情報処理学会論文誌 Vol. 51 No. 3 974–988 (Mar. 2010)

利用者間で接続権限を受け渡し可能なネットワーク制御機構の実現

馬渕充啓[†1] 高田真吾[†1] 小沢健史[†1]
豊岡拓[†1] 松井慧悟[†1] 佐藤聡[†2]
新城靖[†1] 加藤和彦[†1]

多くの大学では，情報コンセントや無線 LAN をいたるところから利用することができるようになりつつある．利用者がすべてのネットワークに対して接続するための権限（接続権限）を持っているとは限らないため，ネットワークを利用できないことがある．それらを利用できる利用者が接続権限を信頼できる他の利用者に手軽に渡すことができれば，他の利用者も手軽にそれらを利用することができる．本論文では，接続権限を利用者間で受け渡し可能なネットワーク制御機構 CaNector について述べる．我々は，ケーパビリティに基づくアクセス制御を用いて CaNector を実現している．CaNector で扱うケーパビリティは，管理権限（接続権限を管理するための権限）ありのケーパビリティと接続権限のみのケーパビリティに分類される．管理権限ありのケーパビリティを保持している利用者は，そのケーパビリティを元に手軽に新しいケーパビリティを作成し，信頼できる他の利用者に渡すことができる．CaNector は，筑波大学大学院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内で約 6 カ月間問題なく稼働している．

Implementation of a Network Control Mechanism that Enables Passing Access Rights among Users

Mitsuhiro Mabuchi,[†1] Shingo Takada,[†1]
Tsuyoshi Ozawa,[†1] Hiraku Toyooka,[†1]
Keigo Matsui,[†1] Akira Sato,[†2] Yasushi Shinjo[†1]
and Kazuhiko Kato[†1]

Information outlets and wireless LANs are becoming available everywhere in many universities. Since a user doesn't always have access rights for all the networks, the user sometimes cannot use a network. If a user who can use them can pass his or her access rights to other trusted users easily, other users can use them readily. This paper describes CaNector, which enables passing access rights among users. We realize CaNector by using capability-based access control. Capabilities in CaNector are classified into two types: the first one is a capability that includes both management rights and access rights, and the second one is a capability that includes only access rights. A user who has a capability that includes management rights can create new capabilities based on his or her own capability and then pass them to another trusted user easily. CaNector has been working for 6 months in Software laboratory in Department of Computer Science, Graduate School of Systems and Information Engineering, University of Tsukuba.

1. はじめに

現在，Web や電子メール等の利用が一般的になっており，さまざまな場所からそれらのツールを利用したい状況がある．たとえば，移動先でのメール確認，Web 上にある授業資料の閲覧，そして，会議中のスケジュール確認等の状況である．多くの大学では，情報コンセントと無線 LAN をいたるところから利用することができるようになってきつつある．しかし，利用者がそれらを用いてネットワークに接続するための権限（接続権限）を持っていないことがある．そのため，利用者がそれらを利用できない場面が少なくない．

利用者がその接続権限を取得しようとする場合，管理者にアカウント発行を申請する手間がかかるという問題がある．このような場合，望ましくない運用として，利用者間でアカウントを共有するというものがある．アカウントを共有して利用した場合，不適切な利用を行った利用者を事後に特定することができないという問題がある．また，そのアカウントが電子メール等の重要なサービスと共通の場合は，そのアカウントを渡すことができない．

利用者が手間をかけずに，手軽に接続権限を取得できるようにするためには，すでに利用できる利用者が接続権限を作成し信頼できる他の利用者に渡すことができればよい．本論文は，利用者間で接続権限を手軽に受け渡し可能なネットワーク制御機構 CaNector について述べる．従来手法で主に用いられている Access Control List（ACL）に基づくアクセス

†1 筑波大学大学院システム情報工学研究科コンピュータサイエンス専攻
Department of Computer Science, Graduate School of Systems and Information Engineering, University of Tsukuba
†2 筑波大学情報環境機構学術情報メディアセンター
Academic Computing & Communications Center, University of Tsukuba

制御[1] では，アクセス制御を実行するために利用者認証を行う必要がある．そのため，利用者は上述したようにアカウントを持つ必要があり，利用者は手軽にネットワークを利用することができない．これに対して，ケーパビリティに基づくアクセス制御を用いることで，アカウント発行を必要とせずケーパビリティを保持している利用者であれば誰でもネットワークの利用を可能にすることが実現できる．そこで我々は，ACL に基づくアクセス制御ではなくケーパビリティに基づくアクセス制御を用いて CaNector を実現する．ケーパビリティに基づくアクセス制御[2)–7)] には，あるケーパビリティを元に別のケーパビリティを作成できるという特徴と利用者間で受け渡しができるという特徴がある．これらの特徴を用いることで，接続権限を信頼できる利用者間で手軽に受け渡すことができる．CaNector で扱うケーパビリティには，管理権限（接続権限を管理するための権限）ありのケーパビリティ（接続権限も含む）と接続権限のみのケーパビリティの 2 種類がある．管理権限ありのケーパビリティを管理者から利用者に委譲することで，その利用者はそのケーパビリティを元に手軽に新しいケーパビリティを作成しそれを信頼できる他の利用者に受け渡すことができるようになる．

ケーパビリティに基づくアクセス制御では，ケーパビリティを保持していれば誰でも使用可能であるため，単純な実装ではケーパビリティの利用者を追跡することができないという問題がある．そこで，CaNector ではケーパビリティを作成する際に，ケーパビリティに配布先の利用者を示す情報を付加するようにしている．また，ケーパビリティの利用に関しては，データベースにログを保存するようにしている．これらにより，CaNector は，不適切な利用を行った利用者を事後に特定できるようにしている．

実験では，CaNector を使用した場合にネットワーク接続にかかる処理時間の計測を行いその有用性を示す．また，従来手法と受け渡しの手軽さについて比較する．CaNector 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 2008 年 12 月から約 6 カ月間にわたりテスト運用を行っている．最後に，CaNector のスケーラビリティやケーパビリティの配布についての考察を行う．

本論文は以下の章で構成される．まず，2 章では，問題解決における要求要件について述べる．3 章では，本機構で用いているケーパビリティに基づくアクセス制御について説明する．4 章では，要求要件に従い実現した CaNector について述べる．5 章では，ケーパビリティを管理するための主な機能について述べる．6 章では，CaNector の評価とそのテスト運用について述べる．7 章では，CaNector に関する考察について述べる．8 章では，本論文のまとめと今後の課題について述べる．

2. 問題解決における要求要件

2.1 想定する環境

ここでは，我々の想定する環境を示す．まず，大学のネットワークを管理・運営する人のことを**管理者**と呼ぶ．次に，そのネットワークを利用する人を**利用者**と呼ぶ．また，利用者は，**内部利用者**と**外部利用者**に分けることができる．内部利用者とは，その大学に属している利用者のことである．外部利用者とは，その大学に属していない利用者のことである．ネットワークに接続するための権限を**接続権限**と呼び，接続権限を管理するための権限を**管理権限**と呼ぶ．

大学内には数多くの情報コンセントや無線 LAN が設置されており，それらのゲートウェイとして L2 や L3 のルーティングポイントがある．ゲートウェイは，WAN につながっており，そこからインターネットへと出ていくことができる．利用者は，PC やモバイル機器等を情報コンセントや無線 LAN を用いてネットワークに接続することができる．また，授業や会議による内部利用者の移動が頻繁に起こり，学会等で外部利用者が頻繁に訪れてネットワークを利用することがある．大学のネットワークは，関係のある人間（教員，学生，事務員，学会等のイベントの参加者等）には利用させたいが，それ以外のまったく無関係の人間には利用させたくない．利用者は，大学のネットワークを利用して，Web サイトの閲覧，Web を用いた情報の検索，あるいは，メールの送受信等の作業を行いたい．

