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

小田 知央[1]　廣川 優[1]　近藤 宇智朗[1]　嶋吉 隆夫[2]　笠原 義晃[2]

概要：電子メールは古くから用いられているメッセージ交換手段で，依然として世界的に広く利用されている．メールサービスを提供するメールホスティングでは，多数の利用者を同一システムに収容するマルチテナント型によりリソース効率を高め，運用コストを低減している．メールホスティングでは利用可能なグローバルIPアドレス数やメール送信の集中管理のため送信サーバは集約されていることが多いが，大量メール送信や送信先の迷惑メール対策により送信キューの輻輳が発生することで，問題を起こしたテナント以外にも影響が波及し，サービス品質の低下や管理コストの増大をまねいている．本研究では，テナントごとの送信キューの分離と，メール送信の集中管理や送信用グローバルIPアドレスの管理を両立する，メール送信集約用の透過型SMTPプロキシを提案する．また，送信キューの分離によってキュー輻輳時の影響範囲が限定される効果を確認するための予備実験と，透過型SMTPプロキシのプロトタイプ実装について述べる．

A Study on Aggregation of Email Transfer and Avoidance of Queue Congestion using a Transparent SMTP Proxy

Tomohisa Oda[1]　Yuu Hirokawa[1]　Uchio Kondo[1]　Takao Shimayoshi[2]　Yoshiaki Kasahara[2]

1. はじめに

電子メールは古くから用いられている電子的メッセージ交換手段である．近年は電子メールを代替する様々なメッセージングツールが利用されているが，依然として電子メールは世界的に広く利用されている．その一方で，電子メールには迷惑メール対策 [1] やアカウント乗っ取りなどによる不正メール大量送信への対応 [2] など，セキュリティ的に対応しなければならない課題が多い．メールサービスには，これらの問題に対応しながらもメールが遅延しないような安定性やセキュリティの担保が求められる．

利用者にメールサービスを提供するメールホスティングでは，多数の利用者を同一システムに収容するマルチテナント型により集積率を高めることで，リソース効率を高めて，低コストでのサービスを実現している．なお，ここで「テナント」とは，ホスティングサービスの利用者（組織）に割り当てられる仮想的な領域であり，メールサービスの場合はメールドメインと，そのドメインで発行された多数のメールアカウントを含んでいる．メールホスティングにおいて，グローバルIPv4アドレス数の制約や送信メールの集中監視といった要請から，外部にメールを送信するサーバは集約し，単一または少数のMTA (Message Transfer Agent) によりメール送信する構成が一般的である．

一方で，外部送信用メールサーバを集約する構成には課題がある．この構成では、送信サーバの送信メールキューがシステム全体で共有されることになるが，そのことに起因する問題が生じる．例えば正常に送信可能な範囲を超える大量のメールが利用者から外部に送信されるなどして，大量のメールが送信メールキューに滞留することがある．同じ事象は，送信先メールサーバにより，迷惑メール対策などを理由に送信サーバからの受信レートが制限された場合にも発生する．メールホスティングにおいて，送信メールキューの滞留は，システム全体でのメール送信の遅延に

[1] GMO ペパボ株式会社
GMO Pepabo, Inc.
[2] 九州大学情報基盤研究開発センター
Research Institute for Information Technology, Kyushu University

つながる．この問題は，少数の送信サーバによる負荷分散だけでは，完全に解決することはできない．また，テナント間での IP アドレスの共有に起因する課題もある．利用者による不正メール送信などによって，メール送信サーバの IP アドレスが拒否リストなどに掲載され，宛先サーバにより接続拒否や受信レート制限を受ける場合がある．その場合，同じ送信サーバを利用する善良な利用者のメールも受信拒否やレート制限の対象となる．もしホスティングのテナント別に送信サーバの IP アドレスを分離できればこの問題は解決できるが，大規模なメールホスティングで収容テナント数と同数のグローバル IPv4 アドレスを用意することは一般的に不可能である．

