卒業論文　　2010 年度 (平成 22 年度)

CoPS :
無線アドホックネットワークにおける ノード協調型攻撃防御機構

指導教員
慶應義塾大学 環境情報学部
徳田 英幸
村井 純
楠本 博之
中村 修
高汐 一紀
重近 範行
Rodney Van Meter
植原 啓介
三次 仁
中澤 仁
武田 圭史

慶應義塾大学 総合政策学部

星 北斗

kanny@ht.sfc.keio.ac.jp

学士論文要旨　　2010 年度 (平成 22 年度)

CoPS: 無線アドホックネットワークにおける ノード協調型攻撃防御機構

論文要旨

近年ノードのみで動的にネットワークを構築でき，インフラの敷設が不要，または最小限に抑えることのできる無線アドホックネットワークが注目を集めている．データ共有やゲームの通信プレイといった用途から，既存インフラが利用不能に陥った際の通信手段など，利用用途，適用範囲は多岐にわたり，将来のユビキタスコンピューティング環境を支える基盤技術として期待されている．

ネットワーク通信における攻撃者による悪意ある攻撃の検知や防御については，情報セキュリティの分野において常に研究が続けられてきた．大量のデータを送信し通信を妨害する DoS (Denial of Service) 攻撃や，ソフトウェアが持つ脆弱性を利用した攻撃など，様々な攻撃手段が存在し，このような攻撃を防ぐために IDS (Intrusion Detection System) をはじめとした様々なセキュリティ機構が研究，開発されている．しかし，無線アドホックネットワークは，管理者を必要としない点やネットワーク構成が動的に変化する点など，既存の通信ネットワークにはない特徴を持つ．加えて，無線アドホックネットワークを構成する機器は小型省電力化に伴いリソースが制限されている．また，パケットドロップ攻撃など，無線アドホックネットワーク特有の攻撃手段も存在する．これらの要因により，既存のネットワーク環境を前提としたこれまでのセキュリティ機構を無線アドホックネットワークにそのまま適用することは困難である．無線アドホックネットワークは主に小型ノードによって構成されており，特に DoS 攻撃のような，リソースを使い果たさせる攻撃には脆弱である．攻撃を放置しておけば，ネットワーク全体に致命的なダメージを与える可能性があり，そのようなノードをネットワークから排除する手法を確立することは今後の無線アドホックネットワークにとって必要不可欠である．

本論文では，攻撃を行っている不正なノードを複数ノードの協調によってネットワーク上から切り離すための機構として，CoPS (Cooperative Attacker Prevention System) を提案する．本機構は，攻撃を行うノードを排除する意思決定を他のノードによる多数決によって行う．また，攻撃を受けて動作不能に陥っているノードが存在する場合，通信可能な他のノードを選出し攻撃を受けているノードに変わって攻撃を行うノードの排除を行う．これにより，動的かつ自律的に構成されたアドホックネットワーク上において攻撃を行う不正なノードを排除することを可能とする．また考案したアルゴリズム，データ送信手法に基づいて設計・実装を行い，最後に評価を行う．評価は，ノード協調アルゴリズムの動作速度，ノード排除による負荷測定，加害ノードへのデータ送信手法の違いによるデータ送信速度差について実験を行った上で定量的評価を行う．本研究は，既存のシステムと比べ，高機能な，あるいは特定の攻撃防御のためのノードを必要としないため，ネットワークを構成するノードにかかる負荷をより公平にすることが可能であり，ノードの関係が対等なネットワークにおいても適用可能な点で有益である．

キーワード：

1 情報セキュリティ　2 モバイルアドホックネットワーク　3 攻撃防御　4 協調制御
5 IEEE 802.11

慶應義塾大学　総合政策学部総合政策学科
星 北斗

Abstract of Bachelor's Thesis

Academic Year 2010

CoPS: A Cooperative Attacker Prevention System for Mobile Ad Hoc Networks

Summary

Several critical attacks, such as DoS and exploiting software vulnerability prevents large-scale deployment of ad hoc networking, because it is critical not only in emergent situation but also in everyday life.

Most technologies to protect from these attacks in infrastructure networks cannot solve this problem due to network and end system availability in ad hoc networks. First, ad hoc networks do not have any specific administrator. Second, logical and physical network topology frequently changes. Third, end systems in ad hoc networks generally have less resources than those in infrastructure networks. The limited resources in end systems make the tolerance against some attacks, such as DoS weaker. Fourth, some attacks are specifically for ad hoc networks. From these reasons, we need a new scheme to separate malicious attackers explicitly from the ad hoc network, rather than to protect end systems by themselves.

In this paper, we propose a Cooperative Attacker Prevention System (CoPS). CoPS can be safely deployed, because CoPS adopts majority rule to decides the node to be separated in order for reliable decision. CoPS ensures robustness for the ad hoc network, because any available node in the network can separate the malicious attacker.

We implement CoPS in Linux, and evaluate it in terms of quickness to separate the malicious node, accuracy of the node-cooperation algorithm and overall performance.

Keyword :

1 Information Security　2 Mobile Ad Hoc Networks　3 Intrusion Prevention System
4 Cooperative Control　5 IEEE 802.11

Hokuto Hoshi
Faculty of Policy Management
Keio University

目 次

図 目 次

表 目 次

第1章　序論

本章では，本論文の研究背景，研究動機について述べ，本研究の目的を提示する．また，論文全体の構成についてまとめる．

1.1 研究背景

本節では，本研究の背景である無線アドホックネットワークの普及，無線アドホックネットワークを狙う攻撃の存在について述べる．また，本論文で述べる無線アドホックネットワークを定義する．

1.1.1 無線アドホックネットワークの普及，攻撃の存在

近年，無線通信技術の発達が急速に進んでいる．2000 年頃より製品化され, 爆発的に普及した無線 LAN は IEEE 802.11n が策定されたことで, その速度は 最大 600Mbps に達した. これは, これまで一般的に利用されてきた有線 LAN 規格において最大 100Mbps の通信速度を持つ, 100BASE-TX を凌ぐ回線速度である. 携帯電話の世界においても, 高速データ転送規格である IMT-2000 は第三世代移動通信規格として定着し, 携帯電話自身が Bluetooth や IEEE 802.11 など別の無線通信規格に対応するなど, 次世代のネットワーク構築に向けた進化を続けている. 無線を利用したセンサや家電なども様々な製品が日々開発されており, もはや日常生活におけるあらゆるものに無線通信技術が搭載され, ネットワークとして結合するような世界は目前に迫っている.

多様な形態をとる次世代ネットワークを構成する技術のなかで，近年無線アドホックネットワークと呼ばれるネットワーク構成が注目を集めている．無線アドホックネットワークとは，無線機器を機器同士で接続することで構成するネットワークである．携帯電話のような移動端末を用いて構成でき，ネットワーク上の通信をノードが中継することで既設インフラによらない通信を可能とする．ネットワークの構成は端末が自律的に行うことが可能であり，特定の管理者などを必要としない．無線アドホックネットワークのトポロジ例を図 1.1 に示す.

無線アドホックネットワークは，通信機器があらゆる場所に遍在するユビキタス・コンピューティング環境を実現していく上で非常に重要な通信技術であり，その特徴を活かした様々なサービスが既に提案されている．しかし，無線アドホックネットワークもコンピュータネットワークの一種であり，それらのサービスを狙った攻撃も当然懸念されるべきであろう．以下に，無線アドホックネットワークを利用したサービス，それらに考えうる攻撃を例示する.

- 携帯ゲーム機の通信対戦システム
  日常生活における無線アドホックネットワークの実用，製品例として，Nintendo DS[1] や PlayStation Portable[2] などの携帯ゲーム機が挙げられる．従来のゲーム機は通信用のケーブルなど，複数のゲーム機を物理的に接続した上で通信を行っていたが，無線通信，並びに無線アドホックネットワークの利用により，ケーブルなどを介さずに通信を実現している．これによって，より大人数での通信プレイやインターネットを介した通信プレイが可能となり，次世代携帯ゲーム機として急速に普及している．しかし，無線通信が可能となり，全く面識の無い第三者と同じネットワークを構成し通信を行う機会も存在するようになった．また，エレクトロニック・スポーツ (e-Sports)[3] と呼ばれるように，ゲームを競技として捉え，プロリーグなどを開催する動きも存在する．こうした状況下において，悪意ある人間が不正に改造されたプログラムなどを用いて，ネットワーク全体，あるいは特定の利用者のゲーム機の通信を妨害する，あるいはソフトウェアのバグを突き他者のデータを破壊する攻撃などを行うことが想定される.

- 山林や農場モニタリングのためのセンサネットワーク
  山林や農場において，位置によってばらつきのある気温や湿度，照度などを観測し活かすことによって，火災の監視や環境の変化による動植物への影響調査，ロボットなどの自動制御による農作物の効率的な栽培などが可能となる．しかしこうした環境においては，景観上や対象地域の広さなどから通信インフラを敷設することが難しい場合が多い．ここで無線アドホックネット

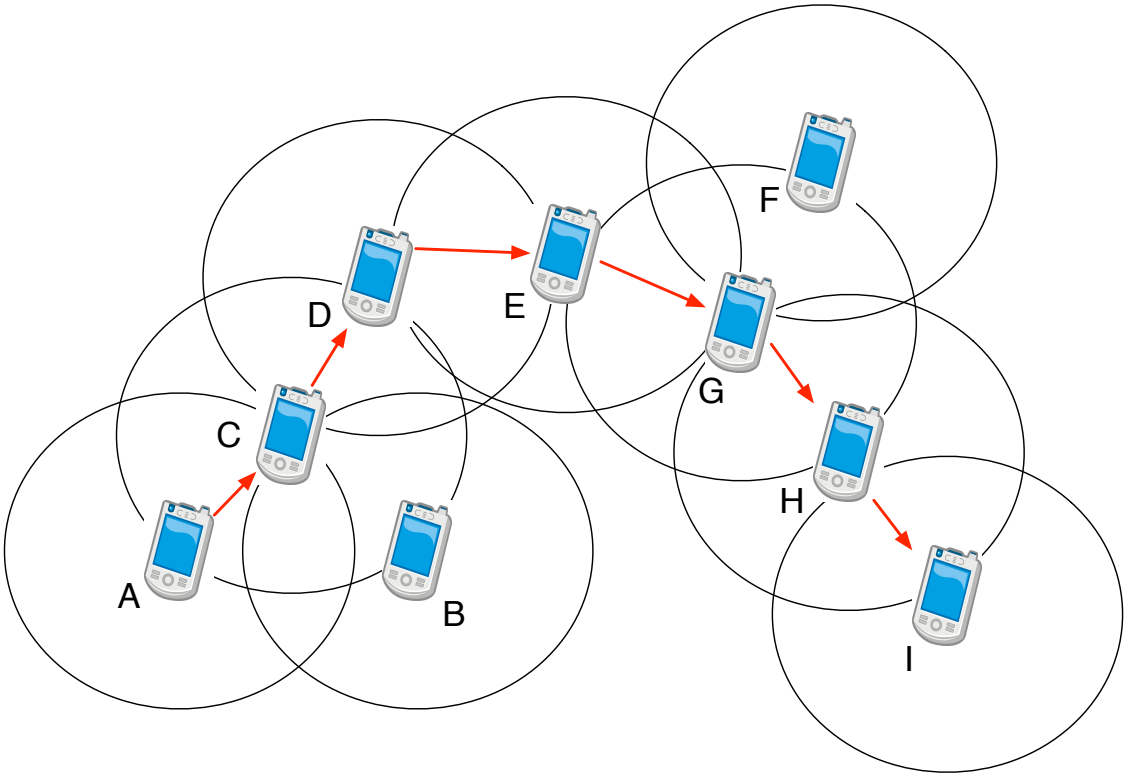

図 1.1: 無線アドホックネットワークのトポロジ例．円は中心のノードの通信可能範囲である．ノード A がノード I と通信する際，通信は C → D → E → G → H の順に中継されて I まで届く．

ワークを利用することによって，センサノードを互いが無線通信可能な範囲に設置するだけで，自律的にネットワークを構成しインフラの存在する場所までネットワークを広げることが可能となる．既にこのような用途のためのセンサ製品も実用化されており，その例として eKo mote[4] などが挙げられる．このような環境において，モニタリングの妨害のため，収集されているデータの盗聴や破壊，改ざんを目的に攻撃を行うことが考えられる．不正なプログラムがインストールされたノードや乗っ取ったノードをネットワークに参加させデータやネットワークを盗聴・破壊する，センサノードそのものを物理的に破壊・妨害する，などが考えられる攻撃手段として挙げられる．

- 救助活動における通信システム
  救助活動においては，怪我人の状況や周囲の環境変化など，変化する情報を総合的に把握しておくことが重要である．しかし，地震や火災などの災害時には，既存の通信インフラは破壊され，利用不可能である場合が多い．こうした状況においても，無線アドホックネットワークを用いることにより，救助隊の心拍数をはじめとしたバイタルデータの共有，周囲の温度など環境をセンシングしたデータの共有，といったシステムを構築することが可能である．このアプリケーションが送受信する情報は救助隊が救助活動を行う上で非常に重要なものであり，データの損失や不達は許されない．しかしこの状況下において，例えば災害がテロリストによって引き起こされたものであるなど，悪意のある者が関与している場合，救助隊員の持つ端末やネットワークに向けて異常な，あるいは大量のデータを送信することで救助活動を妨害するといった攻撃が考え

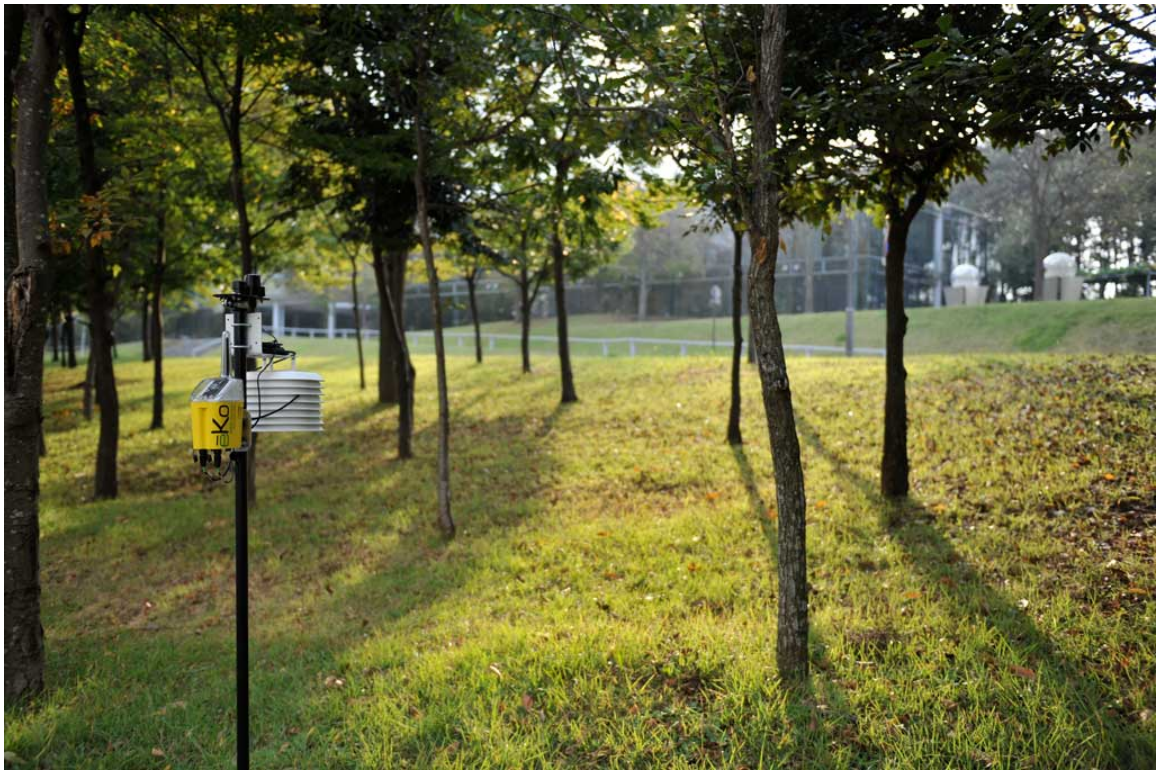

図 1.2: eKo mote[4] を利用した環境モニタリングにおける無線センサノードの実用例

られる．システムと攻撃の想定図を図 1.3 に示す．

こうしたアプリケーションの実現によって，将来の我々の生活はより便利，安全，安心に暮らしていけるものになるが，コンピュータネットワークを利用したシステムは常に攻撃を受ける可能性があると考えるべきである．特に，先述の救助活動における通信システムなど，人の生命に関わるような，あるいは生活基盤に直結する情報を扱うアプリケーションの場合，可用性，信頼性をはじめとしたセキュリティを確保することは必要不可欠である．攻撃に耐え，確実に通信が行えることを保証できない場合，無線アドホックネットワークをミッションクリティカルな用途に用いるのは危険である．

