Stealth-LIN6：匿名性のあるIPv6モビリティ通信プロトコル
Stealth-LIN6 : An IPv6 Mobility Protocol equipped with Anonymity

市川 隆浩　　坂野 あゆみ　　寺岡 文男
慶應義塾大学 大学院 理工学研究科

概 要

本論文では，IPv6 移動通信において盗聴者による通信ノードの特定および位置追跡を防止するモビリティプロトコル *Stealth-LIN6 (SLIN6)* を提案する．SLIN6 はモビリティプロトコル LIN6 をベースとしており，通信毎に IP レイヤにおいて動的に生成されたアドレスを用いることによってノードの匿名性を実現する．また，SLIN6 は特有のプロキシを用いることによってノードの位置秘匿性を実現する．SLIN6 を FreeBSD に実装した．測定の結果，データ通信における SLIN6 パケット処理のオーバヘッドは無視できる程度であることが分かった．また，パケットが SLIN6 プロキシを経由しパケット順序が狂うことにより，TCP スループットが大きく減少することが分かった．

1　はじめに

インターネット環境の普及により，あらゆる機器があらゆる場所で通信が行えるようになった．また，携帯通信機器や無線環境普及により，移動先からもインターネットへの接続が活発に行われるようになった．そこで移動中も通信が途切れない移動透過性が保証されている通信が必要と考えられ，ノードの移動透過性を実現するモビリティプロトコルである Mobile IPv6[1] や LIN6[2] が提案された．今後ますますインターネットに接続する機器は増加すると考えられ，IPv6 を基盤とするモビリティプロトコルの必要性は高まると考えられる．しかし，既存のモビリティプロトコルの IP ヘッダには，通信を行っているノードが接続するサブネットを示す位置指示子と，ノードを一意に識別可能にするノード識別子が格納される．代表的なモビリティプロトコル MIPv6 では HoA（Home Address）がノード識別子であり，CoA（Care of Address）のネットワークプレフィックスが位置指示子である．また，LIN6 では LIN6ID がノード識別子であり，ネットワークプレフィックスが位置指示子である．よって，一般に移動を前提としたモビリティプロトコルを用いた通信には

- 盗聴者による通信者の特定
- 盗聴者によるノード移動履歴の追跡

といった二つの脅威が存在する．これらの脅威に対抗するためには，通信するノードの接続するサブネットを隠す位置秘匿性やノードを識別されないようにする匿名性の実現が必要になる．一つ目の脅威に対しては，通信の盗聴者に通信者の位置指示子やノード識別子を知られないことが必要である．二つ目の脅威に対しては，盗聴者にノードが移動したことを判別できないように，位置指示子の変化を知られないことが必要である．しかし，パケット配送時の経路決定に利用される IP ヘッダは暗号化ができないため，既存の暗号化技術を IP ヘッダに適用することは不可能である．匿名通信技術としては，Mixes[3]，Crowds[4] といったプロトコルが挙げられるが，これらのプロトコルに対してモビリティ機能を組み合わせることは，ルーティング，帯域，遅延，オーバヘッドを考慮すると望ましくない．またモビリティと匿名性を考慮したプロトコルである Host Identify Protocol（HIP）[5] も位置秘匿性やオーバヘッドに問題がある．

本論文ではモビリティプロトコル LIN6 をベースとした位置秘匿性と匿名性のあるモビリティプロコル *Stealth-LIN6*（*SLIN6*）を提案する．SLIN6 はモビリティ機能を有しつつ，最小限のオーバヘッドで上記の二つの脅威に対抗できる匿名通信を提供することを目的とする．また，SLIN6 がその効果を発揮する環境は盗聴者が無線区間で盗聴を行うような一般的な環境を想定しており，ISP（Internet Service Provider）は信頼できるものとする．

2　関連研究

本章では，先に挙げたノードの位置秘匿性と匿名性に関する関連研究を紹介し，モビリティ環境への適応性や問題点について述べる．

2.1　source-rewriting 方式

source-rewriting 方式とは，中間ノードが IP アドレスを次々に書き換えて通信者の匿名性を得る方式である．図 1 に source-rewriting 方式の例を示す．図 1 はノード X とノー

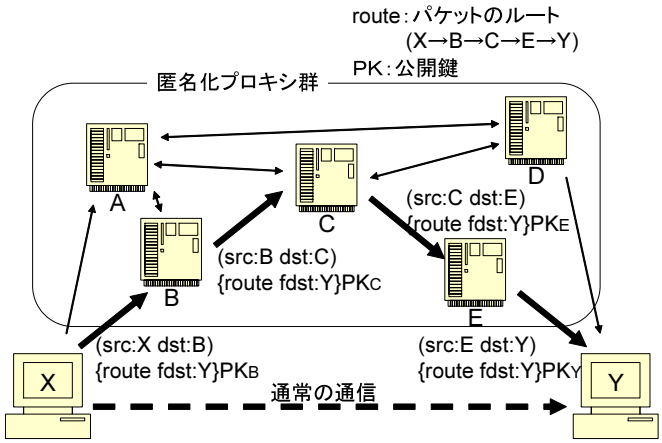

図 1: source-rewriting 方式

ド Y が匿名通信を行う例である．ノード X が送信したパケットはノード B，ノード C，ノード E と次々に IP アドレスの書き換えが行われ，最終的にノード Y に配送される．このアドレスの書き換えが行われる中間ノードを匿名化プロキシと呼ぶ．この匿名化プロキシは，送信者がランダムに選定し，どのような順番でパケットを配送するかをルート情報としてパケットに付与する．匿名化プロキシはこの情報を基にパケットを配送する．配送されるパケットは通常の始点アドレス (src：X)，終点アドレス (dst：B)，最終終点アドレス (fdst:Y)，最終終点ノードまでのルート情報 (route) を持つ．