本稿では，マルチテナント型メールホスティングにおいて，不正メール送信による他の利用者への影響を限定的とする，メール送信サーバの構成方法に関する検討について述べる．提案する方法では，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ことで，メール送信の集約による集中管理と，利用者間でキューの分離によるキュー輻輳の回避を両立する．テナント別送信サーバからインターネットへのメール送信は，透過型 SMTP プロキシを経由して送信する．これにより，テナント別送信サーバでは既存のメール送信ソフトウェアに変更を加えることなく，送信サーバと宛先サーバ間での SMTP コマンド・応答を，ホスティングシステム全体で網羅的に収集，制御できる．また，透過型 SMTP プロキシはキューを持たないことから，コンテナにより送信キューをテナント分離することで，他テナントの影響でメールがキューに滞留することがなく，ホスティングシステム全体でのメールキューの輻輳を回避できる．

本論文の構成を述べる．2 章では，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 プロキシを提案する．4 章では，提案した機能の概念実証のため，送信キュー分離の効果を確認する予備実験と，プロトタイプ実装ついて述べる．5 章でまとめとする．

2. メールホスティングにおけるメール送信機能の課題

本章では，メールサーバ，特に集積度の高いメールホスティングが，他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

電子メールは，各組織が自分のメールドメインを持ち，自組織の利用者にメールアドレスを付与し，各ドメインのサーバ同士がメッセージを交換する分散運用が前提となっている．各メールサーバは DNS (Domain Name System) を利用して，MX レコードから宛先メールアドレスのドメインを担当するサーバを取得し，そのサーバに接続してメールを送信する [3]．一般的に MTA には送信キューが用意されていて，送信すべきメールは一旦キューに入って順に送信処理される．DNS で宛先サーバの情報を取得できたとしても，そのサーバに常時接続できるとは限らず，配送できなかったメールはキューに残して一定時間後に再送を試みたり，長時間配送できなかったメールは破棄して送信元にエラーメッセージを返したりする．サーバが外部に配送するべきメールを受け付ける数が，実際に外部に配送できる数より多くなる状態が輻輳である．なんらかの理由により輻輳が発生すると，キュー長の増大によりメールがキューに滞留し，その MTA による全てのメール送信が遅延する．

メールホスティング環境において，メール配信の遅延はサービス品質に関わる大きな問題である．ネットワークやサーバの能力がアカウント数やメール流量に対して不足することによる輻輳であれば，送信サーバを複数用意して負荷分散することで，輻輳を軽減できる．しかし，アカウントの不正利用による迷惑メール送信では一時的に数万〜数十万通のメールが投入されるような場合があり，通常の負荷分散で完全に輻輳を解決することは困難である．アカウントの不正利用については 1 アカウントが単位時間に送信できるメール数を制限するといった入口対策も重要だが，送信キューの輻輳は様々な理由で起こりうるため，出口側の対策も必要である．

送信キュー輻輳によるメール遅延は，原理的にはテナント毎に送信キューを分離する事で影響範囲を特定のテナント内に限定できる．利用者の MUA (Mail User Agent) からの送信メールを受け付ける MSA (Message Submission Agent) については内部的にテナント分離される場合もある．しかし，不正メール送信への対策やメール送信状況の集中管理などの要請などから，メールを外部に送信する MTA を集約し，送信キューがテナント間で共有される構成が一般的である．我々が過去に提案した高集積マルチアカウント型メール基盤 [4] においても，MSA は軽量コンテナにより高集積とテナント分離を両立させる構成となっていたが，送信メールのウイルス検査や，ホスティング全体の送信メールについて網羅的・集中的に情報を収集して不正メール送信対策等を行う必要性から，送信メールはテナント間で共有のメール送信ゲートウェイコンテナから外部に送信する設計を採用しており，送信キューは集約されていた．送信キューが集約されるメールホスティング環境において，送信キュー輻輳は，高集積であればあるほど，影響範囲が拡大する．

