# メール送信サーバの集約における透過型SMTPプロキシの定量評価

小田, 知央
さくらインターネット株式会社さくらインターネット研究所

嶋吉, 隆夫
岡山大学AI・数理データサイエンスセンター

笠原, 義晃
九州大学情報基盤研究開発センター



出版情報：インターネットと運用技術シンポジウム論文集. 2024, pp.17-24, 2024-11-28. Information Processing Society of Japan
バージョン：

# メール送信サーバの集約における透過型SMTPプロキシの定量評価

小田 知央[1,a)]　嶋吉 隆夫[2]　笠原 義晃[3]

**概要**：電子メールは，オープンなメッセージ交換手段であり，現在も世界中で広く利用されている．共有メールホスティングでは，多くの利用者を同一システムに収容するマルチテナント型の構成を採用することで，リソースの効率を高め，運用コストの削減を実現している．また，グローバルIPv4アドレス使用数の抑制やメール送信の集中管理を目的として，送信サーバはリレーサーバーとして集約されることが一般的である．しかしながら，不正な大量メール送信や送信先サーバーによる迷惑メール対策などの影響から輻輳が発生し，集約サーバーでメール送信キューが伸長することがある．キュー伸長は，原因を発生させた特定のテナントだけでなく他のテナントにも影響を与え，他の利用者の正常なメール送信に大幅な遅延をもたらす場合がある．この結果，共有メールホスティングサービス全体の品質が低下し，サービス提供者にとって管理コストの増加を招いている．この問題に対処するため，筆者らは，送信キューの分離とメール送信の集中管理を両立する「メール送信集約用の透過型SMTPプロキシ」を提案している．本論文では，従来のメールリレー方式と提案手法を比較した実験による，提案手法の定量的な評価結果について述べる．メール送信の輻輳実験では，提案手法により輻輳の影響が限定され，メール送信のレイテンシーに変化がないことを確認した．また，サーバリソース消費実験では，提案手法が従来手法に比べ時間あたりのディスクI/Oの消費が少ないことが分かった．さらに，異なる送信元から同一宛先への同時送信実験において，提案手法は従来手法より短時間で送信完了することを確認した．一方で，提案手法は早く完了する送信元と待たされる送信元が発生し，送信流量の公平性が失われる可能性があることが分かった．

# Quantitative Evaluation of Transparent SMTP Proxy in Email Sending Server Aggregation

Tomohisa Oda[1,a)]　Takao Shimayoshi[2]　Yoshiaki Kasahara[3]

***Abstract:*** Email is a open messaging system that has been widely used worldwide. In shared email hosting, a multi-tenant architecture is commonly employed, where numerous users are accommodated on the same system, enhancing resource efficiency and reducing operational costs. Additionally, due to the limitations on available global IPv4 addresses and the need for centralized email management, it is common to consolidate outbound mail servers as relay servers. However, due to factors such as unauthorized bulk email transmissions and spam mail protection measures taken by destination servers, email sending queues can grow due to congestion at aggregation servers. This queue elongation affects not only the specific tenant that caused the problem, but also other tenants, causing significant delays in the normal sending of emails by other users. As a result, the quality of the entire shared email hosting service declines, and service providers incur increased management costs. To address this issue, we have proposed a "transparent SMTP proxy for email sending aggregation" that combines the separation of sending queues with centralized management of email sending. In the congestion experiment for the outbound message queue, it was confirmed that the proposed method limited the impact of congestion, resulting in no change in email transmission latency. Additionally, in the server resource consumption experiment, the proposed method demonstrated lower disk I/O consumption per hour compared to the existing method. Furthermore, in the simultaneous transmission experiment from different senders to the same recipient, the proposed method completed transmissions in a shorter time than the existing method. However, we found that with the proposed method, some senders may complete their transmissions quickly while other senders may experience delays, resulting in a loss of fairness in the amount of transmission traffic.

## 1. はじめに

電子メールは，インターネットにおける長年の歴史を持つオープンなメッセージ交換手段であり，電子メールに代わる多種多様なメッセージングツールが普及した現在でも世界中で広く使用されている．しかし，電子メールについては，セキュリティの課題がますます重要視されている．迷惑メール対策 [1] やアカウント乗っ取りによる大量の不正メール送信 [2] などが代表的な問題である．メールサービスの運用では，これらのセキュリティ問題に対処すると同時に，遅延などを起こさない安定した運用が要求され，その両立が課題である．

多数の利用者を同一システム内で管理する共有メールホスティングでは，ネットワークやサーバリソースの効率的な利用と，低コストでのサービス提供を目的として，バーチャルドメインかつ複数サーバのマルチテナント型の構成を採用されることが一般的である．ここでのテナントとは，ホスティングサービスの利用者や組織に割り当てられる仮想的な領域を指し，メールサービスの場合，テナントにはメールドメインとそのドメインで管理される複数のメールアカウントが含まれる．

一方で，共有メールホスティングにはいくつかの課題がある．特に，使用できるグローバル IPv4 アドレス数の制限や，送信メールの集中管理の必要性から，外部へのメール送信は，通常，単一または少数の MTA（Message Transfer Agent）をリレーして行われる．しかし，このようなサーバの集約により生じる問題がある．