図 1 の場合，ホップ毎にパケットのデータ部分は公開鍵によって暗号化されるので，最終終点アドレスとルート情報も暗号化される．そのため，盗聴者がルートの途中でパケットを取得しても，匿名化プロキシどうしの通信に見えるため，ノード X とノード Y が通信していることを隠蔽することができる．したがって，途中のパケットを盗聴されても位置秘匿性とノード匿名性が実現される．ただし，暗号化方式は必ずしも公開鍵で行われるとは限らない．source-rewriting 方式の匿名化プロキシのルート選択方法は大きく分けて 2 つあり，Mixes のようにルートが固定長なものと，Crowds のようにルートが可変長なものがある．

source-rewriting 方式では，通常複数の匿名化プロキシを使用しなければならないため，パケットがネットワーク的に非常に冗長なルートを通ることになり，遅延が大きくなる．また各ノードで暗号化，復号化が行われることによる遅延も発生する．また，source-rewriting 方式はノードの移動を前提に設計されていないため，既存のホストモビリティプロトコルとの連係が必要になる．ホストモビリティプロトコルは，帯域の狭い無線環境において利用される可能性が高いと考えられている．したがって，source-rewriting 方式はルート情報などの付与でパケットオーバヘッドが大きくなるため，ホストモビリティプロトコルへの適用は難しい．

表 1: 関連研究の特徴

	source-rewriting	HIP	SLIN6
ノード匿名性	○	△	○
位置秘匿性	○	×	○
オーバヘッド	×	△	○
ルーティング	×	○	△
移動透過性	×	○	○

2.2 HIP

HIP は IPsec を前提とした IPv4，IPv6 両方で利用できるホストモビリティプロトコルである．HIP は新たにトランスポートレイヤとネットワークレイヤの間に HIP レイヤを構築する．HIP のトランスポートレイヤでは，ホストを示す識別子である HI（Host Identifier）がコネクション管理などに利用される．この HI はノードの公開鍵でもある．HIP のネットワークレイヤでは，IPv4 や IPv6 アドレスがパケットの処理や配送に利用される．HIP レイヤはトランスポートレイヤで指定された識別子から，保持するマッピングを基にネットワークレイヤで利用される IPv4 もしくは IPv6 アドレスに変換する．HIP レイヤでの変換により，トランスポートレイヤ以上では HI や HI のハッシュ値である HIT（Host Identity Tag）のみを知っていればよく，IP アドレスの変化による影響を受けないため，移動透過性が実現される．HIP では HIT 情報を利用した Security Parameter Index（SPI）が常にパケットヘッダに付与され，IP アドレスの変化に関係なく通信相手を識別できる．

HIP は，IPsec を前提として設計されているため，上位レイヤのセキュリティプロトコルを利用する場合，重複して暗号化されるという問題点がある．移動端末といった計算速度の遅いマシンの場合，複数回暗号化や復号化を行うことは処理のオーバヘッドとなり，大きな問題点となる．

HIP では，HIT を用いて匿名性を得ようとしている．HI 自体を得ることはできないが，HIT は HI に依存するので，HI の値が同じならば同じ HIT となる．結果として，HIT がノードを示す識別子となる．また，SPI をパケットヘッダに付与するため，HIT により盗聴者による通信ノードの追跡が可能となり，ノードの匿名性に問題があるといえる．一方，HIP にはモビリティサポートのためにパケットを中継する FA （Forwarding Agent）があるが，これは通常通信中のパケット配送には用いられないため，位置秘匿性は提供されない．

2.3 まとめ

本章で挙げた関連研究と，本論文で提案する SLIN6 の特徴を表 1 に示す．

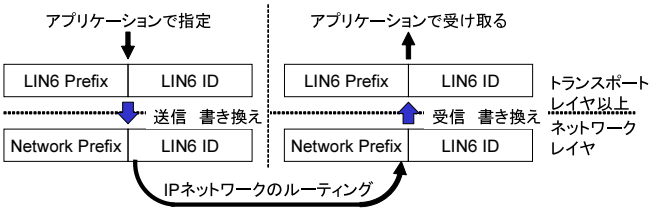

図 2: LIN6 の概要

既存の匿名化技術では，位置の秘匿性と匿名性が同時に得られるとしても，そこにホストモビリティを組み合わせるのはルーティング，帯域，オーバヘッド等を考慮すると適さない．また，HIP はノード識別子が直接読みとられることはないが，パケットヘッダにノードを識別可能な値が含まれてしまうため，結果としてノードが識別されてしまう可能性がある．SLIN6 ではこれらを改善し，移動透過性，位置秘匿性，匿名性を同時に実現させる．

3 モビリティプロトコル LIN6 の概要

LIN6 は位置指示子とノード識別子を分離するという概念を導入しており，ノード識別子は接続しているサブネットにかかわらず一意に定められる．

LIN6 は図 2 に示すように，ネットワークレイヤより上位レイヤにおいて LIN6 汎用識別子を用い，ネットワークレイヤにおいて LIN6 アドレスを用いる．LIN6 汎用識別子の上位 64bits は LIN6 Prefix と呼ばれる位置に依存しない固定値であり，下位 64bits は LIN6ID と呼ばれるノード識別子である．LIN6 アドレスの上位 64bits は現在接続しているサブネットのネットワークプレフィックスであり，下位 64bits は LIN6ID である．

