# 大規模キャンパスネットワークHINET2007へのシングルサインオン機能の実装および評価

藤村喬寿[†1] 田島浩一[†2] 大東俊博[†2]
西村浩二[†2] 相原玲二[†2]

複数のアプリケーションを利用する際の認証を一元化することで、アカウント管理の煩雑さを解消する手段としてシングルサインオン (SSO) が注目されている。一方で、セキュリティ意識の高まりから、企業や教育機関ではネットワークを利用する際に利用者へ認証を要求することが一般的になっている。このネットワークの利用者認証にSSO機能を導入することで、より利便性の高いサービスの提供ができる。ゲートウェイとなるPCに実装された認証システムにSSO機能を導入する試みについては既に議論されているが、近年で増えてきているスイッチベースの認証ネットワークへSSO機能を導入する試みはなされていない。そこで我々は、広島大学キャンパスネットワーク（HINET2007）を例として、スイッチベースの認証ネットワークにSSO機能を導入する方法について議論をし、異なる環境を想定した二つの方式を提案した。しかしながら、これらの方式の性能評価は必ずしも十分とは言えず、実運用での有効性については未だ検討していない。そこで本稿では、これらの二方式の有効性を実機を用いた実験によって評価をし、HINET2007のような大規模キャンパスネットワークにおいて実運用可能な性能を有していることを示す。

# Implementation and Evaluation of Single Sign-On Authentication in Large-scale Campus Network HINET2007

Takatoshi Fujimura ,[†1] Kouichi Tashima ,[†2]
Toshihiro Ohigashi ,[†2] Koji Nishimura [†2] and Reiji Aibara[†2]

SSO is a mechanism to provide many services by only one authentication process, and achieves unification of service accounts. On the other hand, network authentication is requested for security in corporate and education agency. Therefore, network authentication with SSO that allows a user to provide better usability is more expected. Currently, an authentication system using authentication switches is becoming more common. However, no authentication system using authentication switches with SSO has been discussed yet. Then, we have proposed two different mechanisms for above authentication system on HINET2007, which is a large-scale campus network with authentication switches in Hiroshima University. However, the details of performance of the proposed mechanisms have not been evaluated. In this paper, we evaluate the performance of the proposed mechanisms by experiments in the real environment. In addition, we demonstrate that the proposed mechanisms have availability in a large campus network like HINET2007.

## 1. はじめに

Webブラウザがあれば利用でき、環境に依存しにくいメリットから業務作業の効率化や、教育支援などの様々な用途でWebアプリケーションは広く普及してる。それに伴い、利用者が複数のアプリケーションを利用することが想定される。一方で、セキュリティの要求によりアプリケーションで認証が必要となることが多くなっているため、利用者はアプリケーションを利用する度に認証する必要がある。さらに、アプリケーションの運用元が異なる場合では、各アプリケーションからアカウントは個別に発行されるため、利用者が複数のアカウントを管理する必要も生じる。そのようなアカウント管理の煩雑さから、セキュリティ意識の低い利用者が簡易なパスワードを設定する傾向があり、セキュリティの低下が懸念される。それに対して、運用側で登録できるアカウントのパスワードに文字数や形式に制約を設けるなどの対策をしてはいるが、アカウント管理の負荷に起因する問題の根本を解決でき

†1 広島大学大学院総合科学研究科
Graduate School of Integrated Arts and Sciences, Hiroshima University
†2 広島大学情報メディア教育研究センター
Information Media Center, Hiroshima University

ている訳ではない。また、アカウント管理の負荷を軽減するため組織内の複数のアプリケーションでデータベースを共有してアカウントの一元化をすることが一般的になってはいるが、他組織間の場合やアプリケーションを運用している部局間でアカウントの管理ポリシが異なる場合の共有は難しい。その解決策としてシングルサインオン（Single Sign-On : SSO）が注目されている。SSO は一回の認証手続きで利用権限のある全てのサービスの利用を可能にする認証の一元化の技術である。また、認証の一元化に伴ってアカウントも一元化される。SSO の取り組みとして、国立情報学研究所（NII）の主導で Shibboleth[1] を利用した認証連携である学術認証フェデレーション（GakuNin)[2] が構築・運用されている。