外部送信用メールサーバを集約する場合，送信サーバのメール送信キューがシステム全体で共有されるため，アカウント不正利用などによる過剰なメール送信が発生すると，キューが滞留しシステム全体に影響を与えることがある．また，送信先のメールサーバにより受信レートが制限される場合，送信キューにメールが滞留しやすくなり，結果としてシステム全体のメール送信に遅延が生じる場合が多い．このような状況では，単にサーバを複数用意して負荷を分散するだけでは，問題を完全に解決することは難しい．

さらに，共有されるグローバル IP アドレスに関連する問題もある．不正なメール送信により，送信サーバのグローバル IP アドレスが拒否リストに登録されると，他の正当な利用者が送信するメールも同様に拒否される可能性がある．この問題を解決するためには，テナントごとに送信サーバのグローバル IP アドレスを分離することが望ましいが，大規模な共有メールホスティングにおいて，各テナントに個別のグローバル IPv4 アドレスを割り当てることは実質的に困難である．

我々は，恒常性のある高集積マルチアカウント型メール基盤の研究を進めている [3]．この研究の一環として，不正メール送信などによるメール送信遅延の影響を最小限に抑えるメール送信サーバの構成方法を検討し，メール送信ゲートウェイとして透過型 SMTP プロキシを利用する方法を提案している [4]．さらに，その実装を「Warp」として公開している [5]．Warp は，SMTP セッションを透過的に中継するだけであり，メールリレーを行わないため，メール送信キューを持たない．Warp を利用することで，メール送信の集約による集中管理とメール送信キューの分離による輻輳影響の限定を両立することができる．

本論文では，従来手法であるメールリレー方式と，提案手法である透過型プロキシ方式を比較する実験による，提案手法の定量的な評価結果を示す．本論文の構成は次の通り．2 章では，共有メールホスティングサービスにおけるメール送信の課題について述べる．3 章では，この課題を解決するためメール送信集約用の透過型 SMTP プロキシを説明する．4 章では，提案手法と従来手法の比較実験を通じた性能評価結果を述べる．5 章でまとめとする．

## 2. 共有メールホスティングにおけるメール送信課題

本章では，マルチテナント型の共有メールホスティングにおいて，外部のメールサーバにメールを送信する際に考慮すべき課題について述べる．

### 2.1 大量メール送信

MTA は，メールを送信するために送信キューを使用し，送信すべきメールはまずこのキューに格納される [6]．送信先のサーバに常に接続できるわけではないため，配送が一時的に失敗したメールはキュー内に残り，一定時間後に再送が試みられる．長期間配送できないメールは最終的に破棄され，送信元にはエラーメッセージが返される．外部に配送すべきメールの量が，サーバの実際の送信レートを超えると輻輳が発生する．輻輳が発生すると，キューの長さが増し，キュー内にメールが滞留する．結果，キュー内のメール送信が遅延する可能性が高まる．

共有メールホスティング環境において，メール配送の遅延はサービス品質に直結する重大な問題である．輻輳が，アカウント数やメール量に対してネットワークやサーバのリソースが不足していることに起因する場合，送信サーバを増やして負荷を分散させることで，この問題を軽減できる．しかし，一時的に数万〜数十万通ものメールがアカウントの不正利用により送信される場合，通常の負荷分散だ

[1] さくらインターネット株式会社 さくらインターネット研究所
SAKURA internet Research Center, SAKURA internet Inc.
[2] 岡山大学 AI・数理データサイエンスセンター
Center for Artificial Intelligence and Mathematical Data Science, Okayama University
[3] 九州大学 情報基盤研究開発センター
Research Institute for Information Technology, Kyushu University
[a)] t-oda@sakura.ad.jp

けでは輻輳を完全に解決するのは難しい [7]．特に，送信メールキューが集約されている共有メールホスティング環境では，輻輳が発生した際の影響範囲は，高集積であればあるほど広がる傾向がある．

### 2.2 IP アドレスによる配送制限

電子メールはインターネット上の任意のホストから配送されることを前提としているため，外部からメールを受信するメールサーバはインターネット全体からの接続を受け付ける必要がある．一方で，電子メールはフィッシングやマルウェアの配布など，悪意を持った目的で利用されることも多く，インターネット上には悪意を持ったサーバやクライアントが多数存在する．このようなホストからの迷惑メールや不正利用を防ぐために，SMTP セッションの接続元 IP アドレスに基づいてメール受信を制限する技術が多くのメールサーバで利用されている [8]．これらの技術は，メールを受信する側のセキュリティ向上のために必要であるが，メールを送信する側に悪意がない場合でも，不正利用や誤判定によって制限の対象となり，正常なメールの配送に影響を与えることがある．

悪質なホストが使用する IP アドレスを登録したリストを利用し，登録された IP アドレスからの接続やメール受信を拒否する方法は広く利用されている．複数のサーバや監視システムで収集した情報に基づいて拒否すべき IP アドレスの一覧を作成し，それを提供するサービスも存在する [9], [10]．単一サーバでの情報収集には限界があるため，そのような既存の拒否リストをメールサーバで利用する例も多い．

拒否リストは IP アドレスやネットワーク単位で構成されるため，共有メールホスティングで送信に利用しているグローバル IP アドレスが特定の拒否リストに含まれると，そのリストを利用しているサーバにはメールを配送できなくなる．