LIN6 において，LIN6ID と現在のネットワークプレフィックスとの対応づけをマッピングと呼び，このマッピングを管理する機構として *Mapping Agent*（*MA*）と呼ばれるノードを用いる．LIN6 では，通信を開始する際に MA に問合わせを行うことで通信相手に対するマッピングを取得する．LIN6 ノードはこのマッピングをレイヤ間のアドレス変換に利用して通信を行う．

4 SLIN6 の設計

4.1 SLIN6 の設計方針

SLIN6 では，盗聴者はパケットの経路上のどこか一点のみでしかパケットを盗聴できないものと想定する．すなわち，盗聴者が無線区間で盗聴を行うような一般的な環境を想定しており，ISP は信頼性があると想定する．SLIN6 は，そのような環境においてモビリティ機能を有しつつ，送受信者の位置秘匿性，匿名性を実現し，移動の追跡を防ぐことを目的とする．IP ヘッダの暗号化は不可能であることから，SLIN6 では一時的に使用する疑似アドレスを通信毎に複数生成して通信開始時に安全に交換し，その複数のアドレスをパケット毎に使い分けるという方法で匿名性を得る．また，始点ノードと終点ノードで直接通信すると通信者の識別子や位置指示子が分かってしまうので，複数の中継ノードを用意し，パケット毎に中継ノードを使い分けるという方法で位置秘匿性を得る．

4.2 SLIN6 の概要

SLIN6 のパケット配送機構は既存の IPv6 のインフラストラクチャを使用し，匿名性のある通信と既存の通信を明示的に使い分ける機能を備える．SLIN6 では，トランスポートレイヤ以上では SLIN6 汎用識別子と呼ばれる仮想的な IPv6 アドレスによってノード間にコネクションを生成し，ネットワークレイヤでは SLIN6 アドレスと呼ばれる動的に生成される IPv6 インターフェースアドレスがパケット配送に利用される．この SLIN6 汎用識別子と SLIN6 アドレスとの関係を通信毎や接続サブネットの移動毎に動的に変更することでノードの匿名性を実現する．また，SLIN6 では SLIN6 プロキシと呼ばれる中間ホストを導入し，SLIN6 プロキシがパケットを中継することでノードの位置秘匿性を実現する．さらに暗号化されたノード，プロキシ間の情報交換の機構により，SLIN6 における制御メッセージの保護を実現する．SLIN6 はネットワークレイヤに実装され，上位，下位レイヤに対する変更はない．なお，SLIN6 アドレスおよび SLIN6 プロキシは同時に複数使用することが可能であり，脅威に対してより安全に通信することが可能となる．

SLIN6 では，あるノード間で通信を開始する場合，通信要求を行うノードを initiator，通信要求を受けるノードを responder と呼ぶこととする．initiator は通信開始時に通信要求を行うために responder を一意に識別する識別子をあらかじめ知っておく必要があるが，responder は initiator を一意に識別する識別子を必ずしも知っておく必要はない．

SLIN6 では，通信毎に initiator と responder 間で動的に LIN6ID を生成し，一時的にパケット配送に利用する．このノードと関連性のない LIN6ID を *SLIN6ID* と呼ぶ．この SLIN6ID は initiator と responder 間のみで解決できればよいので，initiator，responder ともに SLIN6ID を管理する MA を必要としない．その代わり，SLIN6 ではネットワークレイヤより上位のレイヤにおいて，LIN6ID と同じ働きを持つものとして 64bits の *Connection ID*（*CID*） を用いてホストを識別し，SLIN6 における MA はこの CID を管理する．

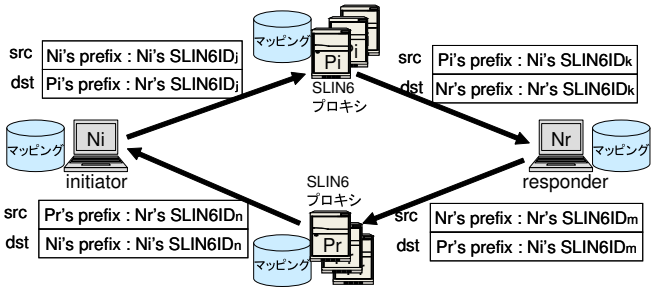

図 3: SLIN6 通信中の動作概要

また，IPv6 において 1 つのインタフェースは複数の IPv6 アドレスを持つことが可能であることを利用し，各ノードは複数の SLIN6ID を持つことができる．通信中はこの複数の SLIN6ID を始点アドレスとしてランダムに選択して使用し，終点アドレスも通信したい相手のが持つ SLIN6ID からランダムに選択して使用する．

位置指示子の隠蔽は，SLIN6 プロキシと呼ばれる中間ホストの導入によって実現する．SLIN6 において，SLIN6 プロキシがノード間のすべてのメッセージを中継し，通信開始時に生成したマッピングを基にパケットのアドレスを書き換え転送する．ノードは使用できる SLIN6 プロキシを複数持つが，すべての SLIN6 プロキシに対して信頼関係があるとする．ノードは通信時に SLIN6 プロキシをランダムに選択することができる．

SLIN6 通信中の動作概要を図 3 に示す．以下，initiator を N_i，responder を N_r，initiator が使用する SLIN6 プロキシを P_i，responder が使用する SLIN6 プロキシを P_r とする．SLIN6 の通信において，IP ヘッダのアドレス部分は図 3 に示す 4 種類となる．ここで，盗聴者はインターネットにおいてパケットが通る経路上の一箇所でしか同時に盗聴できないことを前提とする．図 3 のように，SLIN6 プロキシはマッピングを基にアドレスの SLIN6ID 部分を $SLIN6ID_j$ から $SLIN6ID_k$ に，$SLIN6ID_m$ から $SLIN6ID_n$ に変えて転送できる．したがって，このどの部分を盗聴されても，直接通信を行っているノード間の関連性がアドレスから特定できない．このように，SLIN6 では複数の SLIN6ID の使用によりノードの匿名性を実現し，複数の SLIN6 プロキシによるアドレス書き換えによりノードの位置秘匿性を実現する．