広島大学で運用されている広島大学キャンパスネットワーク HINET2007[3] では、全てのネットワーク利用の入り口で認証の必要がある。具体的には、学内に分散配置している約 450 台の認証機能を持ったネットワークスイッチ (認証スイッチ) で認証を要求し、主にブラウザを利用した Web 認証によってユーザを認証している。このようなネットワーク基盤としての利用者認証に SSO を導入することで、ポータルシステム、e-ラーニングのためのコースウェアなどの学内で展開されているアプリケーションの SSO 化による利便性の向上が期待できる。さらに、組織間の認証連携で SSO 機能を他組織の構成員も利用できるように拡張することで、訪問者がネットワークサービスを利用する際に普段から利用している自組織のアカウントでの認証を提供でき、事前のゲストアカウントの利用申請の手間が解消される。既に著者らは上記のようなサービスの向上を目的として、HINET2007 のようなスイッチベースの認証ネットワークに SSO 機能を導入する方式を異なる環境を想定して二種類提案している[4][5]。しかしながら、これらの方式の性能評価は十分ではなく、実環境において想定している性能を有するかについては検討していない。そこで、本稿では実機を利用した評価実験によって提案した二方式の有効性を評価する。

本稿の構成は以下の通りである。第 2 節ではキャンパスネットワークに SSO 機能を導入する従来の研究とアカウントの一元化に対する試みについて紹介し、HINET2007 のようなスイッチベースの認証ネットワークにはこれらの方式が不向きであることを示す。第 3 節では、以前提案したスイッチベースの認証ネットワークに SSO 機能を導入する二つの方式について概説し、第 4 節でそれらの方式の有効性を実機を用いた実験によって評価する。第 5 章はまとめであり、本論文を総括し、今後の課題について述べる。

## 2. 関 連 研 究

ネットワークの利用者機能の SSO 化の試みとして佐賀大学のキャンパスネットワークで実装されている SSO-Opengate[6] がある。これはゲートウェイにネットワークの認証機能を実装し、認証されていない端末のネットワークアクセスを制限している。その際、認証機能を Shibboleth によって実装することで SSO を実現している。これはゲートウェイとなる PC で利用者認証機能を実装している環境への SSO の実装である。しかし、認証機能を持つネットワークスイッチは、メモリ等の資源に強い制約があるため、このような実装は困難である。アプリケーションでの SSO への対応が難しい場合への実装として、NII によってアプリケーションの認証を代理で行うログインプロキシを Shibboleth 上で実装する Shibboleth Proxy[7] が提案されている。しかし、認証スイッチは、端末の MAC アドレスとアカウント情報の対で認証を管理する。そのため、ログインプロキシを挟まないで、端末本体から認証情報を送信する必要があり、この方式は適用できない。よって、スイッチベースの認証ネットワークでは SSO-Opengate とは異なる型の SSO の実装が必要になる。

アカウントの一元化に対する試みとしては、他組織の構成員に対して無線 LAN 環境を提供するための認証連携である eduroam[8] も注目されている。認証連携の仕組みは、加入している組織間で RADIUS サーバ（Remote Authentication Dial-In User Service Servers）のプロキシツリーを構成する。それによって、他組織での認証の際に認証情報を自組織の RADIUS サーバまでプロキシを行い、IEEE802.1X を利用した認証ができる。また、GakuNin の参加組織の構成員に対しても eduroam での認証を可能にする方法として代理認証方式[9] も提案されている。代理認証方式は GakuNin で認証された利用者に対して eduroam で使用できる一時的なゲストアカウントを発行することで認証を提供する。しかしながら、IEEE802.1X を利用した無線 LAN 環境に特化した認証連携で、スイッチベースの認証ネットワークでは、別の仕組みについて検討する必要がある。

先にも述べたように、SSO-Opengate は PC のゲートウェイによる認証を実装することで、Shibboleth による認証の実装のようなソフトウェアでの機能拡張に優れている。しかしながら、データ転送もソフトウェア処理されるため、CPU 負荷がデータ転送性能に影響する。スイッチベースの認証ネットワークでは、PC のようなソフトウェア面での拡張性は低いが、データ転送がハードウェアで処理されるため、CPU 高負荷時でもデータ転送性能への影響がでない。それに加えて、IEEE802.1q（TagVLAN）の活用での柔軟な仮想

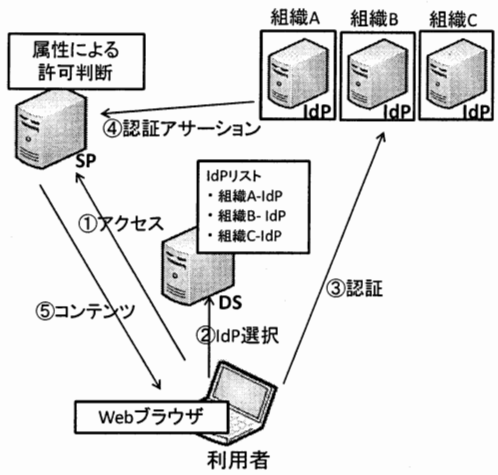


図 1　Shibboleth の認証手順

配線、エッジスイッチによる認証の分散処理やセキュリティの水際対策の実現などのメリットから採用する組織は年々増加している。このような環境への実装の要求を無視することはできない。本研究は、この要求に応えるためのものである。