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信されうることが前提であるため，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，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の利用も多く，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も多く存在する．そ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 [5]．それらの技術は，メールを受信する側のセキュリティ向上のために必要な一方で，メールサービス自体には悪意がない場合でも不正利用や誤判定により制限の対象となることで，正常なメールの配送に影響を受けることがある．

悪質なホストが利用する IP アドレスを登録したリストを用いて，登録 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用いられている．サーバに静的な拒否リストを用意して手動で IP アドレスを登録する方法も用いられるが，手動での拒否リスト管理は煩雑でコストが高いことから，ログの出力を元に一定期間接続を拒否するようなソフトウェアを利用することもある．また，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，それを講読者に提供するサービス [6] もあり，単一サーバでの情報収集には限界があることから，サービスとして提供されている拒否リストを購読してメールサーバで利用する例は多い．拒否リストは IP アドレスやネットワーク単位であるため，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．拒否リストでメール受信を拒否する場合には，リストから除外する方法を提供することが推奨されており，SMTP の応答メッセージなどにその情報を含めて返すのが一般的である [6]．

また，許可・拒否の二択ではなく，レピュテーション（評判）に基づいてメール受信を制御する方法も用いられる．レピュテーションのしきい値を決めて拒否したり，レピュテーションの値に応じて単位時間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一般的に，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 アドレスは評判がよく，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 アドレスが評判が悪くなっていく．

また，ある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するという，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製品もある．例えば，Microsoft Exchange Online では，今まで Exchange Online にメールを送信したことがない IP アドレスからのメールには強い送信レート制限が課せられる [7]．IP スロットリングを実装しているメールサービスに対し，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，最初に低レートでメールを送信し，だんだん流量を増やす（ウォームアップする）必要がある [8]．レート制御の詳細は，攻撃者によって回避されると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はじめてスロットリング対象になっていることがわかる．筆者らの所属する九州大学では，全学のメールサービスを Exchange Online に移行した直後，学内の他のサーバからの転送メールや，安否確認のための一斉送信サービスからのメールがスロットリング対象となったことがある．

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．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，受信側でどのような受信制限を行っ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されるまでは，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，利用者からの問合せなどから抽出し，拒否リストからの除外依頼などの対応を行なう必要がある．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必要である．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，高集積な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と同数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止めたり，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．例えば Exchange Online では拒否リストに入ってもいい高リスク配送プールを別に用意する運用をしている [9]．

3. 透過型 SMTP プロキシ

メールホスティングにおいて，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方法として，メール送信集約用の透過型 SMTP プロキシを提案する．

透過型プロキシとは，クライアントとサーバ間のネットワーク経路に設置され，クライアントとサーバはお互い直接通信しているように見せつつ，セッションに対する付加処理を行うことができるプロキシである．クライアント・サーバ間の通信内容を収集・改変したり，ウェブプロキシであれば通常のウェブキャッシュのようにサーバコンテン

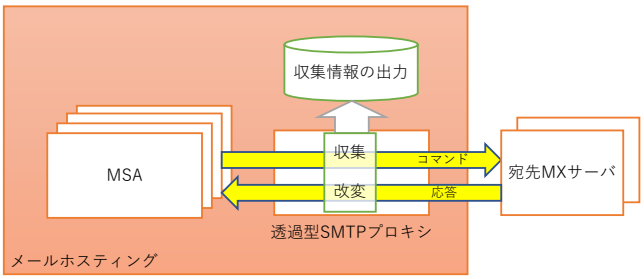

図 1 透過型 SMTP プロキシによるメール送信部の構成
Fig. 1 Mail sending facility using a transparent SMTP proxy.