1.1.2 無線アドホックネットワークの定義

本論文で述べる無線アドホックネットワークは，移動端末をはじめとした無線機器のみによって構成されたネットワークである．ネットワークは IEEE 802.11 を利用できる無線機器を用いて構成され，ネットワーク上通信をノードが中継することによってノード間の通信経路が形成される．

1.2 研究動機

1.1 節で述べたように，無線アドホックネットワークを利用したサービスは様々なものが考案，実用化されている．無線アドホックネットワークによって人々とコンピュータネットワークの距離はより近づくと考えられる．農作物生産の際の補助や，人命救助にも応用でき，電気や水道などの生活インフラと同じく，人々にとって無線アドホックネットワーク，コンピュータネットワークは当たり前のよ

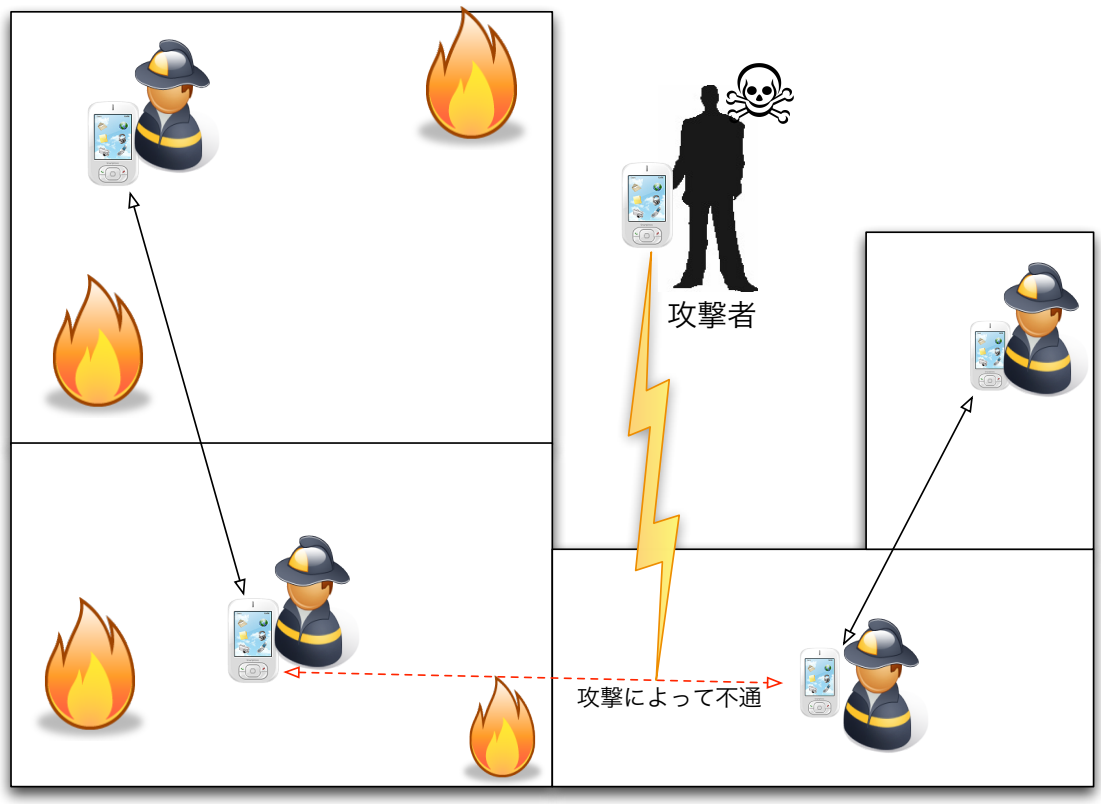

図 1.3: 救助活動における通信システムと想定される攻撃

うに使えなくては困るものとしての地位を得るだろう．そのため，無線アドホックネットワークの実用化に向け，セキュリティ分野における課題の解決は避けては通れない．人々のインフラとして機能するためには，無線アドホックネットワークは信頼できる強固で安全な環境でなければならない．

しかし無線アドホックネットワークにおいては，物理的に無線を使用不能にする電波妨害や，ノードによるルーティングを阻害する攻撃，ソフトウェアの脆弱性への攻撃，乗っ取りなど，インターネットをはじめとした既存のネットワーク環境においても行われてきたものや無線アドホックネットワークに特化したものまで，様々な攻撃が考えられ，それらは解決すべきセキュリティ分野の課題として提示され，いまだ解決のための方策を見出せないものも存在する．

無線アドホックネットワークに対する攻撃として想定され，対抗するための研究がされているものの一つに，大量のデータをネットワーク全体に流すことで，ネットワークを麻痺させるものや，1 つのノードに通信を集中させ，対象ノードのリソースを逼迫させることで機能の停止を狙う，あるいは負荷上昇によりノードの寿命を極端に縮めるといった攻撃などが挙げられる．これらの攻撃は， DoS (Denial of Service) 攻撃と呼ばれ，ネットワークやネットワークを構成するノードの機能停止など重大な問題を引き起こす．従来のネットワークにおいては， DoS はネットワークに顕著な影響を及ぼすが，ユーザノードに影響が及ぶ前にネットワーク管理者が上流で通信を遮断するなどの措置で対抗できる場合が多かった．しかし，無線アドホックネットワークは管理者が存在しない場合が多く，また管理者が存在しなくてもネットワークを構築できることを利点の一つとする技術である．管理者が存在しないネットワークにおいて意思決定ができるのは，ユーザかノード自身である．しかし，無線端末は決してコンピュータリテラシが高い人々のためのものではなく，日常に溶けこみ利用されるもの

であり，従来専門知識を持っていた者が行っていた攻撃の判断や対応をユーザに行わせることは難しい．ノードが自律的に攻撃者の判断，対応を行うことができれば，ユーザの負担は劇的に減少する．

また，2.4 節でも述べるが，無線アドホックネットワークにおけるセキュリティ分野の研究として，攻撃を検知するための研究は多数存在するが，検知した攻撃をどのように防御するか，という研究は少ない．攻撃を検知することができても，その後適切な対応をとれなければ検知に意味はない．攻撃者にとっては対象とするネットワーク，ノードと接続できる時間が長ければ長いほどアドバンテージとなるため，攻撃を行うノードはネットワークから排除し，通信を不可能にさせることが根本的対策として有効である．

これらのことから，無線アドホックネットワークにおいて，こうした攻撃を自律的に素早く検知，排除できる手法を確立することは非常に重要であると考える．

1.3 研究目的

本研究では，無線機器によって動的かつ自律的に構成される無線アドホックネットワークにおいて，複数のノードが協調動作することにより，特定の高機能ノードや管理用ノード，ユーザによる操作を必要とすることなく，ネットワークに対して攻撃を行うノードを自律的に排除する攻撃防御を行うことを目的とする．

1.4 本論文の構成

本論文は，第 2 章において本研究の対象領域と，既に提案されている攻撃防御手法，ならびに無線アドホックネットワーク固有の問題点について述べる．第 3 章では，本研究の機能要件，並びにノード協調アルゴリズム，データ送信手法について述べ，第 4 章では CoPS の設計について，第 5 章では CoPS の実装について述べる，第 6 章では実装したソフトウェアを用いて， CoPS の定量的評価を行う．最後に，本論文のまとめと，本研究の課題と展望について述べ，まとめる．

第2章　無線アドホックネットワークにおける攻撃防御手法

本章では，本研究において問題としている無線アドホックネットワークにおける攻撃防御について述べる．

2.1 無線アドホックネットワークにおいて想定される攻撃手法

本節では，移動端末によって構成された無線アドホックネットワークにおいて想定される主な攻撃手法についてまとめる．

2.1.1 物理的な攻撃

物理的な攻撃としては，IEEE 802.11 が利用する電波周波数帯における電波妨害が挙げられる．ケーブルによる有線ネットワークと比較すると，ケーブルの物理的切断などは不可能であるため，攻撃に至るまでの労力は高いが，実行された場合，ソフトウェア的に対処することは非常に難しい．

また，物理的攻撃としてもう一つ挙げられるのは，ノードに対しての直接的な破壊行為である．人々が所持して利用する端末は意図的な破壊の危険には晒されにくいが，センサノードをはじめとしたある場所に固定して利用するような端末は，人為的に，あるいは事故や環境の変化によって破壊される可能性がある．電波妨害と違い，受けた損傷が攻撃後も残るため，ノードの動作が不可能な状態にされるとソフトウェア的な対処は難しい．

2.1.2 データリンク層プロトコルにおける攻撃

データリンク層における攻撃には，有線ネットワークと同様の攻撃として， ARP スプーフィングによる経路に対する攻撃が挙げられる．また，IEEE 802.11 を利用するネットワーク固有の攻撃として， IEEE 802.11 MAC における管理フレームのフラッディング攻撃なども挙げられる．IEEE 802.11 MAC における管理フレームは，近年利用されている WPA (Wi-Fi Protected Access) などをはじめとしたセキュリティ機構によっても保護されておらず，無線ネットワークを構築する上で注意すべき攻撃となっている．膨大な ARP リクエストを発行し，その応答を盗聴する ARP インジェクション攻撃は， IEEE 802.11 において利用されている認証，データ暗号化機構である WEP への攻撃を簡単に行うための攻撃として知られている [7].

2.1.3 ルーティング，データ中継における攻撃

無線アドホックネットワークにおけるセキュリティの議論において，ルーティングプロトコルに対する攻撃は多く扱われてきた．具体的な攻撃として，攻撃者が隣接するノードからの経路を引き寄せ，データを破棄したり改ざんするブラックホール攻撃 [5] や，ブロードキャストパケットを大量に生成して中継させることでネットワークの帯域幅を溢れさせるフラッディング攻撃 [5]，自身のノードに関するパケットのみ処理することで自身のバッテリをはじめとしたリソースを節約し，他ノードの処理を増加させるセルフィッシュ・ノード [6] といった攻撃が存在する．無線アドホックネットワークは，経路制御をノード自らが行うことで構成するネットワークであり，経路制御に対する攻撃は経路の混乱やネットワークリソースの枯渇，ノードが持つバッテリなど資源の枯渇などを招きやすくなるため，これらの攻撃への対策は研究分野においても重要視されている．

2.1.4 トランスポート層プロトコルにおける攻撃

トランスポート層における攻撃は，有線ネットワークと無線ネットワークにおける差は少ない．代表的なものとして， TCP における 3-way handshake を悪用した SYN flood 攻撃や，UDP セグメントを大量に送出する UDP Flood 攻撃などをはじめとした DoS 攻撃に分類される攻撃が挙げられる．

これらは，従来のようなツリー型ネットワークにおいては，より上流のネットワーク機器でパケットの転送を止める，といった対策をとることも可能であったが，無線アドホックネットワークではルーティング層における攻撃と同じく，ネットワークを構成するノードへの負担が非常に高まり，重大な影響を引き起こすと考えられる．また，大量にパケットを送出するノードを分散させた DDoS (Distributed Denial of Service) 攻撃などが発生すると，その防御は無線アドホックネットワークだけではなく，ツリー型ネットワークにおいても難しいものとなる．

2.1.5 ソフトウェアの脆弱性に対する攻撃

無線アドホックネットワークに限らず，OS のプロトコルスタックやノード上で動く Web サーバやデータベースに存在するバグなどを起因とする脆弱性を利用した攻撃や，SSH に対する総当り攻撃のように，サービス妨害だけでなく，成功した場合ノードの制御そのものを奪ってしまうような攻撃も存在する．これらの攻撃はソフトウェアに依存するものであり，無線アドホックネットワークにおいては，個々のノードにおいて適切にソフトウェア，サービスの管理を行うことや，ファイアウォールなどを用いて通信そのものを遮断するといった対策を行う必要がある．しかし，無線アドホックネットワークに利用される小型ノードなどは，リソース不足などの理由からファイアウォールなどリソースを必要とするソフトウェアを稼働させられない場合も考えられ，ノート PC など高機能なノードのように潤沢なセキュリティ対策を行うことは難しい．

2.2 既存のセキュリティ機構

LAN やインターネットなど，既存のネットワーク環境におけるセキュリティ機構として，IDS (Intrusion Detection System) や IPS (Intrusion Prevention System) をはじめとした攻撃検知，防御に関する手法や，認証手法が存在する．本節では，前節で述べたような攻撃を防ぐため，これまで利用されてきた既存のセキュリティ機構についてまとめる．

2.2.1 IDS (Intrusion Detection System)

ネットワークやコンピュータに対する攻撃検知を行う IDS は 2 種類に分類される．IDS を利用する場合のネットワークトポロジ例を図 2.1 に示す．

- ネットワーク型 IDS (Network IDS, NIDS)
  ネットワーク型 IDS とは，Snort[8] などに代表される，ネットワーク上の通信が集約される経路上にソフトウェアを稼働させたノードを設置しネットワークパケットを収集，通信データを監視することで，ネットワーク全体の攻撃検知を行うものである．ソフトウェアの脆弱性を突くものなど，不正なパケットをあらかじめシグネチャとして登録しておき，ネットワークを流れるパケットと照合することで攻撃を検知する手法が主に利用され，攻撃を検知した場合は攻撃を行うホストからの通信を遮断するなどの防御を行う．比較的小規模なネットワークにおいては一般的な PC を利用してシステムが構築される場合もあるが，ネットワークを流れる全てのパケットをチェックするため動作負荷が高く，ネットワークの規模が大きい場合には専用のハードウェアなどを用いてパケットの解析を行う場合も多い．

- ホスト型 IDS (Host-based IDS, HIDS)
  ネットワーク型 IDS と違い，tripwire[9] などに代表されるホスト型 IDS は，ノートパソコン