すべての利用者がすべての利用者を信頼しているのではなく，ある利用者は少数の他の利用者を信頼しているものとする．ある利用者が他の利用者を信頼し，またその利用者が別の利用者を信頼する．このように形成された信頼の連鎖で結ばれた利用者のグループによってネットワークは利用される．

2.2 想定環境における問題

2.1 節で述べた環境において，利用者が接続権限を手軽に取得できないという問題がある．筑波大学では，このことを示す以下のような状況があった．

(1) ある研究室は毎回同じ部屋でゼミを行っていた．その部屋の無線 LAN には，教員と学生全員が接続することができる．あるとき，諸事情によりゼミの部屋が変更された．変更された部屋の無線 LAN は，教員は接続できるが学生はアカウントが発行されていないため接続することができなかった．接続するためのアカウントを取得するにはその無線 LAN を管理している管理者に申請する必要があるため，学生は手軽に接続権限を取得することができなかった．このため，学生は Web 上にあるゼミの情

報を閲覧することができなかった．

(2) ある研究室に他大学に所属する共同研究者が訪れた．その研究者は，外部利用者なので筑波大学のネットワークの接続権限を保持しておらず，学内のネットワークを利用することができなかった．接続するためのアカウントを取得するには管理者に申請する必要があるため，接続権限を手軽に取得することができなかった．このため，共同研究者はメールの送受信や Web を用いた情報の検索等の作業を行うことができなかった．

(3) 筑波大学において応用物理学会が開催された．参加者は，4 日間で延べ 1 万人だった．このとき，参加者に対しては一時的なアカウントを発行することでネットワークの接続を許可するようにした．しかし，会場が広範囲であったにもかかわらずアカウントの配布箇所を 2 カ所しか設置できなかったため，アカウントを取得するために長い距離を移動する必要があり，参加者は手軽に接続権限を取得することができなかった．そのため，学会参加者は，Web 上にある学会の情報やメール送受信等の作業を行うことができなかった．

これらの状況においては，申請さえすればすべての利用者はアカウントを取得することができる．

2.3 従来手法

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[8]．また，利用者ごとに許可された接続権限を考慮したアクセス制御と情報コンセントの不正利用を防止するための手法がある[9]．そのほかにも，有線 LAN や無線 LAN が混在するマルチベンダ環境に対応するために開発された情報コンセントシステムがある[10]．これらのシステムでは，ネットワークを利用するためのアカウントを発行することができるのは管理者だけである．そのため，2.2 節で述べた例と同じ状況が起こると考えられるため，利用者は手軽に接続権限を取得することができない．

ジョージア工科大学で開発された LAWN では，上述した従来手法での問題を解決するために教員は一時的なアカウントを発行する権限を持っている[11]．教員が一時的なアカウントを発行し内部利用者や外部利用者に配布することが可能なため，2.2 節で述べた例においても利用者はある程度手軽に接続権限を取得することができる．Opengate，文献 9)，そして，文献 10) で述べられているシステムと比較すると，LAWN を用いた方が，利用者は手軽にアカウントを取得することができる．ただし，一時的なアカウントを受け取った利用者はネットワークに接続することはできるが，新たなアカウントを発行することはできない．そのため，2.2 節で述べた (3) の状況においては，学会の参加者はその大学に教員の知り合いがいない場合，手軽に接続権限を取得することはできない．

2.4 要求要件

利用者が手軽に接続権限を取得できるようにするためには，管理者だけでなく利用者が接続権限を作成できるようにする方法がある．近くにいる信頼できる他の利用者に接続権限を渡すことが可能な制御機構を実現するためには，以下の 3 つの要求要件を満たす必要がある．

(1) 利用者間で管理権限を受け渡すことを可能にする：2.2 節の例 (3) の場合に，ある参加者が信頼できる他の参加者に対して接続権限を作成し渡すことが可能になる．そのため，利用者間で管理権限を受け渡し可能にする必要がある．“受け渡す” とは，利用者が自分の権限の一部を他の利用者に利用可能にすることを意味する．利用者は，元々持っていた権限は引き続き利用できる．

(2) 管理権限を受け渡すかどうかを選択可能にする：2.2 節の例 (2) の場合に，共同研究者である外部利用者に管理権限を渡したくないことがある．このような場合に対応するために，管理権限を受け渡すかどうかを選択可能にする必要がある．

(3) 不適切な利用を行った利用者を事後に特定することを可能にする：接続権限を渡した利用者が不適切な利用を行った場合に，事後にその利用者を特定する必要がある．本論文でいう “不適切な利用” とは，スパムメールの送信や不正ログイン等の攻撃のことである．

LAWN は，上述した要求要件を部分的には満たすが不十分である．詳しくは，6.2 節で述べる．

3. ACL とケーパビリティ

ここでは，本機構で用いているケーパビリティに基づくアクセス制御について説明する．

まず，ACL に基づくアクセス制御とケーパビリティに基づくアクセス制御は，アクセスマトリックス（**図 1**）[1] を実現するものである．図 1 の縦軸はアクセスの主体（ユーザ等）を表しており，横軸はアクセスの対象（ファイル等）を表している．ACL は，図 1 中の荒い点線で囲まれたリストのことである．この ACL は，File B に対して，User 1 と 2 は “読み込み” が可能，User 3 は “読み書き実行” が可能であることを示している．このように ACL は，アクセスの主体の識別子とその主体が可能な操作の組で構成されており，オブジェクト側で保持される．そのため ACL では，アクセスの主体を識別する必要がある．

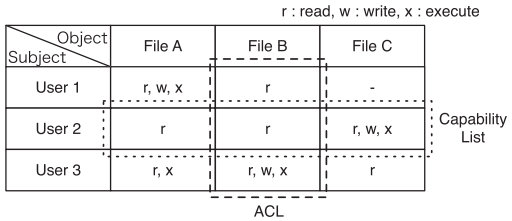


図 1　アクセスマトリックスの例
Fig. 1　An example of an access matrix.

ACL を用いている多くのシステムでは，認証を用いることでアクセスの主体を識別している．一方，ケーパビリティのリストは，図 1 中の細かい点線で囲まれたリストのことである．このケーパビリティのリストは，File A と B に対して “読み込み” が可能で，File C に対しては “読み書き実行” が可能であることを示している．このようにケーパビリティのリストは，オブジェトの識別子とそのオブジェクトに対して可能な操作の組で構成されており，アクセスの主体側で保持される．そのため，あるオブジェクトに対するケーパビリティを保持している利用者は，誰でもそのオブジェクトに対するアクセス権限を行使することができる．ケーパビリティでは，ACL とは異なり主体の認証を行う必要はない．

また，ケーパビリティには，以下の 2 つの特徴がある．

- あるケーパビリティを元にそれよりも操作の限定されたケーパビリティを作成することができる．
- ケーパビリティを利用者間で受け渡すことができる．

本研究では，ケーパビリティのこれらの特徴を用いることで，利用者間で手軽に接続権限を受け渡すことを可能にする．

ケーパビリティに基づくアクセス制御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．ケーパビリティに基づくアクセス制御を用いることで，プロセス間で権限の委譲を行うことが可能になる．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[2]．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[6]．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[7]．本研究では，利用者がネットワークを利用する際のアクセス制御にケーパビリティを用いることで，接続権限を利用者間で手軽に受け渡し可能にする．このように本研究では，従来の研究とアクセスの対象が異なる．