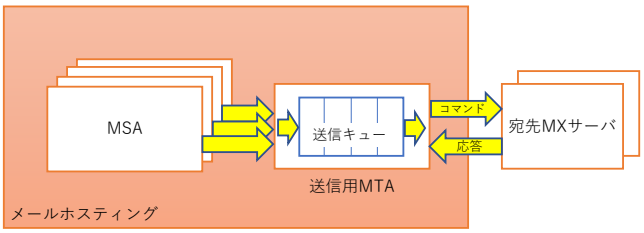

図 2 従来のメールホスティング環境の一般的構成
Fig. 2 Conventional mail hosting environment.

ツをキャッシュしたり，暗号通信をプロキシで終端することによって平文に戻し，通信内容を検査したりすることができる．SMTP 通信を検査する際に透過モードを利用可能なセキュリティ製品がある [10].

3.1 透過型 SMTP プロキシの動作

透過型 SMTP プロキシによるメール送信部の構成を図 1 に示す．「MSA」は，利用者が MUA から送信するメールを受け付けるメールサーバである．従来の一般的なメールホスティング環境の構成（図 2）では，外部への送信メールは，MSA から「送信用 MTA」に転送されてキューに格納されたのち，送信用 MTA から「宛先 MX サーバ」に送信される．一方，提案する構成では，MSA がメールを外部に送信する際に，通常通り DNS で宛先ドメインの MX レコードから「宛先 MX サーバ」の IP アドレスを取得し，SMTP セッションを開始する．しかし，実際にはそのパケットは「透過型 SMTP プロキシ」で一旦受け取る．透過型 SMTP プロキシは，ここで SMTP のコマンドメッセージの内容に対して検査や情報収集などの必要な処理を行う．その後，MSA からのコマンドメッセージを，場合によっては改変を伴って，透過型 SMTP プロキシがバインドする SMTP 送信用 IP アドレスから本来の宛先サーバへと送信する．宛先サーバからの応答メッセージは透過型 SMTP プロキシの SMTP 送信用 IP アドレス宛に送られるので，コマンドメッセージと同様に情報収集や改変などののちに，本来の送信元である MSA に転送する．

最近では，MTA 間の SMTP 通信についても，インターネット上での盗聴を防ぐ目的などから，STARTTLS コマンドによる TLS 暗号化が一般的になっている [11][12]．暗号化が送信サーバと宛先サーバで終端されている場合，経路上でセッションの内容を取得することができない．つまり，MSA が宛先 MX サーバに対して STARTTLS コマンドを発行すると，それ以降のコマンド・応答メッセージは TLS 暗号化されることになり，透過型 SMTP プロキシでは内容を確認できない．しかし，本提案では，TCP セッション情報だけでなく，SMTP コマンド・応答メッセージの内容も情報収集・検査の対象とすることを考えている．そこで，透過型プロキシにより TLS 暗号化を制御することとした．具体的には，宛先 MX サーバからの EHLO コマンド応答に「TLS 暗号化通信可能」を意味する STARTTLS の提示があった場合，宛先 MX サーバとの TLS セッション確立は透過型 SMTP プロキシが行い，TLS 通信を透過型 SMTP プロキシで終端する．これにより，インターネット上の SMTP 通信は暗号化で保護しつつ，MSA からの送信メール情報を一元的に収集できる．このとき，MSA と透過型 SMTP プロキシとの間で暗号化が不要であれば，透過型 SMTP プロキシで EHLO コマンド応答から STARTTLS を削除して MSA に返すことで，透過型 SMTP プロキシと MSA との間は平文で SMTP セッションを開始できる．MSA と透過型 SMTP プロキシとの間も暗号化したい場合は，MSA において任意のホストに対して透過型 SMTP プロキシのサーバ証明書を許可するように設定することで実現できる．

3.2 透過型 SMTP プロキシの特徴

メールホスティングにおいて 2.1 節で述べた大量メール送信による影響を最小限に留めるには，キューをテナントごとに分離し，そのキューから直接外部にメールを送信する必要がある．一方 2.2 節で述べたように，テナントごとに別の IP アドレスを付与するのは現実的でないため，送信 IP アドレスは集約する必要がある．