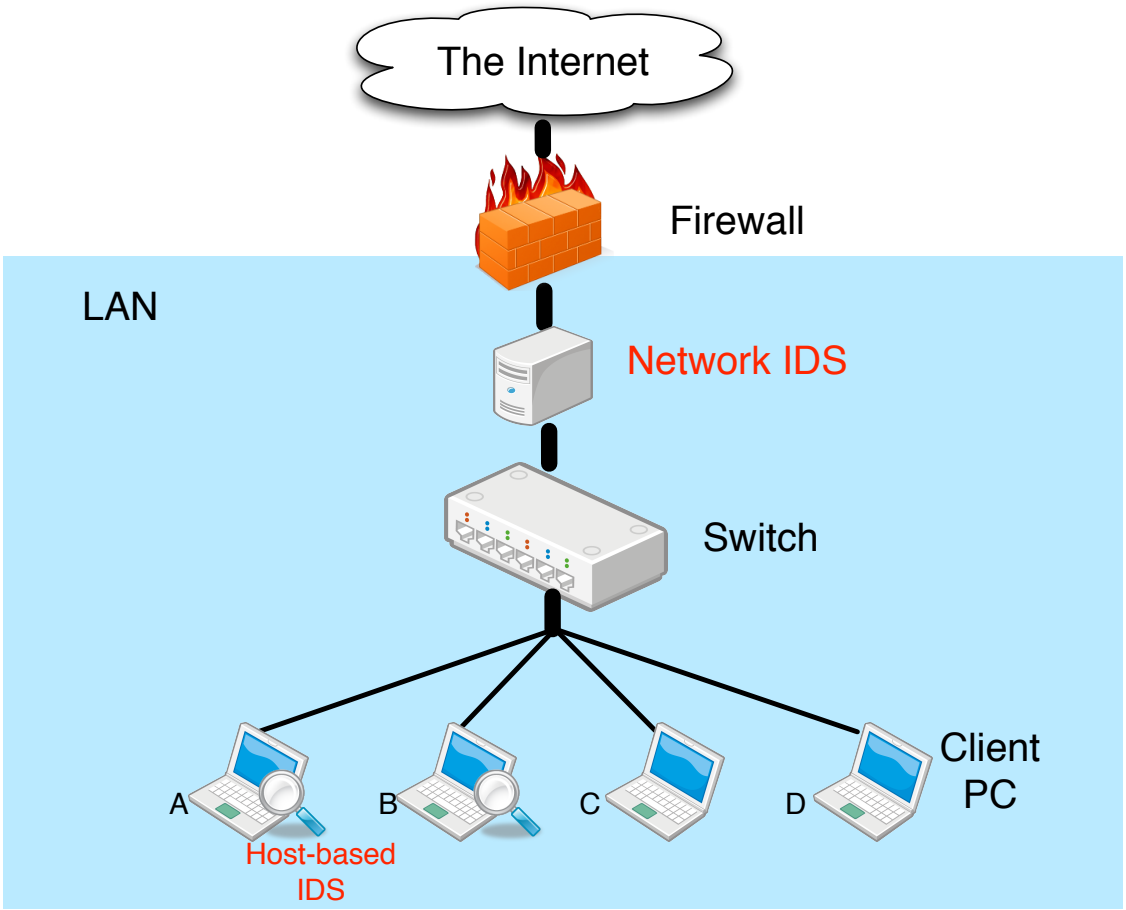

図 2.1: IDS を利用する際のトポロジ例．Client PC において送受信されるデータは全て Network IDS を通り，チェックされる．また，PC A, B には Host-based IDS がインストールされ，ノード上で通信のチェックが行われている．

などの機器上で動作し，動作するコンピュータへの通信，コンピュータ上におけるプロセスなどの振る舞いや，データの改ざんをチェックすることによって IDS がインストールされたホストに対する攻撃検知を行うものである．常にリアルタイム動作して監視を行うものや，データの改ざんチェックのため定期的に起動し動作するものなどが存在する．

2.2.2 ネットワーク認証

コンピュータをネットワークに接続する前に，何らかの認証を行う機構も導入されている．

- IEEE 802.1X[10]
  IEEE 802.1X は， LAN 接続時に利用する認証規格である．ネットワーク上において特定のノードが認証サーバとなり，各機器は LAN 接続後， 802.1X に対応したスイッチを介して認証サーバと認証を完了することで，はじめてネットワークに参加できるようになる．

- WEP (Wired Equivalent Privacy)
  WEP は，IEEE 802.11 の一部として策定されたセキュリティ規格である．2 つのノードが通信

を行う際，予め設定された WEP Key を設定しておくことで認証を行う．また， WEP Key を利用してデータフレームの暗号化を行う．しかし， WEP には様々な脆弱性が存在し [11]，今日では認証機構として推奨されていない．

- IEEE 802.11i[12]
  IEEE 802.11i は，脆弱性の指摘された WEP に代わるセキュリティ規格であり，一般には WPA (Wi-Fi Protected Access) または WPA2 として知られる．WEP に存在した脆弱性を改良した上で，ユーザ認証として先述の IEEE 802.1X を利用することが可能となった．

2.3 無線アドホックネットワーク固有の問題点

既存のネットワーク環境におけるセキュリティ機構として，前節で述べたように様々な攻撃検知，防御に関する手法や，認証手法が存在する．しかし，これら既存の攻撃検知，防御手法や認証手法を無線アドホックネットワークに適用することは難しい．これは，無線アドホックネットワークが特定の管理者を必要とせず自律的，かつ動的にネットワークを構成する技術であること，無線アドホックネットワークを利用する機器の特徴に起因する問題である．

本節では，無線アドホックネットワークにおいて既存の攻撃防御手法を適用するにあたり，発生する問題点をまとめる．

2.3.1 ネットワーク構成の変化

Snort をはじめとしたネットワーク型 IDS や IPS は，ツリー構造のネットワークにおいて，通信が集約する経路上に設置することで機能し，従来利用されてきたネットワークトポロジが変化しないネットワークを対象としている．しかし，無線アドホックネットワークは，自律的かつ動的に構成されるネットワークであり，その構成要素は変化し続ける場合が多い．加えて，無線アドホックネットワークはフラットな構造のネットワークであり，通信が集約される経路は存在しないか，その時の経路によって変動する．したがって，ネットワーク型 IDS や IPS を動作させるノードをあらかじめ用意し，無線アドホックネットワークで利用するためには，動的に構成される経路をある程度限定する必要がある．また，通信の集約点における観測は可能であるが，経路を限定しない場合，あるいは攻撃者が攻撃対象ノードの近くにいる場合など， IDS を動作させたノードを通らない経路で通信が行われた場合，観測は困難である．図 2.2 にネットワーク型 IDS を無線アドホックネットワークにおいて適用した場合の問題点をまとめる．

2.3.2 利用される機器の性能

ネットワーク型 IDS や IPS を無線アドホックネットワーク上で利用することの問題点は先述のとおりであるが， tripwire[9] に代表されるようなホスト型 IDS を利用することは可能である．また，Snort などの IDS を 1 ノードを監視するものとして運用することも可能ではある．これらのソフトウェアはほとんどの場合，デスクトップパソコンやサーバを中心とした高性能なコンピュータを対象としている．

しかし，無線アドホックネットワークは主にノートパソコンや携帯ゲーム機，センサノードをはじめとした小型で移動可能な機器で構成される．これらの機器はバッテリで駆動し，排熱や電力消費を抑えるために機能や性能が制限されている場合が多い．こうした機器にとってホスト型 IDS を動作させることは， CPU をはじめとしたリソースを多く必要とすること，通信を監視する場合には常に動

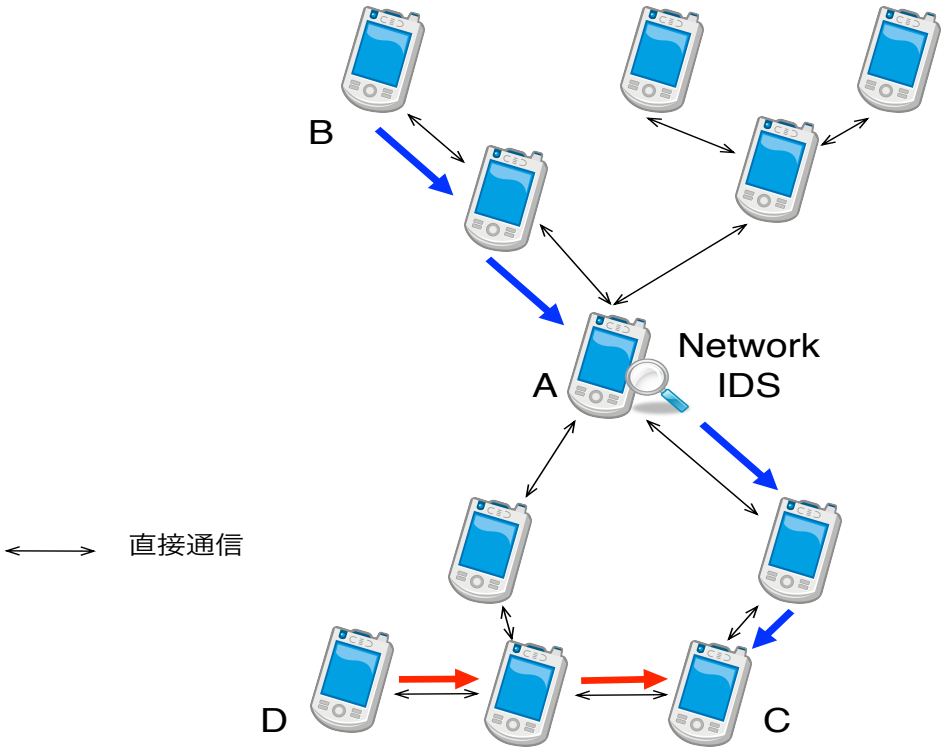

図 2.2: 無線アドホックネットワークにおいてネットワーク型 IDS を利用する際の問題点．ノード B からノード C への攻撃 (青矢印) は IDS が動作しているノード A を通るため観測可能．ノード D からノード C への攻撃 (赤矢印) はノード A を通らないため，観測されず直接攻撃がノード C へ到達する．

作させておく必要があることなどから負担が大きく，バッテリの持続時間や主機能の動作に影響を与える可能性がある．このため無線アドホックネットワークを構成する機器上でプログラムを稼働させ続ける場合，そのプログラムは CPU やメモリなどのリソース要求を少なく抑える必要がある．

2.3.3 ノードの認証

無線アドホックネットワークにおいては，2.2.2 小節で述べた， IEEE 802.1X をはじめとした認証サーバを用いた認証を行うことができない．なぜなら無線アドホックネットワークは場所や構成要素が動的に変化するネットワークであり，認証サーバのような特定のノードの参加を前提とできないためである．WEP や WPA-PSK[13] のように，事前共有鍵を用いた手法であれば認証可能であるが，ノードが自由にネットワークに参加できる無線アドホックネットワークにおいては，事前共有鍵を伝達しなければならないなどの理由から採用は難しい．しかし，悪意のあるノードをネットワークから排除するためには，ネットワーク上においてノードが何らかの形で認証された状態で通信を行う必要がある．無線アドホックネットワークの研究が進んでいる現在，Ariadne[14] などいくつかの認証技術が提案されているが，デファクトスタンダードとして確立された認証手法や実装は存在せず，またノードの参加，離脱が自由な無線アドホックネットワークにおいて認証を行うことは難しい．

2.3.4 異常なノード排除の意思決定

無線アドホックネットワークは，機器が自律的にネットワークを構成するため，既存のインフラやネットワークの管理者を必要とせずにネットワークを利用することのできる技術である．特定の管理者が存在しない場合，異常な動作をするノードをネットワークから排除する，という動作は慎重に行う必要がある．ノードをネットワークから排除する決定を下すのは，あるポリシに従って特権を持つ管理者ではなくネットワークを構成するノードやその利用者である．1 ノードや 1 人の判断のみでは，異常な動作をしていない正常なノードを誤った判断により排除してしまう危険性がある．このため，異常な動作をするノードを排除するための意思決定について，特に無線アドホックネットワークにおいては単一ノードではなく，複数ノードによる信頼性を伴ったプロセスを踏む必要がある．

2.4 既存の手法と対象領域

前節までで述べた問題点を解決するための攻撃検知手法として，既にいくつかの研究成果が発表されている．D. Sterne らの論文 [15] では，クラスタ化され代表者が上位ネットワークとの通信を行うことのできる無線アドホックネットワークにおいて，ノード協調による攻撃検知を実現するネットワークアーキテクチャや，アーキテクチャが攻撃検知に適用される具体的なシナリオについて述べられている．具体的には，各ノードが周辺ノードや自身が中継するデータの流量やパケットヘッダを監視し，それらの情報を上位ノードに集約，処理することで攻撃検知を行うものである．しかしこの論文で提案されている手法は，ネットワーク内に情報を集約することのできる高性能なノードが存在することを前提としており，完全に独立した，フラットなアドホックネットワークにおいて利用することは難しい．

また F. Hugelshofer らの論文 [16] では，センサノードで構成されたネットワークなど，各ノードの性能が低い状況においても動作する軽量なホスト型 IDS である，OpenLIDS の実装と評価について述べられている．OpenLIDS は攻撃検知手法として， Snort などで利用されている，あらかじめ定義されたシグネチャとパケットを照合し検知を行う手法 (シグネチャ型) でなく，コネクションのステータスなどふるまいによって検知を行う手法 (アノマリ型) を利用し軽量化を図っている．

Y. Huang らの論文 [17] では，無線アドホックネットワークにおいて，互いが直接通信可能なノードをクラスタ化し，選出された代表ノードがクラスタ全体の攻撃検知を行う手法について述べられている．この手法は各ノードが IDS を持ち，ノードごとに攻撃検知を行う手法に比べて効率が良いと述べられている．しかしこの論文で提案されている手法は，ネットワーク上のいくつかのノードに負荷が集中するため，携帯ゲーム機のように利用者がノード毎に異なるような状況においては適用することは難しい．

これらの手法は，主に無線アドホックネットワーク上における攻撃検知に主眼がおかれており，検知された攻撃をどのように止めるか，という攻撃防御に関する研究は少ない．本研究は，攻撃が検知された後に無線アドホックネットワークから不正ノードを排除する，攻撃防御手法を対象領域とする．

2.5 本章のまとめ

本章では，本研究において対象とする無線アドホックネットワークにおける主な攻撃についてまとめ，既存の防御手法における問題点，本研究に対象領域について言及した．

第3章　ノード協調動作による攻撃防御機構 CoPS

本章では，2章で述べた本研究の対象領域及び既存手法の比較を踏まえ，機能要件を定める．そして，ノード協調動作による攻撃防御機構 CoPS (Cooperative Attacker Prevention System) についての説明を述べる．

3.1 CoPS の概要

本節では 2 章を踏まえ，機能要件を定めた上で本研究が提案する，ノード協調動作による攻撃防御機構 CoPS の概要について説明する．

3.1.1 想定環境

本研究が想定しているシステムの利用環境は，移動する小型ノードによって自律的に構成される無線アドホックネットワークである．また，以下の条件を仮定する．

1. 攻撃を受けているノードが通信不可能な場合，それを検出する必要があるため，隣接するノードは一定時間ごとに互いが通信可能であることを HELLO メッセージのやり取りによって確認可能である．

2. 隣接ノードを監視し，攻撃を検出した際にネットワーク全体にその通知を行うため．全てのノードはネットワークインタフェースをプロミスキャスモードに設定できるものとする．

3. ノード間に全く信頼関係のない，オープンな無線アドホックネットワークにおいては，特定のノードを排除することが難しいため，ノードがネットワークで通信を行うためには，隣接するノードに 802.11 認証されている必要があり， 802.11 認証を解除された場合にはデータの送受信を行えない．

