認証・検疫ネットワークに対するケーパビリティを用いた接続制御システムの構築

馬渕充啓[†1] 松井慧悟[†1] 高田真吾[†1]
小沢健史[†1] 豊岡拓[†1] 佐藤聡[†1]

持ち込み PC のネットワークの接続に関して，利用者の利便性を向上させることと管理者による管理コストを低減することを同時に実現することは非常に難しい．従来の手法では，利用者が持ち込み PC を用いてネットワークに接続する場合，利用者を特定するために管理者は利用者情報の登録等の管理作業を行う必要がある．これらの管理作業は，主にその組織の情報管理部門が一括して行うため，利用者に対してネットワークへの接続許可を出すには手間がかかる．我々は，利用者に対して接続許可を出す権限を，ネットワーク管理者から利用者へ委譲することが可能なネットワーク接続制御システムを提案する．本システムの特徴は，ネットワークの接続制御に利用者認証ではなくケーパビリティを用いていることである．本システムでは，ケーパビリティに対する全ての操作を Web ブラウザから行うことが可能であり．利用者は特別な設定をすることなく持ち込み PC からネットワークに接続することが可能である．最後に，このシステムを筑波大学システム情報工学研究科コンピュータサイエンス専攻所属のソフトウェア研究室においてテスト運用した結果について報告する．

Implementation of a Capability-based Connection Control System for an Authentication and Remediation Network

Mitsuhiro Mabuchi,[†1] Keigo Matsui,[†1] Shingo Takada,[†1]
Tsuyoshi Ozawa,[†1] Hiraku Toyooka[†1] and Akira Sato[†1]

When guests connects their client PCs to a network, it is a challenging task for network administrators to achieve user-friendliness while reducing administrators' efforts. In conventional systems, when a guest user connects his/her client PC, an administrator needs to perform some administrative operations including registration of a name, address, and email address to identify the guest user. If these administrative operations are delayed, the guest user cannot start working with a host user.

We propose a network connection control system that enables administrators to delegate a right to connect to the network to host users. The key feature of our system is to use capability-based access control instead of authentication-based one in order to control network connections. Our system enables host and guest users to handle capabilities with Web browsers. (Guest users only require running web browsers to connect his/her client PCs to the network.) We report experimental results in a laboratory in Department of Computer Science, Graduate School of Systems and Information Engineering, University of Tsukuba.

1. はじめに

近年，ネットワーク利用者に対する空間的制限が無線 LAN や情報コンセントの設置数の増加によりなくなってきている．そのため，利用者に与えられた権限に応じてネットワーク接続を制御する仕組みが必要になってきている．現在，企業や大学等の大規模な組織では，認証・検疫ネットワークを用いて利用者を特定することによりネットワークへの接続制御を行っている．このとき，利用者は，自分の権限の範囲内でのみネットワークに接続することができる．たとえば，HTTP プロトコルでのみ通信可能な権限等である．

この接続制御方法では，利用者を認証するための情報を事前にシステムに登録する必要がある．そのため，持ち込み PC のネットワークの接続に関して，利用者の利便性を向上させることと管理者による管理コストを低減することを同時に実現することは非常に難しい．従来の手法では，利用者が持ち込み PC を用いてネットワークに接続する場合，利用者を特定するため

†1 筑波大学大学院システム情報工学研究科
Graduate School of Systems and Information Engineering, University of Tsukuba

に利用者情報の登録等の管理作業を行う必要がある．これらの管理作業は，主にその組織の情報管理部門が一括して行うため，利用者に対してネットワークへの接続許可を出すには手間がかかる．たとえば，大学では，共同研究者や学会開催等により外部の利用者が多数訪れ持ち込み PC をネットワークに接続することが頻繁にある．その度に，事前登録，あるいは，一時的なネットワークの設置などを行うことは管理者にとって非常に大きな手間となる．

本論文では，その様な管理にかかる手間を削減するために，接続許可を出す権限をネットワーク管理者から利用者に委譲可能にする接続制御システムを提案する．本システムの特徴は，ネットワークの接続制御において利用者認証ではなく，ケーパビリティを用いていることにある[1)5)4)6)8)10)12)]．本システムは，中央サーバ（ケーパビリティ管理，DHCP，そして，DNS）と外部ネットワークとの通信路を開閉する L3 スイッチから構成される．本システムでは，L3 スイッチのパケットフィルタリング機能を用いて，あるパケットがそのスイッチを通ることができるかどうかを検査することで通信路の開閉を行っている．このようなスイッチを本論文では，ゲートウェイ・スイッチと呼ぶ．