送信用 MTA ではなく透過型 SMTP プロキシを用いることで，キューを持つことなく，少ないグローバル IP アドレスで集約してメールを送信可能である．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だが，送信メールの集中管理には別の仕組みが必要となる．

さらに，送信キューは MSA ごとに持つ構成となるため，先行研究のように 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，キュー長の増大が他のテナントに影響を与えることがない．透過型 SMTP プロキシで収集する情報に基づいて，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は，透過型 SMTP プロキシで通信レートを制御するなどの対応も可能である．

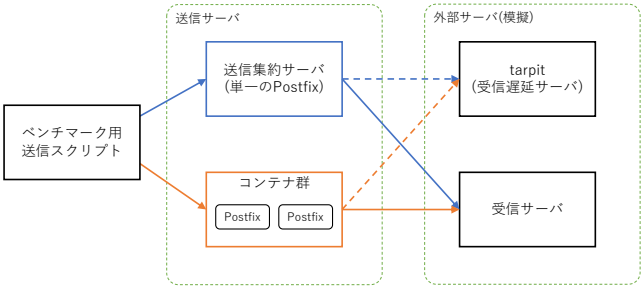

図 3 実験の構成
Fig. 3 Preliminary experimental system.

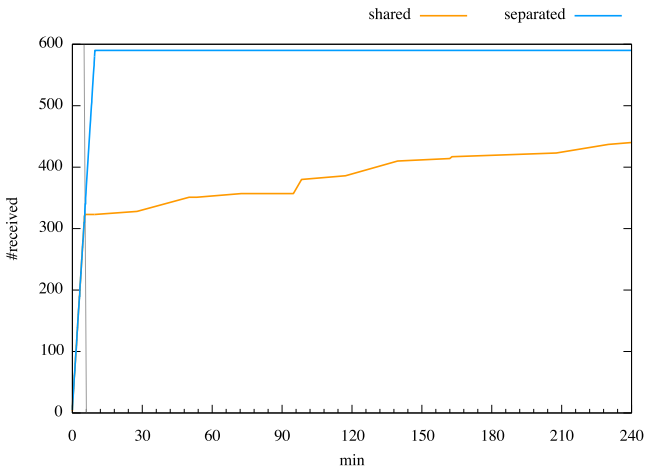

図 4 メール配信数の時間経過
Fig. 4 Time series of message delivery counts.

透過型 SMTP プロキシでは，送信に利用する IP アドレスを集中的に管理するだけでなく，宛先 MX サーバからの SMTP 応答も集中管理できる．このことから，例えばある送信用 IP アドレスが拒否リストに登録されたり，スロットリングの対象となったときに，応答メッセージや応答タイミングなどを分析することでこれを検知可能である．例えば，複数の送信用 MTA を用意して別個の送信用 IP アドレスを用いる構成においてある送信用 IP アドレスが拒否リストに登録された場合，当該 MTA のキューにメールが滞留することになり，キュー内のメールは送信用 IP アドレスを付け替えるまで送信できず，アドレス付け替え後に順次送信されるのを待つ必要がある．しかし，透過型 SMTP プロキシは自身がキューを持たないことから，透過型 SMTP プロキシでのメール送信遅延は発生しない．さらには，透過型 SMTP プロキシに複数の送信用 IP アドレスを設定しておけば，拒否リストに登録された IP アドレスの使用を停止するだけで，メール送信が継続できる．

4. 概念実証

4.1 予備実験

予備実験として，送信キューの分離によってキュー輻輳時の影響範囲が限定される効果を確認するための実験を行った．実験の構成を図 3 に示す．実験は，Ubuntu 20.04 LTS 上で実施した．「ベンチマーク用送信スクリプト」は，多数の利用者からのメール送信を模擬するスクリプトである．従来法では送信用 MTA サーバでキューが共有されることから，ベンチマーク用送信スクリプトは単一の Postfix 経由でメールを送信する（キュー共有）．一方，提案手法の模擬構成（キュー分離）では，MSA 別にキューが分離されることになるので，コンテナを利用して複数の Postfix を MSA として起動し，ベンチマーク用送信スクリプトは複数 MSA 経由でメールを送信する．