SLIN6 においては，通信中のパケットには特別な情報やヘッダを付加しないのでパケットオーバヘッドがなく，またパケットを中継するプロキシは 1 パケットにつき 1 つであるため，位置秘匿性を得るための経路の冗長性としては最低限のレベルであると考える．

SLIN6 汎用識別子は上位 64bits が SLIN6 prefix とよばれる固定値であり，下位 64bits は CID である．ネットワークレイヤでは，現在のノードが接続しているネットワークプレフィックスと SLIN6ID によって SLIN6 アドレスが生

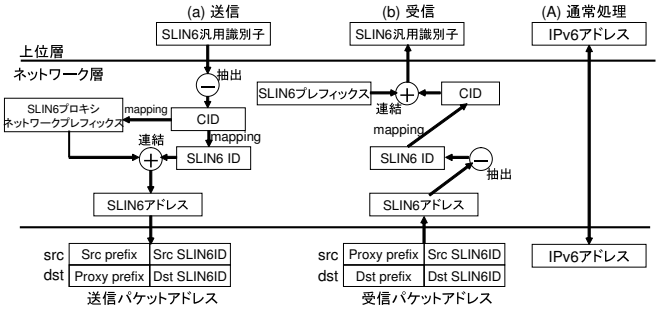

図 4: SLIN6 ノードにおける送受信処理

成され，これを使用する．これによって移動透過性を実現する．SLIN6ID の上位 24bits は SLIN6 に割り当てられた OUI（Organizationally Unique Identifier）であり，SLIN6 アドレスか否かはここを見れば判断できる．各 SLIN6 ノードや SLIN6 プロキシはカーネル空間にマッピングテーブルを持ち，

- CID
- SLIN6ID（複数個可能）
- 現在接続するサブネット
- SLIN6 プロキシのアドレス（複数個可能）
- セキュアなマッピング交換に使用する暗号鍵

といった要素のマッピングを保持する．

4.3 SLIN6 のパケット処理

4.3.1 SLIN6 ノード

SLIN6 ノードにおける送信時の処理を図 4(a) に示す．N_i の CID を CID_N_i，N_r の CID を CID_N_r と表す．対象となる SLIN6 汎用識別子の下位 64bit から CID_N_r を得る．次に，CID_N_r を鍵としてマッピングテーブルを検索する．その結果，その通信で使われているマッピングエントリを発見することが可能であり，N_i と N_r の SLIN6ID とネットワークプレフィックスと使用プロキシのアドレスが求められる．もしマッピングにある SLIN6ID，または使用プロキシが複数ある場合はその中からランダムに選ぶ．この N_i の SLIN6 アドレスを始点アドレス，使用プロキシのネットワークプレフィックスと N_r の SLIN6ID を連結させた SLIN6 アドレスを終点アドレスとしてパケットを配送する．

SLIN6 ノードにおける受信時の処理を図 4(b) に示す．対象となる LIN6 アドレスから LIN6ID 部分を抽出し，OUI により SLIN6ID または既存の LIN6ID かを判断する．SLIN6ID であった場合，この SLIN6ID を鍵にマッピング

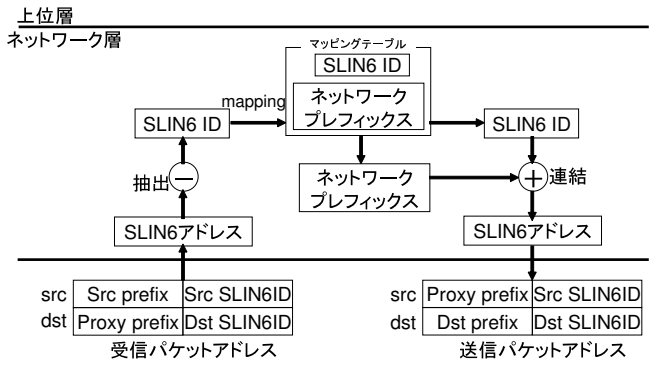

図 5: SLIN6 プロキシにおける送受信処理

テーブルを検索し，対応した SLIN6 汎用識別子を得る．また，既存の LIN6ID であった場合，既存の LIN6 の処理を行う．

SLIN6 ノードにおける通常の IPv6 アドレスの送受信処理を図 4(A) に示す．パケット送信時にトランスポートレイヤで指定された IPv6 アドレスが SLIN6 汎用識別子または LIN6 汎用識別子でない場合，通常の IPv6 アドレスとして処理される．また，パケット受信時にネットワークレイヤでの IPv6 アドレスが SLIN6 アドレスまたは LIN6 アドレスでない場合，通常の IPv6 アドレスとして処理される．

4.3.2 SLIN6 プロキシ

SLIN6 プロキシにおける転送時の処理を図 5 に示す．SLIN6 プロキシは受信パケットの終点アドレスからノード識別子部分を抽出し，OUI により SLIN6ID か否かを判断する．SLIN6ID であった場合，この SLIN6ID を鍵にマッピングテーブルを検索し，対応したネットワークプレフィックスと SLIN6ID を得る．もし SLIN6ID が複数あった場合，その中からランダムに選択する．そしてパケットの始点アドレス，終点アドレスを図 5 のように書き換えて送信する．