4. CaNector の概要

この章では，2 章で述べた問題を解決するための接続機構である CaNector の設計と実現方法について述べる．

4.1 CaNector の設計

2.4 節で述べた要求要件を満たすために，以下のように CaNector を設計する．要求 (1) を満たすために，CaNector は，利用者がネットワークを利用する際にケーパビリティに基づくアクセス制御[2)–7)] を用いる[⋆1]．ケーパビリティに基づくアクセス制御には，あるケーパビリティを元に別のケーパビリティを作成できるという特徴と利用者間で受け渡しができるという特徴がある．したがって，ケーパビリティを保持している利用者は誰でも新しいケーパビリティを作成し，信頼できる他の利用者に渡すことで要求 (1) を満たすことができる．信頼できる利用者にのみケーパビリティを配布することで，信頼の連鎖を構築できる．CaNector は，信頼の連鎖で構築されたグループ内でのみ使用されるべきである．ケーパビリティの配布に関しては，7.2 節で詳しく考察する．

要求 (2) を満たすために，CaNector で扱うケーパビリティには，管理権限あり（接続権限を含む）のケーパビリティと接続権限のみのケーパビリティの 2 種類を用意する．CaNector における管理権限とは，ケーパビリティの作成，編集，そして，削除等の管理作業を行うことができる権限のことであり，接続権限とはネットワークに接続するための権限のことである．管理権限ありのケーパビリティを保持している利用者は，元のケーパビリティの接続権限に基づいて誰でも新しいケーパビリティを作成することができる．したがって，接続権限のみのケーパビリティを保持する利用者はケーパビリティを作成することができないため，要求 (2) を満たすことができる．CaNector におけるケーパビリティに関しては，4.3 節において詳しく述べる．

要求 (3) を満たすために，CaNector では，まず，ケーパビリティを他の利用者に渡すときには，必ず自分のケーパビリティを元に新しいケーパビリティを作成するようにする．この結果，1 つのケーパビリティは，たかだか 1 人にしか利用されない．そして，ケーパビリティの利用に関する記録を IP アドレスと MAC アドレスを利用された日時とともにデータベースに記録するようにする．ケーパビリティの利用とは，ケーパビリティを用いたネット

⋆1 内部的には利用者から見えないゲートウェイにおいて，パケットごとのアクセス制御に関しては ACL を用いて行っている．詳しくは，4.2 節で述べる．

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．また，ケーパビリティを作成するときには，利用者を特定するための情報を付加するようにする．CaNector では，データベースに保存したログを管理画面で閲覧可能にすることで，どのケーパビリティを用いてどのような操作を行ったかを特定することができる．したがって，これらにより不適切な利用を行った利用者を事後に特定することができるため，要求 (3) を満たすことができる．CaNector における利用者の特定方法に関しては，4.4 節において詳しく述べる．

4.2　CaNector の構成

図 2 と**図 3** に CaNector の構成を示す．CaNector は，**ゲートウェイ**と**ケーパビリティ管理サーバ（CMS: Capability Management Server）**で構成される．

ゲートウェイは，パケットフィルタリング機能を持つスイッチ，または，ルータであり，あるパケットがそのゲートウェイを通ることができるかどうかを制御することで通信路の開放閉鎖を行う．図 2 と図 3 に示した専用スイッチも Linux ルータも現在の実装では，MAC レベルで通信路の開放閉鎖を行っている．CMS は，ケーパビリティの管理や利用機能，DNS，そして，DHCP サービスをクライアントである PC やモバイル機器に提供する．図 2 に示した LAN 内のクライアントは，デフォルトでは CMS にのみアクセス可能となるようにゲートウェイを設定する．図 3 に示した LAN 内のクライアントは，デフォルトでは LAN の外に出ることができないようにゲートウェイを設定する．

図 2 と図 3 を用いて，CaNector の実際の動作について説明する．どちらの場合も，利用者のクライアントが情報コンセント，もしくは，無線 LAN に接続すると，DHCP によって IP アドレスが割り当てられる．次に，利用者は Web ブラウザを用いて CMS にケーパビリティを送信する．CMS は受け取ったケーパビリティの有効性と要素（有効期限や使用回数）をチェックし正しい場合，ゲートウェイに対して通信を許可するように設定を行う．CMS 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．

CaNector の特徴として，ゲートウェイと CMS が完全に分離していることがあげられる．そのため，現在 CaNector では，通信路の開放閉鎖に関しては，スイッチ専用のアプライアンス（Extream Network 社製の eXtreme Summit 150-24t [12]）と iptables [13] の 2 つで実現しているが，他の機種やパケットフィルタリングを行う汎用 OS で動作するプログラムに変更することも可能である．

CMS には，ゲートウェイの違いを吸収するモジュールが存在する．このモジュールは主に，指定されたゲートウェイに接続し，パケット通過の許可エントリの追加・削除を行うそのゲートウェイ特有のコマンドを実行している．そのため，ゲートウェイを変更する場合，ゲートウェイ特有のコマンドを送信するモジュールを実装すればよい．

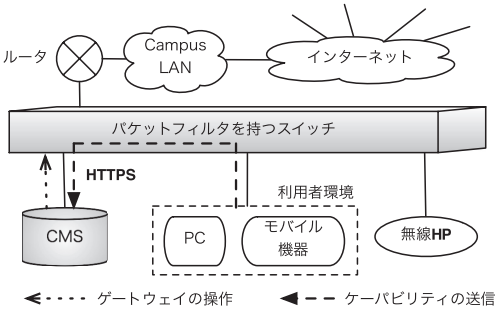


図 2　専用スイッチを用いた CaNector の構成
Fig. 2　The architecture of CaNector using dedicated switch.

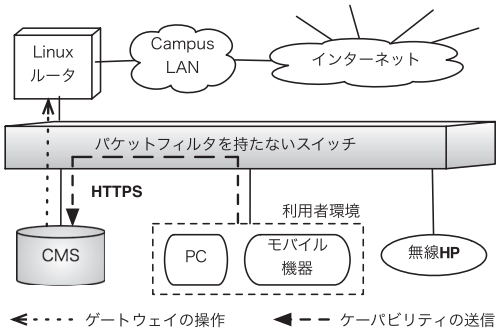


図 3　Linux ルータを用いた CaNector の構成
Fig. 3　The architecture of CaNector using Linux router.