外部の利用者に対しては，内部のサーバにアクセスできないようにするなどの制限を与える必要がある．そのため，本システムでは，ケーパビリティの属性として宛先ポート番号，有効期限，そして，使用回数等を設定可能にする．また，悪質な利用に関しては，事後に利用者を特定できるようにする必要がある．そこで，本システムでは，ケーパビリティ利用時の利用者記録をとるための機能も提供する．

実験では，ケーパビリティを使用したネットワーク接続制御にかかる処理時間を計測した．また，本システムは，筑波大学システム情報工学研究科コンピュータサイエンス専攻ソフトウェア研究室で約 1ヶ月に渡りテスト運用を行った．それらの結果を利用して，本システムの評価を行った．

本論文は以下の章で構成される．2 章では，本システムに必要な機能とその実現方法について述べる．3 章では，本システムの概要について述べ，4 章では，その実装について述べる．5 章では，本システムを用いた実験とそのテスト運用について述べる．6 章で本研究の関連研究について述べ，7 章で本論文のまとめを行う．

2. 提案システムに必要な機能と実現方法

従来の接続制御システムでは，利用者認証を用いているため利用者毎に登録作業を行う必要がある[11)]．外部からの利用者に接続許可を出す場合は，身分を確認し一時的なアカウントを作成してそれを渡す運用が行われている．この場合，ネットワーク管理者の利用者登録・削除等の管理作業や外部の利用者によるアカウント作成のための手続き等の手間がかかる．

上述した手間を削減するために，本システムは，利用者に容易にネットワークへの接続許可を出すことができるようにすることを目指す．この場合，容易にとは，利用者登録やアクセス制御設定，そして，それに伴う利用者による利用のための手続き等の管理に必要な作業の手間を削減することを意味する．また，本システムは，管理権限を一般の利用者に委譲可能にしているため信頼できるコミュニティにおいて使用されることを想定している．信頼できるコミュニティとは，具体的には学校や研究室等である．

管理作業の削減を実現するために，本システムでは，利用者認証の代わりにケーパビリティを用いることでネットワークに対する接続許可があるかどうかを調べる．ケーパビリティとは，アクセス対象であるオブジェクトの識別子とそのオブジェクトに対して可能な操作から構成されるものである[1)5)4)6)8)10)12)]．ケーパビリティを用いることで，利用者の特定をしないため利用者登録を行う手間が省かれる．それに伴い，登録削除の手間も省くことができる．

本システムで扱うケーパビリティには，以下に示す 2 種類の権限がある．

- ケーパビリティ管理：新しいケーパビリティの作成やケーパビリティの編集・削除等の管理作業を行うことができる権限．
- ネットワーク接続：ネットワークに対して接続を行うことができる権限．

ケーパビリティの管理にかかる手間は，ケーパビリティ管理権限を分散することで複数の利用者で分担することが可能になる．ケーパビリティ管理の権限を持つケーパビリティを利用者に渡すことで，それを持つ全ての利用者はケーパビリティを自由に作成し配布することが可能性になる．これは組織のネットワーク管理ポリシーに反する可能性が高いため，本システムでは，ケーパビリティ管理とネットワーク接続の権限を分割することで管理権限を持つ利用者を選択することができるようにしている．

本システム に必要な機能を以下に示す．

- ケーパビリティの使用や管理を行うユーザ・インタフェース．
- ケーパビリティの検証を行う機能．

- 通信路の開閉を行う機能.
- ケーパビリティの利用を記録する機能.

本システムでは，ケーパビリティの使用や管理を行うインタフェースには，Web ブラウザを用いる．Web ブラウザは，ネットワークを利用する利用者にとって使い慣れた物であり，多くの PC に最初からインストールされているため特別な設定をすることなく使用することができる．また，ケーパビリティの入力方法として，Web ブラウザを用いたインターフェース以外に QR コードを用いる．これは，スマートフォン等のモバイル機器でケーパビリティを入力する手間を削減するためである．

ケーパビリティの検証を行う機能は，Web サービスとして実現する．ケーパビリティは，簡単に推測されないようにするために乱数を含んでいる[3)4)6)]．この機能は，Web ブラウザに入力されたケーパビリティがサーバに保持されているかどうかを検証する．本システムでは，ケーパビリティに宛先ポート，有効期限，そして，使用回数等の制限を付加することを可能にする．制限の検証は，ケーパビリティ自体の検証が終了した後に行う．