4. 第 3.2 節において述べる加害ノード排除の意思決定は，複数ノードの投票によって行うため，各端末が送信したパケットは ID で識別可能であり，これを改竄することはできない．

3.1.2 機能要件

本研究においてシステムの利用環境として想定している，移動する小型ノードが自律的に構成する無線アドホックネットワークという環境において，ネットワークに対して攻撃を行うノードを排除するという目的を達成するためには，以下の要件を満たす必要がある．

- 攻撃を行うノードの特定
  本研究において，攻撃を行うノードをネットワーク上から排除するために，まずネットワーク上の情報を収集し，攻撃を行っているノードを特定する必要がある．想定環境において攻撃者を含む全てのノードは移動している可能性があるため，複数のノードで観測を行う必要がある．

- ノードの誤排除防止
  本研究は，特定のノードをネットワーク上から排除，追放するものであり，誤った動作によって正常な動作をするノードがネットワーク上から排除されることがあってはならない．そのため，そのノードが本当に排除する必要のあるノードかどうかを判断するプロセスが必要となる．

- 攻撃を行うノードの排除
  ノード特定後，実際にノードをネットワーク上から排除する必要がある．攻撃を受けているノードが排除動作を行えることは保証されないため，ネットワーク上の他ノードが攻撃を受けているノードに代わりノードの排除を行えることが求められる．

- 軽快な動作
  本研究の想定環境におけるノードは，小型かつバッテリで動作することができるような，移動可能な無線端末である．また，ノードの性能は均一であり，処理を全て任せることのできるような高性能ノードは存在しない．2 章で述べたように，全てのノード上で，処理に負荷がかかる従来の IDS を常に動作させることは性能低下や稼働時間の短縮に繋がるため難しい．このため，ノードが協調することによって負荷のかかる処理を最小限に抑え，CPU やメモリなどのリソース要求を少なく抑えることが要求される．

3.1.3 定義

本論文において，ネットワーク上で動作しているノードのうち， CoPS の動作に関わるノードを次のように定義する．

- 加害ノード
  悪意をもった不正なノードであり，ネットワーク上ノードに大量のデータを送信，ソフトウェアの脆弱性を突くなどの攻撃を行うノード．
- 被害ノード
  加害ノードより攻撃をうけているノード．
- 投票管理ノード
  加害ノードを特定したノードであり，投票動作の管理を行う．
- 補助ノード
  被害ノードに代わって加害ノードの通信を止め，ネットワークから切断するノード．

3.2 ノード協調アルゴリズム

本節では，CoPS のノード協調アルゴリズムについて述べる．

2.3.1 小節で述べたように，従来利用されてきたネットワーク型 IDS や IPS を無線アドホックネットワーク上でそのまま適用することは無線アドホックネットワークのトポロジが動的に変化すること，またフラットなネットワーク構造であることから難しい．また，2.3.2 小節で述べたように，無線アドホックネットワークに利用される機器は機能や性能が制限されている場合が多い．

この問題を解決するため，本論文ではネットワーク上各ノードが相互を監視し，被害ノードではなく，ネットワーク上の他ノードの動作によって攻撃を行う加害ノードの通信を停止させるノード協調アルゴリズムを提案する．ノード協調アルゴリズムの動作イメージを図 3.1 に示す．

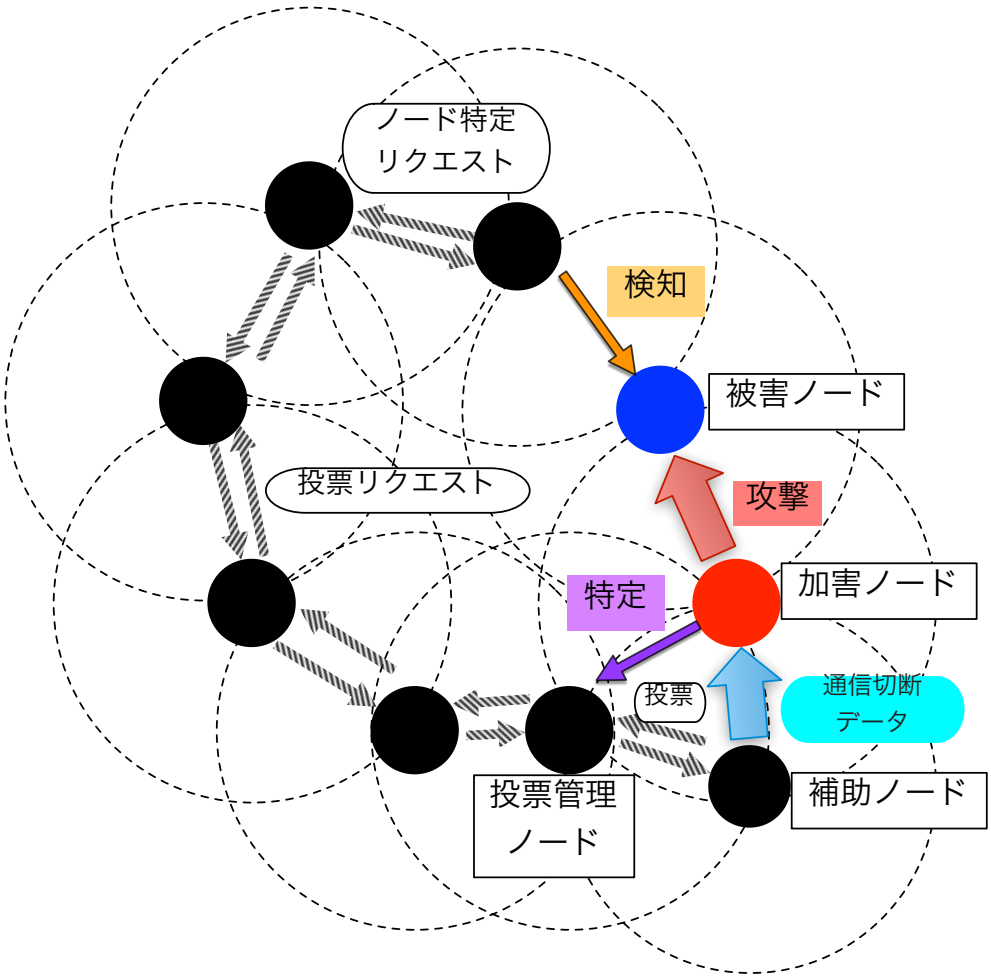

図 3.1: ノード協調アルゴリズム動作イメージ

本論文で提案するノード協調アルゴリズムは以下の 7 段階からなる．

1. 攻撃の検出
   被害ノードが，自身が攻撃を受けていることを検出できた場合，加害ノードの情報を含めて他のノードへ多数決を求めるメッセージを送信する．被害ノードが攻撃を検出できない場合，あるいは他のノードへ通信できない場合，通信を定期的に監視している隣接ノードが被害ノードへの攻撃を検知する．

2. 加害ノードの特定
   被害ノード周辺のノードが，ネットワーク上の他ノードが攻撃を受けていることを検知し，プロミスキャスモードで通信を監視し加害ノードを特定する．特定できない場合，被害ノードの情報を含んだノード特定リクエストをネットワーク内にブロードキャストする．ノード特定リクエストを受信した場合も通信を監視し特定動作を行うが，特定できなかった場合は既にメッセージはブロードキャストされているため，何も行わずに処理を終了する．

3. 加害ノード排除の多数決
   加害ノードを特定したノードは投票管理ノードとなり，加害ノード情報を含め，多数決を求める

メッセージをネットワーク内にブロードキャストする．多数決を求められたノードは，自身が加害ノードの通信を監視できる範囲にいるか確認し，確認できる場合は加害ノードが本当に攻撃を行っているかを確認し，排除の可否について投票を行うメッセージをブロードキャストする．

4. 加害ノード排除の意思決定
   投票管理ノードは一定時間待機した上で受信した投票メッセージを集計し，加害ノードを監視可能なノードのうち排除について賛成を投じたノードが票を投じた 8 割以上であった場合，加害ノード排除のための動作を開始する．また，他のノードも同じ計算を行い，加害ノードを排除する投票の結果を自身で判断しておく．

5. 自身の通信可否判定，被害ノードデータ中継
   本手法では IEEE 802.11 MAC フレームを用いてデータ送信を行うため，補助ノードは加害ノードに直接通信できる距離になければならない．投票が行われている間に加害ノードが移動している可能性があるため，まず投票管理ノードが加害ノードに直接通信可能か確認する．通信不可能な場合は隣接するノードにさらに加害ノードと被害ノードのデータを転送する．ただし，前項で加害ノードを排除する投票の結果，排除しないと判断したノードはノードデータの中継を行わない．これは，投票の結果にかかわらず排除動作が行われることを防ぐためである．

6. 補助ノード決定
   加害ノードと被害ノードのデータを受け取り，自身が加害ノードに直接通信可能であった場合，自身を補助ノードとして設定する．また，補助ノードの決定通知をネットワーク全体にブロードキャストする．

7. 通信切断データ送信
   通信切断のためのデータを IEEE 802.11 MAC フレームを用いて加害ノードに送信する．

上記ノード協調アルゴリズムを，擬似コードを用いて図 3.2 に示す．図 3.2 に示した擬似コードは，単体のノード上で動作するアルゴリズムであり，自身が被害ノードを検出するか，他ノードから被害ノード，加害ノードの情報が転送されることで動作する．加害ノードへの通信可否などを判定し，加害ノードに通信停止のためのデータを送信するか，隣接するノードへ被害ノード，加害ノードの情報を転送することで終了する．

```
victim: 被害ノード,  attacker: 加害ノード,  supporter: 補助ノード
  repeat
    if detect victim then
      set interface to promiscuous mode
      detect attacker
      send vote request message to all nodes on network
      if receive threshold% allow votes of nodes that can connect directly to attacker then
        break_flag ← 1
      end if
    end if
    if receive attacker info then
      break_flag ← 1
    end if
    if receive vote request message then
      break_flag ← 2
    end if
  until break_flag is 1 or 2
  if can connect directly to attacker then
    if break_flag is 1 then
      set me as supporter
      send info of supporter to all nodes on network
      send data for prevention to attacker
    else if break_flag is 2 then
      observe attacker
      if vote request message is true then
        send allow vote to all nodes on network
      else
        send deny vote to all nodes on network
      end if
    end if
  else
    if receive threshold% allow votes of nodes that can connect directly to attacker then
      send info of victim and attacker to neighbors
    end if
  end if
```

図 3.2: ノード協調アルゴリズムの擬似コード

3.3 通信切断データ送信手法

本節では，CoPS の通信切断データ送信手法について述べる．本論文において提案する CoPS は，加害ノードへ通信切断のためのデータを送信する手法として， データリンク層のプロトコルである IEEE 802.11 MAC (Media Access Control)[18] を利用する．一般的に行われる DoS 攻撃などは， TCP や ICMP など，ネットワーク層より上のレイヤにおいて行われる場合が多いが，補助ノードが加害ノードに攻撃を止めるためのデータを TCP などによって送信したとしても，ネットワーク層以下の通信状態の影響を受けて処理が行われない可能性がある．また， Y. Hu らの論文 [14] において，セキュリティを考慮したアドホックネットワーク用ルーティングプロトコルの提案がされているが，IP や MAC フレームなど， ルーティングプロトコルより下位層のプロトコルにおいて攻撃が行われた場合にも対応することは難しい．このため，より下位層において攻撃への対策を行うことが有効である．したがって CoPS では， IEEE 802.11 MAC の一つである， DeAuthentication フレーム (DA フレーム) を利用して加害ノードの通信を停止させる．DA フレームのフレームフォーマットを図 3.3 に示す．

フレーム制御コード	デュレーション	宛先 MAC アドレス	送信元 MAC アドレス	BSSID	シーケンス制御コード	理由コード	FCS (Frame Check Sequence)

図 3.3: IEEE 802.11 MAC[18] における DeAuthentication フレームフォーマット

無線アクセスポイントやクライアントなど， IEEE 802.11 を利用する無線デバイスは通信開始時に認証を行う．インフラストラクチャモード (無線クライアントがアクセスポイントへ通信を行うモード) においては認証確立後にアクセスポイントがクライアントへ ID を付与するアソシエーションを行ってから通信を開始するが，無線アドホックネットワークに利用されるアドホックモードにおいてはアソシエーションは存在せず，認証が確立した段階で通信を開始する．

通信を終了する際，またはデバイス同士の距離が離れ，通信不可能となる際に DA フレームが送信され，認証を解除する．DA フレームは，インフラストラクチャモードで構成される管理されたネットワークにおいて，フレームの送信元アドレスを偽装し，不正に設置されたアクセスポイントの MAC アドレスからのフレームとしてクライアントに送ることでアクセスポイントとクライアント間の認証を解除し，通信を停止させる手法 [19] として利用されている．さらに，ネットワークへの攻撃手法としても利用されている．J. Bellardo らの論文 [20] では，インフラストラクチャモードにおいてクライアントの MAC アドレスを偽装し， DA フレームをアクセスポイントに大量に送信することで通信を妨害する攻撃について述べられている．CoPS は，不正ノードの通信を切断する手法としてこれらの手法と同様に， DA フレームを補助ノードが加害ノードに対し被害ノードからのフレームとして送信することで切断を行う．これにより，被害ノードのリソースの状態によらず，両ノード間における通信切断を実現する．

ただし，アドホックモードにおける認証は IEEE 802.11 MAC の規格上ではオプションとなっており，認証を行わずに通信を開始することも可能とされている．現在利用されている無線 LAN カードドライバにおいては，アドホックモード時には認証を行わず通信を開始する実装が多く，DA フレームを利用しても無視される可能性がある．このため，3.1.1 小節で述べたように，ネットワーク上ノードは通信開始前に必ず 802.11 認証を行うという条件のもと， CoPS を提案する．