4.4 SLIN6 の通信開始手順

SLIN6 では，各ノード間や MA との情報交換，プロキシの転送設定に SLIN6 メッセージを用いる．SLIN6 メッセージは UDP を利用し，固定長の SLIN6 メッセージヘッダと可変長の SLIN6 メッセージボディから形成される．SLIN6 メッセージボディはメッセージタイプにより含まれる内容が異なる．通信開始時に SLIN6 メッセージを交換することにより，通信に必要なマッピングが各 SLIN6 ノード，SLIN6 プロキシに登録される．

SLIN6 の通信開始手順を図 6 に示す．N_r のマッピングを管理する MA を MA_{N_r}，N_i の SLIN6 プロキシを P_i，N_r の SLIN6 プロキシを P_r とする．SLIN6 は図 6 の (0) から

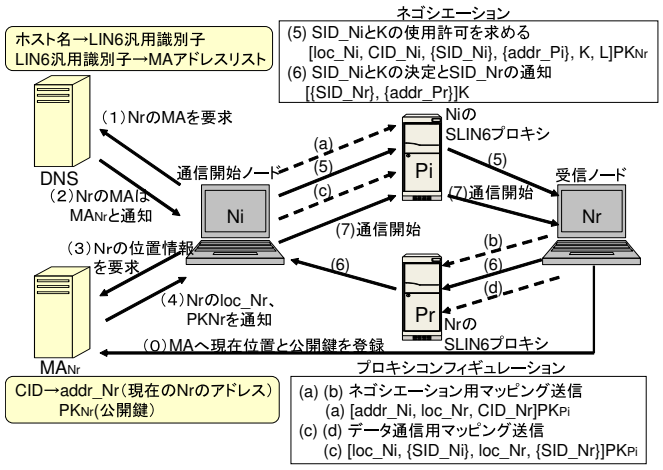

図 6: SLIN6 の通信開始手順

(7) の手順によって匿名通信を開始する．

N_r は responder 機能を持つので，N_r のマッピングを管理する MA_{N_r} へ CID_N_r，現在の IPv6 アドレス $addr\_N_r$，N_r の現在の公開鍵 PK_{N_r} を登録する（図 6(0)）．一定時間が経過した時または N_r が他のサブネットに移動しネットワークプレフィクスが変更された時に，N_r はこの現在位置の登録を再び行う．

N_i が N_r と SLIN6 を利用した通信を行う場合，N_i はまず DNS へ N_r のマッピングを管理する MA_{N_r} の IPv6 アドレスを問い合わせる（図 6(1)）．DNS は CID_N_r と対応する MA_{N_r} の IPv6 アドレスを通知する（図 6(2)）．

次に，N_i は MA_{N_r} へ N_r の位置情報を要求する（図 6(3)）．MA_{N_r} は N_i に N_r の現在の IPv6 アドレス $addr\_N_r$ と公開鍵 PK_{N_r} を通知する（図 6(4)）．このように，N_i は $addr\_N_r$ と PK_{N_r} を取得することができる．

次に N_i は MA_{N_r} から取得した情報を基に $addr\_N_r$ と CID_N_r を利用し，SLIN6ID を決定する処理に入る．N_i は loc_N_i を現在のネットワークプレフィックスとし，N_i の SLIN6ID である SID_N_i を複数ランダムに生成する．また，CID である CID_N_i と，N_i と N_r 間で使う共通鍵 K をランダムに生成し，共通鍵 K のライフタイムを L とする．そして，使用するプロキシ P_i を現在 N_i が利用できるプロキシリストの中からランダムに決定する．そのプロキシの IPv6 アドレスを $addr\_P_i$ とする．これらのパラメータは N_i 上のマッピングテーブルで保持される．その後，N_i は $[loc\_N_i, \{SID\_N_i\}, CID\_N_i, \{addr\_P_i\}, K, L]PK_{N_r}$ を含むメッセージを N_r にプロキシ P_i を中継させて N_r へ送信する（図 6(5)）．ここで $[loc_{N_i}, \{SID\_N_i\}, CID\_N_i, \{addr\_P_i\}, K, L]PK_{N_r}$ とは，loc_{N_i}，$\{SID\_N_i\}$，CID_N_i，$\{addr\_P_i\}$，K，L を PK_{N_r} を用いて暗号化することを意味しており，$\{SID\_N_i\}$ とは，SID_N_i が複数あることを意味している．N_r は N_i からのメッセージを受信すると，秘密鍵 SK_{N_r} でメッセージを復号化し，$\{SID\_N_i\}$ が N_r 上で一意であるかを確認した

後，N_r の SLIN6ID である SID_N_r を複数ランダムに生成する．そして，使用するプロキシ P_r を現在 N_r が利用できるプロキシリストの中からランダムに決定する．そのプロキシの IPv6 アドレスを $addr\_P_r$ とする．これらのパラメータは N_r 上のマッピングテーブルに保持される．次に，N_r は N_i へ $[\{SID\_N_r\}, \{addr\_P_r\}]K$ を含むメッセージをプロキシ P_r を中継させて N_i へ送信する（図 6(6)）．メッセージ受信した N_i は $[\{SID\_N_r\}, \{addr\_P_r\}]K$ を共通鍵 K で復号化し，N_i 内のマッピングテーブルを検索する．N_i は，検索により発見されたエントリに $\{SID\_N_r\}$，$\{addr\_P_r\}$ を追加する．SLIN6 では，この一連の処理をネゴシエーションと呼ぶ．ネゴシエーションを行うことによって，通信を行う両ノードがマッピングを持つことができ，図 3 で示したような SLIN6 アドレスを用いた匿名通信を開始することができる（図 6(7)）．