4.3　CaNector におけるケーパビリティ

CaNector では，ケーパビリティに以下の要素を含ませる．

- **通信範囲**：宛先 IP アドレス（通信可能な宛先 IP アドレス）や宛先ポート番号（通信可能な宛先ポート番号）等の通信範囲を表す．
- **有効期限**：ケーパビリティの使用可能な期限を表す．開始時刻と終了時刻の 2 つの要素から構成される．
- **使用回数**：ケーパビリティの使用可能な回数を表す．DHCP の有効期間を 1 回と数える．

- **管理権限**：管理権限の有無を表す．
- **利用者情報**：利用者を特定することができる情報（名前，所属等）を表す．この情報を用いて，管理者はケーパビリティから利用者を追跡できる．
- **識別子**：ケーパビリティの識別子を表す．
- **上位のケーパビリティの識別子**：ケーパビリティを作成する際に用いられた元のケーパビリティを識別するために用いる．

通信範囲と有効期限は，外部利用者のネットワーク利用を許可するときに用いる．たとえば，共同研究者である外部利用者には内部サーバへのアクセスを禁止したいという要望や，学会参加者には開催期間のみネットワークの利用を許可したいという要望を満たすために用いる．通信範囲を指定することで，通信可能な宛先 IP アドレスや宛先ポート番号を制限することができるため，外部利用者による内部サーバへのアクセスを禁止できる．有効期限を指定することで，使用可能な期限を限定することができるだけでなく，未来に有効になるケーパビリティを事前に作成することができる．また，使用回数制限を指定することで，ゲートウェイに対してパケット通過の許可エントリを追加する回数を制限することができる．使用回数 1 回につき，DHCP の有効期限の間ネットワークに接続可能となる．ただし，DHCP の有効期限よりもケーパビリティの有効期限が短い場合，ケーパビリティの有効期限が優先され使用できなくなる．これらの要素は，外部利用者にケーパビリティを渡す際に，権限の範囲を制限するためと，万が一漏洩した場合にも被害の拡大を防ぐために利用される．また，有効期限のみを用いて使用回数制限を設定しない運用も可能である．

ケーパビリティは，乱数により作成した識別子によって識別される．乱数を用いる理由は，識別子の偽造を防ぐためである[4),5),14)]．利用者は，Web ブラウザを用いて CMS へ識別子を送信することで，ケーパビリティを持っていることを示す．ケーパビリティを配布するには，配布したい利用者にその識別子を渡せばよい．配布方法としては，識別子を印刷，あるいは，手書きして手渡す方法やメールで渡す方法が考えられる．本機構の乱数と利用者認証における ID とパスワードの組みを比較すると，ID とパスワードは利用者本人が決定することができるため，乱数と比べて覚えやすいという利点がある．ただし，パスワードは ID から類推されないようにする必要があり，また，より頑健にするためには利用者本人でも覚えにくいランダムな文字列を用いる必要がある．このような覚えにくいパスワードは，パスワード管理システムを用いて管理されることがある．また，初期パスワードを乱数で生成し印刷して配るという運用も広く行われている．そのため，本機構の乱数と利用者認証の ID とパスワードの組みの安全性に関しては同程度であると考えられる．

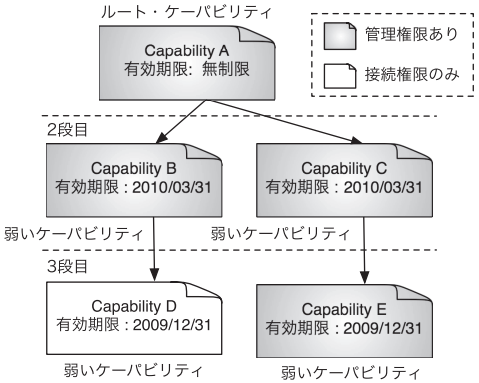


図 4　ケーパビリティの階層構造の一例
Fig. 4　An example of capability hierarchy.

あるケーパビリティを元にして作成されるケーパビリティは，元のケーパビリティよりも権限を弱くする必要がある．このように，権限が元のケーパビリティの権限よりも制限されているケーパビリティのことを，本論文では，**弱いケーパビリティ**と呼ぶ．弱いケーパビリティの例としては，たとえば，元のケーパビリティよりも使用回数が少ないものや有効期限が短いものがあげられる．弱いケーパビリティを作成すると，ケーパビリティは DNS の管理構造のように階層構造を形成する[15),16)]．**図 4** にケーパビリティの階層構造の例を示す．

ケーパビリティは，管理者，および，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．管理者によって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 4 の Capability A）．本論文では，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よって作成されるすべてのケーパビリティは，弱いケーパビリティとなる．たとえば，図 3 では，Capability B，C は，有効期限が付加されており，Capability A よりも権限が弱くなっていることが分かる．また，Capability D は，有効期限が Capability B よりも短くなっており，さらに，管理権限がなくなっている．Capability E は，Capability C よりも有効期限が短くなっている．このように，Capability B，C，D，E 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，上位のケーパビリティよりも権限が弱くなっている弱いケーパビリティである．また，管理権限なしの Capability D は，ケーパビリティの作成を行うことはできないため，階層の葉に位置する．このように，ケーパビリティを作成していくうちにケーパビリ

ティは階層構造を形成していく．

組織の役割による階層構造とケーパビリティの階層構造について考える．管理者によって作成されるルート・ケーパビリティは，教員等の組織の役割による階層構造の上位に位置する人に配布する．それらの管理権限ありのケーパビリティから作成される弱いケーパビリティは，学生等の組織の役割の階層構造の下位に位置する人に配布する．また，接続権限のみのケーパビリティは，主に外部利用者に配布する．このようにすることで，組織の役割による階層構造に沿ったケーパビリティを利用者に配布することができる．

4.4　利用者の特定方法

CaNector では，不適切な利用を行っている利用者を事後に特定するために，ケーパビリティの作成時に配布先の利用者に関するデータを付加し，ケーパビリティの利用ログをデータベースに保存するようにする．利用者を特定するというのは，事後に利用者を追跡し個人を判別することである．4.3 節で述べたように，ケーパビリティの要素の 1 つである “利用者情報” に利用者に関する情報（名前，所属等）を入力することができる．また，CaNector では，ケーパビリティの利用ログとして以下の要素をデータベースに保存する．

- **日時**：操作された日時を表す．
- **操作**：操作内容を表す．
- **操作を行ったケーパビリティ**：操作に使用されたケーパビリティを表す．
- **操作対象のケーパビリティ**：編集や削除等の操作の対象となったケーパビリティを表す．
- **IP アドレス**：操作を行ったクライアントの IP アドレスを表す．
- **MAC アドレス**：操作を行ったクライントの MAC アドレスを表す．

不適切な利用が行われた場合，多くの場合，管理者には利用先の管理者から IP アドレスと時間で照会依頼がくる．管理者は，その時間と IP アドレスや MAC アドレスを用いてどのケーパビリティが利用されたかを特定する．そして，そのケーパビリティの “利用者情報” に入っているデータ等を参照することで，ケーパビリティの利用者を特定することができる．利用者情報が，直接利用者を特定することができない情報の場合，管理者はそのケーパビリティを作成した利用者を特定し誰に配布したかを聞く．この作業を再帰的に行っていくことで，管理者は不適切な利用を行った利用者を特定することができる．

5.　CMS の機能

この章では，CMS が提供する以下の機能について述べる．

- ケーパビリティを用いたネットワーク接続．

図 5　ケーパビリティの識別子入力ページのスクリーンショット
Fig. 5　The screenshot of the page to input an identifier of a capability.