送信キューでの輻輳を発生させることを目的に，本実験では送信先の 2 ドメインのうち 1 つを tarpit として動作させる．tarpit は迷惑メール対策の 1 つで，相手が悪意のあるサーバと判定した場合などにセッションは切断せずに応答の送出を極端に遅らせ，送信サーバによるメール送信の完了を遅延させる仕組みである [13]．本実験では，tarpit サーバとして mxtarpit[*1]を使用した．なお，本実装では標準で SMTP 応答を 1 文字につき 3 秒かけて返す設定となっていたが，実験にかかる時間を考慮し 0.25 秒に変更している．これにより，tarpit 宛のメール送信には通常以上に時間を要することとなり，結果的にメール送信レートが低下することでキューの輻輳が発生する．また，tarpit の影響を明確に可視化するため，実験開始 5 分後に受信遅延サーバ宛メールを 80 通送信している．キュー分離構成では，あるテナントが不正メールを送信している状況を想定し，通常のメールと受信遅延サーバ宛メールはベンチマーク用送信スクリプトから別の MSA へと送信する．

本実験でのメール配信数のグラフを図 4 に示す．横軸は実験開始からの時間（分）であり，青がキュー分離の構成で配信完了した通常メールの積算通数，黄がキュー共有の構成で配信完了した通常メールの積算通数である．tapirt 宛メールの送信開始時点を灰色縦線で示している．キュー分離構成では，tarpit 宛メールの送信開始前後でレート変化なくメールが受信できていることが分かる．一方，キュー共有構成では，tarpit 宛メールの送信開始後には配信レートが著しく低下しており，大きな配信遅延が生じていることが分かる．これは，tarpit へのメール送信に大きく時間が掛かることから，Postfix の送信キューにメールが滞留し，さらには Postfix でのメール受理が不能になり，送信スクリプトからのメール送信にも遅れが生じたことによる．この結果から，送信キューを分離することで，メール送信遅延の影響範囲が限定されることが確認できた．

4.2 プロトタイプ実装

提案する透過型 SMTP プロキシが実現可能であることを示すため，プロトタイプを開発した．実装言語は go 言

*1 https://github.com/martinh/mxtarpit/

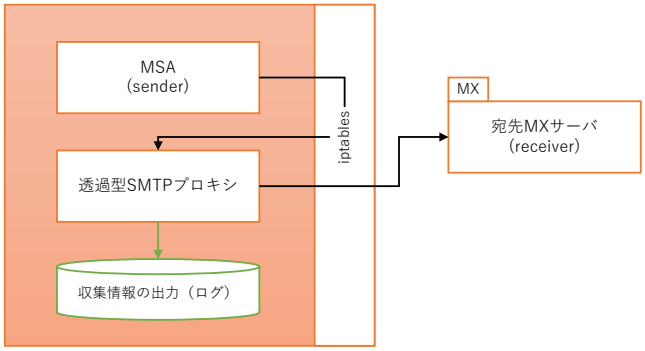

図 5 プロトタイプ検証環境の構成
Fig. 5 Prototype test bed.