## 3. スイッチベースの認証ネットワークへのSSO 実装

### 3.1 HINET2007

広島大学では、2008 年度から HINET2007 の運用を開始した。HINET2007 ではセキュアでスケーラブルなネットワークを構築するため約 450 台の認証スイッチによるスイッチベースの認証ネットワークを採用している。その認証スイッチによって、全ての場所で HINET2007 を利用する端末へ何らかの認証を要求している。それによって認証スイッチで、認証前の端末のアクセスを制限し、セキュリティへの要件を満たしている。また、主要フロア毎に設置してある認証スイッチによる認証の分散処理や、Web 認証機能の活用による混在する OS への柔軟な対応によって約 2 万人の広島大学構成員が利用する大規模ネットワークのスケーラビリティへの要件を満たしている。

### 3.2 Shibboleth

SSO 機能を実現する方法は複数存在するが、教育機関での認証連携である GakuNin を利用できること、ブラウザを利用した仕組みのために Web 認証と親和性が高いことから HINET2007 では Shibboleth を利用して SSO 機能を導入する。Shibboleth は Internet2 によって開発された SAML を実装した認証のための属性交換を行うミドルウェアである。その構成は、サービスの利用認証を行う IdP（Identity Provider）、Web アプリケーションの提供と利用権限の確認を行う SP（Service Provider）、SP に対して複数の IdP が用意されている場合に IdP のリストを提供する DS（Discovery Service）で構成されている。IdP と複数の SP、または SP と複数の IdP が認証連携することで連携した SP の間では統一されたアカウント情報の利用を可能にする。また、事前に IdP と SP 間で受け渡すアカウント情報を利用ポリシで定めて信頼関係を構築することで必要最小限のアカウント情報での認証を実現する。この一連の認証手順によって組織を越えた認証連携においてもセキュアで個人のプライバシに配慮できる。その認証手順について図 1 に示す。

表 1　方式 A・B の比較

| | 方式 A | 方式 B |
|---|---|---|
| 学内者 | 利用可 | 利用可 |
| 訪問者 | 利用不可 | 利用可 |
| 一時アカウント | 不要 | 必要 |