SLIN6 では，パケットを送信する前に必ず SLIN6 プロキシにアドレス書き換え設定を行うためのマッピングを登録する．これをプロキシコンフィギュレーションと呼ぶ．プロキシコンフィギュレーションには二種類あり，一つはネゴシエーションメッセージを送る前のコンフィギュレーション，もう一つは SLIN6 アドレスを用いた通信を行う前のコンフィギュレーションである．N_i は N_r にネゴシエーションメッセージ（図 6(5)）を送信する前に，$addr\_N_i$, loc_N_r, CID_N_r, を含むメッセージ $[addr\_N_i, loc\_N_r, CID\_N_r]PK_{P_i}$ を P_i へ送信する（図 6(a)）．ここでの PK_{P_i} とは，P_i の公開鍵のことである．一方，N_r は N_r にネゴシエーションメッセージ（図 6(6)）を送信する前に，$addr\_N_r$, loc_N_i, CID_N_i を含むメッセージ $[addr\_N_r, loc\_N_i, CID\_N_i]PK_{P_r}$ を P_r へ送信する（図 6(b)）．これらのプロキシコンフィギュレーションによって，プロキシはネゴシエーションメッセージを転送するためのマッピングを持つことができる．

ネゴシエーションが終わると，N_i は N_r に loc_N_i, $\{SID\_N_i\}$，loc_N_r，$\{SID\_N_r\}$ を含むメッセージ $[loc\_N_i, \{SID\_N_i\}, loc\_N_r, \{SID\_N_r\}]PK_{P_i}$ を P_i へ送信する（図 6(c)）．一方，N_r は N_i に loc_N_r, $\{SID\_N_r\}$, loc_N_i, $\{SID\_N_i\}$ を含むメッセージ $[loc\_N_r, \{SID\_N_r\}, loc\_N_i, \{SID\_N_i\}]PK_{P_r}$ を P_r へ送信する（図 6(d)）．これらのプロキシコンフィギュレーションによって，プロキシは SLIN6 アドレスを用いたデータパケットを転送するためのマッピングを持つことができる．

4.5 SLIN6 の移動処理

ノード A とノード B が SLIN6 を用いた匿名通信を行っており，ノード A が他のサブネットに移動したときの移動処理を図 7 に示す．ノード A を N_A，ノード B を N_B，現在のノード A のネットワークプレフィックスを loc_N_A，ノード B のネットワークプレフィックスを loc_N_B とする．N_A が他のサブネットに移動し，ネットワークプレフィクスが loc_N_A' に変更された場合，まず N_A は位置情報を管理する MA に対して移動後に得た IPv6 アドレス $addr\_N_A'$ を送信して位置情報を更新する（図 7(0)）．そして，N_A は P_A に対して新しいマッピングでプロキシコンフィギュレーションを行い（図 7(a)），N_B にプロキシ P_A を中継させてマッピングアップデートメッセージを送信する（図 7(1)）．このマッピングアップデートメッセージは，$[loc\_N_A', \{SID\_N_A\}, K, L]K$ であり，移動前に使用していた SLIN6 アドレスを用いて送信する．

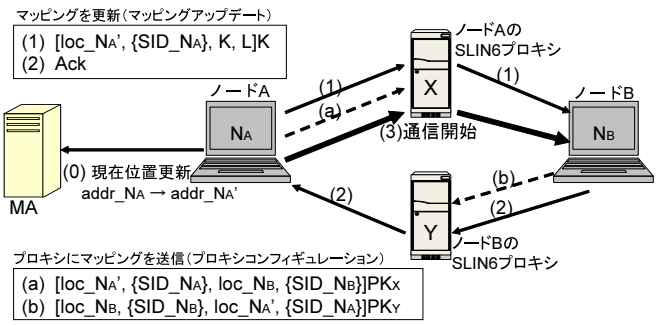

図 7: SLIN6 の移動処理

N_B は受信したマッピングアップデートメッセージからマッピングテーブルを検索し，取得した共通鍵 K を利用してマッピングアップデートメッセージを復号化した後，マッピングアップデートメッセージ内の共通鍵 K との整合性を確かめる．共通鍵 K が一致したならば，N_B はマッピング登録メッセージを送信した相手が正しい N_A であることを認証できる．この認証後，N_B は自身のマッピングテーブルを更新し，P_B に対して新しいマッピングでプロキシコンフィギュレーション（図 7(b)）を行った後，N_A に確認応答する（図 7(2)）．

N_A と N_B が同時に移動した場合，initiator であるノードが responder の MA に現在のネットワークプレフィクスを問い合わせ，再びネゴシエーションを行う．

4.6 SLIN6ID の衝突

通信開始時，または移動時に同一リンク内で同じ SLIN6ID が存在する可能性がある．その際は，Duplicated Address Ditection（DAD）[6] により SLIN6ID の重複を発見する．重複が発見されると，ノードは新たに SLIN6ID を生成し，通信相手ノードにマッピングを送信する．

5 SLIN6 の実装

LIN6 の実装を参考に，カーネル空間とユーザ空間に必要な機能を追加した形で SLIN6 を実現した．実装したオペ

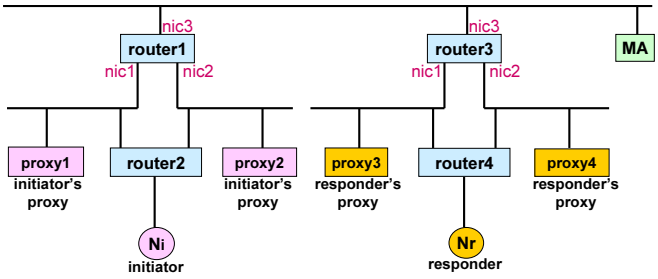

図 8: 実験のネットワークトポロジ

レーティングシステムは FreeBSD 5.4-release である．通信状態の情報をカーネル空間にキャッシュするために，カーネル空間にマッピングテーブルを生成した．また，ユーザ空間には SLIN6 における制御メッセージの送受信やカーネルと情報交換を行うデーモンを実装した．