語である[*2]．なお，本実装では，TLS 対応を簡略化するために，MSA と透過型 SMTP プロキシを同一ホスト上で動作させることを前提とし，3.1 節で述べた方法を用いて，宛先 MX サーバと透過型 SMTP プロキシとの間だけで TLS 通信を行い，MSA と透過型 SMTP プロキシとの間は平文で通信する実装とした．また，収集した情報はログとして出力する実装としている．

本プロトタイプの動作確認実験を行った模擬環境の構成を図 5 に示す．MSA からのパケットを透過型 SMTP プロキシプロセスにリダイレクトするために Linux の iptables を使用している．

透過型 SMTP プロキシを起動し，MSA (sender) から宛先 MX サーバ (receiver) にメールを送信した際のプロキシでのセッションログの例を図 6 に示す．図 6 で時刻情報に続く「<-」は sender に送信された応答（プロキシで改変されたものを含む），「->」は receiver に送信されたメッセージ（プロキシで改変されたものを含む），「|<」は receiver から送信されプロキシで終端された応答，「|>」はプロキシから receiver に送信されたメッセージ，「>|」は sender から送信されプロキシで終端されたメッセージである．

図 6 の 1，2 行目で平文での SMTP 接続を開始している．5 行目で receiver からの応答には TLS による暗号通信の開始が可能であることを示す「STARTTLS」が含まれているが，6 行目で sender に返す際にはその行を削除することで，sender は平文のまま SMTP 通信を続ける．7 行目で sender とは独立にプロキシから STARTTLS コマンドを発行し，TLS セッションを確立するとともに，10 行目で再び EHLO コマンドを送る事で SMTP セッションを再開する．その応答である 12 行目は sender に返す必要がないため破棄している．8 行目で sender から受け取ったメール送信のためのコマンドは一旦プロキシで保持しておき，TLS セッションが確立した後に 14 行目で receiver に送信している．なお，実際にはこれらのメッセージは暗号化されて送信される．その後も，引き続き通常の SMTP セッションを継続することで，メール送信を完了している．

[*2] https://github.com/linyows/warp/

この実験により，宛先 MX サーバとの間で TLS 暗号化通信を行う場合でも，透過型 SMTP プロキシで SMTP セッションの情報を収集・検査可能であることが確認できた．

5. まとめ

本研究では，高集積マルチテナント型メールホスティングにおけるメール送信機能で解決すべき課題を述べ，その解決方法としてメール送信集約用の透過型 SMTP プロキシを提案し，キュー分離によるメール送信遅延の問題回避を確認する予備実験，および，プロトタイプ実装と動作検証を行った．これらの実験により，提案手法の概念が実証できた．

今回実装した透過型 SMTP プロキシのプロトタイプは機能が限られているが，今後，実装を改良していく予定である．例えば，透過型 SMTP プロキシでは，宛先 MX サーバからの応答を収集できるだけでなく，各 SMTP コマンド・応答メッセージの送受信に掛かった時間などの情報も出力できる．これにより，受信拒否や一時受信不能による明示的な受信制限だけでなく，tarpit やそれに類する通信レートによる受信制限も検知可能だと予想している．また，アカウント，テナントや宛先 MX サーバ，送信メールの疑わしさなどによって送信用 IP アドレスを使い分けること，拒否リストの掲載を理由に受信拒否したという応答メッセージを受け取った場合に当該 IP アドレスを自動的に利用停止するなどといった機能拡張が考えられる．

さらに，本研究では送信用 MTA の代わりに透過型 SMTP プロキシを用いる方法を提案しているが，透過型 SMTP プロキシは既存のシステムにも容易に追加できることから，例えば，既存の送信用 MTA と同一ホスト上で透過型 SMTP プロキシを動作させて情報収集に利用することも可能である．今後，ホスティングサービスの実運用に影響を与えない範囲で透過型 SMTP プロキシを導入し，実際の SMTP コマンド・応答の情報を収集し，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ていくことを検討している．