### 3.3 SSO 実装のための二つの方式

著者らは既に HINET2007 のようなスイッチベースの認証ネットワークへ SSO 機能を導入する方式を二種類提案している。

**方式 A**：JavaScript を利用し、ページの遷移や認証情報の送信を制御してネットワークの利用者認証と SSO を連続して行う。特定の LDAP(IdP) 上に認証情報が存在する利用者のみを対象とした方式。

**方式 B**：通常の Shibboleth SSO で認証許可された利用者に対してネットワーク認証用のアカウントを動的に生成し、ネットワークの利用者認証を行う方式。

方式 A は下記の手順で行われる。動作手順を図 2 に示す。

**(a1)**　ブラウザを起動し任意の URL へのアクセス
任意の URL へアクセスすると認証のためのページにリダイレクトされる。

**(a2-3)**　SSO を利用するためのリンクの選択
リンクを選択すると、JavaScript で書かれた SSO 専用ページが返る。

**(a4)**　認証情報の認証スイッチへの送信
認証情報を入力して送信すると、JavaScript の非同期通信（Ajax）によって認証スイッチへ送信される。

**(a5)**　SSO のための認証情報が送信する
認証スイッチでの認証を確認後に IdP へ認証情報が送信される。SP へ認証情報を通知するリダイレクトページが返る。

**(a6)**　認証結果の SP への通知
SP で利用権限が確認され、認証完了を通知するページが返る。

方式 A では JavaScript による制御で、入力された認証情報を Ajax よるネットワーク利用者認証と HTTP

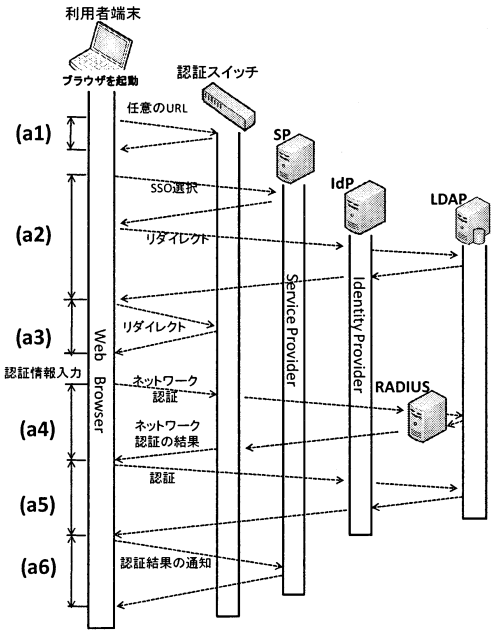


図 2　方式 A の処理フロー

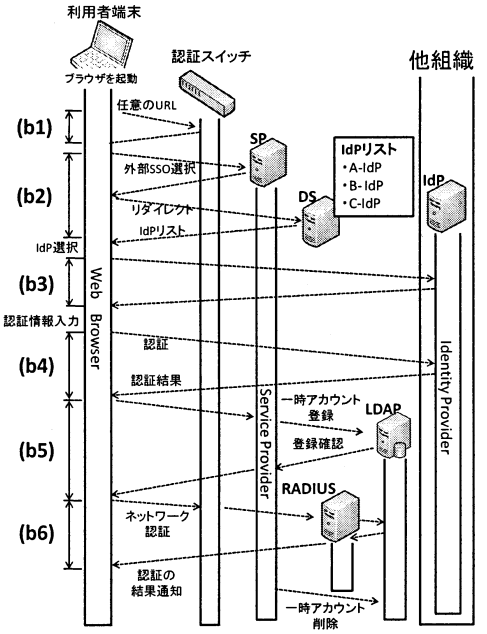


図 3　方式 B の処理フロー

による SSO 認証の 2 回の認証で使用して、ネットワークの認証の手順の中に SSO を組み込んでいる。このようにして、ネットワークの利用者認証の際に SSO が自動的に行われる。

方式 A のメリットとして、認証情報の送信を制御する SSO のための専用ページと、SSO のための Shibboleth-SP、IdP の設置のみで実装が可能であり、導入のコストが小さい点にある。

方式 B は下記の手順で行われる。動作手順を図 3 に示す。

**(b1)** ブラウザを起動し任意の URL へアクセス
任意の URL へアクセスすると認証のためのページにリダイレクトされる。

**(b2)** SSO を利用するためのリンクを選択
リンクを選択しサービスを提供すると、DS へリダイレクトされる。

**(b3)** 自組織の選択
DS で IdP を選択すると IdP から認証情報を入力するページが返る。

**(b4)** 認証情報の送信
IdP で認証すると、SP へ認証情報を通知するリダイレクトページが返る。

**(b5)** SP へ認証結果の通知と一時アカウント登録
SP で利用権限が確認され、一時アカウントが登録される。その認証情報を認証スイッチに送信するリダイレクトページが返る。

**(b6)** 一時アカウントによるネットワークの認証
一時アカウントの情報が認証スイッチに送信され、認証完了を通知するページが返る。

方式 B では SSO で認証の確認できた利用者に一回ネットワークの認証をする間だけ有効なアカウント（以後、一時アカウント）を発行して、その認証情報を自動的に送信することでネットワークの認証を提供する。HTTP リクエストはプロトコル上の問題からリダイレクトによるデータ送信が制約される。そのため、この方式では HTML に書き込まれた認証情報を JavaScript を利用することで自動的に送信している。この際、HTML に書き込まれた情報から真のアカウントとパスワードの漏えいすることを回避するために、一時アカウントを利用している。この一時アカウントは登録されて一定時間の経過後に CGI とは別のプロセスによって削除することで LDAP に無効なアカウントが溜まらないようになっている。

方式 B のメリットとして、組織間で認証連携された認証システムの SSO をベースにすることで、広島大学にアカウントを持たない他組織の構成員への認証提供できる点にある。その際の利用者の情報は、GakuNin 内でユニークに定められている eduPersonTargetedID というハッシュされた匿名の属性情報のみを SP に要求することで、プライバシに関しては一定の配慮が可能である。また、不正利用などで何らかの調査が必要な場合は、利用者の組織へ IdP のログの開示を請求することで、不正利用者の追跡が可能となる。一方で、

