# DHCPv6サーバを用いたNDPにおける攻撃の検知手法の提案

柏田 拓哉[1,a)]　今泉 貴史[2,b)]

**概要**：近隣探索プロトコル（Neighbor Discovery Protocol:NDP）は同一リンク上でIPv6パケットを送信するために必要な機能を提供する．NDPにはメッセージの妥当性を確認する仕組みが存在しないことから，攻撃者は改ざんしたメッセージを送ることで，容易に近隣ノードのキャッシュを操作することができてしまう．NDPを保護する手法の研究の1つに，DHCPv6サーバを利用するものがあるが，この研究の手法ではNDPの仕様に変更を加えており実用が難しい．そこで本研究では，NDPの仕様に変更は加えずに，DHCPv6サーバを利用してNDPを保護する手法を提案する．

**キーワード**：IPv6，近隣探索，NDP，DHCPv6

# Proposal of attack detection method in NDP using DHCPv6 server

Kashiwada Takuya[1,a)]　Imaizumi Takashi[2,b)]

***Abstract:*** Neighbor Discovery Protocol (NDP) provides the necessary functions to send IPv6 packets on the same link. Since there is no mechanism for checking the validity of messages in NDP, attackers can easily manipulate the cache of neighboring nodes by sending tampered messages. One of the researches on the method of protecting NDP uses a DHCPv6 server, but this research method has changed the protocol and is difficult to put into practical use. In this study, we propose a method to protect NDP using DHCPv6 server without changing the protocol.

***Keywords:*** IPv6，Neighbor discovery，NDP，DHCPv6

## 1. はじめに

インターネット利用者数の加速度的な増加により，IPv4ではIPアドレスの不足を筆頭に様々な問題が生じている．こうした問題を解決すべくIPv6が誕生し，現在IPv6を利用したネットワークの使用が急速に広まっている．

近隣探索プロトコル（Neighbor Discovery Protocol:NDP）はIPv6において最も重要なプロトコルであり，IPv4におけるARPやICMPルータ発見に相当する機能を実現するほか，ノードのIPv6アドレス自動生成などIPv4にはなかった機能も提供する．NDPではICMPv6メッセージを用いて情報をやり取りするが，このときに通信相手を認証する仕組みが存在しない．すなわち，攻撃者から送信元IPv6アドレスなどが偽装されたメッセージが送られてきても，受信側はメッセージの改ざんを検知することができない．このことがNDPを悪用した様々な攻撃の原因となっている．

従って，IPv6ネットワークはNDPの脆弱性に由来する数多くの攻撃が発生する危険を抱えており，攻撃者の脅威からノードを守るための対策手法が必要とされている．標準化された保護手法として既にSEND[1]が存在するが，これは計算コストが重く実用に堪えないといった問題があることから，SENDの利用は現在でも一般的であるとは言い

[1] 千葉大学大学院融合理工学府
〒263-8522 千葉県千葉氏稲毛区弥生町1-33
[2] 千葉大学統合情報センター
〒263-8522 千葉県千葉氏稲毛区弥生町1-33
a) t.kashiwada@chiba-u.jp
b) imaizumi_takashi@faculty.chiba-u.jp

難い [2]．そのため，より実用的な NDP の保護手法が研究されてきた．

こうした研究の 1 つに Ahmed らの研究 [3] がある．この研究では DHCPv6 サーバを利用した比較的軽量なアルゴリズムを提案しているが，NDP の仕様に大きく変更を加える手法になっていることから，実際に用いるのは難しい．そこで，本研究では，NDP の仕様に変更を加えない形で DHCPv6 サーバを利用して攻撃を検知する手法を提案する．

本論文の残りは以下のような構成となっている．2 章で NDP と NDP を悪用した攻撃について説明し，3 章で NDP の保護を目的とした研究を紹介する．4 章で提案手法のアルゴリズムについて述べ，5 章でそれを評価し，6 章では既存手法との比較等の考察を行う．最後に 7 章でまとめを述べる．

## 2. 近隣探索プロトコル

RFC4861[4] では NDP が提供する機能とそのプロセス，使用されるメッセージのフォーマットなどが定義されている．NDP の主な機能の 1 つとして，IPv6 アドレスからリンク層のアドレスを取得する，アドレス解決機能がある．IPv4 では Address Resolution Protocol（ARP）[5] を用いてアドレス解決を行っていたが，IPv6 の NDP では以下の 2 種類の ICMPv6 メッセージを用いてこれを実現する．