SLIN6 ノードのカーネルはパケットのアドレスの SLIN6 汎用識別子と SLIN6 アドレスの変換処理を行う．MA，SLIN6 プロキシ，SLIN6 ノード間の SLIN6 メッセージの交換はユーザ空間のプログラムで処理し，カーネル空間に取得したデータが渡される．

SLIN6 プロキシのカーネルはパケットの IP ヘッダ内の SLIN6 アドレスの書き換え，転送処理を行う．SLIN6 ノードとの SLIN6 メッセージの交換はユーザ空間のプログラムで処理し，カーネル空間に取得したデータが渡される．

6 SLIN6 の性能評価

SLIN6 が実ネットワークにおいて動作することを確認した．そして，従来の IPv6 パケット処理と比べたネットワークレイヤにおける SLIN6 のパケット処理時間のオーバヘッドを測定した．その結果，SLIN6 ノードにおいて送信処理は μsec オーダで +12.47%，受信処理は $10\mu sec$ オーダで +44.69% となったが，実ネットワークでの Round Trip Time（RTT）が $msec$ オーダであることを考慮するとこのオーバヘッドは無視できる．また，SLIN6 プロキシにおいて受信からアドレスを書き換えて送信までの処理は時間は $134\mu sec$ であり，$100\mu sec$ オーダであることからこのオーバヘッドも無視できる．

次に，図 8 および表 2 に示す実験環境において SLIN6 の TCP スループットを測定した．スループットの測定には netperf を使用し，1 分間 N_i から N_r へ 32768bytes のデータパケットを送信し，N_r から N_i へ ack パケットを送信した．図 8 において，矢印が示すフローの方向に対して dummynet によって遅延を発生させた．SLIN6 プロキシとして N_i は proxy1，proxy2 を使用し，N_r は proxy3，proxy4 を使用した．SLIN6ID はそれぞれ 3 つずつ使用した．

TCP スループットの測定結果を図 9 に示す．図 9 の縦軸は TCP スループットを示し，横軸は遅延の組み合わせを示す．遅延の組み合わせの内容は（*router1* が *nic1* で受信するパケットに付加する遅延，*router1* が *nic2* で受信するパケットに付加する遅延，*router3* が *nic1* で受信するパケットに付加する遅延，*router3* が *nic2* で受信するパケットに付加する遅延）である．例えば（10，10，10，50）は router1 の nic1，nic2，router3 の nic1 へ向かうパケットの遅延が 10（ms）であり，router3 の nic2 へ向かうパケットの遅延が 50（ms）であることを表している．パターン a は SLIN6 を用いず通常の通信方式を用いた測定，その他のパターンは SLIN6 を用いた測定である．

表 2: 実験に使用したマシン

machine	CPU	memory
Ni (initiator)	Pentium-M 1.2GHz	512MB
Nr (responder)	Pentium-3 800MHz	640MB
proxy,router,MA	VIA C3 800MHz	512MB

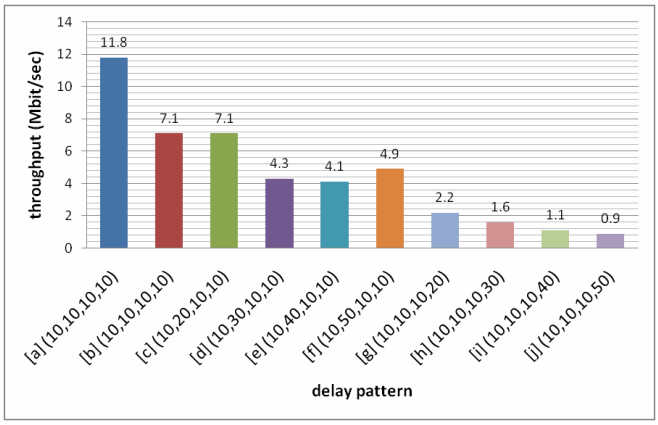

図 9: SLIN6 の TCP スループット

図 9 のパターン a とパターン b を比較すると，本実験において SLIN6 を用いた場合は通常の場合に比べ TCP スループットが 3 分の 2 程度に減少することが分かる．また，パターン b とその他の SLIN6 を用いたのパターンとを比べると，パケットに順序違いが発生すると TCP スループットがさらに減少することが分かる．また，パターン c，d，e，f では N_r に到着するデータパケットに順序違いが発生し，パターン g，h，i，j では N_i に到着する ack パケットに順序違いが発生することを考えると，データパケットに順序違いが発生する場合より ack パケットに順序違いが発生する場合の方が，TCP スループットが大きく減少することが分かる．

7 考察

7.1 盗聴者に対する位置秘匿性と匿名性

SLIN6 において，盗聴の可能性があるパケットの IP ヘッダ情報として，図 3 に示したように 4 種類が考えられる．以降の議論では，盗聴者は同時には 1ヶ所でしか盗聴できないことを前提とする．

SLIN6ID は動的に生成，変更することにより，ノードとの関連性を発見することが困難になる．また，複数の SLIN6ID を用い，さらに SLIN6 プロキシによって SLIN6ID が書き換えられるので相関を得ることも困難になる．また図 3 のどの IP ヘッダの情報を見ても，SLIN6 がパケットを中継しているため始点または終点アドレスのどちらか片方のプレフィックスは必ず SLIN6 プロキシのものとなる．SLIN6 プロキシも複数使用できるので，どのサブネット同士が通信を行っているかを判別することは困難となる．したがって，initiator と responder は盗聴者に対して位置秘匿性と匿名性があると言える．