表 2　測定に使用したマシン

| | |
|---|---|
| SP | |
| CPU | Intel(R) Pentium(R) 4 CPU 1.90GHz |
| Memory | 1,035,192 kB |
| OS | CentOS (2.6.18-164.11.1.el5) |
| Package | Shibboleth-SP,Tomcat,Apache, LDAP(一時アカウント) |
| IdP | |
| CPU | Intel(R) Pentium(R) 4 CPU 1.60GHz |
| Memory | 644,900 kB |
| OS | CentOS (2.6.18-164.11.1.el5) |
| Package | Shibboleth-IdP,Tomcat,Apache, LDAP(認証用アカウント) |
| DS | |
| CPU | Intel(R) Pentium(R) 4 CPU 1.60GHz |
| Memory | 644,900 kB |
| OS | CentOS (2.6.18-164.11.1.el5) |
| Package | Shibboleth-DS,Tomcat,Apache |
| RADIUS | |
| CPU | Intel(R) Pentium(R) 4 CPU 2.80GHz |
| Memory | 507,252 kB |
| OS | CentOS (2.6.18-164.11.1.el5) |
| Package | FreeRADIUS |
| Client × 4 | |
| CPU | Intel(R) Core(TM) i7 CPU 920 @ 2.67GHz |
| Memory | 6,125,536 kB |
| OS | CentOS (2.6.18-164.11.1.el5) |
| Package | Jmeter |
| 操作 PC | |
| CPU | Intel(R) Atom(TM) CPU N270 @ 1.60GHz |
| Memory | 1,019,384 kB |
| OS | Ubuntu 9.10 ( 2.6.31-22-generic) |
| Package | dsh |
| 認証 SW | |
| Product | Alaxala AX2400S |
| CPU | PowerPC 533MHz |
| Memory | 262,144kB |

eduPersonTargetedID の属性情報はログから即座に利用者を判別したい場面では有効とはいえない。そこで、自組織の IdP などアカウント情報をそのまま利用できる場面では、SP へ eduPersonTargetedID ではなく eduPersonPrincipalName の属性情報を要求しすることで可読なアカウント名（学籍番号など）を利用することで、ログによる利用者管理を容易にすることも可能である。

## 4. 性能評価

HINET2007 で SSO 機能を安定的に運用するためには、認証要求の集中時の処理性能が重要になってくる。実際に HINET2007 では、始業時間帯の事務端末の稼動開始に伴う集中した認証要求が起きている。利用状況に関する事前調査（H.22.7.20）の結果からも一日の認証要求集中ピークとして 8 時 20 分〜35 分の間（15 分間）で約 700 台の端末からの認証要求があ

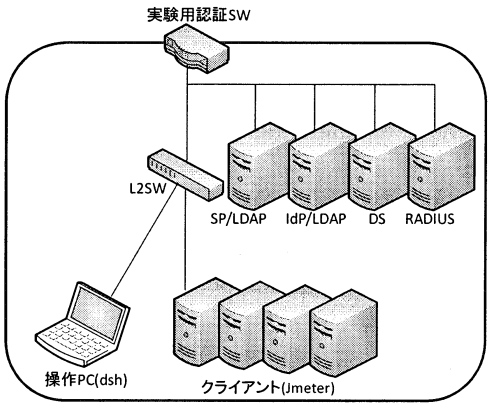


図 4　測定環境

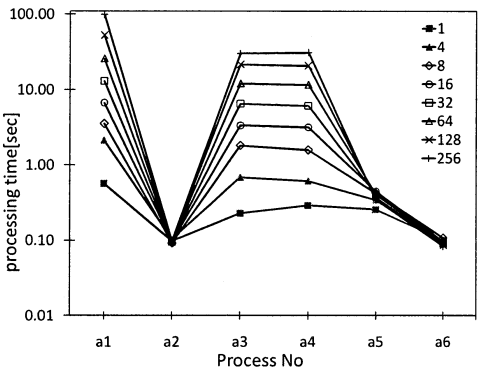


図 5　方式 A の性能測定

ることがわかっている。このような業務日に見られる集中した認証の処理が HINET2007 で運用する上では求められる。そこで本節では、この値を目安として方式 A・B、それぞれの SSO 機能の認証要求の集中時の処理性能について示す。測定のために使用したマシンスペックを表 2 にネットワーク構成を図 4 に示す。

### 4.1 方式 A の処理性能の測定

まず、方式 A の性能測定について述べる。二つの測定を行った。一つ目に、アクセス集中時の方式 A の処理時間の内訳を明らかにするために、一台の認証スイッチに接続された複数のクライアントが図 2 に沿った動作を同時にした時の各プロセスの処理時間を測定した。二つ目に、アクセス集中時の Shibboleth の処理時間を明らかにするために、SSO のための認証を複数のクライアントが同時にした時の処理時間を測定した。測定はクライアント PC で Jmeter[*1] を起動し、図 2 に沿った動作をするシナリオファイルを実行させて HTTP リクエストの送信から HTTP レスポンスが