**近隣要請（Neighbor Solicitation:NS）**
ノードによって発信され，別のノードのリンク層アドレスを要求する．

**近隣広告（Neighbor Advertisement:NA）**
NS に対する応答として発信され，ノードのリンク層アドレスを通知する．

IPv6 では近隣キャッシュ（Neighbor Cache）で IPv6 アドレスとリンク層アドレスの対応が管理される．ある IPv6 アドレスにパケットを送信する際には，まず近隣キャッシュを参照して対応する IPv6 アドレスを検索する．もし近隣キャッシュに対応する IPv6 アドレスが存在しなければ，ノードは NS をマルチキャストし，NA が返されるのを待つ．ここで，NS の IPv6 ヘッダ内の宛先 IPv6 アドレスフィールドにはアドレス解決したい IPv6 アドレスに対応するマルチキャストアドレスが設定されるが，アドレス解決したい IPv6 アドレス自体は ICMPv6 ヘッダ内のターゲット IPv6 アドレスフィールドに設定される．NS を受信した当該 IPv6 アドレスの持ち主は，NS の送信元 IPv6 アドレスに対して NA をユニキャストすることで応答する．ここで，通知すべきリンク層アドレスは ICMPv6 ヘッダ内のターゲットリンク層アドレスオプションに設定され，対応する IPv6 アドレス（NA の送信元 IPv6 アドレス）が ICMPv6 ヘッダ内のターゲット IPv6 アドレスフィールドに設定される．NDP のアドレス解決のプロセスを**図 1** に示す．

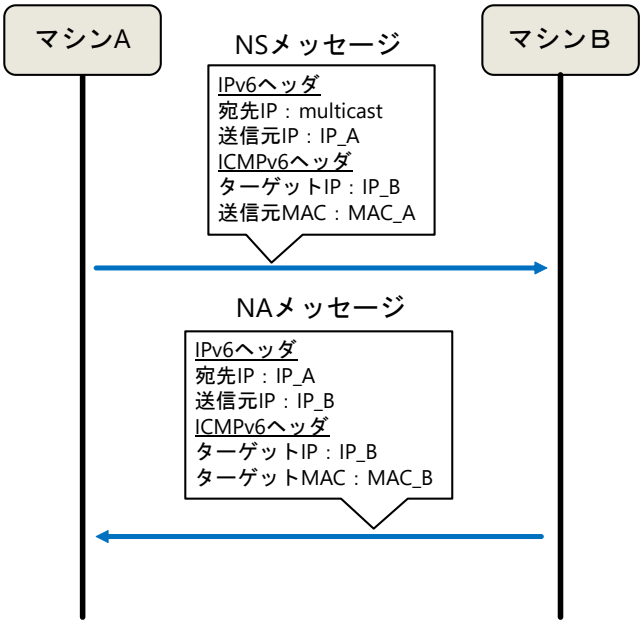


**図 1** アドレス解決のプロセス
**Fig. 1** Address resolution process

### 2.1 自発的な近隣広告

通常，NA は NS に対する応答として送信される．しかし，ノードのリンク層アドレスが変更されたとき，ノードはこの変更を近隣ノードのキャッシュに素早く反映させるために，NA をマルチキャストすることができる．このような NA は自発的な近隣広告（Unsolicited NA: UNA）と呼ばれる．UNA を受信したノードは，自身の近隣キャッシュの対応エントリを受信した情報に基づいて更新する．UNA は本来，アドレス解決の回数を減らしてネットワークのパフォーマンスを向上させるために用いられる．

### 2.2 近隣キャッシュの汚染

RFC3756[6] では，攻撃者が任意のリンク層アドレスを設定した NS または NA を送信することで，ノードの近隣キャッシュが上書きされ，なりすましや DoS 攻撃が成功してしまう問題が報告されている．

ノードは基本的に NS を受け取った際に，該当するエントリが近隣キャッシュに存在しなければ，新しくエントリを作成する．また，既存のエントリと受信した NS が示す送信元リンク層アドレスが異なる場合，そのエントリを上書きする．このような挙動から，攻撃者は IPv6 アドレスまたはリンク層アドレスを詐称した NS を送信するだけで，近隣ノードの近隣キャッシュに任意のエントリを作成あるいは上書きすることが可能である．なお，NA は受信時に該当するエントリが存在しない場合，新しくエントリを作成することはないが，エントリが存在し NA の送信元リンク層アドレスと異なる場合は，NS と同様にそのエントリ