7.2 盗聴者の追跡に対する耐性

SLIN6 はマッピングを含む制御メッセージを暗号化しており，SLIN6ID を動的に変更してよいことから SLIN6ID を追跡される可能性は極めて低い．また SLIN6 プロキシにより，SLIN6ID が書き換えられるので SLIN6ID の相関を得ることも困難となる．したがって，initiator と responder は盗聴者の追跡に対して耐性があると言える．

7.3 通信開始時のオーバヘッド

SLIN6 の通信開始時にはネゴシエーション，プロキシコンフィギュレーションといった処理のオーバヘッドが発生する．この通信開始時の処理において，動的に生成した SLIN6 アドレスをインターフェースに付与する際は，DAD 処理によって大きな遅延が発生する．しかし，この遅延は通信開始時のみで通信中は発生しないので，位置秘匿性，匿名性を得ることを考えると許容できると考える．

7.4 送受信時のオーバヘッド

SLIN6 のアドレス書き換え処理によるオーバヘッドは処理時間が非常に小さいため無視できることが分かった．よって，SLIN6 の通信におけるパケット送受信時の遅延は，SLIN6 プロキシがパケットを中継することによってパケットが冗長経路を通ることによる遅延である．しかし，各パケットは 1 つの SLIN6 プロキシのみに中継されればよく，位置秘匿性を得ることを考えるとこのオーバヘッドは最小限である．

7.5 SLIN6ID の衝突検知処理

通信開始時または移動時に，使用する SLIN6ID が同一リンク上で衝突しないか DAD により確かめる必要がある．これはインタフェースに付与するすべての SLIN6 アドレスにおいて行われるので，より多くの SLIN6ID を用いる場合はその SLIN6ID の数だけ DAD を行う必要があり，処理時間のオーバヘッドが増大する．また衝突が検知された場合は，SLIN6ID の再決定をし新しいマッピングを送信しなければならない．しかし，SLIN6ID は 40bits 空間であり既存の IPv4 の 2^8 倍のアドレス空間を保持できるので，SLIN6ID の衝突確率はほぼ無視できる．

7.6 実用性の考察

SLIN6 通信において，パケットの順序違いが発生する度合は SLIN6 プロキシの接続位置や使用数によって異なる．使用する SLIN6 プロキシの数が多く各々の経路における RTT（Round Trip Time）の差が大きいと，パケットの順序違いの割合は大きくなり TCP スループットに大きな影響が出ると考えられる．特に ack パケットの順序違いによる影響は，図 9 で分かるように顕著に現われてしまう．これは，データを送信するノードが受信した ack パケットによって次のデータパケットの送信を決定する TCP の特性により，送信待ちを行ったり無駄な再送を行ったりすることが原因と考えられる．

よって，SLIN6 プロキシの使用数を増やすほど位置秘匿性はより向上するが，TCP スループットはより低下してしまうので，どちらを優先させるかで SLIN6 プロキシの使用数や使用位置を検討する必要がある．

一方，使用する SLIN6ID の数については，DAD 処理による遅延には影響があるが，マッピングを交換した後のデータ通信中においては TCP スループットにほとんど影響を及ぼさない．

7.7 今後の課題

SLIN6 通信において，responder になるすべてのノードは匿名でのアクセスを許可している．実運用を想定した場合，どのようなノードに対して匿名でのアクセスを許可するかポリシを決める必要がある．また，SLIN6 プロキシの使用方法や運用方法の考察と，様々な SLIN6 プロキシの配置におけるトランスポートレイヤの評価，様々なアプリケーションに対する適正度を評価する必要がある．

8 結論

SLIN6 は移動透過性を保証するモビリティプロトコル LIN6 に必要な機能を加え，移動透過性を失うことなく以下の 2 つの匿名性や位置秘匿性を実現できた．

- 盗聴者に対する両端ホストの匿名性，位置秘匿性
- 盗聴者による追跡を防ぐ移動ノードの匿名性，位置秘匿性

SLIN6 では，通信開始時にネゴシエーション，プロキシコンフィギュレーションを行うことによって位置秘匿性，匿名性を有する通信を行うことができる．SLIN6 通信中は匿名通信用の情報や IP ヘッダがパケットに付加されず，パケット処理のオーバヘッドは無視できる．また，SLIN6 プロキシの配置次第では，TCP スループットに大きな影響が出ることが分かった．SLIN6 の実運用を考えると，SLIN6 プロキシ運用方の決定や匿名通信ポリシの決定などの課題が挙げられる．

参考文献

[1] D. Johnson, C. Perkins and J. Arkko. Mobility Support in IPv6. RFC3775, Jun. 2004.

[2] Mitsunobu Kunishi, Masahiro Ishiyama, Keisuke Uehara, Hiroshi Esaki and Fumio Teraoka. LIN6 : A New Approach to Mobility Support in IPv6. In *Proceedings of the Third Inernational Sympsium on Wireless Personal Multimedia Communications*, pages 1079–1084, Nov. 2000.

[3] David Chaum. Untraceable Electronic Mail, Return Addresses and Digital Pseudonyms. *Communications of the ACM*, pages 84–88, Feb. 1981.

[4] Michael K. Reiter and Aviel D. Rubin. Crowds: Anonymity for Web Transaction. *ACM Transactions on Information and System Security*, pages 1(1):66–92, Nov. 1998.

[5] Pekka Nikander, Jukka Ylitalo and Jorma Wall. Integrating Security, Mobility, and Multi-homing in a HIP Way. In *Proceedings of Network and Distributed Systems Security Symposium (NDSS)*, pages 87–99, Feb. 2003.

[6] S. Thomson and T. Narten. IPv6 Stateless Address Autoconfiguration. RFC2462, December 1998.