*1 http://jakarta.apache.org/jmeter/

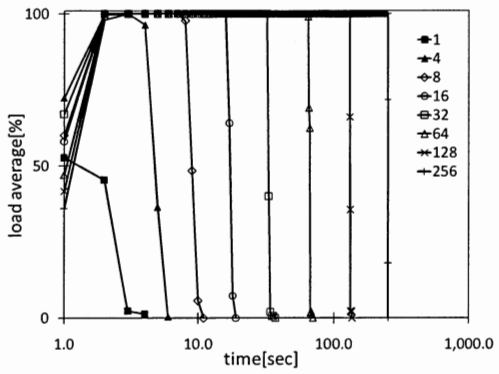


図 6　方式 A における認証スイッチの CPU 負荷

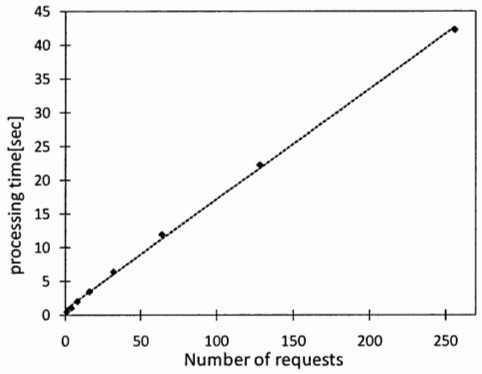


図 7　Shibboleth の処理（a2、a5、a6）にかかる時間

戻るまでの時間を測定した。認証用アカウントとして LDAP には事前に 1000 アカウント登録した。同時の認証要求で 4 以下のクライアントの実験では実機によるシミュレーション、それ以上の実験では Jmeter の擬似クライアントを複数生成する機能で 4 台の実機で 8、16、32、64、128、256 クライアントをシミュレーションした。クライアントに対して同時に制御コマンドを実行するために dsh[*2] を操作 PC から使用して 4 台の実機上で Jmeter を同時起動させた。Jmeter の設定で各アクセスでの HTTP レスポンスのタイムアウトは 120 秒に設定した。1 から 256 クライアントまでの各クライアント数で 10 回ずつ試行した測定値の平均値を図 5 に示す。縦軸は対数で HTTP レスポンスが戻るまでの時間、横軸にプロセスの番号を示している。処理が完了した否かは、HTTP レスポンスより判断した。この測定において 1〜128 クライアントの同時認証要求で設定した時間 (120 秒) で各プロセスは処理された。しかし、256 クライアントのときは 19 %の確率で（a1）がタイムアウトしており測定不能であった。その他の箇所で、タイムアウトや処理の失敗はなかったことを確認している。

測定の結果、128 クライアントからの同時認証を平均 96 秒、最大 130 秒で処理できることから、4 節の冒頭で述べた始業時間帯の認証要求（15 分間で約 700 の認証要求）に耐えられることが期待できる。また、内訳から処理時間で支配的なのは（a1、a3、a4）の認証スイッチの部分であることが分かった。一方で、SSO 機能のための Shibboleth の認証部分である（a2、a5、a6）の値は、どのクライアント数でもほぼ一定であった。これは、特に (a1) における処理で認証スイッチの処理部分でリクエストのフローが鈍化するためだと考えられる。ここで、別途測定した方式 A の性能測定中の認証スイッチの CPU 使用率を図 6 に示す。縦軸は CPU 使用率で、横軸は対数で認証要求の集中からの時間を示している。認証スイッチの負荷の増大が処理時間へ影響を与えたことが推測できる。

利用状況の事前調査から始業時間帯の 700 台の認証は 174 台の認証スイッチで処理されたことがわかっている。そのため、実環境では分散処理によって、リクエストのフローが図 5 よりも円滑に処理され Shibboleth へのアクセスがより密になることが想定される。その点を明確にするために、同様の方法で Shibboleth（a2、a5、a6）の処理における耐性を測定した。結果を図 7 に示す。128 クライアントの時の処理時間に注目すると、認証スイッチは (a1) のみでも 52 秒、Shibboleth は全体で 22 秒であった。この結果から、認証スイッチと比較して Shibboleth の処理時間の増加は緩やかである。認証スイッチへの負荷が集中した時と比較すると Shibboleth へ負荷が集中した時の処理時間の小さいことがわかる。

### 4.2　方式 B の処理性能の測定

次に、方式 B の性能測定について述べる。二つの測定を行った。一つ目に、アクセス集中時の方式 B の処理時間の内訳を明らかにするために、一台の認証スイッチに接続された複数のクライアントが図 3 に沿っ