通信路の開閉に関しては，L3 スイッチを用いて行う．本システムでは，L3 スイッチのパケットフィルタリング機能を用いて，あるパケットがそのスイッチを通ることができるかどうかを検査することで通信路の開閉を行っている．このようなスイッチを本論文では，**ゲートウェイ・スイッチ**と呼ぶ．ケーパビリティとその制限の検証が全て通った場合，その PC の IP アドレスから ARP テーブルを用いて MAC アドレスを取得する．その後，その MAC アドレスを用いてスイッチに ACL を書き込む．

本システムでは，管理権限は管理者だけでなく，一般の利用者が保持する可能性がある．そのため，事後に利用者を特定するために，ケーパビリティの利用に関する記録を IP アドレスと MAC アドレスとともにデータベースに記録する．また，本システムでは，データベースに保存したログを管理画面で閲覧できるようにする．ケーパビリティの使用とは，ケーパビリティを用いたネットワーク接続，ケーパビリティの作成，編集，そして，削除等である．

3. 提案システムの概要

3.1 システムの構成

図 1 に本システムの構成を示す．本システムは，中央サーバとゲートウェイ・スイッチで構成される．中央サーバは，ケーパビリティ管理，DHCP，そして，

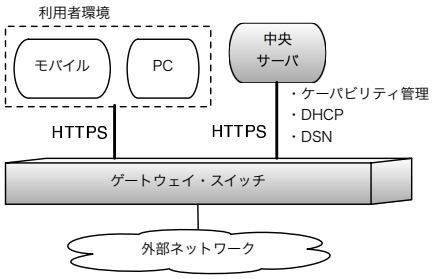

図 1 本システムの構成.
Fig. 1 The architecture of our system.

DNS を LAN 内のクライアントに提供する．LAN 内のクライアントが，中央サーバに自由にアクセス可能となるようにゲートウェイ・スイッチを設定する．また，LAN の外にはアクセスできないようにゲートウェイ・スイッチを設定する．外部ネットワークに接続する場合，クライアントは，ゲートウェイ・スイッチとを通る．また，中央サーバとクライアント間の通信が盗聴されケーパビリティが漏洩することを防ぐために，HTTPS を用いて通信を暗号化している．

ケーパビリティの使用や管理は，Web ブラウザを用いて中央サーバにアクセスすることで全て行うことができる．このため，利用者は特別な設定を行うことなく持参した PC やモバイル機器を用いて容易に利用することができる．

本システムの特徴として，中央サーバと通信路の開閉に用いているゲートウェイ・スイッチでは機能が分離していることが挙げられる．そのため，現在本システムでは，ハードウェアであるゲートウェイ・スイッチを用いて通信路の開閉を行っているが， iptables[2)] 等のパケットフィルタリングを行うソフトウェアに変更することも可能である．

3.2 ケーパビリティの階層

図 2 にケーパビリティの階層構造について示す．ケーパビリティは，管理者と管理権限ありのケーパビリティを保持する利用者の両方によって作成される．管理者に作成されたケーパビリティは，階層の最も上に位置する（図 2 の Capability A）．このようなケーパビリティを**ルート・ケーパビリティ**と呼ぶ．

管理権限ありのケーパビリティを保持する利用者に作成される全てのケーパビリティは，作成される際に入力された元のケーパビリティよりも制限される．たとえば，Capability B，C，E，そして，F は，それぞれ別のケーパビリティを元に作成されているためそれらよりも権限が制限されている．このように制限が増すことで，元のものよりも権限が減ったケーパビリ

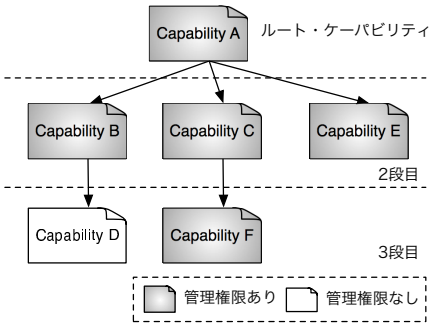

図 2 ケーパビリティの階層構造.
Fig. 2 The hierarchy of capabilities.

