. Isomura, Y. Kawahara and T. Asami, “Possible Overload Protection of IMS Network at Emergent Situation,” IEICE Society Conference, BS-6-47, Sept. 2011.
- [7] W. Zhu, A. K. A. Hamid, M. Mizutani, M. Isomura, Y. Kawahara and T. Asami, “Overload Protection with Dynamic Queuing Structure for SIP Voice Call,” IEICE Society Conference, BS-3-5, Mar. 2012.

-
- [8] W. Zhu, Y. Kawahara and T. Asami, “Analysis of the Impact of Queue Length on Throughput for SIP Voice Call with Dynamic Queuing Structure,” IEICE Society Conference, BS-5-8, Sept. 2012.