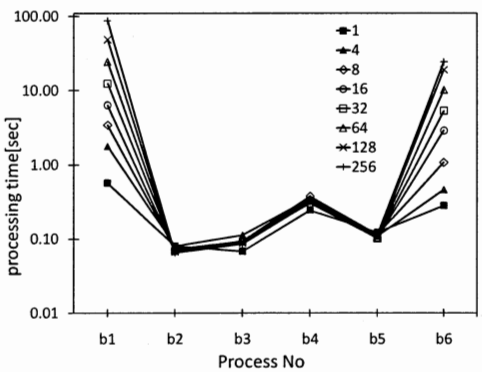


図 8　方式 B の性能測定

⋆2 http://www.netfort.gr.jp/ dancer/software/dsh.html

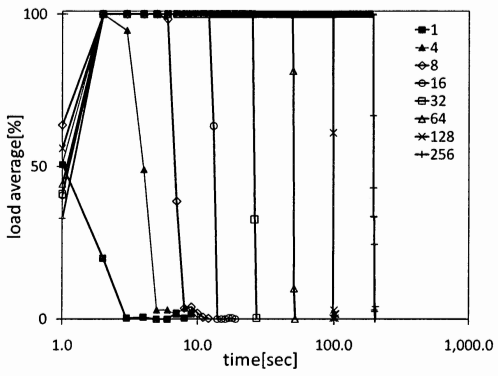


図 9　方式 B における認証スイッチの CPU 負荷

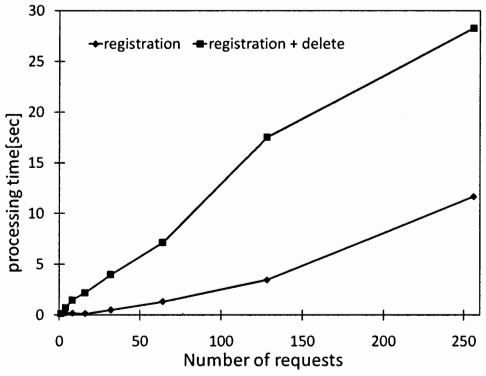


図 10　一時アカウント登録処理 (a5) の性能測定

た動作を同時にした時の各プロセスの処理時間を測定した。二つ目に、アクセス集中時の CGI の処理時間を明らかにするために、CGI に複数のクライアントが同時にアクセスした時の処理時間を測定した。方式 A の性能測定と同様の方法で図 3 に沿った動作をするシナリオファイルを実行させて測定した。一時アカウントの登録用の LDAP に登録してあるアカウントは試行前に全て削除している。また、測定は CGI で一時アカウントが登録される際に無効になった一時アカウントの削除の処理が生じない状況で行った。方式 B の性能測定の結果を図 8 に示す。縦軸は対数で HTTP レスポンスが戻るまでの時間、横軸にプロセスの番号を示している。処理が完了した否かは、HTTP レスポンスより判断した。

測定の結果、128 クライアントから同時認証を平均 68 秒、最大 100 秒で処理できることから 4 節の冒頭で述べた始業時間帯の認証要求に耐えられることが期待できる。また、処理時間の内訳から方式 B でも認証スイッチの認証に関わる（b1、b6）の部分が処理時間に大きく影響していることがわかった。そして、SSO 機能のために追加した（b2、b3、b4、b5）の部分はクライアント数に依らずほぼ一定であった。この場合も、特に (b1) の認証スイッチの処理でフローが鈍化しているためだと考えられる。ここで、別途測定した方式 B の性能測定中の認証スイッチの CPU 使用率を図 9 に示す。縦軸は CPU 使用率で、横軸は対数で認証要求からの時間を示している。方式 B においても認証スイッチの負荷の増大が処理時間へ影響を与えたことが推測できる。

また、複数の認証スイッチで認証管理がされている実運用環境を想定するにあたって方式 B では図 7 の結果に加えて CGI の高負荷時の処理時間も考慮する必要がある。そこで、(b5) で動作する一時アカウントを登録する CGI へアクセスが極端に集中した時の処理性能を測定した。測定は CGI で一時アカウントが登録される際に無効になった一時アカウントの削除の処理が起きない状況と、一時アカウントの登録の際に削除の処理が重なって起きる状況の二つの状況で測定した。前者は、試行毎に一時アカウント登録用 LDAP のアカウントは全て削除し、さらに一時アカウント削除のプロセスを実行しない状態での測定である。また後者は、事前に LDAP に 500 アカウント登録し有効期限の過ぎた一時アカウントの削除が起こるタイミングで CGI にアクセスした。この時、一時アカウントは 2 秒ごとに 10 アカウントずつ削除されている。それぞれの状況での性能測定の結果を図 10 に示す。CGI の処理が完了した否かは、HTTP レスポンスより判断した。