ティのことを**弱いケーパビリティ**と呼ぶ．これにより，利用者が階層的にケーパビリティを作成することができるため，管理者の管理作業を減らすことが可能になる．たとえば，大学等での研究室内部ネットワークについて考える．まず，管理者が先生に対して無制限の Capability A を作成し渡す．次に，先生がこの Capability A に有効期限を付加した Capability B，C，そして，E を作成し学生達に渡す．さらに，Capability B を保持した学生がそれに使用回数制限を付加し管理権限をなしにした Capability D を作成し外部の利用者に渡す．また，Capability C を保持した学生がさらに有効期限を制限した Capability F を作成し別の学生に渡す．このようにして，ケーパビリティ作成の権限を制限しながら下位層の利用者に委譲することができるため，管理作業を管理者や他の利用者間で分担することが可能になる．管理権限なしのケーパビリティは，ケーパビリティの作成を行うことはできないため，階層の最も下に位置する．

3.3 ケーパビリティの管理

ここでは，管理者によるケーパビリティの作成方法と管理権限ありのケーパビリティを保持する利用者によるケーパビリティの作成方法について述べる．ケーパビリティの本体は，偽造を防ぐため乱数を用いて生成している[3)4)6)]．ケーパビリティの管理操作として以下のものがある．

- **作成**：新しいケーパビリティを作成する．管理者に作成される場合，ルート・ケーパビリティが作成される．管理権限ありのケーパビリティを保持する利用者に作成される場合，元のケーパビリティの下に位置するケーパビリティが作成される．ケーパビリティが作成される時，同時にこのケーパビリティを含んだ URL から QR コードも作成される．
- **編集**：ケーパビリティの制限等を編集する．ただし，元のケーパビリティの制限を越えた編集はできない．たとえば，元のケーパビリティの使用回数が１０回であるとき，その下に位置するケーパビリティの使用回数を１０回より大きくすることはできない．
- **削除**：ケーパビリティを削除する．また，そのケーパビリティよりも下位層に位置する全てのケーパビリティが削除される．削除されたケーパビリティは使用できなくなるため，削除することでケーパビリティを無効化することができる．

上述した管理操作について，本システムのスクリーンショットを用いて詳しく説明する．図 3 には，管理者によるケーパビリティの管理を行う際のスクリーンショットを示す．この管理画面にいくには，管理者としての ID とパスワードを入力する必要がある．管理者の場合，全てのルート・ケーパビリティとそれらから派生したケーパビリティが表示される．図 4 には，管理権限ありのケーパビリティを保持する利用者によるケーパビリティの管理を行う際のスクリーンショットを示す．この管理画面にいくには，管理権限ありのケーパビリティを入力する必要がある．管理権限ありのケーパビリティを保持する利用者の場合，入力されたケーパビリティと，それよりも下位層に位置するケーパビリティ全てが表示される．図 4 の上部に表示されているケーパビリティが，入力されたケーパビリティである．また，自分のケーパビリティに対しては操作（編集，削除等）を行うことができないようになっている．

各ページの下部にある ”ケーパビリティの作成” リンクをクリックすると，ケーパビリティの制限を入力するページ（図 5）が表示される．まず，管理権限ありのケーパビリティを作成する場合は，チェックボックスにチェックをする．チェックがない場合，管理権限なしのケーパビリティが作成される．制限を入力後，ページの下部にある ”作成” リンクをクリックすることでケーパビリティの作成が完了する．元のケーパビリティに制限がある場合，入力された制限と元のケーパビリティの制限を比較し，制限が強くなっていれば作成を許可する．しかし，弱くなっている場合は，作成を拒否し再度制限の入力を求める．

ケーパビリティに付加された制限を編集する場合，ケーパビリティの右側に表示されている ”編集” リンクをクリックする．すると，制限を入力するページ（図 5 とほぼ同じ.）が表示されるので，そこに新しい制限を入力し編集の ”確定” リンクをクリックすること

図 3 管理者用管理ページのスクリーンショット.
Fig. 3 The screenshot of the manager's page.

図 4 利用者用管理ページのスクリーンショット.
Fig. 4 The screenshot of an inner member's page.

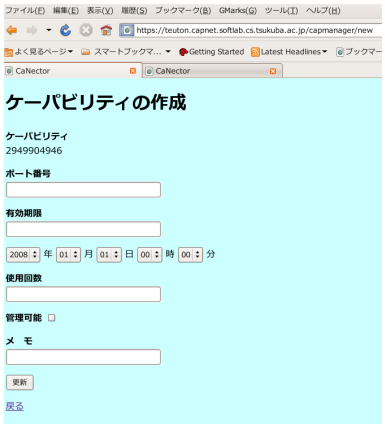

図 5 ケーパビリティ作成ページのスクリーンショット.
Fig. 5 The screenshot of the page for creating capabilities.