- ケーパビリティの管理．
- ログの閲覧．

また，CMS の各機能は，Web ブラウザを用いたインタフェースから利用することができる．そのため，特別なソフトウェアのインストールは必要ない．

5.1　ケーパビリティを用いたネットワーク接続方法

CMS では，ケーパビリティの利用方法として以下の 2 種類を用意する．

- 利用者が，Web ブラウザでケーパビリティの識別子を**図 5** に示すポータルページ（CMS のページ）に対して手で入力する．
- 利用者が，ケーパビリティの識別子を含む URL にアクセスする．

Web ブラウザを用いる場合，利用者は，CMS のケーパビリティ入力ページにケーパビリティの識別子を入力し “connect” ボタン（図 5）をクリックする．すると，CMS に識別子が送信され，CMS は受け取った識別子を用いてケーパビリティ本体を取り出し有効期限や使用回数の検証を行う．取り出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，再帰的にルート・ケーパビリティにまでさかのぼってすべてのケーパビリティを検証する．

有効期限に関してはチェックをするだけであるが，使用回数制限に関しては弱いケーパビリティを使用した場合，元のケーパビリティの使用回数もチェック時に減らされる．このとき，弱いケーパビリティの使用回数は残っていても元のケーパビリティの使用回数が残っていない場合，その弱いケーパビリティを使用することはできない．このような事態を避けるために，元のケーパビリティを保持している利用者がその使用を止め，それを元に新たに作成したケーパビリティを使用するようにすればよい．たとえば，利用者 A が使用回数制限 100 回を持ったケーパビリティから使用回数制限 10 回のケーパビリティを作成して利用者 B に配布する場合，以下のような運用をすることもできる．

- 利用者 A は，元のケーパビリティ（使用回数制限 100 回）から 10 回利用できる弱いケーパビリティを作成し，利用者 B に配布する．
- 利用者 A は，元のケーパビリティ（使用回数制限 100 回）から 90 回利用できる弱いケーパビリティを作成し，自分はこの弱いケーパビリティを使用する．

すべての検証に通った場合，CMS は，ゲートウェイにその利用者の PC の許可エントリを追加する．その後，ブラウザには，操作が成功したことと使用したケーパビリティの権限情報が表示される．

URL にアクセスを行う場合，URL に含まれた識別子がパラメータとして CMS に送信される．CMS は，その識別子を用いてケーパビリティを取り出しその検証を行う．後の CMS の動作は，上述した場合と同じである．また，この URL をエンコードした QR コード（**図 6**）を利用することで，URL を手で入力する手間を省くことが可能になる．

ゲートウェイに追加された許可エントリは，ケーパビリティの有効期限が過ぎた後に CMS 内の cron によって削除される．また，DHCP のリース期間が終了した際にも，その IP アドレスを使用していたクライアントの許可エントリを削除する．

5.2　ケーパビリティの管理

ここでは，管理者，および，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方法について述べる．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**：ケーパビリティを作成する．ケーパビリティが作成されるとき，同時にこのケーパビリティの識別子を含んだ URL をエンコードした QR コードも作成される．また，同じ権限のケーパビリティであれば，1 度の操作で指定個数作成できる．作成されたケーパビリティは，CMS 内のデータベースに保存される．
- **編集**：ケーパビリティの権限を編集する．ただし，元のケーパビリティの権限を超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 回以上にすることはできない．
- **削除**：ケーパビリティを無効化する．削除の操作がされた場合，CMS はデータベース内の対象となるケーパビリティの項目を削除するのではなく，無効となった記録を保管する作業を行う．無効となったケーパビリティよりも下位層に位置するすべてのケーパビリティが使用できなくなる．
- **ログの閲覧**：ケーパビリティの利用に関するログを表示する．これについては，5.3 節で詳しく述べる．

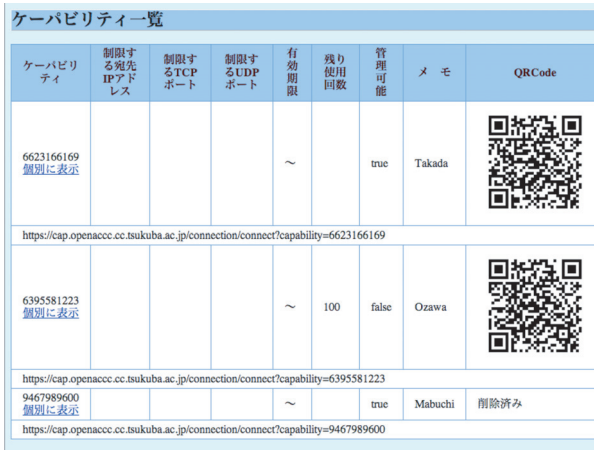


図 6　管理者用のケーパビリティ管理ページのスクリーンショット
Fig. 6　The screenshot of capability management page for administrators.

上述した管理操作について，CMS のスクリーンショットを用いて詳しく説明する．図 6 には，管理者がケーパビリティを管理する際のスクリーンショットを示す．管理者の場合，すべてのルート・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 7 には，管理権限ありのケーパビリティを保持する利用者がケーパビリティを管理する際のスクリーンショットを示す．この管理画面にアクセスするためには，管理権限ありのケーパビリティの識別子を CMS のページに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，入力された識別子を用いて取得し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すべてが表示される．図 7 の上部に表示されているケーパビリティが，入力された識別子を用いて取得したケーパビリティである．また，

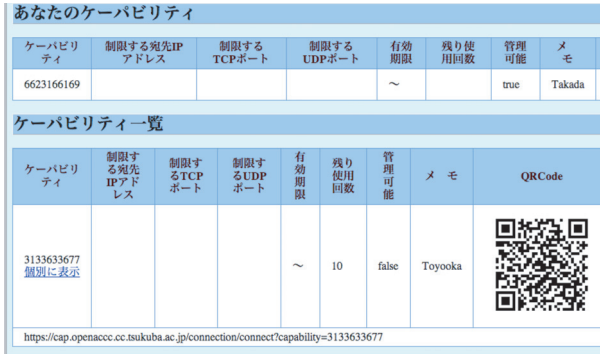


図 7　利用者用のケーパビリティ管理ページのスクリーンショット
Fig. 7　The screenshot of capability management page for regular users.

入力された識別子を用いて取得されたケーパビリティ（自分が保持する管理権限ありのケーパビリティ）に対しては操作（編集と削除）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，ケーパビリティの権限を入力するページ（**図 8**）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，接続権限のみのケーパビリティが作成される．権限を入力後，“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．入力された権限を元のケーパビリティの権限と比較し，権限が元のケーパビリティの権限よりも制限されている場合のみ作成を許可する．元の権限より制限されていない場合は，作成を拒否し再度権限の入力を求める．ケーパビリティの識別子をデータベースに保存する場合，データベース内にすでに同じ識別子がないことをチェックする．同じ識別子が存在している場合，別の一意な識別子を作成し直して保存する．また，同じ権限のケーパビリティを複数一括で作成したい場合，作成個数を指定のフィールドに入力する．一括で作成する場合は，“利用者情報”としては 1 つの情報しか入力することができない．たとえば，学会名やイベント名等である．そのため，学会やイベント等でケーパビリティを配布する際に，ケーパビリティと利用者を対応付ける情報を学会やイベント等の事務局が管理する．具体的な方法としては，紙にケーパビリティの識別子と名前を書いてもらう方法や，事務局が編集機能を用いてそのケーパビリティの“利用者情報”を変更する方法が考えられる．

ケーパビリティに付加された権限を編集する場合，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，権限を入力するページ（図 8 とほぼ同じ）が表示されるので，そこに新しい権限を入力し編集の“更新”リンクをクリックすることで権限の編集が終了する．ここでも，入力された権限を元のケーパビリティの権限と比較し，権限が元のケーパビリティの権限よりも制限されているかどうかを検査する．また，削除する場合，“削除”リンクをクリックすると削除確認が表示され，OK を選択すると無効化される．後にケーパビリティのアクセスログやデータ（“利用者”）を参照する場合があるため，無効化されたケーパビリティは利用できなくなるが内部のデータベース内からは削除されない．

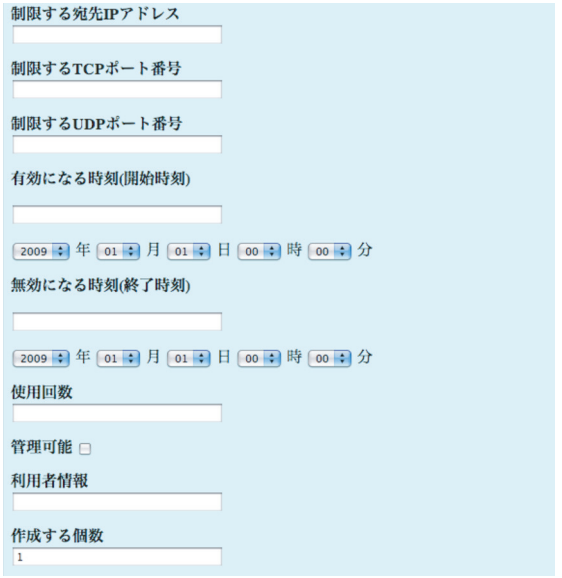


図 8　ケーパビリティ作成ページのスクリーンショット
Fig. 8　The screenshot of the page for creating capabilities.