を上書きする．

いくつかの研究 [3], [7], [8] では，このような攻撃の疑いがある NS，NA が観測された際に，そのターゲット IPv6 アドレスで改めてアドレス解決を試みて，これに対し返ってくる近隣広告の情報を確認して攻撃どうか判断する手法をとっている．以後，本稿ではこのように攻撃の疑いがあるパケットに反応して送信される近隣要請を NS_PROBE と呼ぶ．

### 2.3 重複アドレス検出に対する DoS 攻撃

IPv6 ノードがネットワークに参加するために自身へ IPv6 アドレスを設定した際，ノードはその IPv6 アドレスがネットワークで使用されていないことを確認しなければならない．この動作は重複アドレス検出（Duplicate Address Detection）と呼ばれる．重複アドレス検出に失敗（アドレスの重複を確認）すると，ノードは IPv6 アドレスを設定できずネットワークへ参加することができない．

重複アドレス検出では，自身が使用する予定の IPv6 アドレスについて NS を送信し，リンク層アドレス解決を行う．このとき NA がネットワークから返ってこなければ，アドレスは重複していないとしてその IPv6 アドレスで通信を開始する．このため，本来その IPv6 アドレスを使用していないノードでも，送信元 IPv6 アドレスを偽装した NA を返すことで，容易に他人の重複アドレス検出を失敗させることができる．

もう一つ，重複アドレス検出が失敗する条件がある．それは，他のノードが同時に同じ IPv6 アドレスに対して重複アドレス検出を行っていて，NA を受信する前にその NS を受信したときである．このような重複アドレス検出の失敗も，攻撃者が重複アドレス検出を装った NS を送信することで引き起こすことができる．

## 3. 関連研究

2.2 節，2.3 節への対策として提案される手法は，プロトコルに変更を加えるものとそうでないものに分けられる．

### 3.1 プロトコルに変更を加える手法

#### 3.1.1 SEND

NDP にセキュリティを提供する仕組みとして，RFC 3971[1] で標準化されている SEND（SEcure Neighbor Discovery）がある．SEND では新たな ICMPv6 メッセージオプションを導入することによって近隣探索メッセージに RSA 署名を施すことを可能にし，メッセージの改ざんや送信元アドレスの詐称などを防止している．しかし，メッセージの署名及び検証によるオーバーヘッドが大きく，実装も不十分な状況であることから，SEND の利用は現在でも一般的ではない．

#### 3.1.2 DHCPv6 サーバを用いる手法

DHCPv6 サーバを用いて NDP の脆弱性を緩和する手法として Ahmed らの研究 [3] が存在する．この研究では NS_PROBE に対し 2 つ以上の近隣広告を受信したとき，その IPv6 アドレスについて DHCPv6 サーバへリース情報を問い合わせることで，攻撃者の MAC アドレスを特定する．

リース情報の問い合わせには，RFC5007[9] で定義される 2 種類の DHCPv6 メッセージが用いられる（**表 1**）．IPv6 アドレスを用いて LEASEQUERY を行うことで，DHCPv6 サーバからは対応する DHCP Unique Identifier（DUID）が返ってくる．RFC3315[10] で定義される DUID のうち，DUID-LLT および DUID-LL についてはリンク層アドレス情報を含むため，IPv6 アドレスに対応するリンク層アドレスを把握することができる．

この研究の手法は，SEND に比べると署名などの計算コストの大きい処理無しに攻撃者の特定が可能である．しかし一方で，新たな ICMPv6 メッセージオプションを追加したり、NS・NA のやり取りの手順が増えていることから，NDP の仕様に大きく変更が加わってしまっている．従って、実用するのは現実的ではない．

### 3.2 プロトコルに変更を加えない手法

#### 3.2.1 NDPMon

NDPMon[11] はネットワーク内の通信を監視して，ノードの IP-MAC アドレスペアを独自テーブルで保持する．不正なアクティビティを検出した際にアラートを出したり，ユーザ定義のスクリプトを自動実行することができる．

NDPMon の問題点として，ノードの IP-MAC アドレスペアを学習している間は攻撃が検出できない点や，学習中に攻撃が発生した場合にシステムが失敗する点が挙げられる [7], [8].