で制限の編集が終了する．ここでも，入力された制限が元のケーパビリティの制限より強くなっているかどうかを検査する．本システムでは，有効期限や使用回数制限等の制限があり，特に使用回数制限の編集は繊細であるため，編集された制限は下位層のケーパビリティに伝播しないようになっている．たとえば，ある親ケーパビリティの使用回数制限が１０回であり，それぞれ使用回数制限が５回の子ケーパビリティが２個ある場合を考える．ここで，親ケーパビリティの使用回数を１回減らした場合，どちらの子から１回減らせばよいかということは本システムでは判断できない．そのため，制限の検証を行う再帰的に行うことで，あるケーパビリティが上位のケーパビリティよりも弱い制限で利用されることを防いでいる．また，削除する場合，各ケーパビリティの右側にある ”削除” リンクをクリックすると削除確認が表示され，OK を選択すると削除される．ケーパビリティを削除した場合，そのケーパビリティは使用できなくなる．また，削除されたケーパビリティの下位層に位置する全てのケーパビリティが削除されるため，それらのケーパビリティも使用できなくなる．

3.4 ケーパビリティの使用方法

本システムでは，ケーパビリティを用いたネットワーク接続方法として以下の 2 種類を用意する．

- Web ページ入力.
- QR コード.

Web ページ入力を用いる場合，利用者は，本システムのトップページの入力フォームにケーパビリティを入力し ”connect” ボタンをクリックする．すると，システムにケーパビリティが送信され検証が行われる．入力されたケーパビリティに上位のケーパビリティがある場合，再帰的に行われルート・ケーパビリティにまでさかのぼって検証される．全てのケーパビリティが存在し制限の検証が成功した場合，システムはスイッチに ACL を記述する．その後，図 6 のページが表示される．このページには，利用者が入力したケーパビリティ，ポート番号，有効期限，そして，使用回数等の情報が表示される．

QR コードを用いてケーパビリティを使用する場合，その QR コードを Web カメラやモバイル機器のバーコード読み取り機器を用いることで，ケーパビリティが入力された URL を取得することができる．この URL にアクセスすることで，ケーパビリティが中央サーバに送信される．後のシステムの動作は，上述した場合と同じである．

ネットワーク接続を切断する場合，図 6 の下部にある切断リンクをクリックする．すると，システムはス

図 6 接続後のスクリーンショット.
Fig. 6 The screenshot after connecting.

イッチに記述された ACL を消す．その後，切断されたことがブラウザに表示される．切断後，利用者は，外部のネットワークにアクセスすることはできなくなる．

4. 提案システムの実装

ケーパビリティ管理システムは，Ruby on Rails を用いて実装した．

4.1 データ構造

本システムで扱っているケーパビリティのデータ構造を以下に示す．

- **ケーパビリティ**：ケーパビリティを表す．
- **上位のケーパビリティ**：元のケーパビリティを表す．ルート・ケーパビリティの場合は，”/”が保存される．
- **宛先ポート番号**：通信可能なポート番号を表す．
- **有効期限**：ケーパビリティの有効期限を表す．
- **使用回数**：ケーパビリティの使用回数を表す．
- **階層**：そのケーパビリティまでの階層を表す．たとえば，図 2 を用いて説明すると，最上位のケーパビリティの場合，”/capabilityA”となり，3 段目の場合，”/capabilityA/capabilityB/capabilityD”となる．
- **フラグ**：管理権限の有無を区別するために使用する．
- **メモ**：そのケーパビリティについての備考を表す．

本システムで扱っているログのデータ構造を以下に示す．

- **ケーパビリティ**：使用されたケーパビリティを表す．
- **日時**：ケーパビリティが使用された日時を表す．
- **IP アドレス**：利用者の使用している IP アドレスを表す．
- **MAC アドレス**：利用者の使用している機器の MAC アドレスを表す．
- **操作対象のケーパビリティ**：ケーパビリティの作成，編集，そして，削除等の操作を行ったの対象となったケーパビリティが記録される．
- **操作**：ケーパビリティを用いて行った操作を表す．