また，許可・拒否の二択ではなく，レピュテーションに基づいてメール受信を制御する方法も用いられる [11]．レピュテーションに応じて，接続を拒否したり，単位時間あたりに受け取るメールの流量を制限したりする．メールサーバに対するレピュテーションを提供するサービスでは，長くインターネット上に存在し正常なメールを送出している IP アドレスはレピュテーションが良く，悪意のあるメールの送信元としてセキュリティ対策機器で検知された IP アドレスはレピュテーションが悪くなる [12]．

さらに，特定のメールサービスにとってなじみのない IP アドレスからのメールには受信レート制御を行う IP スロットリングという仕組みを導入しているサービスやサーバー製品も存在する [13]．新しい IP アドレスから継続的にメールを送信したい場合には，最初に少ない流量でメールを送信し，徐々に流量を増やす必要がある [14]．レート制御の詳細は，攻撃者によって回避されるのを防ぐため非公開となっており，メール受信を拒否されて初めてスロットリング対象になっていることがわかる．

このように，現状の電子メールシステムでは迷惑メール対策などのセキュリティ対策として，送信元の IP アドレスに基づいて受信側でさまざまな制限が行われている．共有メールホスティングでは，送信サーバの IP アドレスが制限対象になると，利用者に多大な影響がある．メール送信側から見ると，送信先でどのような受信制限が行われているかは一般的に分からない．また，実際に接続拒否や一時的なメール受信拒否を受けるまで，制限対象になったことも分からない．管理者は，送信先で受信制限を受けているかをエラーメールやログ，利用者からの問い合わせなどから抽出し，拒否リストからの除外依頼などの対応を行う必要がある．サービス品質に大きな影響を与えるため，速やかな検知と迅速な対応が求められる．

もしテナントごとに別個のグローバル IP アドレスを割り当てることができれば，ある IP アドレスが制限対象になっても他のテナントには影響しない．しかし，近年はグローバル IPv4 アドレスの確保が困難で高コストであることから，共有メールホスティングサービスでグローバル IPv4 アドレスをテナントごとに用意して完全なテナント分離を実現することは事実上不可能である．現実的には，メール送信に利用する複数のグローバル IP アドレスのプールを用意し，テナント間で共有する方法がとられる．状況により特定 IP アドレスの利用を一時的に停止したり，新しい IP アドレスをプールに追加する際に事前にウォームアップしたりするなど，限られた IP アドレスをやりくりして運用する必要がある．

## 3. 透過型 SMTP プロキシ

我々は，共有メールホスティングにおけるメール送信の集中管理や情報収集機能を維持しつつ 2 章で述べた問題に対処する手段として，メール送信集約用の透過型 SMTP プロキシの提案と開発を進めている [4]．

透過型プロキシは，クライアントとサーバの間に配置され，両者が直接通信しているかのように見せかけつつ，通信内容を収集したり，セッションに対して追加の処理を行ったりするプロキシである．HTTP の透過型プロキシでは，通信内容の収集・改変やコンテンツのキャッシュ，暗号通信の終端による通信内容の検査などの機能を持つものがある．SMTP については，透過型でメール検査をするセキュリティ製品が存在する [15]．

図 1 に従来の共有メールホスティング環境の構成を示す．MSA（Message Submission Agent）は，MUA（Message User Agent）からのメールを受け付ける役割を持つサーバであり，負荷分散などの理由で複数のサーバを用いる．外部に送信するメールは，MSA から集約された送信用 MTA

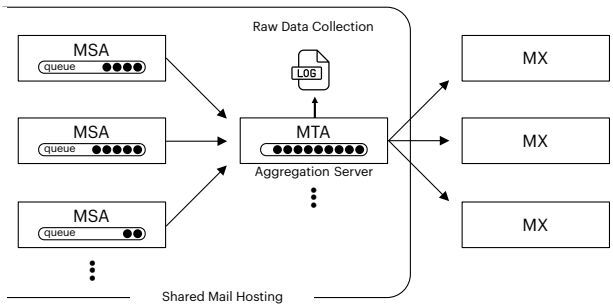



図 1: リレー MTA による集約のメール配送
**Fig. 1** Email delivery using aggregated relay MTA

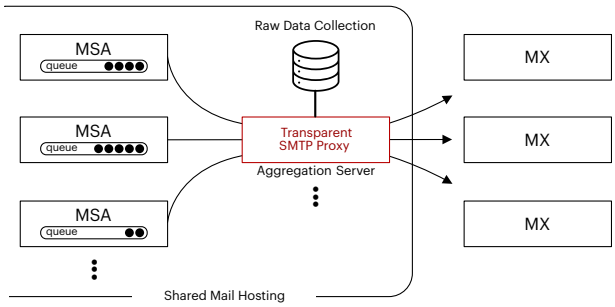



図 2: 透過型 SMTP プロキシによる集約のメール配送
**Fig. 2** Email delivery using transparent SMTP proxy

に転送され，キューに格納された後，送信用 MTA から宛先である各 MX（Mail Exchanger）サーバに送信される [6], [16].