#### 3.2.2 その他の NIDS 型の手法

NDPMon と同様の NIDS 型の手法として，Ferdous らの研究 [7] がある．この手法では NDPMon のようにテーブルで IP アドレスと MAC アドレスの対応を保持しつつも，攻撃の疑いがあるパケットを受信した際に NS_PROBE を送信してこれを判定する．

## 4. 提案手法

本研究では，プロトコルには変更を加えずに，DHCPv6 サーバを利用して攻撃を検知する手法を提案する．提案手法は有線のイーサネットで接続された同一リンク上のノードに作用するシステムであり，別リンクの同システムや DHCPv6 リレーエージェントの存在は考慮しない．また，提案手法は以下の前提条件に依るものとする．

- リンク層アドレスには MAC アドレスが使われている．
- ノードの IPv6 アドレスは DHCPv6 サーバからリース

表 1　メッセージ定義
**Table 1**　Message definition

| 番号 | メッセージ名 | 内容 |
|---|---|---|
| 14 | LEASEQUERY | リース情報の問い合わせ |
| 15 | LEASEQUERY-REPLY | LEASEQUERY に対する応答 |

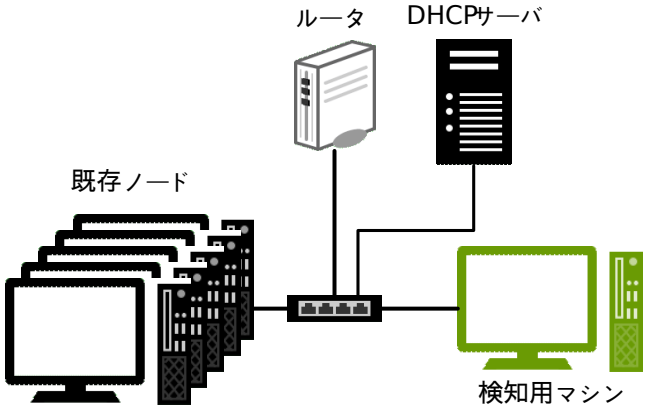


図 2　提案手法の稼働環境
**Fig. 2**　Operating environment of the proposed method

され，DHCPv6 サーバはネットワーク内のすべてのノードのリース情報を管理する．

- 提案手法を実装したシステムは図 **2** のように監視対象のノードと同一リンク上に存在し，通信用のポートとは別にミラーポートに接続され，ネットワーク上の全てのパケットを観測できる．
- システムは稼働時点で，DHCPv6 サーバから入手したリース情報によって，稼働直前のネットワーク内の正しい IP-MAC マッピングをテーブルとして保持している．
- 攻撃者は提案手法を実装したシステムと同一リンク上に存在し，DHCPv6 サーバからリースされた正規の IPv6 アドレスを用いずに，もしくはリースされる前に攻撃を行う．
- 攻撃者ではないノードは，必ず DHCPv6 サーバからリースされた正規の IPv6 アドレスで通信する．リース期限切れを起こした IPv6 アドレスも通信には用いない．
- MAC アドレスを変更するノードは，UNA を送信する前に，新しい MAC アドレスに対して IPv6 アドレスを DHCPv6 サーバからリースされている．また，UNA による近隣キャッシュの更新は，必ず IPv6 アドレスリース直後に最初に行う．

システムはネットワーク上のパケットをミラーポートから受け取り，観測された NS を Algorithm 1，NA を Algorithm 2 で処理することで，それらがアドレスを詐称していないかを確認する．システムは DHCPv6 サーバのリース情報に由来するネットワーク内の IP-MAC アドレスペアをテーブルとして保持する．アルゴリズムではまず，テーブルの情報と NS・NA のアドレス情報を照らし合わせることで，アドレスが詐称されていないか確認する．ほとんどの場合，ノードは DHCPv6 サーバからリースされた IPv6 アドレスを同じ MAC アドレスのまま利用し続けるので，テーブルの情報と齟齬は発生せず正常な通信であると判断される．一方，ネットワーク内で攻撃者が自分にリースされた IPv6 アドレス以外のアドレスで通信しようとすれば，テーブルの情報と齟齬が発生し異常が発生していることを検知できる．ただし，DHCPv6 サーバから新たにリースされた IPv6 アドレスなどはシステムが保持するテーブルに反映されていない場合が考えられる．そこで，異常を検知したときに LEASELQUERY を用いて DHCPv6 サーバに問い合わせることで，攻撃かどうかを判断する．攻撃ではないと判明すればその情報をテーブルに反映させることで，システムが DHCPv6 サーバの最新のリース情報を把握できたことになる．以上の動作を繰り返すことで，システムがネットワーク内の正しい IP-MAC アドレスペアを把握し続けて，ネットワーク内のアドレス詐称を検知することが可能になる．