4.2 機　　能

ケーパビリティ管理サーバは，ケーパビリティに対して以下の操作を行う．

- **ケーパビリティの作成**：ケーパビリティとして乱数の自動生成を行い．入力されたポート番号，有効期限，そして，使用回数等の値とともにデータベースに保存する．制限されたケーパビリティを作成する場合，元のケーパビリティよりも制限が強くならないといけないため，入力された制限がデータベースに格納される際に元のケーパビリティの制限よりも強くなっているかどうかを検証する．制限が強くなっている場合，ケーパビリティを作成しデータベースに格納する．また，その際に上位のケーパビリティが識別できるようにするために元のケーパビリティとそのケーパビリティまでの階層も同時に保存される．
- **ケーパビリティの検証**：入力されたケーパビリティが，データベース内にあるかどうかを調べる．存在した場合は操作を続行し，存在しない場合は権限なしとみなし操作を中断する．また，上位のケーパビリティが存在する場合，ルート・ケーパビリティまで順にデータ・ベース内に存在するかどうかを確認する．
- **制限の検証**：ケーパビリティの検証が終わった後，ケーパビリティに付加された制限を取り出す．有効期限の場合，現在時刻と比較し現在時刻よりも大きければ有効であるとみなす．使用回数の場合，0 より大きければ有効とみなし現在の値から 1 減らした値を保存する．また，上位のケーパビリティが存在する場合，ルート・ケーパビリティまで順に同様にして制限を検証する．
- **スイッチに対する ACL の記述**：ケーパビリティと制限の検証が通った場合，ARP を用いて IP アドレスから MAC アドレスを取得する．その後，スイッチに Telnet を用いてアクセスを行い ACL に ACCEPT を記述する．その時，ポート番号の制限がある場合，そのポート番号のみ ACCEPT するような ACL を記述する．

4.3 ケーパビリティ利用記録の取得

本システムでは，ケーパビリティ管理の権限を保持するのが管理者だけではないため，SYSLOG ではな

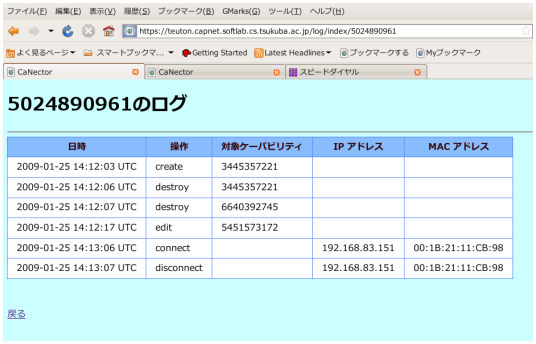

図 7 ログ表示ページのスクリーンショット.
Fig. 7 The screenshot of the page for displaying logs.

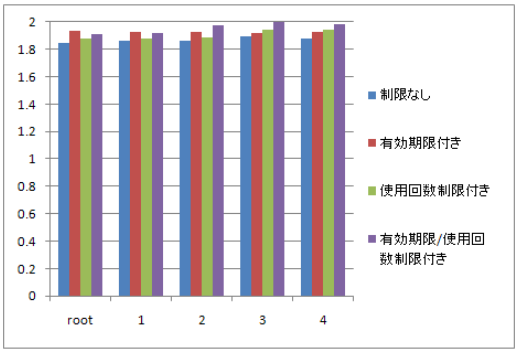

図 8 接続処理にかかる時間.
Fig. 8 Processing times of "connect" operation.

く通常の利用者でもアクセス可能なデータベースに利用記録を保存する．また，ケーパビリティ管理画面においてケーパビリティ毎のログを表示するようにする．

ログの表示画面を図 7 に示す．ログは，日時を用いてソートされた順で表示される．ケーパビリティに対する操作（create，destroy，edit）の場合，ログには操作対象のケーパビリティが表示される．ケーパビリティを用いてネットワーク接続の操作（connect，disconnect）を行う場合，ログにはその PC の IP アドレスと MAC アドレスが表示される．

4.4 スイッチの ACL の消去方法

現在，実現している方法として以下の 2 つがある．

- **明示的な消去**：利用者が切断を明示的に指定することで，システムがスイッチの ACL を消去する．
- **時間的な消去**：利用者がケーパビリティを用いてスイッチに ACL を書き込んだタイミングで，クロンに一定時間経過後にその ACL を消去するタスクを追加する．一定時間経過後，クロンはそのタスクにしたがいスイッチの ACL を削除する．
- **有効期限による消去**：利用者がケーパビリティを用いてスイッチに ACL を書き込んだタイミングで，クロンに有効期限終了後にその ACL を消去するタスクを追加する．有効期限が終了し次第，クロンはそのタスクにしたがいスイッチの ACL を削除する．

5. 実験とテスト運用

5.1 マイクロベンチマーク

利用者がケーパビリティを入力し，接続操作が完了するまでの時間が長い場合，利用者にとっては大きな負担となる．節で述べたようにケーパビリティには階層があり，ケーパビリティの検証やその制限の検証は再帰的に行われる．そこで，ルート・ケーパビリティから 4 段目のケーパビリティまでの 5 種類のケーパビリティについて，それらの処理時間がどの程度かかるか調べるために，プログラムを用いて要求を行い応答が返ってくるまでの時間を計測した．また，以下の 4 つの場合に分けて計測を行った．