3.4　本章のまとめ

本章では，本研究の機能要件を定めた上で，CoPS におけるノード協調アルゴリズム，加害ノードへのデータ送信手法について述べた．

第4章　CoPS の設計

前章では，本研究において提案するノード協調動作による攻撃防御機構 CoPS について説明を行った．本章では，CoPS の設計について述べる．

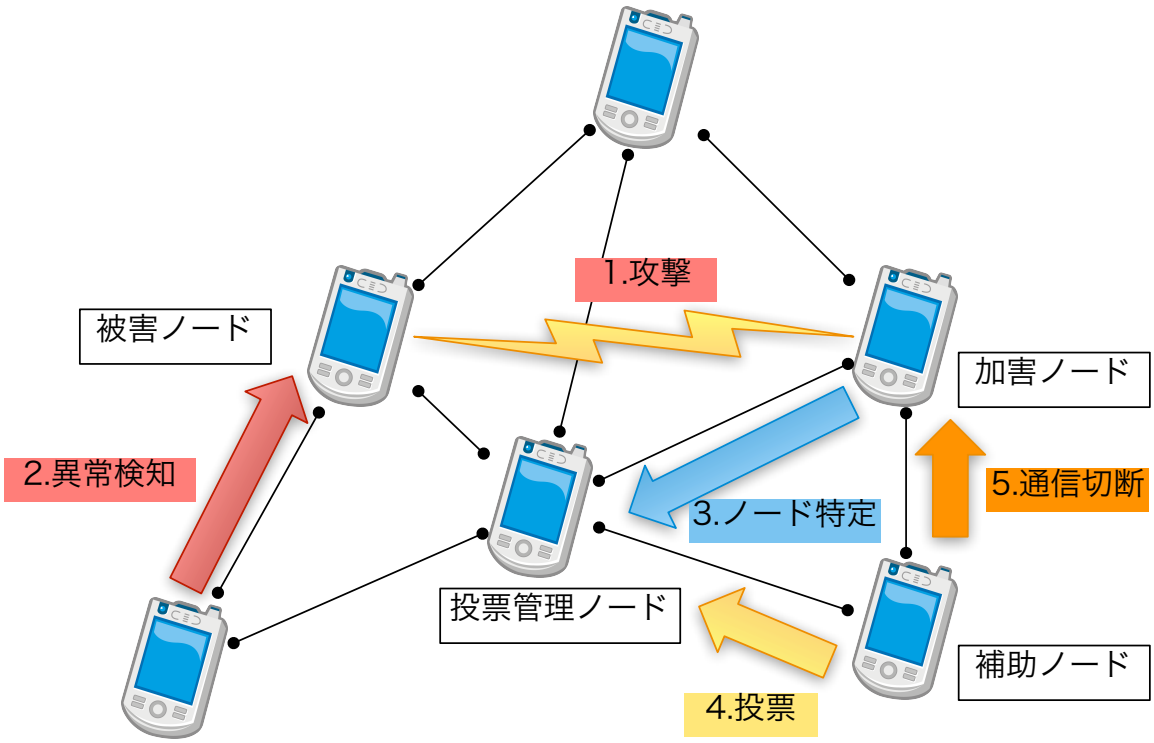

図 4.1: CoPS 動作イメージ図

4.1 設計概要

本研究では，前章で説明した，ノード協調動作による攻撃防御機構 CoPS の設計を行う．CoPS は 3.1.1 小節で説明したように，小型の無線端末を用いた無線アドホックネットワークを想定環境としている．CoPS は，ネットワーク上で不正なデータを送信するノードを特定し，ネットワーク上から排除するまでを特定の管理者を必要とせず，ノードが協調して自律的に行う機構である．CoPS は，隣接するノードがそれぞれ互いの状況を監視していることを前提とし，異常を検知するとネットワークをモニタリングして攻撃を行う加害ノードを特定する．加害ノードが本当に攻撃を行っており，排除すべきかどうかを複数ノードによる投票によって判断し，被害を受けているノードが，あるいは加害ノードへ直接通信することのできるノードが補助ノードとなって加害ノードをネットワークから排除する．CoPS の動作イメージを図 4.1 に示す．

4.2 想定する攻撃

3.2 小節で述べた， CoPS のノード協調アルゴリズムは攻撃の種類によらずネットワーク上から異常なノードを排除できると考えられる．しかし，ノード排除動作を開始するためには攻撃を検知，識別できる必要がある．本研究は攻撃の検知ではなく，検知された加害ノードのネットワークからの排除を目的としているため，本論文では攻撃を判別するアルゴリズムについては設計しない．

4.3 モジュール設計

本節では，CoPS を構成するモジュールの設計について述べる．CoPS は，加害ノード判定モジュール，ノード排除投票モジュール，補助ノード決定モジュール，通信停止モジュールの 4 つのモジュールで構成される．また，CoPS を利用するノードはこれら全てのモジュールを組み込まれた状態で利用される．加害ノード判定モジュールは，通信をモニタリングし加害ノードを特定，攻撃を行っているかを判定する．ノード排除投票モジュールは，加害ノード判定モジュールの判定に基づき，加害ノードとされているノードを排除すべきか投票を行う．排除を行う場合，実際に排除を行う補助ノードを補助ノード決定モジュールが決定し，通信停止モジュールは加害ノードへ通信停止のためのデータを送信する．図 4.2 に CoPS のシステム構成図を示す．

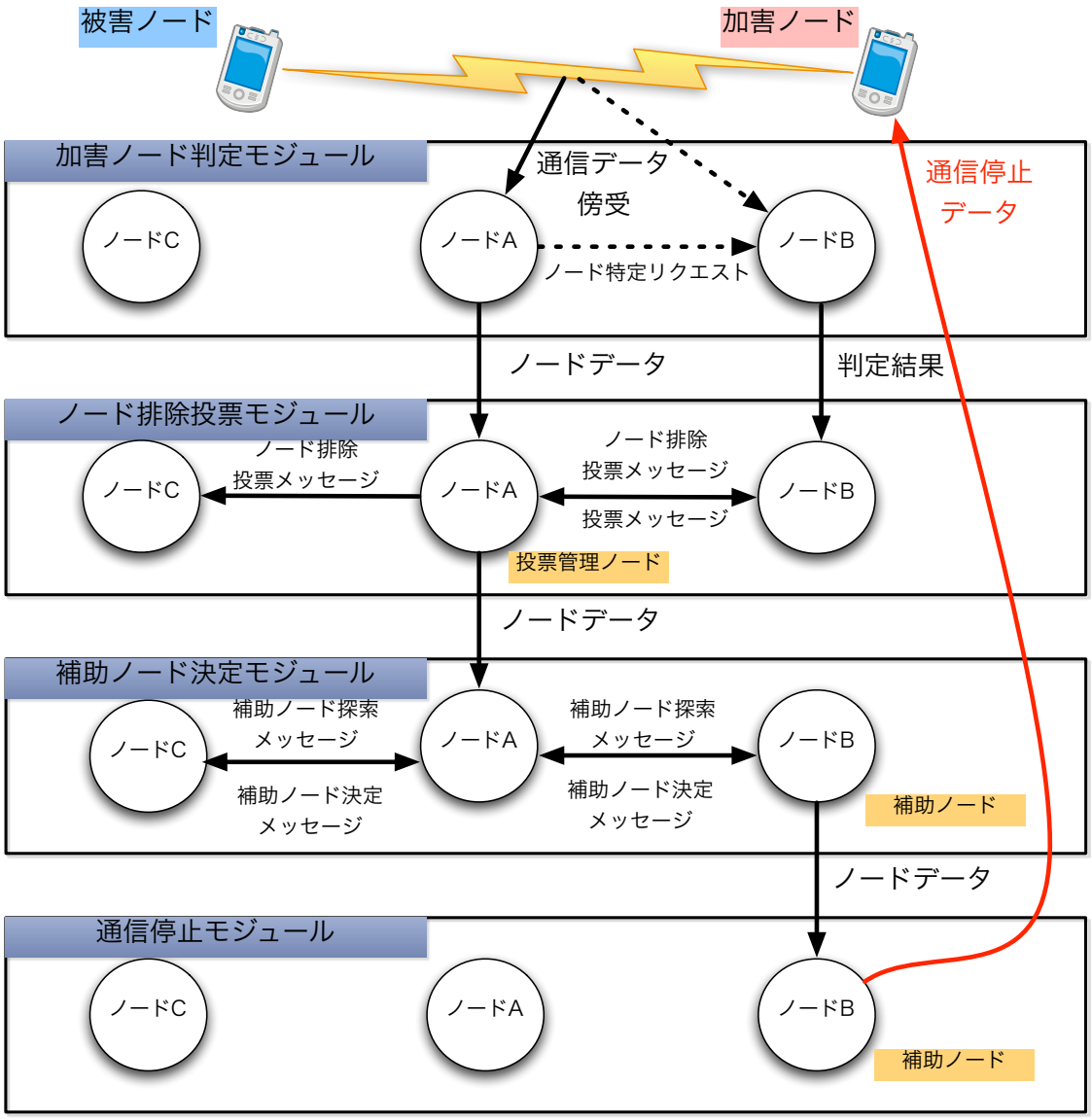

図 4.2: CoPS システム構成図．点線はノードの特定に失敗した際の動作を示す．ノード A が加害ノードを特定し，投票後ノード B が補助ノードに設定され加害ノードに通信停止データを送信する．

4.3.1 加害ノード判定モジュール

ネットワーク内のノードが攻撃を受けていることが本モジュールに伝えられると，ネットワークインタフェースをプロミスキャスモードに設定し，周辺の通信モニタリングを行う．特定ノードからのあらかじめ設定された閾値を超えるデータ送信を発見した場合，その送信ノードを加害ノードとし，通信内容から加害ノードと被害ノードの IP アドレスと MAC アドレスを取得する．加害ノード判定モジュールは動作するきっかけによって，通信モニタリングを行う際に対象とするパケットと動作が異なる．

- 自身が攻撃を受け，IDS などから通知を受けた場合
  自身が受けている通信を監視する．
- 監視関係にある隣接ノードが応答しなくなった通知を受けた場合
  応答しなくなったノードを宛先とする通信を監視する．攻撃を観測できなかった場合，ノード特定リクエストをネットワークにブロードキャストする．
- ノード特定リクエスト，ノード排除投票メッセージを受けた場合
  ノード特定リクエストに含まれる被害ノードの MAC アドレス，もしくはノード排除投票メッセージに含まれる，加害ノードと被害ノードの MAC アドレスを含む通信を監視する．

その後，被害ノード，特定した加害ノードの IP アドレスと MAC アドレスを補助ノード決定モジュールに通知する．

4.3.2 ノード排除投票モジュール

本モジュールは，加害ノード判定モジュールの判定に基づき，設定された加害ノードを排除すべきか決定する投票の管理を行う．他ノードからの投票要求や他ノードからの投票メッセージを待ち受けるため，常に起動している．本モジュールの動作イメージを図 4.3 に示す．

自身が攻撃を受けている場合，あるいは隣接ノードが応答しなくなった場合など，自らが投票を行う最初のノードである場合は，判定に必要な情報を含んだ投票要求メッセージをネットワーク内にブロードキャストする．投票要求メッセージに含まれる情報は以下である．

- 加害ノードの IP アドレス
- 加害ノードの MAC アドレス
- 被害ノードの IP アドレス
- 被害ノードの MAC アドレス
- 自身の IP アドレス
  投票を行う投票管理ノードとなる．
- 投票識別 ID
  ネットワーク内に複数の投票に関するメッセージが流れた場合，投票のグループを識別できなくなる可能性があるため．この ID は重複を避けるためにランダムな英数字で作成される．

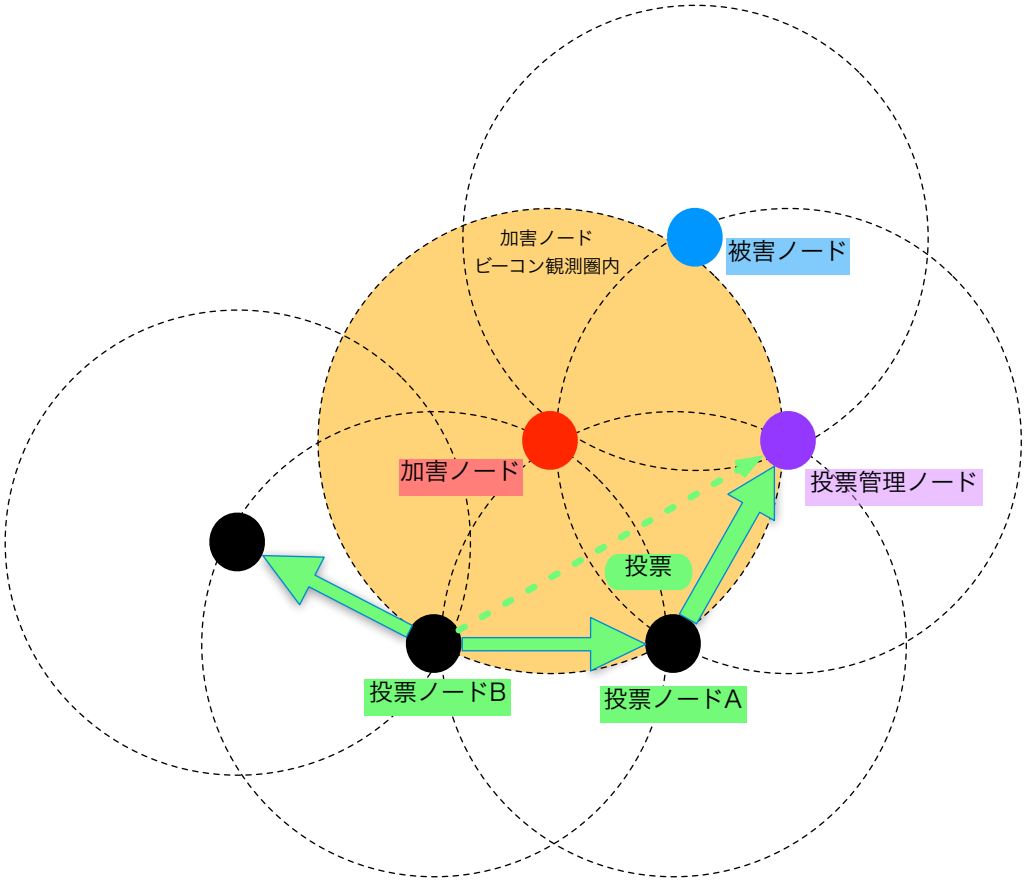

図 4.3: ノード排除投票モジュール 動作イメージ．加害ノードに隣接する A, B が投票ノードとなり，通信を監視し投票を行っている．投票ノード B の投票内容は実際には A をホップして投票管理ノードへ届く (図破線矢印)． A や B の投票内容は他のノードにも記録されている．

他ノードからの投票要求メッセージを受信した場合は，まず加害ノード，被害ノードへの通信監視可否を判定するため， IEEE 802.11 MAC においてノードの存在を知らせるために一定時間で発信されるビーコンフレームを観測する．どちらかの MAC アドレスからのビーコンフレームを観測できた場合は，加害ノード判定モジュールに加害ノードと被害ノードの MAC アドレスを通知し，加害ノードが本当に攻撃を行っているか判定させる．また，ビーコンフレームを観測できなかった場合は加害ノード判定を行わない．受け取った投票識別 ID は，投票要求メッセージに含まれた情報とともにテーブルに書きこむ．加害ノード判定モジュールは受け取った情報を用いて監視を行い，本モジュールに結果を返却する．その後，受け取った結果を含んだ投票メッセージを，投票の相互監視のため，ネットワーク内にブロードキャストする．投票メッセージに含まれる情報は以下である．

- 投票識別 ID
- 投票結果 (true or false)

他の投票メッセージを受信した場合，自身に届いた投票要求メッセージの中で該当する投票識別 ID のものがないかテーブルを検索する．同じ投票識別 ID がない場合，その投票パケットは偽造された可能性が高いためこれを無視する．投票識別 ID がある場合，該当する ID の投票数と true の数を増やす．

また，自身が投票を開始したモジュールである場合，時間を計測し，一定時間を超えたら投票数と true の数をカウントする．true の数が全投票数のうち，設定された割合に到達した場合，投票によりネットワークからの追放が決まったと判断し，加害ノード，被害ノードの情報，投票識別 ID を補助ノード決定モジュールに渡す．

4.3.3 補助ノード決定モジュール

本モジュールは，加害ノード，被害ノードの情報，投票識別 ID を得て動作する．ノード排除投票モジュールから通知を受けた場合，加害ノードの MAC アドレスからのビーコンフレームを観測し，観測できた場合は直接通信が可能とし，自身を補助ノードと設定して，自ノードの情報を補助ノード決定メッセージとしてネットワーク内にブロードキャスト送信する．補助ノード決定メッセージに含まれる情報は以下である．

- 自身 (補助ノード) の IP アドレス
- 投票識別 ID

その後通信停止モジュールに加害ノード，被害ノードの MAC アドレスを引き渡す．通信が不可能な場合には，被害ノード，加害ノードの IP アドレスと MAC アドレスを含んだ補助ノード探索メッセージをブロードキャスト送信する．補助ノード探索メッセージに含まれる情報は以下である．