### 4.1　用語の定義

この節では後述する疑似アルゴリズムで用いる略語の定義を行う．

*NS* 近隣要請（のパケット）
*NA* 近隣広告（のパケット）
*SIP* 送信元 IP アドレス
*SMAC* 送信元 MAC アドレス
*TIP* ターゲット IP アドレス
*TMAC* ターゲット MAC アドレス
$NS_{SIP}$ 近隣要請パケットの送信元 IP アドレス

### 4.2　アルゴリズム

Algorithm 1 はネットワーク上で観測された NS を処理する．まず，そもそも $NS_{SIP}$ がテーブルに存在しない場合，それはシステム稼働後に新たに加わった IPv6 アドレスか，DHCPv6 サーバでリースしていない不正な IPv6 アドレスのどちらかと考えられる．よって LEASEQUERY を発行して，リース情報が存在すれば攻撃ではないと判断し，テーブルを更新する．リース情報が存在しないならば，ネットワーク上に存在しない不正な IPv6 アドレスを用いた攻撃であると判断する．

$NS_{SIP}$ がテーブルに存在する場合は，$NS_{SMAC}$ が対応

するエントリの MAC アドレスと一致するか確かめる．一致していれば当然この NS は問題ないと判断できるが，一致しない場合は $NS_{SMAC}$ が $NS_{SIP}$ になりすまそうとしていると判断する．なお，$NS_{SMAC}$ が MAC アドレス変更後に $NS_{SIP}$ を利用している可能性は，前提条件より変更直後の UNA でテーブルに正しいエントリが予め存在しているはずであることからあり得ない．

**Algorithm 1** NS の処理アルゴリズム

```
Input NS_SIP, NS_SMAC
Output 検証結果
if テーブル内に NS_SIP が存在 then
  if 対応エントリの MAC アドレス = NS_SMAC then
    検証結果：NS_SMAC は NS_SIP の正当なリース先
  else
    検証結果：攻撃を検知
  end if
else
  NS_SIP について LEASEQUERY を発行
  if リース情報が存在しない or MAC アドレスが一致しない
  then
    検証結果：攻撃を検知
  else
    検証結果：NS_SIP は新規の正規ノード
    テーブルにエントリを追加
  end if
end if
```

Algorithm 2 はネットワーク上で観測された NA を処理する．Algorithm 1 と異なる点として，テーブル内に $NA_{TIP}$ が存在し，かつ $NA_{TMAC}$ が対応エントリの MAC アドレスと一致しない場合に，必ずしも攻撃とは限らないことがある．MAC アドレスを変更したノードがそれまでと同じ IPv6 アドレスを新しい MAC アドレスにもリースされる可能性があるからである．よって，Algorithm 2 では LEASEQUERY を用いてこれを判断する．

**Algorithm 2** NA の処理アルゴリズム

```
Input NA_TIP, NA_TMAC
Output 検証結果
if テーブル内に NA_TIP が存在 then
  if 対応エントリの MAC アドレス = NA_TMAC then
    検証結果：NA_TMAC は NA_TIP の正当なリース先
  else
    NA_TIP について LEASEQUERY を発行
    if リース情報が存在しない or MAC アドレスが一致しない
    then
      検証結果：攻撃を検知
    else
      検証結果：NA_TMAC は NA_TIP の正当なリース先
      テーブルにエントリを追加
    end if
  end if
else
  NA_TIP について LEASEQUERY を発行
  if リース情報が存在しない or MAC アドレスが一致しない
  then
    検証結果：攻撃を検知
  else
    検証結果：NA_TIP は新規の正規ノード
    テーブルにエントリを追加
  end if
end if
```