これに対し，提案する透過型 SMTP プロキシを用いたメール送信の構成を図 2 に示す．MSA が外部にメールを送信する際には，あたかも MSA が直接インターネットにメール配送するときのように MX サーバへと SMTP セッションを開始するが，実際にはこのセッションは，IP ルーティングの設定により透過型 SMTP プロキシによって一旦捕捉される．透過型 SMTP プロキシは，ここで SMTP コマンドメッセージを検査し，必要に応じて情報を収集し，場合によってはメッセージを改変する．その後，透過型 SMTP プロキシは，自身がバインドする SMTP 送信用 IP アドレスを使用して，MSA からの要求メッセージを透過的に宛先 MX に送信する．MX サーバからの応答メッセージも透過型 SMTP プロキシに送られ，ここで再度情報収集や改変が行われた後，元の MSA に転送される．

透過型 SMTP プロキシは，MTA ではなく，MSA からの SMTP セッションを中継するだけである．そのため，再送制御，キュー管理といった送信制御機能は前段に配置された MSA が担当する．透過型 SMTP プロキシはキューを保持せず，送信メールの集約に特化することで，IP アドレスの節約と集中管理が可能になる．また，キュー管理を MSA が行うことで，輻輳によるキュー伸長の影響は，キューを増大させたアカウントを担当する MSA に限定される．先行研究 [3] のように MSA をテナントごとに分離すれば，あるテナントが大量にメールを送信したとしても，そのキュー伸長が他のテナントに影響を与えることはない．

なお，Warp は，SMTP コマンドや応答メッセージの内容も検査・収集することも目的としていることから，SMTP の TLS 通信暗号化 [17], [18] されたセッションに対して TLS 通信を終端するための機能も実装している．

キューを持たない透過型 SMTP プロキシは，既存の送信構成をなるべく変更することなく，ルーティングの変更によって導入することが可能である．共有メールホスティングのような多くのメール送信サーバーが稼働している環境では，送信用 MTA を用いた集約に比べ容易に導入ができる．送信 IP アドレスを集約するだけであれば，SNAT（Source Network Address Translation）でパケットの送信アドレスを付け替えることでも実現可能であるが，SMTP コマンドレベルの情報収集や，収集から得られた送信メールの集中管理が難しいという制約がある．一方，透過型 SMTP プロキシを使用すれば，収集する情報に基づき，不正利用が疑われるテナントやアカウントからのメール送信に対して個別の通信レート制御やグローバル IP を使い分けるなどの対応も可能である．

このように，透過型 SMTP プロキシには，再送制御やキュー管理をしないことによる利点がある．しかしその一方で，制御しないことによる弊害も考えられる．例えば，多数のテナントが同一メールドメイン宛に同時に送信するケースが該当する．個人用メールには大手メールプロバイダーが提供する無料のサービスが利用されることが多いため，外部の同一ドメイン宛に多数のテナントが同時に送信することが想定される．この場合，複数の MSA が同一の宛先サーバに対して接続を試みる．受信サーバから見れば単一の送信サーバから多数の同時接続があるように見え，宛先サーバが設定する同一クライアントの同時接続数の制限により何らかの問題を生じる可能性がある．こうしたケースにおいて，従来手法の集約送信サーバは，MSA からのメールをキューに格納した上で，同一宛先への同時接続数制限を行うため，一般的に問題は生じないが，透過型 SMTP プロキシの場合，MSA が個別に接続するため問題が生じる可能性がある．

## 4. 実験と評価

メール送信のゲートウェイにあたる集約サーバが送信メールキューを管理しないことが送信パフォーマンスにどのような影響を与えるのか，また実用性があるかについて，従来手法と提案手法である Warp を比較し議論する．

従来手法として，代表的な MTA である Postfix を使って MSA からリレーを行う．実験環境の概要を図 3 に示す．本番環境を模して，複数の送信ドメインと複数の受信ドメインを使用し，MUA として送信プログラムが各 MSA に接続してそれぞれ異なる宛先ドメインへメールを送信す

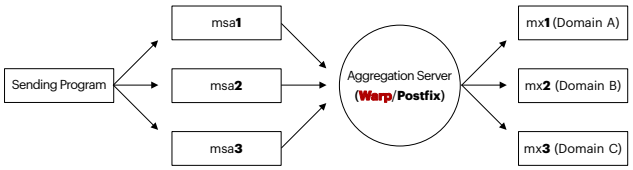



図 3: 性能比較実験環境の構成

**Fig. 3** Experimental environment for performance comparison.

る．比較対象である Warp と Postfix を動かす集約サーバのスペックは，CPU Intel Xeon E3-1220 v6 3GHz 4Core，メモリ 8GB である．MSA と MX および送信プログラムは単一物理ホストで動作させ，各 MSA，各 MX はコンテナ上で動作させた．そのスペックは，CPU Intel Xeon Silver 4208 2.1GHz 8Core，メモリ 32GB である．使用するホストの OS は，全て Ubuntu 22.04.3 LTS，Kernel は 5.15.0-91-generic で，Postfix のバージョンは 3.6.4 である．

実験は，商用サービスの専用物理サーバを用いて別サブネット間で行った．メールの送信宛先には実験用サブドメインを公開 DNS に登録して使用した．なお，MSA と Warp の通信は，iptables とルーティングを使って外部 25 番ポート宛先パケットのみが転送されるようにした．比較実験は以下の 3 つを行った．