5.3　ログの閲覧

保存されたログは，ケーパビリティ管理画面（**図 9**）で閲覧可能である．あるケーパビリティのログを表示した場合，そのケーパビリティの下位層に存在するすべてのケーパビリティのログが表示される．

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（作成，削除，編集）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

図 9　ログ表示ページのスクリーンショット
Fig. 9　The screenshot of the page for displaying logs.

ティを用いてネットワーク接続の操作（接続，切断）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．また，表示されているログは，図 9 のテキストフィールドに時間や IP アドレスの値を入力することでフィルタリングすることができる．

6. 評価とテスト運用

この章では，CaNector の有用性を評価するために以下の 2 つの評価を行う．

- **処理時間**：ケーパビリティの権限をチェックし，ゲートウェイにルールを追加するためにかかる処理時間を計測する．
- **従来手法との比較**：接続権限の取得の手軽さに関して従来手法と比較を行う．

6.1　処理時間の計測

利用者がケーパビリティの識別子を CMS に入力してから，ゲートウェイの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．4.3 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権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，以下の 4 つの場合に分けて処理時間の計測を行った．

- **制限なし**：すべてのケーパビリティに制限がない．
- **有効期限付き**：すべてのケーパビリティに有効期限がある．
- **使用回数制限付き**：すべてのケーパビリティに使用回数制限がある．
- **有効期限/使用回数制限付き**：すべ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ある．

表 1　クライアントマシンとサーバマシンの性能
Table 1　Performances of the client machine and the server machine.

	クライアント	CMS
CPU	Core 2 Duo 2.2 GHz	Core i7 2.66 GHz
メモリ	2.0 GB	3.0 GB
OS	Ubuntu 8.0.4 (Kernel 2.6.24)	Cent OS 5.3 (Kernel 2.6.29.3)
NIC	1000BASE-T	1000BASE-T

表 2　ゲートウェイマシン（Linux）の性能
Table 2　The performance of the gateway (Linux).

	ゲートウェイ
CPU	Pentium D 2.8 GHz
メモリ	1.0 GB
OS	Cent OS 5.3 (Kernel 2.6.18)
NIC	1000BASE-T
iptables	v1.3.5

この実験で用いたクライアントと CMS に用いたマシンの性能を**表 1** に示す．今回使用したゲートウェイは，eXxtreme Summit 150-24t と Linux マシンで iptables を用いたものの 2 つである．LAN は，1 Gbps である．iptables を動作させたマシンの性能を**表 2** に示す．要求を行うプログラムとしては，wget を用いた．

計測した結果を**図 10** と**図 11** 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（単位：秒）を表している．また，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．

ゲートウェイが eXtreme Summit 150-24t の場合，CaNector の処理時間は最も遅いもので約 1.98 秒である．また，CMS が Telnet で eXtreme Summit 150-24t にルールの追加を行う処理時間は，約 1.2 秒だった．処理時間のほとんどは，HTTPS を用いたクライアントと CMS 間の通信時間と CMS が Telnet でゲートウェイを操作する時間である．

ゲートウェイが iptables の場合，CaNector の処理時間は最も遅いもので約 0.5 秒である．ゲートウェイが eXtreme Summit 150-24t の場合と比べて約 4 倍から 5 倍速くなっている

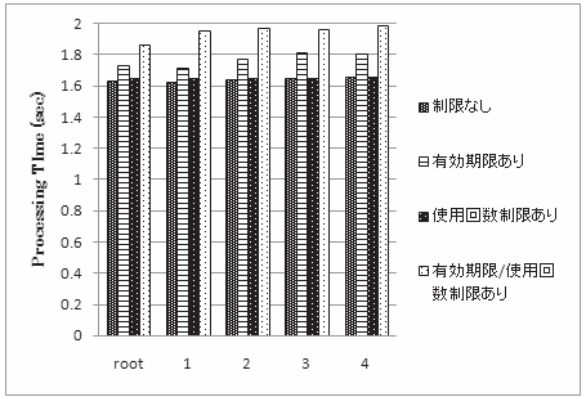


図 10　CaNector の処理時間（eXtreme Summit 150-24t）
Fig. 10　Processing times of CaNector (eXtreme Summit 150-24t).

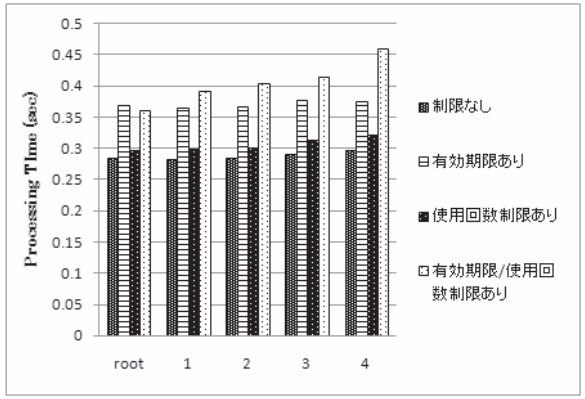


図 11　CaNector の処理時間（iptables）
Fig. 11　Processing times of CaNector (iptables).

ことが分かる．

ゲートウェイが eXtreme Summit 150-24t の場合は若干遅いが，Telnet での操作がボトルネックになっているためこれ以上の改善は難しい．これに対して，ゲートウェイが iptables の場合は十分な処理速度であることが分かる．どちらの場合においても，実用には問題のない速度であるといえる．

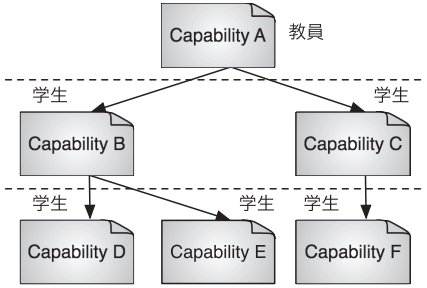


図 12　ケーパビリティの利用例（1）
Fig. 12　The example of using capabilities (1).