- 加害ノードの IP アドレス
- 加害ノードの MAC アドレス
- 被害ノードの IP アドレス
- 被害ノードの MAC アドレス
- 投票識別 ID

補助ノード探索メッセージを受信したノードは加害ノードの MAC アドレスからのビーコンフレームを観測し，通信可能な場合は補助ノード決定メッセージを送信する．補助ノード決定メッセージを受信した場合，投票をカウントするテーブル内の該当する投票識別 ID を検索し，エントリをテーブルから削除する．

4.3.4 通信停止モジュール

本モジュールは，補助ノード決定モジュールから加害ノード，被害ノードの MAC アドレスを得て動作する．3.3 節で述べたように，メッセージ送信には IEEE 802.11 MAC における DeAuthentication フレームを利用し，送信元 MAC アドレスを被害ノードのものに書き換えたフレームを加害ノードへ送信する．

4.4 送信されるメッセージ量

4.3 節において述べたように， CoPS はネットワーク上ノード全体の通信によって加害ノードの特定や加害ノード排除の意思決定を行う．そのため，動作においてメッセージをブロードキャスト送信する必要性が複数回生じる．以下の 2 つのシナリオにおいて，CoPS が攻撃を検知してから加害ノードの排除を完了するまでに要する送信メッセージの総量を示す．

- シナリオ A
  被害ノードの隣接ノードが攻撃の検知，加害ノードの特定に成功し，またこの隣接ノードが補助ノードとなる場合．4.3.1 小節で述べたノード特定リクエストや，4.3.3 小節で述べた補助ノード探索メッセージは送信されない．被害ノード自身が加害ノードを特定して通信を行った場合を除いてメッセージ量は最小となる．

- シナリオ B
  被害ノードの隣接ノードが加害ノードを特定できず，またこの隣接ノードが補助ノードとなれない場合．この場合は，ノード特定リクエストや補助ノード探索メッセージなど， CoPS が用いるメッセージ全てが送信される．考えうる最悪のシナリオであり，メッセージ量は最大となる．

まず，ネットワーク全体のノード数を x とする．また，CoPS が利用するブロードキャスト送信は，自身以外のネットワーク上全ノードにデータ送信を行うものとし，その際のホップ数は考慮しないものとする．ブロードキャスト送信を行う際のメッセージ量は送信するノード数に等しいため， $x-1$ とおける．

次に，加害ノード，被害ノードの隣接ノードであり，排除投票に参加するノード数をノード全体の k % とすると，排除投票に参加するノード数 n は式 4.1 のように定義できる．

$$n = \frac{k}{100}x = 0.01kx \tag{4.1}$$

シナリオ A において送信されるメッセージは，以下の 3 つである．

- 投票要求メッセージ
- 投票メッセージ
- 補助ノード決定メッセージ

4.3 節で述べたように，これら 3 つのメッセージは全てブロードキャスト送信される．投票要求メッセージ，補助ノード決定メッセージは投票管理ノードからブロードキャストされるため，メッセージ量はそれぞれ $x-1$ とおける．また，投票メッセージは排除投票に参加するノードがブロードキャストするメッセージであるため，メッセージ量 m は式 4.2 のようにおける．

$$m = n(x-1) = 0.01kx(x-1) \tag{4.2}$$

これより，シナリオ A において送信されるメッセージの総量 Y は式 4.3 のように示される．

$$Y = (x-1) + (x-1) + 0.01kx(x-1) = (2 + 0.01kx)(x-1) \tag{4.3}$$

次に，シナリオ B において送信されるメッセージは，以下の 5 つである．

- 加害ノード特定リクエスト
- 投票要求メッセージ
- 投票メッセージ
- 補助ノード探索メッセージ
- 補助ノード決定メッセージ

加害ノード特定リクエストは，最初に被害ノードの異常を検知したノードがブロードキャストするためメッセージ量は $x-1$ である．また，補助ノード探索メッセージも投票管理ノードからブロードキャストするためメッセージ量は $x-1$ となる．これより，シナリオ B において送信されるメッセージの総量 Y は式 4.4 のように示せる．

$$Y = (x-1) + (x-1) + (x-1) + (x-1) + 0.01kx(x-1) = (4 + 0.01kx)(x-1) \tag{4.4}$$

シナリオ A, B において，排除投票に参加するノード数 k をノード全体の 30 % と仮定し，ノード台数 x を変化させた場合のメッセージ量を，図 4.4 に示す．

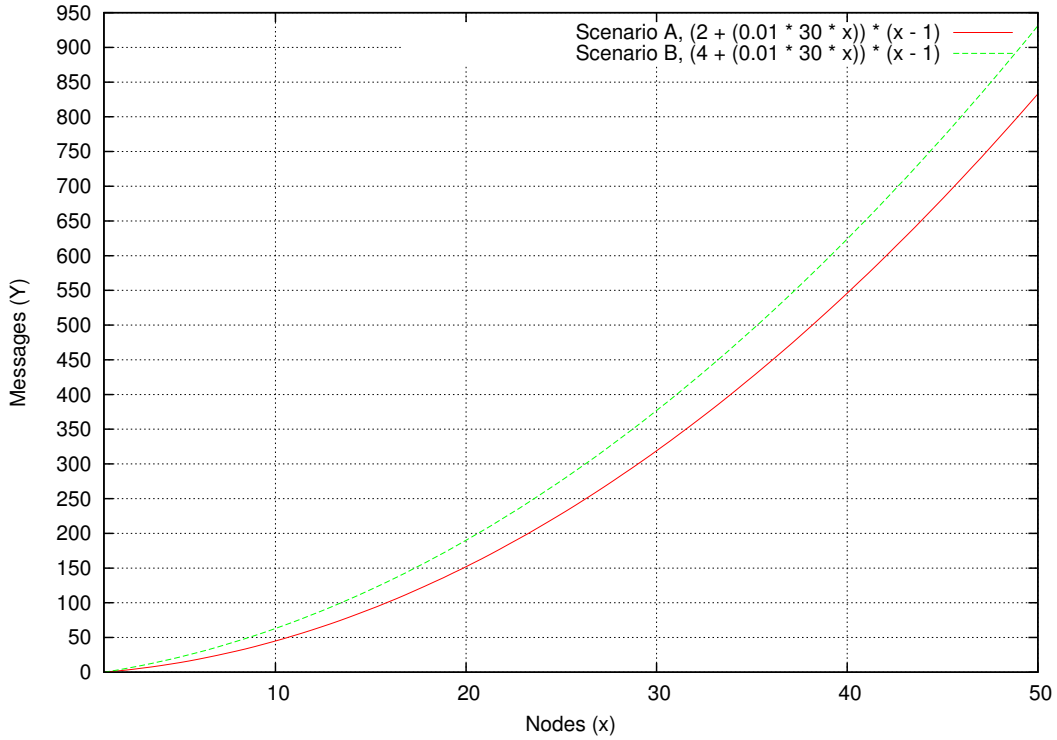

図 4.4: ネットワーク全体のノード数と CoPS が送信するメッセージ量

また，同じくシナリオ A, B において，ネットワーク全体のノード数 x を 30 と仮定し，排除投票に参加するノード数の割合 k を変化させた場合のメッセージ量を，図 4.5 に示す．

4.5 本章のまとめ

本章では， CoPS の設計や構成について述べた．

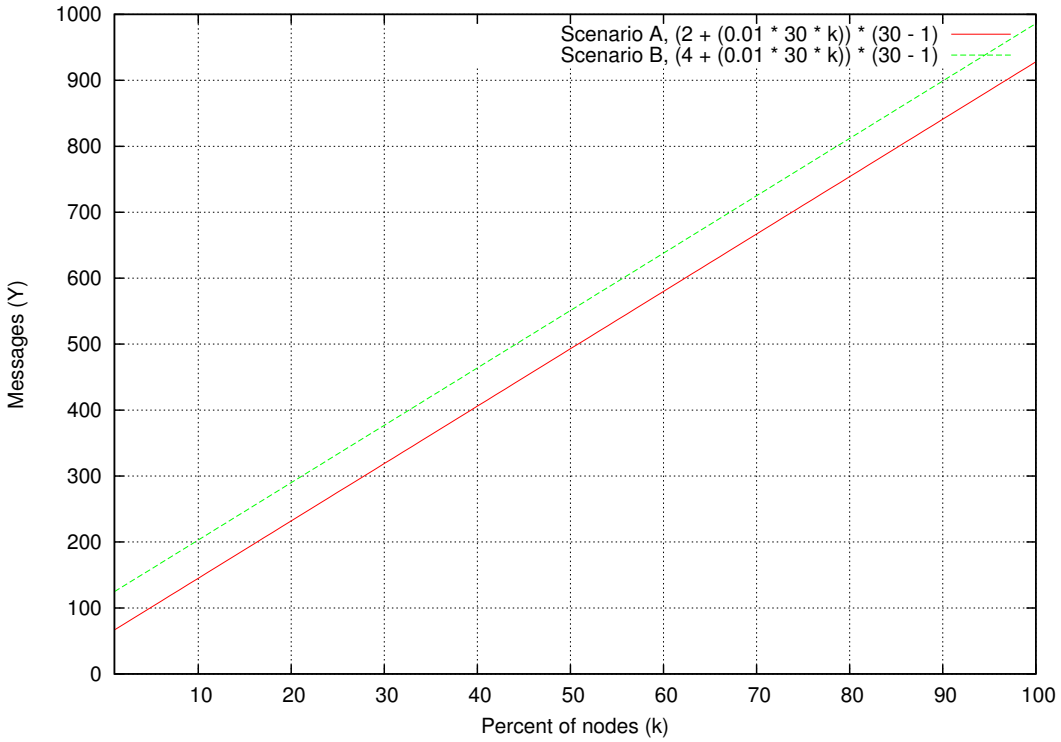

図 4.5: 排除投票に参加するノードの割合と CoPS が送信するメッセージ量

第5章　CoPS の実装

本章では，4 章にて述べた設計に基づき，CoPS の実装について述べる．

5.1 概要

本研究では 4 章で説明した設計を元に，CoPS の実装を行った．本章では， CoPS の実装について述べる．まず， CoPS が対象とする攻撃について説明し，今回の実装で対応した攻撃について述べる．次にハードウェア構成を述べ，ソフトウェア構成と各モジュール，評価用ドライバの実装について述べる．

5.2 対象とする攻撃

本研究は，攻撃を検知した後にノードの協調動作により不正なノードをネットワークから排除することを目的としている．そのため，攻撃パケットを観測した上で該当する攻撃を検知することができれば， CoPS を利用することが可能である．2.1 節でも述べたように，無線アドホックネットワークにおいて考えられる攻撃は様々であるが， CoPS の動作タイミングはノード上で動く攻撃検知アルゴリズムに依存する．本論文においては，検知対象を UDP および ICMP によるパケットフラッディング攻撃として実装を行った．

5.3 開発環境

本節では CoPS の実装に使用したハードウェア環境とソフトウェア環境について述べる．

5.3.1 ハードウェア構成

CoPS は小型の移動可能な無線端末を動作ハードウェアとして想定している．また，5.4.4 小節で述べるが，通信停止モジュールを実装するため，実装に利用するマシン全てに Atheros 社製 802.11 チップを搭載した無線 LAN カードを装着し，実装を行った．

5.3.2 ソフトウェア構成

次に，ソフトウェア構成について述べる． CoPS は Linux ディストリビューションの 1 つである Ubuntu 上に，C 言語と Ruby を用いて実装を行った．通信停止モジュールを動作させるドライバ，また評価用に変更を行ったドライバとして Atheros 社製無線 LAN チップ用オープンソース Linux ドライバである MADWifi[21] を利用している．実装に使用したソフトウェア構成を表 5.1 に示す．

表 5.1: ソフトウェア構成

要素	詳細
OS	Ubuntu 8.04 LTS (Hardy Heron) / Linux kernel 2.6.24
使用言語	C, Ruby 1.8.7
パケットキャプチャライブラリ	libpcap[22] 1.1.1, PacketFu[23] 1.0.0
無線 LAN カードドライバ	MADWifi[21] 0.10.5.6 Revision 4112

5.3.3 MADWifi

本論文の実装では，評価用ドライバとして Atheros 社製無線 LAN チップの Linux ドライバである MADWifi[21] を利用している．MADWifi はオープンソースソフトウェアであり，Atheros 社製無線 LAN チップを搭載した無線 LAN カードを Linux 対応にするドライバとして普及している．本研究では，一般的なドライバには実装されていない， 802.11 認証が行われている実験用アドホックネットワークの作成や， DA フレームの扱いの変更，通信停止モジュールが DA フレームを意図的に改変して送信するといった動作を行うため，コードが公開され，改変可能な MADWifi と Atheros 社製無線 LAN チップの組み合わせを利用する．

5.4 各モジュールの実装

本節では CoPS を構成する各モジュールの実装について述べる．

5.4.1 加害ノード判定モジュール

加害ノード判定モジュールは，被害ノード，もしくは被害ノードと加害ノードの MAC アドレスを受けて，ネットワークインタフェースをモニタモードに設定し，10 秒間通信の監視を行う．ノード特定リクエストを受信するため常に起動しており，リクエストを受信した場合も同じように監視を行う．10 秒間の通信のうち，被害ノードに向けたもの，もしくは加害ノードから被害ノードに向けられたパケットを記録する．5.2 節で述べたが，本論文において対象とする攻撃は UDP および ICMP によるパケットフラッディング攻撃であるため，送信元別にパケット数を記録し，一定以上を超えた場合に攻撃が行われていると判定する．本論文においては，10 秒間で 200 パケット以上の送信が確認された場合に攻撃が行われている判定をするよう実装を行った．使用言語は Ruby である．監視にはパケットキャプチャ用ライブラリである libpcap[22] の Ruby ラッパである PacketFu[23] を用いて実装を行った．

5.4.2 ノード排除投票モジュール

ノード排除投票モジュールは，他のノードから投票要求メッセージを受信した場合，ネットワークインタフェースをモニタモードに切り替え，加害ノードの MAC アドレスからのビーコンフレーム取得を試みる．フレームが取得できた場合，加害ノードは隣接ノードであるため，受信した被害ノード，加害ノードのアドレスを加害ノード判定モジュールに通知する．その後，加害ノード判定モジュールの結果を得て投票メッセージを生成し，ブロードキャスト送信する．ビーコンフレームが取得できなかった場合は加害ノードは隣接ノードではないと判断し，何も行わない．自身が攻撃を検知した最初のノードである場合，加害ノード判定モジュールから加害ノード，および被害ノードの IP アドレスと MAC アドレスを受信し，自身を投票管理ノードとして投票要求メッセージをネットワークにブロードキャスト送信する．4.3.2 小節にて述べた投票 ID は，タイムスタンプにランダムな英数字 3 字を付加したものを生成する．また，ノード排除投票モジュールは投票 ID によって紐付けられた投票結果テーブルを保持し，自身が投票管理ノードである場合，管理する投票の結果によって補助ノード決定モジュールに通知する．使用言語は Ruby である．