(1) 輻輳の再現時の受信数とレイテンシーの変化
(2) サーバリソースの消費
(3) 同時接続時の受信時間分布

### 4.1 輻輳の再現時の受信数とレイテンシーの変化

本実験では，2.1 節で述べた大量送信による輻輳を再現する．実験環境では，実際のインターネットのトラフィック量を再現するのは難しく，また，宛先である受信サーバの応答速度なども異なる．そこで，Postfix のインメモリ送信キューである active キューの上限値（`qmgr_message_active_limit`）を既定値の 20,000 から 1/100 の 200 に変更することで少ないトラフィックで輻輳が生じるようにした [19]．宛先サーバの迷惑メール対策による受信制限を模擬するため，mx3 を tarpit として動かした．ここで tarpit とは，迷惑メール送信者に対する報復措置として，通信セッションを即座に切断せず，応答の送出レートを極端に遅くし，セッションの継続時間を極端に長くすることで不正活動を制限する手法である [20], [21]．受信サーバが故意に応答を遅らせるケースを模擬し，mx3 を tarpit として動かした．送信プログラムは，表 1 に示す 3 種類を同時に，送信通数分を連続して送信し，それらを 10 秒ごと計 60 回繰り返した．この送信パターンは，実際のサービスにおいて，1 回の送信通数は少なく時間を分けて送信することが多く，稀に多くの送信があることを模擬したものである．

表 1: 輻輳再現実験のメール送信パターン
**Table 1** Email sending patterns for simulating congestion

| 送信ホスト | 受信ホスト | 送信通数 | 送信回数 |
|---|---|---|---|
| msa1 | mx1 | 10 | 60 |
| msa2 | mx2 | 1000 | 60 |
| msa3 | mx3 (tarpit) | 10 | 60 |

図 4 は，提案手法の Warp と，Postfix を用いた従来手法における mx1 と mx2 の累積受信メール数の推移を示している．横軸が経過時間，縦軸は受信したメールの通数である．Postfix の場合，400 秒あたりからメールの配送レートが低下していることがわかる．図 5 は，Postfix のキューステータスの推移を示しており，同様に 400 秒あたりから incoming キューが増加している．これは，active キュー長が上限値を超えたため，incoming キュー長が増加し始め，tarpit によって送信できなかったメールが deferred キューに蓄積された結果である [22]．これらの結果から，従来手法では大量送信による送信メールキューの伸長が発生し，送信レートに影響を与えている．一方，Warp の場合，mx1 と mx2 の受信数は線形に増加している．tarpit である mx3 宛への送信は，msa3 のキュー伸長を引き起こすが，その影響が mx1，mx2 への送信に波及していないことが確認された．つまり，メールを中継する Postfix では配送遅延が生じる状況でも，パケットを転送する Warp では遅延が発生していない．

輻輳状態における，メールレイテンシーの変化を確認する．一般的に，メールレイテンシーとは，メールが送信者から受信者に到達するまでの時間的な遅延を指す．本実験では，メール送受信の中で発生する転送における遅延を測定し，Warp と Postfix の性能を比較する．ここでは，送信元である MUA が設定した Date ヘッダーから最後の受信サーバが記録した Received ヘッダーまでの時間を End-to-End レイテンシーと定義し，MSA が付加した Received ヘッダーから最後の受信サーバが記録した Received ヘッダーまでの時間をリレーレイテンシーと定義する．なお，End-to-End レイテンシーは，利用者が認知するレイテンシーであるが，MSA によるオーバーヘッドが含まれる．

輻輳状態における，送信時間での平均レイテンシーの集計結果を図 6 に示した．図 6a, 図 6b はそれぞれ，End-to-End レイテンシー，リレーレイテンシーについて，1 秒間の平均の推移を示している．横軸は送信時間の経過，縦軸は平均レイテンシーを対数表示している．図 6c は，リレーレイテンシーの分布をヒストグラムにより示している．横軸はレイテンシーでビン幅は指数的に増やしている．縦軸はメール数を表している．

図 6a，図 6b では，200 秒を過ぎたあたりから Postfix の平均レイテンシーが長くなっている．一方，Warp による平均レイテンシーは約 1 秒程度に収まっている．なお．レ

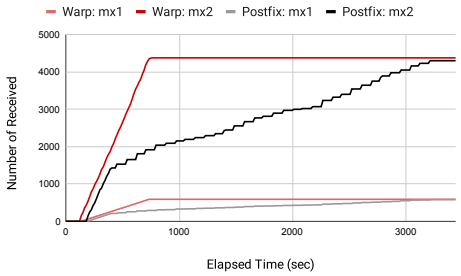



図 4: 輻輳再現時の累計受信数推移
**Fig. 4** Cumulative of received messages under congestion

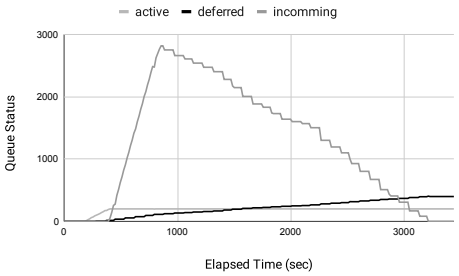



図 5: 輻輳再現時の Postfix キュー長推移
**Fig. 5** Changes in queue lengths of Postfix under congestion