- **制限なし**：全てのケーパビリティに制限がついていない場合．
- **有効期限付き**：全てのケーパビリティに有効期限が付加されている場合．
- **使用回数制限付き**：全てのケーパビリティに使用回数制限が付加されている場合．
- **有効期限／使用回数制限付き**：全てのケーパビリティに有効期限と使用回数制限の両方が付加されている場合．

この実験で用いたクライアント・マシンと中央サーバの環境を表 1 に示す．また，要求を行うプログラムとして wget を用いた．

計測した結果を図 8 に示す．横軸はケーパビリティの階層を表しており，縦軸は処理時間を表している．処理時間の単位は秒で，これらの処理時間は１０回同じ実験を繰り返したものの平均時間である．また，Telnet でスイッチの操作を行う処理時間は，約 1.2 秒だった．

本システムの処理時間は 2 秒前後であり，ケーパビリティの階層が深くなるにつれ処理時間は大きくなる傾向があることがわかる．この時間は，スイッチに ACL を書き込むための処理時間であり，ネットワーク接続処理を行うことはそれほど多くはないため問題ない大きさだと言える．

5.2 テスト運用

本システムは，現在，筑波大学システム情報工学研究科コンピュータサイエンス専攻のソフトウェア研究室でテスト運用を行っている．主に無線 LAN に持ち込み PC やスマートフォン等のモバイル機器を接続

表 1 実験環境.
Table 1 Environments for the experiment..

	クライアント	中央サーバ
CPU	Pentium Core 2 Duo 2.2GHz	Pentium D 2.8GHz
メモリ	2.0GB	512MB
OS	Mac Ubuntu 8.04 (Linux Kernel 2.6.24)	Cent OS 5.2 (Linux Kernel - Xen 2.6.18)
LAN	Gigabit Ethernet	Gigabit Ethernet

表 2 利用者の評価.
Table 2 The evalution by users.

	平均点
接続制御システムとしての使い易さ	3.4
ケーパビリティ管理機能の使い易さ	2.7

して利用している．同時使用台数は 10 台程度で，約 1ヶ月間にわたって運用している．テスト運用中を行うことで，いくつかのプログラム上の間違いを修正することができた．

5.3 調 査 結 果

テスト運用期間中に本システムを使用した 7 名の利用者に対してアンケート調査を行い，以下の 2 項目に関して 4 段階（4 が最高で，1 が最低）で評価してもらった．

- 接続制御システムとしての使い易さ．
- ケーパビリティの管理機能の使い易さ．

各選択項目の点数の平均を表 2 に示す．接続制御システムとしては，一定の評価を得ることができた．しかし，ケーパビリティ管理機能としては，改善の余地があることが分かった．コメントとしては，ケーパビリティ作成の際に，元のケーパビリティの制限を表示するや， Javascript 等を用いて直感的に入力できるようなインタフェースを用意するといい等があった．

6. 関 連 研 究

既存の接続制御システムとして，利用者認証と利用者記録を行うゲートウェイシステムとして佐賀大学で開発された Opengate がある[11]．Opengate では，外部利用者に接続許可を出す場合，ネットワーク管理者が一時的なアカウントを用意しそれを外部利用者に渡す運用を行っている．本システムでは，ネットワーク管理者から接続許可を出すための権限を利用者に委譲しているため，管理者ではなく管理権限を持つ利用者が他の利用者に対して接続許可を出すことができる．これにより，ネットワーク管理者の手間や利用者による事前の手続きの手間を削減することができる．

システムが既に信頼している利用者が信頼する利用者に関しては，システムは間接的にその利用者を信頼するという信頼の輪モデルという研究がある[7]．このモデルを用いたシステムでは，登録された利用者に対して，新しい利用者を登録する権限を委譲することができる．新しく登録された利用者は，自分を登録してくれた利用者の権限よりも弱い権限でシステムを利用することができる．本システムの想定する使用環境は，このモデルの考え方と似ている．しかし，このシステムでは利用者は利用者認証に基づいてシステムにアクセスしているのに対して，本システムでは利用者を認証することはせずにケーパビリティを用いてシステムにアクセスすることを可能にしている．