5.4.3 補助ノード決定モジュール

補助ノード決定モジュールは，自身が投票管理ノードであった場合，ノード排除投票モジュールから通知を受信する．起動するとネットワークインタフェースをモニタモードに切り替え，加害ノードの MAC アドレスからのビーコンフレーム取得を試みる．フレームが取得できた場合は自身が加害ノードの隣接ノードであり， IEEE 802.11 MAC を使った通信が可能であるため，補助ノード決定メッセージをネットワークにブロードキャストし，通信停止モジュールを起動する．できなかった場合は補助ノード探索メッセージをブロードキャスト送信する．．他のノードからの補助ノード探索メッセージを受信した場合，前述のように加害ノードが隣接ノードであるかの判定を行い，隣接ノードである場合には補助ノード決定メッセージをブロードキャスト送信する．また，隣接ノードでなかった場合には何も行わない．モジュール全体の使用言語は Ruby であり，ビーコンフレーム取得部のみ C 言語と libpcap を利用している．

5.4.4 通信停止モジュール

通信停止モジュールは，指定された MAC アドレスに DA フレームを送信するモジュールである．オープンソースの WEP 解析ツールである Aircrack-ng[24] の一部機能を改良して実装を行っており，送信元，宛先 MAC アドレスを指定して起動することで送信元を指定された MAC アドレスに書き換えた DA フレームを送信する．通常の DA フレームと区別するため， 3.3 節にて述べた，DA フレームに含まれる理由コードにおいて，確保されているが未割り当ての数値 (100) を格納して送信する．

5.5 評価用ドライバ

CoPS の評価のため， MADWifi の IEEE 802.11 プロトコルスタック上に，アドホックモード利用時において DA フレームを受信した際に通信を終了する処理を追加した．これは， MADWifi や Intel 社製無線 LAN カードドライバ [25] などの Linux 用無線 LAN カードドライバが，アドホックモード時に認証を行わず DA フレームを無視してしまう仕様となっていたためである．5.4.4 小節で述べた，CoPS で利用する理由コード (100) を含んだ DA フレームを受信することによって通信を終了する．

5.6 本章のまとめ

本章では，4 章にて述べた設計に基づき，まず CoPS 実装の概要を述べ，今回の実装で対応した攻撃について説明した．また，実装環境について説明し，最後に各モジュールや評価用ドライバの実装について述べた．

第6章　評価

本章では，CoPS の評価について述べ，それに対する考察を行う．

6.1 評価方針

本研究では，実装したソフトウェアを用いて実験を行い，ノード協調アルゴリズムの動作速度，ノード排除による負荷測定，加害ノードへのデータ送信手法の違いによるデータ送信速度差について定量的に評価する．評価項目は以下の 3 つである．

- ノード協調アルゴリズムの動作速度
  この評価実験は， CoPS が攻撃防御機構としての役割を果たす上で，加害ノードを認識してからノード協調アルゴリズムにより補助ノードが選定されるまでにどの程度の時間を要するか評価することを目的として行う．実験用に構築した無線アドホックネットワーク上において，実際の攻撃を行い，ノード協調アルゴリズムが完了するまでの時間を測定することで評価を行う．

- ノード排除による負荷測定
  CoPS は，小型の移動端末で構成された無線アドホックネットワークにおいても，端末の主機能に影響を与えることなく動作するよう提案する攻撃防御機構である．この実験は， CoPS が動作するうえでノードにかかる負荷を計測し，リソースの乏しい小型端末においても問題なく動作するか考察するために行う．

- 加害ノードへのデータ送信手法の違いによるデータ送信速度差
  本研究において，加害ノードへのデータ送信手法として IEEE 802.11 MAC を用いたデータ送信手法を提案した．ここでは，加害ノードが攻撃を行っている状況下で，提案手法と他の手法によるメッセージ送信実験を行い，メッセージ伝達速度と到達性について評価を行う．

6.2 実験環境

本章で述べる実験では，複数台の PC を用いて無線アドホックネットワークを構築した．本節では，実験のため構築したネットワーク環境について具体的に述べる．

6.2.1 ネットワークトポロジ

実験にあたって構築したネットワークは，5 台の PC を用いて， IEEE 802.11b によって相互接続することによって構成される．それぞれが互いに直接通信することのできる，フルメッシュ型の無線アドホックネットワークである．このうち加害ノード，被害ノード，補助ノードが 1 台ずつ存在し，残る 2 台は投票に使われるノードである．実験で構成したネットワークトポロジ図を図 6.1 に示す．

6.2.2 ハードウェア構成

実験に使用した PC のうち，加害ノード，被害ノードとなった PC のハードウェア構成を表 6.1 に示す．また，補助ノードとなった PC のハードウェア構成を表 6.2 に示す．

6.3 ノード協調アルゴリズムの速度評価

本節では，CoPS が加害ノードを認識してから意思決定を行い，補助ノードを選定するまでのノード協調アルゴリズムの動作速度について定量的に評価する．加害ノードを認識してから排除するまで

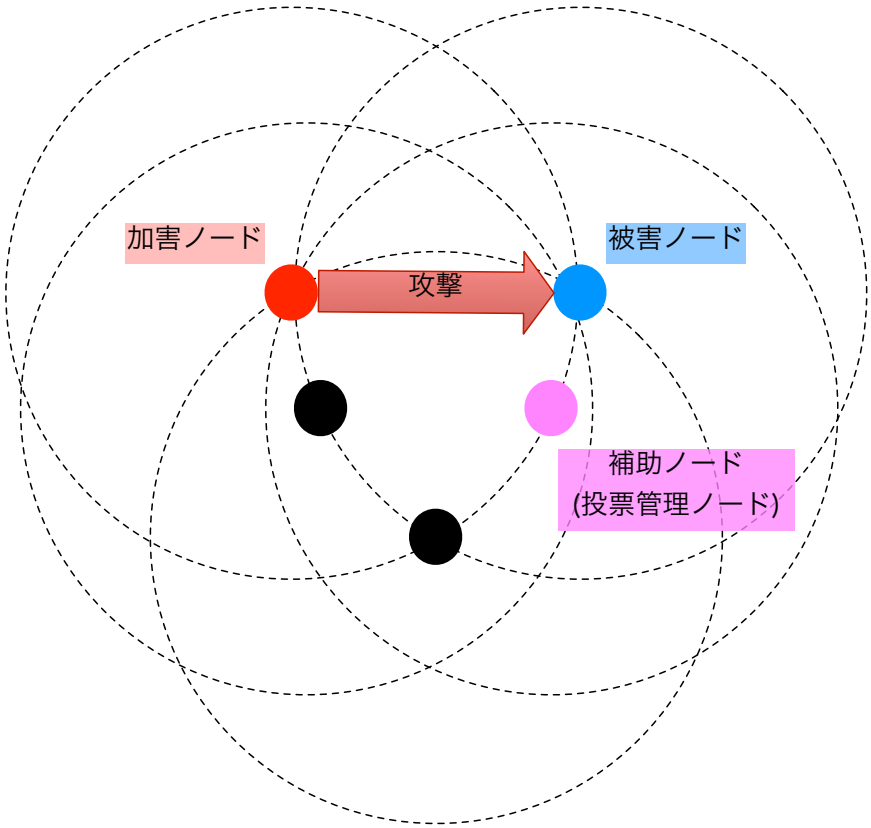

図 6.1: 評価実験において構成したネットワークトポロジ図

のノード協調アルゴリズムは，加害ノードがネットワークに大きな影響を及ぼす前に素早く動作する必要がある．

6.3.1 実験方法

実装した各モジュールならびに評価用ドライバを用いて，ノード協調アルゴリズムにおける，加害ノードの認識から通信停止データ送信までを動作させる実験を行う．具体的には，加害ノードから被害ノードへ UDP ヘッダのみのパケットを大量に送信する DoS 攻撃を行い，攻撃を受けていないノードの加害ノード特定モジュールを動作させ，通信停止データが送信されるまでの時間を測定する．また，被害ノードが攻撃を受けていることを最初に検知したノードが投票管理ノード，並びに補助ノードになるものとした．そのため，アルゴリズムは以下のように動作する．

- 加害ノード特定モジュールの起動
  加害ノード特定モジュールが起動し， 10 秒間通信の監視を行い，投票要求を行う．
- 投票
  投票要求を受けた他のノードが加害ノード特定モジュールを起動して 10 秒間通信の監視を行う．また，監視結果によって投票を行う．

表 6.1: 加害ノード，被害ノードのハードウェア構成

要素	加害ノード	被害ノード
モデル	Sony Vaio SZ VGN-SZ73B	IBM Thinkpad T42p
CPU	Intel Core2 Duo T7200 @ 2.00GHz	Intel Pentium M @ 2.10GHz
メモリ	2GB PC2-4200	1GB PC-3200
HDD	100GB	80GB
無線 LAN カード	NEC Aterm WL54AG (Atheros AR5212)	NEC Aterm WL54AG

表 6.2: 補助ノードのハードウェア構成

要素	補助ノード
モデル	Sony Vaio T VGN-T91PS
CPU	Intel Pentium M 753 @ 1.20GHz
メモリ	1GB PC-3200
HDD	60GB
無線 LAN カード	NEC Aterm WL54AG (Atheros AR5212)

- 補助ノード探索
  投票を受けた投票管理ノードは，補助ノードの探索を行う．本論文における実験ではフルメッシュ型の無線アドホックネットワークを用いているため，投票管理ノードから直接通信が可能である．そのため，10 秒間通信を監視した上で通信可能と判断し，補助ノード決定メッセージ，通信停止データを送信する．補助ノード探索メッセージは送出されない．

測定は，攻撃を行っている状況下で，補助ノードにおいて加害ノード特定モジュールを動作させた時間，通信停止データが送信された時間を 10 回計測し，平均を計算することで行った．送信した UDP ヘッダのサイズは 28 byte である．

6.3.2 実験結果

表 6.3: アルゴリズム動作時間 (単位: s)

	測定結果
最大	32.541
最小	30.390
平均	**31.145**
標準偏差	0.770

表 6.3 に，加害ノードモジュールが起動してから通信停止データが送信されるまで計測された時間を実験結果として示す．アルゴリズムの動作にかかった時間は，最小で 30.39 秒であり，最大 32.541 秒であり，平均すると 31.145 秒であった．

6.3.3 考察

6.3.1 小節で述べたように，加害ノード判定モジュールや補助ノード決定モジュールが判定を行うために通信を 10 秒間監視しており，アルゴリズム内でこれらのモジュールが動作すると，1 回につき 10 秒を要する．つまり，表 6.3 に示した実験結果は，これらのモジュールが通信を監視する回数や，判定を行うために必要な秒数に大きく影響されている．実験では加害ノード判定モジュールは 2 回，補助ノード決定モジュールは 1 回動作しているため，結果のうち 30 秒はこれらのモジュールの動作によるものと言える．メッセージ通信によるオーバーヘッドは非常に低いため，この結果は，加害ノード判定モジュールや補助ノード決定モジュールによる判定アルゴリズムの変更によって改善されるものと言える．

6.4 ノード排除による負荷測定

本節では，CoPS が動作することによってノードにかかる負荷を計測する．本論文において，CoPS を Linux 上で動作するソフトウェアとして実装したため，CoPS がターゲットとする小型端末に近い性能のノード上で実験を行った．

6.4.1 実験方法

補助ノードにおいて，実装した各モジュールのうち単体で実行可能な加害ノード判定モジュールについて，10 秒動作させた場合の負荷計測を time コマンドを用いて 10 回行い，平均値を算出した．加害ノード判定モジュールの負荷計測は，監視する攻撃のパケット数による変動を考慮するため，UDP ヘッダのみのパケットを大量に送信している状態と UDP ヘッダを 1 秒に 1 パケット送信している状態においてそれぞれ行い，使用しているライブラリである PacketFu[23] の動作負荷との関係を考慮するため，パケットの解析を行わない状態での負荷計測も実施した．また，6.3 節で述べた実験と同じようにノード排除アルゴリズムを動作させ，システム全体を動作させた場合の負荷計測も同様に行った．動作時間は 6.3.2 小節で述べたように，約 30 秒である．

6.4.2 実験結果

表 6.4 に大量のパケットを送信した場合の動作負荷，表 6.5 に 1 秒間に 1 パケットを送信した場合の動作負荷を示す．実装した加害ノード判定モジュールを使用した場合，PacketFu を用いたパケットキャプチャのみを行っている状態に比べて 3 秒ほどユーザ時間の増加がみられた．また，システム時間は 3 秒ほど減少した．

6.4.3 考察

本実験により，実装したモジュールによるパケットの解析を行った場合，ユーザ時間が増加することがわかった．モジュールの実装には Ruby を用いており，パケットごとの解析を行っているためにユーザ時間が増加したものと考えられる．加害ノード判定モジュールのみ動作させた場合と，システム全体を動作させた場合のリソース使用量に差があまり見られず，加害ノード判定モジュール以外のモジュールのリソース消費は少ないものと考えられる．

加害ノード判定モジュールのみを動作させた場合の CPU 使用率はほぼ 98% であったが，システム全体を動作させた場合の CPU 使用率はほぼ 30 % であった．加害ノード判定モジュールはパケットの

表 6.4: パケットを大量に送信した場合の動作負荷 (単位: s)

	ユーザ時間 (パケット解析無し)	システム時間 (パケット解析無し)
最大	3.70	6.82
最小	3.44	6.46
平均	**3.58**	**6.60**
標準偏差	0.09	0.15
	ユーザ時間 (パケット解析有り)	システム時間 (パケット解析有り)
最大	7.18	3.18
最小	6.02	2.64
平均	**6.63**	**2.97**
標準偏差	0.47	0.21
	ユーザ時間 (システム全体)	システム時間 (システム全体)
最大	7.29	4.16
最小	6.51	3.56
平均	**6.82**	**3.90**
標準偏差	0.32	0.22

表 6.5: 1 秒に 1 パケットを送信している場合の動作負荷 (単位: s)

	ユーザ時間 (パケット解析無し)	システム時間 (パケット解析無し)
最大	3.55	6.49
最小	3.42	6.35
平均	**3.49**	**6.42**
標準偏差	0.04	0.05
	ユーザ時間 (パケット解析有り)	システム時間 (パケット解析有り)
最大	3.70	6.82
最小	3.42	6.41
平均	**3.60**	**6.63**
標準偏差	0.11	0.16
	ユーザ時間 (システム全体)	システム時間 (システム全体)
最大	3.80	6.09
最小	3.23	5.70
平均	**3.56**	**5.89**
標準偏差	0.21	0.15

キャプチャをプロミスキャスモードで無差別に行い解析するため，解析中のみ負荷が高くなるが限られた時間であり，システム全体としての負荷は低い．また，攻撃が行われていない際のリソース消費はないため， CoPS がターゲットとする小型端末においても十分動作させることが可能と考えられる．

6.5 加害ノードへのデータ送信手法の速度評価

本節では，3.3 小節にて説明した加害ノードへの IEEE 802.11 MAC を用いたデータ送信手法のメッセージ伝達速度について定量的に評価する．加害ノードを特定し排除を決定した際には，できる限り早くネットワークから排除する必要がある．また，加害ノードは被害ノードに向けて大量にデータを送信するような攻撃を行っている可能性があり，IP 層より上位のプロトコルでデータを送信しても受信されない可能性がある．このため，加害ノードに対して通信切断データを素早く，確実に送信できる必要がある．