イテンシーの変化幅の値は，Warp の方が Postfix よりかなり小さく，図 6c は Postfix のレイテンシーが分散していることを示す．

これらの結果からも，Warp のメール配送は，輻輳の影響がテナント間で波及しないことがわかる．

### 4.2 サーバリソースの消費

同じ SMTP でのメール配送処理において，Warp と Postfix の間でリソースの使い方にどのような差があるかを実験で確認する．表 2 に，本実験でのメール送信のパターンを示す．3 種類を同時に連続して送信する．送信通数を変化しながら，10 秒ごとに 4 回送信する．1 回ごとの送信通数は，$10^3$ からはじめて $10^6$ まで増加していく．最終的に，合計 1,111,000 通のメールが各ホストから送信される．Postfix は，すべての設定を既定値にしている．

図 7 は，本実験で得られた CPU 使用率，メモリ消費量，およびディスク I/O の推移を示している．図 7a の縦軸は CPU 使用率で，図 7b の縦軸はメモリ使用量，図 7c の縦軸はディスク I/O における読み書きデータ量である．いずれの図も横軸は経過時間である．

CPU 使用率は Warp の方がやや低い．メモリ消費に関しては大きな違いは見られない．これは，同じ SMTP における送信処理であるためだと考えられる．一方で，ディスクの読み書きにおいて顕著な違いが表れた．Postfix はメールキューをファイルで管理する特性上，ディスク I/O を多く必要としているようである．Warp のディスク I/O はロギングにより発生している．また，総計 3,333,000 通の送受信が完了するのに，Warp は 158 分，Postfix は 187 分を要した．

Warp は SMTP セッションの中継処理だけを行うことから，メールリレーを行う従来手法に比べてリソース消費のオーバーヘッドが小さいと言える．

### 4.3 同時接続時の受信時間分布

Warp は，送信メールキューを分散管理することによって輻輳の影響範囲を限定し，メールの配送遅延が該当テナント以外へ波及しないようにしている．一方で，送信メールキューを全体管理しないことによって発生する課題が考えられる．そこで，集約サーバがキューを持たないことのデメリットとして，複数 MSA が同時に同一ドメイン宛にメールを送信する状況で生じる問題を実験により検証する．

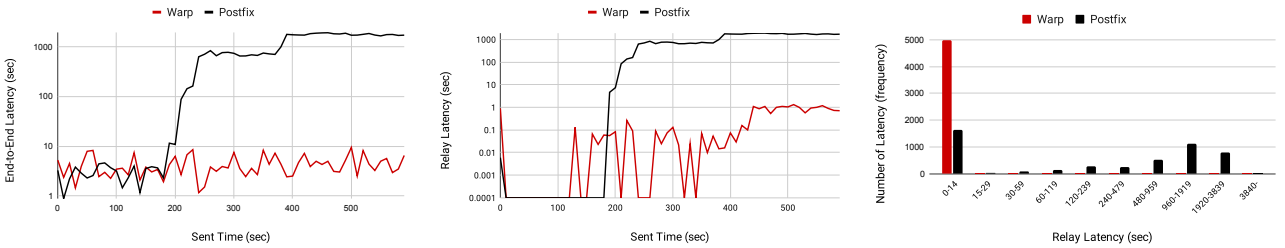
表 2: サーバリソース消費量の評価用メール送信パターン
**Table 2** Sending program configuration for evaluating server resource consumption

| 送信ホスト | 受信ホスト | 送信通数 | 送信回数 |
|---|---|---|---|
| msa1 | mx1 | $10^{n+2}$ ($n$:1-4) | 4 |
| msa2 | mx2 | $10^{n+2}$ ($n$:1-4) | 4 |
| msa3 | mx3 | $10^{n+2}$ ($n$:1-4) | 4 |

本実験では，同一の宛先受信サーバに対して，複数のテナントからの同時接続数の総数が，宛先受信サーバの同時接続数制限を超える状況を再現し，Warp と Postfix においてメールの受信挙動を確認する．mx1 に対して異なるテナントである msa1 から msa3 までが一斉に送信する状況をシミュレートした．表 3 は本実験でのメール送信パターンを示している．想定する状況を再現するため，mx1 は同一クライアントの同時接続数を制限する設定である `smtpd_client_connection_count_limit` を既定値の 50 から 10 に変更した．なお, 各 MSA の配送並列数の初期値である `initial_destination_concurrency` と配送並列数制限値である `default_destination_concurrency_limit` は既定値の 5 と 20 である．