6.2　従来手法との比較

2.2 節の 3 つの例を用いて，CaNector と LAWN を比較する．

(1) に CaNector を適用した場合について**図 12** を用いて説明する．この例では，大学院生 2 人と学部生 3 人の計 5 人に対してケーパビリティを配布することを想定する．まず，管理者が，教員に渡す場合，“利用者情報” として配布先の教員の名前を指定して無制限の Capability A を作成し，その識別子を印刷する等の方法で渡す．次に，Capability A を保持する教員が，大学院生に渡す場合，“利用者情報” として配布先となる 2 人の大学院生の名前を指定してゼミ時間中有効な Capability B，C を作成し，その識別子を紙に書く等の方法で渡す．さらに，それぞれ Capability B，C を保持する大学院生が，学部生に渡す場合，“利用者情報” として配布先となる学部生の名前を指定して Capability D，E，F を作成し，その識別子を紙に書く等の方法で渡す．このように CaNector を適用することで，教員は一部の学生（この場合は，大学院生）に対してのみケーパビリティを作成して配布すればよく，学生との間で負荷を分散することができる．一方，(1) に LAWN を適用した場合，アカウント発行可能な教員が，5 人すべての学生に対してアカウントを発行し渡す必要があり，CaNector に比べて教員の負荷が大きい．

(2) に CaNector を適用した場合について**図 13** を用いて説明する．まず，管理者が，教員に渡す場合，“利用者情報” として配布先の教員の名前を指定して無制限の Capability A を作成し，その識別子を印刷する等の方法で渡す．次に，Capability A を保持する教員が，学生に渡す場合，“利用者情報” として配布先の学生の名前を指定して学生の在籍期間中有効な Capability B，C を作成し，その識別子を紙に書く等の方法で渡す．さらに，Capability B を保持した学生が，共同研究者に渡す場合，“利用者情報” として配布先の共同研究者の

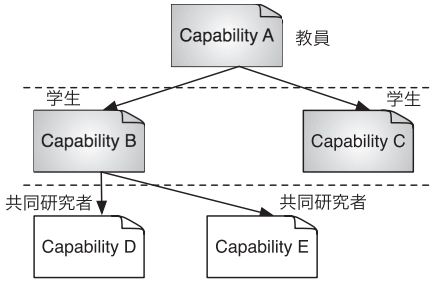


図 13　ケーパビリティの利用例（2）
Fig. 13　The example of using capabilities (2).

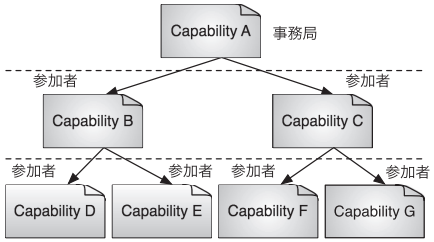


図 14　ケーパビリティの利用例（3）
Fig. 14　The example of using capabilities (3).

名前を指定して共同研究者の滞在期間中有効で管理権限のない Capability D，E を作成し，その識別子を紙に書く等の方法で渡す．このように学生であっても，共同研究者に対して接続権限を渡すことができる．一方，(2) に LAWN を適用した場合，アカウント発行可能な教員が，共同研究者に対してアカウントを発行し渡す必要がある．教員が研究室に不在の場合には，共同研究者はネットワークを利用できない．

(3) に CaNector を適用した場合について**図 14** を用いて説明する．まず，管理者が学会開催中有効な Capability A を作成し，その識別子を印刷した紙を事務局に渡す．次に，Capability A を保持する事務局は，Capability A と同じ権限を持つケーパビリティを大量に作成しその識別子を印刷する．事務局は参加者にその紙を渡す際に，参加者に自分の名前を別紙に記入してもらう．また，ケーパビリティを保持する参加者が，知り合いの参加者に渡す場合，“利用者情報” としてその参加者の名前を指定したケーパビリティを作成し，その識別子を紙に書く等の方法で渡す．一方，(3) に LAWN を適用した場合，事務局が大量の一時的なアカウントを発行し紙に印刷する．参加者は，事務局によって設置されたアカウント配布場所に移動して受け取る必要がある．

6.3　テスト運用

CaNector 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 PC やスマートフォン等のモバイル機器を接続して利用している．同時使用台数は 10 台程度で，2008 年 12 月から約 6 カ月間にわたって運用している．現在も問題なく動作している．

また，2009 年 3 月 5，6 日に開催された第 4 回情報処理学会インターネットと運用技術研究会の会場においてデモを行った．利用していただいた参加者の方にアンケートを記入してもらったところ，接続権限を作成ができるという機能に関しては好評であった．その反面，利用終了の検知方法やケーパビリティの入力が面倒であるという指摘をうけた．現在，CaNector では，5.1 節で述べたように DHCP のリース期間が終了した際に，その IP アドレスを使用していたクライアントの許可エントリを削除するようにしている．また，Opengate の Java Applet を用いた利用の終了検知を CaNector に取り入れることも可能であると考えられる．入力が面倒であるという指摘に関しては，従来手法におけるユーザ ID とパスワードの入力にかかる手間とそれほど変わらないと考える．CaNector の場合，PC 起動時にケーパビリティを自動的に SSL で認証された CMS に送信するような設定をしておくことで，入力の手間を省くことができる．また，QR コードがあるためモバイル機器を用いた場合は入力の手間を省くことができる．

7. 考　　察

7.1　スケーラビリティ

CaNector においてスケーラビリティのネックになるポイントとしては，ゲートウェイと CMS が考えられる．ゲートウェイがスケーラビリティに及ぼす問題は，以下の 3 つが考えられる．

- ゲートウェイに追加可能なルール数には上限がある．
- ルールを追加するための命令がシリアライズされ処理が遅くなる．
- ゲートウェイの通信帯域には上限がある．

これらの問題を解決するために，ゲートウェイを利用者の人数に合わせて部屋ごとに分散配置を行うことが考えられる．1 台のゲートウェイに対して利用者数がそれほど多くなければ，スケーラビリティを確保することができる．

CMS では，このサーバへのアクセスが集中することが問題であると考えられる．そのため，CMS への負荷を分散するために，CMS を異なるコンピュータで立ち上げアクセスを振り分ける方法が考えられる．

7.2　ケーパビリティの配布

CaNector では，各利用者は，作成したケーパビリティを自分が信頼している利用者にのみ配布する．各利用者は自分が信頼している利用者にのみケーパビリティを配布することで，利用者間で信頼の連鎖（trust chain）が形成される．信頼の連鎖でつながった利用者を，CaNector とネットワークの管理者は間接的に信頼する．このように信頼されたグループの中でのみ，ケーパビリティは配布され使用されるべきである．

自分のケーパビリティをそのまま他の利用者に配布した場合，同じケーパビリティを複数の利用者が使用することになり，不適切な利用を行った利用者を事後に特定することができなくなるという問題がある．そのため，CaNector においてはケーパビリティをそのまま配布するのではなく，必ず新しいケーパビリティを作成して配布する必要がある．また，ケーパビリティを配布した利用者は，その配布した利用者が不適切な利用を行った場合に，その利用者を特定するための責務を負うことになる．つまり，利用者はケーパビリティを受け取ると同時に，そのケーパビリティ，また，そのケーパビリティから派生するすべてのケーパビリティの管理や利用者の特定に対しての義務も引き受けることになる．このように，CaNector では，管理や特定もネットワークに接続するための権限と同様に下位層の利用者に委譲されていくことになる．これらの義務を果たすために，利用者は，自分がケーパビリティを配布した利用者を把握しておく必要がある．ただし，管理権限なしのケーパビリティの場合，新しいケーパビリティを作成することはできないため，これらの義務を負うことはない．このようなことを周知したうえで，CaNector は使用されるべきである．

信頼の連鎖を構築するという点で類似したものとして SSL 等で用いられている X.509[17] の木構造がある．X.509 はインターネットという広い範囲で使用されるの対して，CaNector は組織内 LAN というより狭い範囲で利用されることを想定している．そのため，CA よりも不正利用による影響は少ないと考えられる．したがって，上述した程度の運用基準を遵守したうえで利用されれば，CaNector の運用は破綻しないと考えられる．