## 5. 提案手法の評価

2.2 節，2.3 節で述べた攻撃について，具体的な攻撃例を用意し，提案手法がこれを正常に処理できるかどうかを確かめる．

**1. なりすまし攻撃**

攻撃者 X がホスト A の IPv6 アドレス（$IP_A$）に，自分の MAC アドレス（$MAC_X$）を結びつけるような NS を，近隣ノードに送信したとする．このとき，$NS_{SIP} = IP_A$，$NS_{SMAC} = MAC_X$ となる．

NS がシステムに観測されると，Algorithm 1 による処理が開始する．まず，$IP_A$ はホスト A にリースされた正規の IPv6 アドレスなので，テーブルに存在する．このとき $IP_A$ のエントリの MAC アドレスは $MAC_A$ であるはずなので，$MAC_X \neq MAC_A$ であり攻撃を検知する結果となる．実際，これは攻撃だったので，正常に処理が完了して

いることになる．

**2. 未使用 IPv6 アドレスを使用した DoS 攻撃**

攻撃者 X が DHCPv6 サーバでリースされていない IPv6 アドレス（$IP_Y$）に，ホスト A の MAC アドレス（$MAC_A$）を結びつけるような NS を，近隣ノードに送信したとする．このとき，$NS_{SIP} = IP_Y$，$NS_{SMAC} = MAC_A$ となる．

NS がシステムに観測されると，Algorithm 1 による処理が開始する．まず，$IP_Y$ はリースされていない IPv6 アドレスなので，テーブルに存在しない．従ってシステムは $IP_Y$ について LEASELQUERY を発行する．DHCPv6 サーバはこれを受けて，リース情報が存在しないことを伝える．よって攻撃を検知する結果になる．実際，これは攻撃だったので，正常に処理が完了していることになる．

**3. 重複アドレス検出に対する DoS 攻撃**

攻撃者 X がホスト B の重複アドレス検出に対し，ホスト B の使用予定の IPv6 アドレス（$IP_B$）とホスト B の MAC アドレス（$MAC_B$）を使った NA で応答する．このとき，$NA_{TIP} = IP_B$，$NA_{TMAC} = MAC_B$ となる．

NA がシステムに観測されると，Algorithm 2 による処理が開始する．$IP_B$ はリース済みの IPv6 アドレスではあるが，この時点ではシステムのテーブルに存在しない．従ってシステムは $IP_B$ について LEASELQUERY を発行する．DHCPv6 サーバはこれを受けて，$IP_B$ のリース先が $MAC_B$ であることを伝える．よって $IP_B$ は新規の正規ノードであり攻撃は発生していないと判定する．実際には攻撃は発生していたので，正常に処理できなかったことになる．

## 6. 考察

### 6.1 既存手法との比較

3.2.2 項で紹介した Ferdous らの研究は，1. ネットワーク内の信頼できる IP-MAC アドレスペアをテーブルとして保持し，2. 場合によってはシステムが能動的に通信を行って攻撃かどうかを判定する，という 2 点が提案手法と共通する．そこで，Ferdous らの研究を「既存手法」として，提案手法と比較し，考察する．

既存手法における近隣キャッシュ汚染の検知アルゴリズムを，提案手法と比較する形で要約すると，Algorithm 3, 4 のようになる．

AUTH テーブルは既存手法によって安全であると判断された IP-MAC アドレスペアを保持している．$NS_{SIP}$ がこのテーブル内に存在すれば，$NS_{SMAC}$ が攻撃者かそうでないかが直接判断できる．そうでない場合に，NS_PROBE を発行して NA を受信し，情報を NA テーブルに格納する．そして NA テーブルを参照して，$NS_{SIP}$ に対応する MAC アドレスが $NS_{SMAC}$ でなければ，攻撃が発生していると判断する．

**Algorithm 3** 既存手法の NS 処理アルゴリズム

```
Input NS_SIP, NS_SMAC
Output 検証結果
if AUTH テーブル内に NS_SIP が存在 then
  if 対応エントリの MAC アドレス = NS_SMAC then
    検証結果：NS_SMAC は NS_SIP の正当な持ち主
  else
    検証結果：攻撃を検知
  end if
else
  NS_SIP について NS_PROBE を発行
  受信した NA の TIP, TMAC を NA テーブルへ格納
  if NA テーブル内に NS_SIP が存在 then
    if 対応エントリの MAC アドレス ≠ NS_SMAC then
      検証結果：攻撃を検知
    end if
  end if
end if
```