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).

```
2021/02/06 14:50:48 connected from 192.168.30.40:57493
2021/02/06 14:50:48 connected to 192.168.30.50:25
2021/02/06 14:50:48 <- 220 receiver ESMTP Postfix (Ubuntu)\r\n
2021/02/06 14:50:48 -> EHLO sender\r\n
2021/02/06 14:50:48 |< 250-receiver\r\n250-PIPELINING\r\n250-SIZE 10240000\r\n250-VRFY\r\n250-ETRN\r\n250-STARTTLS\r\n250-ENHANCEDSTATUSCODES\r\n250-8BITMIME\r\n250-DSN\r\n250-SMTPUTF8\r\n250 CHUNKING\r\n
2021/02/06 14:50:48 <- 250-receiver\r\n250-PIPELINING\r\n250-SIZE 10240000\r\n250-VRFY\r\n250-ETRN\r\n250-ENHANCEDSTATUSCODES\r\n250-8BITMIME\r\n250-DSN\r\n250-SMTPUTF8\r\n250 CHUNKING\r\n
2021/02/06 14:50:48 |> STARTTLS\r\n
2021/02/06 14:50:48 >| MAIL FROM:<root@sender> SIZE=327\r\nRCPT TO:<root@receiver> ORCPT=rfc822;root@receiver\r\nDATA\r\n
2021/02/06 14:50:48 |< 220 2.0.0 Ready to start TLS\r\n
2021/02/06 14:50:48 |> EHLO sender\r\n
2021/02/06 14:50:48 pipe locked for tls connection
2021/02/06 14:50:48 |< 250-receiver\r\n250-PIPELINING\r\n250-SIZE 10240000\r\n250-VRFY\r\n250-ETRN\r\n250-ENHANCEDSTATUSCODES\r\n250-8BITMIME\r\n250-DSN\r\n250-SMTPUTF8\r\n250 CHUNKING\r\n
2021/02/06 14:50:48 tls connected, to pipe unlocked
2021/02/06 14:50:48 -> MAIL FROM:<root@sender> SIZE=327\r\nRCPT TO:<root@receiver> ORCPT=rfc822;root@receiver\r\nDATA\r\n
2021/02/06 14:50:48 <- 250 2.1.0 Ok\r\n250 2.1.5 Ok\r\n354 End data with <CR><LF>.<CR><LF>\r\n
2021/02/06 14:50:48 -> Received: from sender (localhost [127.0.0.1])\r\n        by sender (Postfix) with SMTP id 45B113EA9B\r\n for <root@receiver>; Sat,  6 Feb 2021 14:50:48 +0000 (UTC)\r\nFrom: <root@sender>\r\nTo: <root@receiver>\r\nDate: Sat,  6 Feb 2021 14:50:48 +0000 (UTC)\r\nMessage-Id: <a77e.0003.0000@sender>\r\nSubject: Hi, Receiver from Sender\r\n\r\nXXXXXXXXXX\r\n.\r\nQUIT\r\n
2021/02/06 14:50:48 <- 250 2.0.0 Ok: queued as 76DAD4113D\r\n221 2.0.0 Bye\r\n
2021/02/06 14:50:48 connections closed
```

図 6 プロキシでのセッションログ

Fig. 6 A proxy session log.

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

[4] 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).

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

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

[7] Exchange_Outlook サポートチーム：IP スロットリングについて，Microsoft（オンライン），入手先 ⟨https://social.msdn.microsoft.com/Forums/vstudio/ja-JP/27b2ffd7-66c9-4c07-b390-dfe59a52e3c4/ip-1247312525124831248812522125311246412395123881235612390?forum=exchangeteamjp⟩（参照 2021-04-30）.

[8] ：IP ウォームアップを行う，構造計画研究所（オンライン），入手先 ⟨https://sendgrid.kke.co.jp/docs/Tutorials/D_Improve_Deliverability/ip_warmup.html⟩（参照 2021-05-07）.

[9] Microsoft 365: Outbound delivery pools, Microsoft (online), available from ⟨https://docs.microsoft.com/en-us/microsoft-365/security/office-365-security/high-risk-delivery-pool-for-outbound-messages?view=o365-worldwide⟩ (accessed 2021-04-30).

[10] MailChannels: Transparent Filtering, (online), available from ⟨https://www.mailchannels.com/transparent/⟩ (accessed 2021-04-30).

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

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

[13] 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).