ケーパビリティは初期のマルチプロセッサや分散オペレーティング・システムの研究で使用された．Hydra では，ケーパビリティはオブジェクトへの参照として使用された[12]．Mach では，ケーパビリティは通信ポートのアクセスを制御するために使用された[1]．Amoeba は，ユーザ・プロセスが一般的なプロセス間通信を用いてケーパビリティを受け渡すことを許可している[8]．近年，携帯電話のためのオペレーティング・システムとして開発された OS である Symbian OS は，資源の保護をするためにケーパビリティを使用する[10]．ケーパビリティの偽造を防ぐために，Hydra や Mach ではケーパビリティを OS カーネル内に格納し，Amoeba では一方向関数を用いて暗号化している，Symbian OS では，ケーパビリティは実行可能ファイルに組み込まれ変更することはできない．ケーパビリティは，OS だけでなくデータベースやファイルシステムのアクセス制御としても使用された．HomeViews では，データベースの view へのアクセス制御としてケーパビリティが用いられている[4]．

本システム は，ケーパビリティを乱数で表現することでその偽造を防いでいる[3][4][6]．また，本システムは，ネットワークに接続するためにケーパビリティを使用している．

7. ま と め

本論文では，認証・検疫ネットワークに対するケーパビリティを用いたネットワーク接続制御システムについて述べた．本システムの特徴は，ネットワークの接続制御において利用者認証ではなく，ケーパビリティ

を用いていることである．また，ネットワークへの接続許可を出すことができる権限を，ネットワーク管理者から利用者に委譲可能にしたことである．これにより，管理者でなくとも管理権限のある利用者であればケーパビリティを作成することができるため，他の利用者に対して容易に接続許可を出すことが可能になる．

現在，本システムは約 1ヶ月間のテスト運用を行っているが，システムの実用性を示すためにはより長く，そして，より多クくの利用者に利用してもらい意見を聴く必要がある．今後の課題としては，利用可能場所を拡大し多くの人が利用可能な環境を構築することと，利用者の意見を取り入れることによるシステムの改善である．また，ケーパビリティ管理画面において，ケーパビリティを木構造として表示することやアンケートのコメントにあった機能を実現することも今後の課題である．

参 考 文 献

1) M. Accetta, R. Baron, W. Bolosky, D. Rashid, A. Tevanian, and M. Young: ”Mach: A New Kernel Foundation for UNIX Development”, Proceeding of USENIX Summer Conference, pp.93-112, 1986.
2) O. Andreasson: ”Iptables Tutorial 1.2.0”, 2005.
3) M.Anderson, R.D.Pose, and C.S.Wallace: ”A Password-Capability System”, The Computer Journal, 29(1):1-8, 1986.
4) R.Geambasu, M.Balazinska, S.D.Gribble, and H.M.Levy: ”HomeViews: Peer-to-Peer Middleware for Personal Data Sharing Applications”, Proceedings of the 2007 ACM SIGMOD international conference on Management of data, pp.235-246, 2007.
5) H.M.Levy: ”Capability-Based Computer Systems”, Digital Press, 1984.
6) M. Mabuchi, Y. Shinjo, A. Sato, and K. Kato: ”An Access Control Model for Web-Services that Supports Delegation and Creation of Authority”, Proceeding of Seventh International Conference on Networking, pp.213-222, 2008.
7) 正岡元，菊池豊: ”信頼の輪モデルに基づいたシステム利用権限の委譲による個人認証手法”, 情報処理学会 分散システム/運用技術研究会, 2003-DSM-29, pp.51-56, 2003.
8) S. J. Mullender, G. Rossum, A. S. Tenenbaum, R.van Renesse, and H.van Staveren: ”Amoeba: A Distributed Operating System for the 1990s”, IEEE Computer, Vol.23, pp.44-53, 1990.
9) A. Silberschatz, P. B. Galvin, and G. Gagne: ”Operating System Concepts”, Wiley, 2008.
10) J.Stichbury, and M.Jacobs: ”The Accredited Symbian Developer Primer, Fundamental of Symbian OS”, John Wiley & Sons Inc, 2006.
11) 渡辺義明，渡辺建次，江藤博文，只木進一: ”利用と管理が容易で適用範囲の広い利用者認証ゲートウェイシステムの開発”，情報処理学会論文誌, vol.42, No.12, pp.2802-2809, 2001.
12) W. Wulf, E. Cohen, W. Corwin, A. Jones, R.Levin, C.Pierson, and F.Pollack: ”HYDRA: The Kernel of a Multiprocessor Operating System”, Communication of the ACM, Vol.17, No.6, pp.337-345, 1974.