図 8a と図 8b は，各 MSA から送信されたメールを mx1 で受信した数と時間経過である．横軸は時間経過，縦軸はメール受信数である．

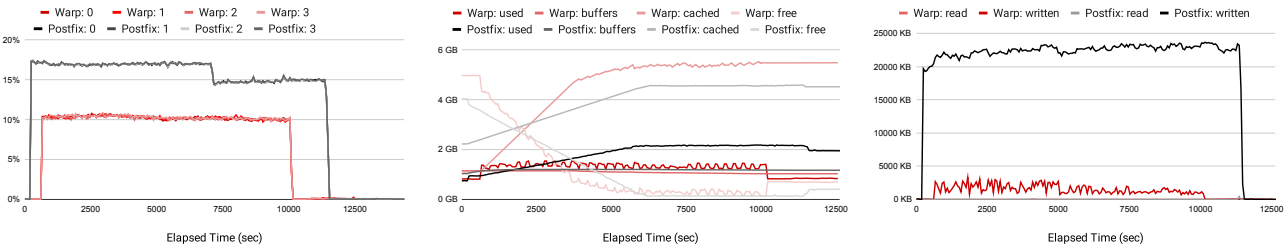
Postfix は，配送並列数を送信先サーバの応答をみて調整する送信制御を行う．各 msa から受け取ったメールは同一のキューに受信順に格納された後に順番に送信されることから，mx1 の制限に応じてそれぞれの MSA は同じ流量で配送している．Warp では，MSA 間で送信完了時間にばらつきが生じている．最も早い msa3 は 39 秒，最も遅い msa2 は 64 秒であり，約 1.6 倍の差が生じている．各 MSA




(a) Average end-to-end latency (b) Average relay latency (c) Relay latency histogram

図 6: 輻輳再現時の平均レイテンシーの推移とレイテンシー分布

**Fig. 6** Changes in average latency and latency distribution under congestion




(a) CPU usage (b) Memory usage (c) Disk I/O usage

図 7: Warp と Postfix におけるサーバリソース使用量の推移比較

**Fig. 7** Comparison of server resource consumption between Warp and Postfix

は，他の MSA の動作とは無関係に Warp を経由して mx1 へとセッションを確立しようとする．それゆえ，mx1 の接続数制限範囲のコネクションは先着順となり，msa 間でコネクション数の不公平が生じる．また，SMTP では単一セッションで複数のメールが送信できる．この実験では，実験初期段階で mx1 への多くのセッションを確立できた msa3 が，メールを優先的に送信している．一方，全 MSA からのメールが受信完了するまでの時間は，Warp が 64 秒に対して Postfix は 299 秒を要した．

Warp は，Postfix に比べ早く処理が完了するものの，送信完了時間にばらつきが生じた．Postfix には，コネクションの再利用を行う SMTP 接続キャッシュ機能がある [23]．本実験では，送信完了した MSA は SMTP セッションを終了した．多くのメールアカウントが MSA に紐づく実環境では，MSA は継続的に大手プロバイダーの MX へメールを送信し，SMTP セッションを，設定された再利用分の時間で保持するとことが考えられる．つまり，共有メールホスティング環境において送信流量の公平性が失われる可能性があると言える．

## 5. まとめ

本論文では，マルチテナント型である共有メールホスティングにおけるメール送信機能に関する課題を挙げ，提案するメール送信集約用の透過型 SMTP プロキシについて，従来のメールリレー方式と比較する実験を行い，提案

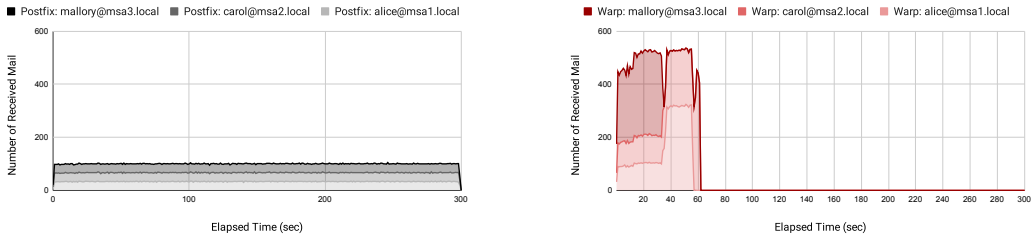
表 3: 同時接続実験のメール送信パターン

**Table 3** Email sending patterns for simultaneous connections

| 送信ホスト | 受信ホスト | 送信通数 | 送信回数 |
|---|---|---|---|
| msa1 | mx1 | 10,000 | 1 |
| msa2 | mx1 | 10,000 | 1 |
| msa3 | mx1 | 10,000 | 1 |

手法の定量的な評価について述べた．実験では，Warp において輻輳の影響範囲が限定的であり，レイテンシーに変化がないことを明らかにした．また，透過型のプロキシであることからディスク I/O を比較的に使用しないことを示し，従来手法のチューニングを行わない場合でも送受信にかかる時間が短縮されることを確認した．一方で，複数テナントから同一ドメインへの同時送信のケースにおいて，送信流量の偏りが発生することがわかった．この課題については，同一宛先に対して複数の IP アドレスを用いてラウンドロビン方式で送信するなどの機能追加によって，メール受信サーバが行う同時接続数制限を緩和することで，不公平性を軽減できる．

Warp は，メール送信サーバの集約の特性を活かし，受信サーバが返す SMTP コマンドの応答情報を収集することが可能であり，メール送信システム全体の可観測性を高めることができる．共有メールホスティングを想定環境としているが，クラウドコンピューティングサービスや VPS サービスのように，ユーザによってメールサーバを構築し




(a) Postfix　　(b) Warp

図 8: 同時接続実験におけるメール受信数の時間経過

**Fig. 8** Changes in message reception rate under simultaneous connections

管理する環境においても，ユーザの手を介さずゲートウェイとして導入することが可能である．今後は，実運用に必要な機能を追加し，受信制限の自動検知や不正メール送信の自動防止のために分析を進める予定である．

**謝辞**　本研究は JSPS 科研費 JP20K11791 の助成を受けたものです．

## 参考文献

[1] 松井一乃，　金高一，加来麻友美，　池部実，吉田和幸：milter の組合せによる低配送遅延を目指した spam 対策メールサーバの設計と導入の効果について，情報処理学会論文誌，Vol. 55, No. 12, pp. 2498–2510（オンライン），入手先 ⟨https://ci.nii.ac.jp/naid/110009851536/⟩ (2014).