8. ま と め

本論文では，ケーパビリティを用いて実現した，接続権限を利用者間で受け渡し可能なネットワーク制御機構 CaNector について述べた．ケーパビリティには，管理権限ありと接続権限のみの 2 種類のケーパビリティがある．CaNector を用いることで，管理権限ありのケーパビリティを保持する利用者が，手軽に新しいケーパビリティを作成し他の利用者に配布することができる．これにより，利用者が接続権限を他の利用者から手軽に取得することができる．また，データベースにログを保存することとケーパビリティに配布した利用者の情報を付加することで，不適切な利用を行った利用者を事後に特定可能にした．さらに，利用者は事前に特別なソフトウェアをインストールする必要なく，ブラウザを用いて Web ページにアクセスすることで CeNector を利用することができる．

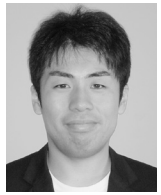
現在，CaNector は 2008 年 12 月から約 6 カ月間のテスト運用を行っているが，実用性を示すためにはより長く，そして，より多くの利用者に利用してもらい意見を聴く必要がある．今後の課題としては，利用可能場所を拡大し多くの人が利用可能な環境を構築することと，ケーパビリティの効率的な配布の実現である．

参 考 文 献

1) Silberschatz, A., Galvin, P.B. and Gagne, G.: *Operating System Concepts*, Wiley (2008).
2) Accetta, M., Baron, R., Bolosky, W., Golub, D., Rashid, R., Tevanian, A. and Young, M.: Mach: A New Kernel Foundation For UNIX Development, *Proc. USENIX Summer Conference*, pp.93–112 (1986).
3) Levy, H.M.: *Capability-Based Computer Systems*, Digital Press (1984).
4) Geambasu, R., Balazinska, M., Gribble, S.D. and Levy, H.M.: HomeViews: Peer-to-Peer Middleware for Personal Data Sharing Applications, *Proc. 2007 ACM SIGMOD International Conference on Management of Data*, pp.235–246 (2007).
5) Mabuchi, M., Shinjo, Y., Sato, A. and Kato, K.: An Access Control Model for Web-Services that Supports Delegation and Creation of Authority, *Proc. 7th International Conference on Networking* (*ICN08*), pp.213–222 (2008).
6) Mullender, S.J., van Rossum, G., Tenenbaum, A.S., van Renesse, R. and van Staveren, H.: Amoeba: A Distributed Operating System for the 1990s, *IEEE Computer*, Vol.23, No.5, pp.44–53 (1990).
7) Wulf, W., Cohen, E., Corwin, W., Jones, A., Levin, R., Pierson, C. and Pollack, F.: HYDRA: The Kernel of a Multiprocessor Operating System, *Comm. ACM*, Vol.17, No.6, pp.337–345 (1974).
8) 渡辺義明，渡辺健次，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).
9) 石橋勇人，山井成良，安倍広多，阪本　晃，松浦敏雄：利用者ごとのアクセス制御を実

現する情報コンセント不正利用防止方式，情報処理学会論文誌，Vol.42, No.1, pp.79–88 (2001).
10) 西村浩二，秋成秀紀，野村嘉洋，相原玲二：遠隔機器制御プロトコルを用いた有線/無線 LAN 用情報コンセントシステム，情報処理学会論文誌，Vol.43, No.2, pp.662–670 (2002).
11) Georgia Institute of Technology: LAWN: Local Area Walkup/Wireless Network. http://www.lawn.gatech.edu/index.html (accessed 2009-12-21)
12) Extreme Networks: Summit X150 シリーズ，Extreme Networks Data Sheet (2007).
13) Andreasson, O.: Iptables Tutorial 1.2.2 (2006).
14) Anderson, M., Pose, R.D. and Wallace, C.S.: A Password-Capability System, *The Computer Journal*, Vol.29, No.1, pp.1–8 (1986).
15) Mockapetris, P.: Domain Names – Concepts and Facilities, RFC1034 (standard) (Nov. 1987).
16) Mockapetris, P.: Domain Names – Implementation and Specification, RFC1035 (standard) (Nov. 1987).
17) TELECOMMUNICATION STANDARDIZATION SECTOR OF ITU: ITU-T Recommendation X.509 (2005).

(平成 21 年 6 月 15 日受付)
(平成 21 年 12 月 17 日採録)



馬渕　充啓（学生会員）

1982 年生．2005 年筑波大学第三学群情報学類卒業．2007 年筑波大学大学院システム情報工学研究科コンピュータサイエンス専攻博士前期課程修了．現在，同上博士後期課程 3 年．修士（工学）．オペレーティングシステム，分散システム，ネットワークセキュリティ，アクセス制御に興味を持つ．

高田　真吾（学生会員）

1985 年生．2008 年筑波大学第三学群情報学類卒業．現在，筑波大学大学院システム情報工学研究科コンピュータサイエンス専攻博士前期課程に在学中．仮想計算機モニタを用いた OS の起動制御の研究に従事．

小沢　健史（学生会員）

1985 年生．2008 年中央大学情報工学科卒業．現在，筑波大学大学院システム情報工学研究科コンピュータサイエンス専攻博士前期課程に在学中．仮想計算機，オペレーティングシステムに関する研究に従事．

豊岡　　拓（学生会員）

1985 年生．2008 年筑波大学第三学群情報学類卒業．現在，筑波大学大学院システム情報工学研究科コンピュータサイエンス専攻博士前期課程に在学中．仮想計算機環境における OS のファイル入出力に関する研究に従事．

松井　慧悟

1985 年生．2009 年筑波大学大学院システム情報工学研究科コンピュータサイエンス専攻博士前期課程修了．現在，(株）日立製作所．大学では，アクセス制御の研究に従事．

佐藤　　聡（正会員）

1967 年生．1991 年筑波大学第三学群情報学類卒業．1996 年筑波大学大学院工学研究科単位取得退学．同年広島市立大学情報科学部助手．2001 年筑波大学システム情報工学研究科講師．現在，同大学学術情報メディアセンター勤務．博士（工学）．キャンパスネットワークの企画管理運用，ネットワークデータベース，言語処理等の研究に従事．電子情報通信学会，ACM-SIGMOD-JAPN 各会員．

新城　靖（正会員）

1965 年生．1993 年筑波大学大学院工学研究科電子・情報工学専攻博士課程修了．同年琉球大学工学部情報工学科助手．1995 年筑波大学電子・情報工学系講師，2003 年同助教授．2007 年准助教授．オペレーティング・システム，分散システム，仮想システム，並行システム，情報セキュリティの研究に従事．博士（工学）．日本ソフトウェア科学会，ACM，IEEE CS 各会員．

加藤　和彦（正会員）

1962 年生．1985 年筑波大学第三学群情報学類卒業．1989 年東京大学大学院理学系研究科情報科学専攻中退．1992 年博士（理学）（東京大学大学院理学系研究科）．1989 年東京大学理学部情報科学科助手，1993 年筑波大学電子・情報工学系講師，1996 年同助教授，2004 年筑波大学大学院システム情報工学研究科教授，現在に至る．オペレーティングシステム，分散システム，仮想計算環境，セキュリティに興味を持つ．1990 年情報処理学会学術奨励賞，1992 年同研究賞，2005 年同論文賞，2004 年日本ソフトウェア科学会論文賞，各受賞．