測定結果から、128 クライアントの結果に注目すると CGI の処理時間はアカウントの登録と削除が重ならない状況では約 3 秒、重なる状況では約 18 秒を要する。この値は極端に CGI へのアクセスが集中した時の処理時間の最良値と最悪値とみることができる。この結果より、必要に応じて一時アカウントの削除処理をコントロールし最良値に近づけることで CGI の処理の与える影響は大幅に小さくできることがわかった。

### 4.3　方式 A と方式 B の性能比較

方式 A は、一時アカウントの登録・削除に伴う処理時間が発生しないため、既に学内にアカウントを持つ利用者を対象とする SSO サービスとして設計した。しかし、測定の結果、両方式の処理時間に大きな差がないことが分かった。これは、方式 A で (a3) の SSO 専用ページを提供するため処理が認証スイッチの負荷を増加させたことが影響している。一方、当初懸念された (b5) の CGI 処理時間は、一時アカウントの削除処理を制御することで抑制できることが分かった。

以上の結果、より一般性のある方式 B のみを採用すれば良いことが判明した。ただし、学内利用者に対して IdP の選択画面を表示することは望ましくないため、サービス提供の際には方式 B を元に、DS で IdP 選択画面を提供する「訪問者用リンク」と、IdP を固定した「学内者専用リンク」を準備する予定である。

## 5. ま と め

本稿では、我々が提案しているHINET2007へSSO機能を導入するための二方式について概説し、実機を用いた実験により認証集中時の処理性能を評価した。性能評価の結果より、現在の広島大学のネットワークにおけるピーク時、すなわち15分間に700台の端末からの認証要求が生じたとしても二方式とも正常に処理ができることがわかった。したがって、両方式のどちらを用いたとしても実運用に耐えられることが期待できる。また、両方式の比較の結果、GakuNinによる認証連携に対応していること及び、処理性能に優位性があることから、方式BによるSSO機能の実現が望ましいと考えられる。今後は、方式Bの試験運用を通して、学内運用における有効性の検証を行い全学展開を行う予定である。

**謝辞** 本研究にあたってシステムの設計、実装の議論にご参加していただいた広島大学情報メディア教育研究センターの関係者に心から感謝いたします。特に、本システムの試験運用にあたり、貴重な時間を割いてサーバ等の構築に協力して下さいました、近堂徹先生に心から感謝の気持ちと御礼を申し上げたく、謝辞にかえさせていただきます

## 参 考 文 献

1) Internet2 Middleware Architecture Committee for Education (MACE) Directory Working Group, http://middleware.internet2.edu/dir/
2) 国立情報学研究所 UPKI イニシアチブ：学術認証フェデレーション, https://upki-portal.nii.ac.jp/SSO
3) 相原玲二, 西村浩二, 岸場清悟, 田島浩一, 近堂徹：利用者認証機能を持つ大規模キャンパスネットワークの構築, 電子情報通信学会2008年総合大会 BS-8-7, pp.116-117 (2008).
4) 藤村喬寿, 西村浩二, 相原玲二：大規模キャンパスネットワークにおけるSSO認証の設計と実装, 電子情報通信学会研究報告. IA. インターネットアーキテクチャ, Vol.109, No.299, pp. 13-18 (2009).
5) 藤村喬寿, 田島浩一, 大東俊博, 西村浩二, 相原玲二：学術認証フェデレーションに基づくキャンパスネットワークの認証機構, 情報処理学会研究報告. IOT. Vol.2010-IOT-8, No.37, (2010).
6) 大谷誠, 江藤博文, 渡辺健次, 只木進一, 渡辺義明：シングルサインオンに対応したネットワーク利用者認証システムの開発, 情報処理学会論文誌. Vol.51, No.3, pp.1031-1039, (2010).
7) 国立情報学研究所 UPKI イニシアチブ：Web アプリケーションのシボレス化, https://upki-portal.nii.ac.jp/docs/fed/technical/sp/WebApp/
8) 国立情報学研究所 ネットワーク運営・連携本部 認証作業部会 eduroam グループ：http://www.eduroam.jp/
9) 山田一郎, 鈴木孝明, 大和純一, 若山永哉, 後藤英昭, 曽根秀昭：セキュアかつ低コストなキャンパスローミングを実現するための代理認証, 電子情報通信学会研究報告. IA. インターネットアーキテクチャ, Vol.109, No.299, pp.37-40, (2009).