[2] Tsuzaki, Y., Matsumoto, R., Kotani, D., Miyazaki, S. and Okabe, Y.: A Mail Transfer System Selectively Restricting a Huge Amount of E-Mails, *2013 International Conference on Signal-Image Technology Internet-Based Systems*, pp. 896–900 (online), DOI: 10.1109/SITIS.2013.146 (2013).

[3] 松本亮介，小田知央，笠原義晃，嶋吉隆夫，金子晃介，栗林健太郎，岡村耕二：精緻に制御可能な恒常性のある高集積マルチアカウント型のメール基盤，マルチメディア，分散協調とモバイルシンポジウム 2018 論文集，Vol. 2018, pp. 1383–1389（オンライン），入手先 ⟨http://id.nii.ac.jp/1001/00193543/⟩ (2018).

[4] 小田知央，　廣川優，近藤宇智朗，嶋吉隆夫，笠原義晃：透過型 SMTP プロキシによるメール送信集約とキュー輻輳回避の検討，マルチメディア，分散協調とモバイルシンポジウム 2021 論文集，Vol. 2021, No. 1, 情報処理学会，pp. 1479–1485（オンライン），入手先 ⟨http://id.nii.ac.jp/1001/00213000/⟩ (2021).

[5] Tomohisa Oda: Warp: Transparent SMTP Proxy, (online), available from ⟨https://warp.linyo.ws⟩ (accessed 2024-09-03).

[6] Klensin, D. J. C.: Simple Mail Transfer Protocol, RFC 5321 (2008).

[7] Gilly, K., Juiz, C. and Puigjaner, R.: An up-to-date survey in web load balancing, *World Wide Web*, p. 105–131 (online), available from ⟨https://doi.org/10.1007/s11280-010-0101-5⟩ (2010).

[8] Lindberg, G.: Anti-Spam Recommendations for SMTP MTAs, RFC 2505 (1999).

[9] Sergeant, M. and Lewis, C.: Overview of Best Email DNS-Based List (DNSBL) Operational Practices, RFC 6471 (2012).

[10] Levine, J.: DNS Blacklists and Whitelists, RFC 5782 (2010).

[11] Dr. Nathaniel S. Borenstein, M. K.: A Reputation Query Protocol, RFC 7072 (2013).

[12] : AbuseIPDB, AbuseIPDB LLC. (online), available from ⟨https://www.abuseipdb.com⟩ (accessed 2024-09-03).

[13] Microsoft Learn: Exchange Online Protection limits, Microsoft (online), available from ⟨https://learn.microsoft.com/en-us/office365/servicedescriptions/exchange-online-protection-service-description/exchange-online-protection-limits⟩ (accessed 2024-09-03).

[14] Microsoft Learn: Warm-up process for marketing senders, Microsoft (online), available from ⟨https://learn.microsoft.com/en-us/dynamics365/customer-insights/journeys/warmup-process-email-marketing⟩ (accessed 2024-09-03).

[15] MailChannels: Transparent Filtering, (online), available from ⟨https://www.mailchannels.com/transparent/⟩ (accessed 2024-09-03).

[16] Crocker, D.: Internet Mail Architecture, RFC 5598 (2009).

[17] Hoffman, P. E.: SMTP Service Extension for Secure SMTP over Transport Layer Security, RFC 3207 (2002).

[18] Melnikov, A.: Updated Transport Layer Security (TLS) Server Identity Check Procedure for Email-Related Protocols, RFC 7817 (2016).

[19] Postfix: The Postfix Home Page, (online), available from ⟨https://www.postfix.org/⟩ (accessed 2024-09-03).

[20] 笠原義晃，小田知央，嶋吉隆夫：インターネットにおける電子メール送信 SMTP 通信への応答の調査方法の検討，研究報告インターネットと運用技術（IOT），Vol. 2023-IOT-63, No. 6, 情報処理学会，pp. 1–8（オンライン），入手先 ⟨http://id.nii.ac.jp/1001/00227557/⟩ (2023).

[21] Hunter, T., Terry, P. and Judge, A.: Distributed Tarpitting: Impeding Spam Across Multiple Servers, *17th Large Installation Systems Administration Conference (LISA 03)*, San Diego, CA, USENIX Association, (online), available from ⟨https://www.usenix.org/conference/lisa-03/distributed-tarpitting-impeding-spam-across-multiple-servers⟩ (2003).

[22] Dent, K. D.: *Postfix: The Definitive Guide*, O'Reilly Media, Inc. (2003).

[23] Postfix: Postfix Connection Cache, (online), available from ⟨https://www.postfix.org/CONNECTION_CACHE_README.html⟩ (accessed 2024-09-03).