**Algorithm 4** 既存手法の NA 処理アルゴリズム

```
Input NA_TIP, NA_TMAC
Output 検証結果
if NS テーブル内の NS_TIP = NA_TIP then
  if AUTH テーブル内に NA_TIP が存在 then
    if 対応エントリの MAC アドレス = NA_TMAC then
      検証結果：NA_TMAC は NA_TIP の正当な持ち主
    else
      検証結果：攻撃を検知
    end if
  else
    NA_TIP について NS_PROBE を発行
    受信した NA の TIP, TMAC を NA テーブルへ格納
    if NA テーブル内に NA_TIP が存在 then
      if 対応エントリの MAC アドレス ≠ NA_TMAC then
        検証結果：攻撃を検知
      end if
      AUTH テーブルを更新
    end if
  end if
else
  UNA ハンドラへ処理を移行
end if
```

表 2　評価の結果
**Table 2**　evaluation results

| | 評価 1 | 評価 2 | 評価 3 |
|---|---|---|---|
| 提案手法 | ○ | ○ | × |
| 既存手法 | ○ | ○ | × |

NS テーブルはネットワーク上で観測された NS の情報を保持している．$NS_{TIP} = NA_{TIP}$ は $NA_{TIP}$ がノードの NS に対する応答であるかどうかを判断する条件で，条件に当てはまる場合は、NS 処理アルゴリズムと同様の処理が行われる．条件に当てはまらない（NS に対する応答ではない）場合，それは UNA であると判断されて，別のアルゴリズム（UNA ハンドラ）によって処理される．

5 章で提案手法に対して実施した評価を，既存手法にも行う．結果は**表 2** の通り．

どちらも重複アドレス検出に対する DoS 攻撃には対応していない．ここで，攻撃ではなく本当に IPv6 アドレスの重複が検出された場合について考える．RFC4862[12] では，リンク層アドレスを用いて生成した IPv6 アドレスの重複を検出したシステムは普通，エラーを出力して管理者による復旧を待つことになっている．つまり既存手法では，少なくとも RFC4862 に則るならば，IPv6 アドレスの重複が検出された際は手動による復旧を行う必要がある．一方で提案手法は，IPv6 アドレスの重複が検出された際はホスト B が DHCPv6 サーバに DECLINE メッセージが送信され，再度 IPv6 アドレスの割り当てが行われる．よって，提案手法は既存手法と比べ，ネットワーク内で IPv6 アドレスが実際に重複した際の動作で優れていると言える．

### 6.2　DHCPv6 snooping 機能との比較

多くの L2 スイッチには DHCPv6 snooping と呼ばれる機能が実装されている．これはスイッチ上で DHCPv6 パケットを観測し，その IPv6 アドレスや MAC アドレスをポート番号と関連づけることで，DHCPv6 で設定された IPv6 アドレスではないアドレスを検出する仕組みである．

ポートベースでパケットを強力にフィルタリングすることから，提案手法と同等の前提条件において，5 章で行った評価を全てクリアできる．しかし，DHCPv6 snooping はスイッチに実装される機能であるため，ホストに導入される想定の提案手法とは導入箇所が異なる点で差別化される．

### 6.3　DHCPv6 が利用できない端末への対応

Android をはじめとするいくつかの端末では，DHCPv6 が利用できないことが考えられる．このような端末は DHCPv6 環境下において，DHCPv6 サーバによる IPv6 アドレスの配布を受けないまま，IPv6 アドレスを自動設定し通信を開始してしまう可能性がある．

この場合提案手法では，端末の IPv6 アドレスが DHCPv6 サーバから配布されたアドレスではないことから，攻撃と誤検知してしまう．一方で既存手法であれば，DHCPv6 サーバの有無に関わらず独自テーブルで IP-MAC アドレスペアを管理していることから，DHCPv6 が利用できない端末の存在にも対応できる．

### 6.4　リンクローカルアドレスに関する問題

IPv6 ネットワークでは，同一リンクのノードと通信する際はリンクローカルアドレスと呼ばれる IPv6 アドレスを，別のリンクのノードと通信する際はグローバルアドレスと呼ばれる IPv6 アドレスを利用する．本研究で想定するステートフルアドレス自動生成の環境下では，ノードがグローバルアドレスを DHCPv6 サーバに割り当ててもらうために，リンクローカルアドレスを自動生成しなければならない．

ここで，提案手法ではネットワーク内で観測した IPv6 アドレスについて DHCPv6 サーバを問い合わせているが，ネットワーク内のノード同士で通信するパケットでみられる IPv6 アドレスはリンクローカルアドレスであり，これは DHCPv6 サーバでリース情報として保持されているグローバルアドレスとは異なる．そこで提案手法は，ノードの DHCPv6 サーバとのやり取りをスヌーピングして，リンクローカルアドレスとグローバルアドレスの対応を記録する必要がある．提案手法の問い合わせではこの記録を基にグローバルアドレスを問い合わせているものとする．

加えて，自動生成されたリンクローカルアドレスとグローバルアドレスが対応する前の通信，すなわち自動生成した直後のリンクローカルアドレスについての重複アドレス検出，およびそれに続く DHCPv6 サーバへのリース要求のパケットについては，本提案手法のアルゴリズムでは攻撃と誤検知してしまうため，例外として処理しなくてはならない．

## 7.　まとめ

本稿では，近隣キャッシュが攻撃者によって汚染される問題に注目し，DHCPv6 サーバの情報を利用してネットワーク内の疑わしい通信を検知する手法を提案した．提案手法はテーブルで保持する IP-MAC アドレスペアの情報を利用しながら，場合によっては DHCPv6 サーバに問い合わせを行うことで，正当な MAC アドレスの変更を誤検知することなく攻撃を検知する．

今後の課題として，今回の提案手法はルータに関係する攻撃を考慮していないことから，それらに手法を対応させることなどが考えられる．

## 参考文献

[1] Arkko, J., Kempf, J., Zill, B. and Nikander, P.: Secure neighbor discovery (SEND)(RFC 3971), *Internet Engineering Task Force* (2005).
[2] Gont, F.: Rfc 6980-security implications of ipv6 fragmentation with ipv6 neighbor discovery, *Interet Engineering Task Force, Request for Comments* (2013).
[3] Ahmed, N., Sadiq, A., Farooq, A. and Akram, R.: Securing the Neighbour Discovery Protocol in IPv6 Stateful Address Auto-Configuration, *2017 IEEE Trustcom/BigDataSE/ICESS*, IEEE, pp. 96–103 (2017).
[4] Narten, T., Nordmark, E., Simpson, W. et al.: RFC 4861, *Neighbor discovery for IP version*, Vol. 6 (2007).
[5] Plummer, D. C.: RFC 826: An ethernet address resolution protocol, *InterNet Network Working Group* (1982).
[6] Nikander, P., Kempf, J., Nordmark, E. et al.: IPv6 neighbor discovery (ND) trust models and threats, *RFC 3756*, IETF (2004).
[7] Barbhuiya, F. A., Biswas, S. and Nandi, S.: Detection of neighbor solicitation and advertisement spoofing in IPv6 neighbor discovery protocol, *Proceedings of the 4th International Conference on Security of Information and Networks*, ACM, pp. 111–118 (2011).
[8] Kumar, N., Bansal, G., Biswas, S. and Nandi, S.: Host based IDS for NDP related attacks: NS and NA Spoofing, *2013 Annual IEEE India Conference (INDICON)*, IEEE, pp. 1–6 (2013).
[9] Brzozowski, J., Kinnear, K., Volz, B. and Zeng, S.: DHCPv6 Leasequery, RFC 5007, RFC Editor (2007).
[10] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C. and Carney, M.: RFC3315: Dynamic host configuration protocol for IPv6 (DHCPv6), *Standards Track, http://www. ietf. org/rfc/rfc3315. txt* (2003).
[11] Beck, F., Cholez, T., Festor, O. and Chrisment, I.: Monitoring the neighbor discovery protocol, *2007 International Multi-Conference on Computing in the Global Information Technology (ICCGI'07)*, IEEE, pp. 57–57 (2007).
[12] Thomson, S.: RFC 4862: IPv6 stateless address autoconfiguration, *http://www. ietf. org/rfc/rfc4862. txt* (2007).