6.5.1 実験方法

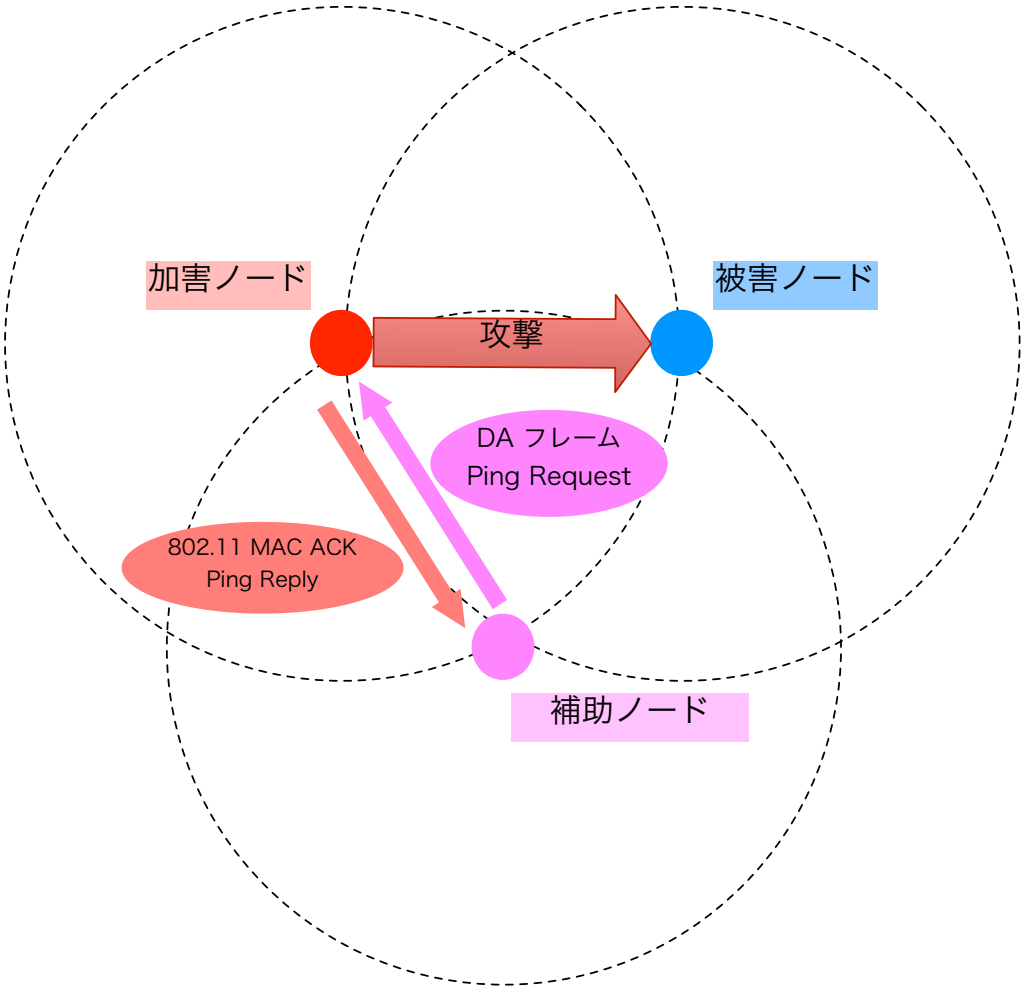

図 6.2: データ送信手法の速度評価実験において構成したネットワークトポロジ図

実装した通信停止モジュールならびに評価用ドライバを用いて，補助ノードが DoS 攻撃を行う加害ノードに対して DA フレームを送信する実験を行う．具体的には， DoS 攻撃が行われている状況を想定し，加害ノードから被害ノードへ実際に攻撃を行う．攻撃を行っている状態で，加害ノードへの通信停止データ送信手法として，本論文において提案する IEEE 802.11 MAC DA フレームを利用してメッセージを送信する場合と，IEEE 802.11 MAC より上位レイヤである ICMP を利用してデータを送信する手法の比較を行った．

測定は，加害ノードから被害ノードへ何もデータが送られていない状態， UDP ヘッダのみのパケットを大量に送信している状態， UDP ヘッダとデータ部のあるパケットを大量に送信している状態の 3 つの状況において，補助ノードがデータを送信し，加害ノードからの確認応答が戻ってくるまでの時間を補助ノードにおいて 10 回計測し，平均を計算することで行った．実験のため構成したネットワークのトポロジを図 6.2 に示す．送信した UDP ヘッダのサイズは 28 byte であり， UDP データ部のサイズは 1500 byte である．また， DA フレームのサイズは 30 byte であり， ICMP パケットの送信には Ping プログラムを利用したため ICMP パケットサイズは， MAC ヘッダ 14 byte，IP ヘッダ 20 byte， ICMP ヘッダ 8 byte，ICMP データフィールド 56 byte を合計し 98 byte である．

表 6.6: IEEE 802.11 MAC DA フレーム応答時間 (単位: μs)

	DoS 攻撃無し	UDP ヘッダ + データ	UDP ヘッダ
最大	0.84	31.42	27.43
最小	0.81	14.56	7.02
平均	**0.83**	**24.56**	**20.59**
標準偏差	0.07	4.40	6.38

表 6.7: Ping 応答時間 (単位: μs)

	DoS 攻撃無し	UDP ヘッダ + データ	UDP ヘッダ
最大	1240.0	27400.0	N/A
最小	578.0	15300.0	N/A
平均	**668.2**	**22390.0**	**N/A**
標準偏差	191.3	3549.8	N/A

6.5.2 実験結果

表 6.6 に DA フレームを送信した際の応答時間，表 6.7 に Ping プログラムを利用した際の応答時間を実験結果として示す．DA フレームを用いた提案手法では，加害ノードから被害ノードへの送信データが無い状態での応答時間が平均 0.83 μs であるのに対し， Ping プログラムを利用した ICMP による手法では平均 668.2 μs であった．また，UDP ヘッダとデータを大量に送信している状態においても，提案手法の応答時間が平均 24.56 μs であるのに対し， ICMP による手法では平均 22390 μs (22.39 ms) と大きく低下している．さらに， UDP ヘッダのみを大量に送信している状態では， ICMP による通信はできなかった．

6.5.3 考察

表 6.6，表 6.7 の実験結果から，大量にデータを送信する DoS 攻撃が行われている状況において，提案手法は ICMP など上位層のプロトコルを利用した手法に比べ，ネットワーク層以上の処理状態の影響を受けず素早く通信を行うことが可能であることがわかった．また， UDP ヘッダのみを大量に送信した際に ICMP では通信ができなかったことから，通信を停止させるための通信は DA フレームを用いる手法が有効であるとわかった．

6.6 本章のまとめ

本章では，実装した CoPS を用いて実験を行い，その評価結果について示し，結果についての考察を行った．実験の結果， CoPS は加害ノードを判定し，大量のデータが送出されている状況でも確実に加害ノードをネットワークから排除することができた．また，システム全体の負荷も低く， CoPS がターゲットとする小型端末においても動作可能であると考えられることがわかった．しかし，設計，実装の過程で一般に利用されている無線 LAN ドライバのアドホックモード実装にはばらつきがあり，評価用に実装したドライバ以外では効果がないことが判明している．また，実験をフルメッシュ型の無線アドホックネットワークにおいて行っており，実際に利用が想定されるマルチホップ可能なルーティングプロトコルが適用された無線アドホックネットワークにおいて評価できていない．このため，アドホックネットワーク用ルーティングプロトコルが適用された環境において実験を行うとともに，実際にノードをネットワークから排除する手法についてさらに改善していく必要があると考えている．

第7章　結論

本章では，本研究の総括として，本研究の今後の課題，展望についてまとめ，その上で本論文の結論を述べる．

7.1 今後の課題と展望

本論文では，無線アドホックネットワークにおけるノード協調による攻撃防御機構を提案，実現した．また，本論文において提案する手法が他の手法に比べて優位であるという評価結果を得た．本研究の今後の課題，及び展望として以下の 3 点を挙げる．

1. 加害ノードの実装を伴わないノード排除

   本論文において， IEEE 802.11 MAC を利用した加害ノードへのデータ送信手法を提案し，実装を行った．評価として， ICMP など上位レイヤのプロトコルを利用するより高速に，より確実にメッセージを伝送できる結果を得たが，実装の過程で，多くの無線 LAN ドライバはアドホックモード時に 802.11 認証を無視した状態で通信を行うことが判明した．これは， IEEE 802.11 の仕様上，アドホックモードの際には認証を省略しても通信を開始することができると決められており，それに従った実装が行われているものと考えられる．このことにより，現状の CoPS 実装は，加害ノード側に評価用ドライバを導入し，アドホックモードにおいても DA フレームを受信することで通信を停止するという環境下においてのみ正常に動作するという制約を持っている．しかし， WEP や WPA-PSK などの事前共有鍵による認証を行ってしまうと，事前共有鍵のない，自由に出入りできる無線アドホックネットワークにおいてノード排除を行うことが難しくなってしまう．この問題を解決する為には，無線アドホックネットワークにおいて，事前共有鍵なしに認証を行ったうえで，ノードの排除も行うことができる認証機構を導入する必要がある．

2. 攻撃の検知も含んだ攻撃防御機構の実現

   本研究は，無線アドホックネットワークにおいて攻撃を検知した際に，ノードが協調して加害ノードの特定と意思決定，排除を行うことを目的としており，実際に攻撃を検知，識別する手法については対象としていない．しかし，将来実用化されていく無線アドホックネットワークにおいて本研究を利用するためには，攻撃を適切に検知，識別することのできる，攻撃検知手法との連携が必要不可欠である．リソースの乏しい小型ノードを対象とした既存研究 [16] なども存在し，また既存のソフトウェアに関してもこれから軽量化が進んでいくと考えられる．本研究はそれらと正確に連携することで，ノードが協調して攻撃検知から排除までを自律的に実現可能な機構であると考えており，実現に向けてさらに研究を進めていきたい．

3. より小型なノード，マルチホップ可能な無線アドホックネットワークにおける実装評価

   本研究は，携帯電話など，小型移動端末を対象にした研究であり，負荷計測などからリソースが乏しい状況においても問題なく稼働するであろうという評価を得た．しかし，本論文においては実装上の問題からノート PC 上に実装を行い，評価したため，携帯電話やセンサノードなど本来の小型ノードにおいて実装した際に問題なく稼働するかは未評価である．また，フルメッシュ型の無線アドホックネットワークにおいて実験を行っており，マルチホップ可能な無線アドホックネットワーク環境における実験評価が済んでいない．近年， Android[26] などをはじめとした，より PC で利用される OS に近い小型端末，組み込み端末用 OS が脚光を浴び，研究や実用化が進んでいる．こうした端末に CoPS を実装し，より想定する利用形態に近い無線アドホックネットワークにおいて評価を行いたいと考えている．

7.2 本論文のまとめ

本論文では，小型移動端末によって構成された無線アドホックネットワークにおける攻撃防御を目的として，複数ノードの協調による攻撃防御機構 CoPS の提案と実装を行った．

様々な利用形態が想定され，未来の情報インフラとして期待される無線アドホックネットワークにおいて，より強固なセキュリティ機構を構築することは非常に重要であり，悪意のもと，あるいはソフトウェアの誤動作などによりネットワークやノードに大きな影響を及ぼす不正なノードを排除する攻撃防御機構は大変有用であると考えられる．本研究では，その特徴から従来利用されてきた攻撃防御機構の適用が難しい無線アドホックネットワークにおいて，ノードを協調動作させ，攻撃を受けたノードが動作不能に陥ったり，攻撃を行っているノードが絶えず動きつづけているようなネットワークにおいてもノードを適切に排除できる攻撃防御機構を実現した．既存研究やソフトウェアと連携することで，攻撃の検知から排除までを自律的に行うことも可能である．

CoPS の評価として，ノード排除の速度やシステム動作負荷，大量のデータを送信するノードへのデータ送信手法などにおいて有効な評価を得たが，ネットワークから排除されるノードの側にも実装が必要となってしまう点など，改善の余地はまだ多く，それらを解決していくことでより効果の高い攻撃防御機構を実現することが可能であると考察する．本論文よりさらなる改善を加え，将来利用される無線アドホックネットワークがより安全に稼働するための基盤とすることが本研究の最終目標であり，今後もさらに研究を続けていき，目標達成を目指したい．

謝辞

本研究を進めるにあたり，絶えず丁寧な御指導を賜りました，慶應義塾大学環境情報学部 教授 徳田英幸博士に深く感謝致します．また，貴重な御助言を頂きました慶應義塾大学環境情報学部 准教授 高汐 一紀博士，慶應義塾大学環境情報学部 専任講師 中澤 仁博士，慶應義塾大学 政策・メディア研究科特別研究助教 間 博人博士に感謝致します．徳田・村井・楠本・中村・高汐・重近・バンミーター・植原・三次・中澤・武田合同プロジェクトの皆様には，丁寧な御指導や貴重な御助言を頂きました．特に，斉藤 匡人氏，森 雅智氏，本多 倫夫氏，金澤 貴俊氏には研究室加入時より本日まで，研究方針や論文執筆においてたくさんの御指導と御助言を頂きました．ここに多大なる感謝と尊敬の意を表します．

さらに，徳田研究室 ECN 研究グループにて多くの時間を共に過ごした，蛭田 慎也氏，天野 賢二氏，Dao Thanh Chung 氏，加藤 碧氏，ECN 研究グループ OB の大日野 舞氏，卒論生としてお互い励ましあい研究を進めてきた瀧本 拓也氏，望月 剣氏，Nguyen Thuy Le 氏，丹羽 亮太氏，西 和也氏，堀川 哲郎氏，上田 真央氏，中原 洋志氏，Vu Dinh Long 氏，様々な御指導をいただいた伊藤 友隆氏，野沢 高弘氏，米川 賢治氏，唐津 豊氏，井村 和博氏をはじめとした先輩方，同じ研究室メンバーとして楽しい日々を送った西條 晃平氏，山内 亜里穂氏をはじめとした後輩諸氏，慶應義塾大学徳田・高汐・中澤研究室全ての皆様に心から感謝致します．

また，研究室から離れての活動で多くの時間を共にした， Scuba Diving Team SevenSeas の皆様，a cappella singers K.O.E. の皆様， CNS コンサルタントの皆様，入学時よりお世話になった田島 悠史氏，深澤 瑠衣子氏をはじめとした SF 関係の皆様，研究室を越えて卒論執筆を励ましあった秋山 博紀氏をはじめとした同学年の友人，大学生活を共にした本当にたくさんの友人，先輩，後輩たちに深く感謝致します．

最後に，これまでの大学生活を精神的にも経済的にも支え励まし続けてくれた家族である，父 隆行，母 直美，弟 智也，祖父，祖母，そして私と繋がる全ての方々に心からの感謝と敬愛を表し，本論文の謝辞と致します．

平成 23 年 1 月 19 日
星 北斗

参照論文

- 星 北斗, 森 雅智, 金澤 貴俊, 斉藤 匡人, 間 博人, 徳田 英幸.
  “CoPS: 無線アドホックネットワークにおける ノード協調型攻撃防御機構”
  情報処理学会 マルチメディア，分散，協調とモバイル (DICOMO2010) シンポジウム論文集,
  pp. 463-469 (2010).
  2010 年 7 月

参考文献

[1] Nintendo DS http://www.nintendo.co.jp/ds/

[2] PlayStation Portable http://www.jp.playstation.com/psp/

[3] 日本 e スポーツ学会 http://j-ess.jp/

[4] eKo mote http://www.xbow.jp/eKo.html

[5] Kannhavong. B, Nakayama. H, Nemoto. Y, Kato. N, Jamalipour. A. A survey of routing attacks in mobile ad hoc networks. Wireless Communications, IEEE, pp. 85-91, 2007.

[6] Marti. Sergio, Giuli. T. J, Lai. Kevin, Baker. Mary. Mitigating routing misbehavior in mobile ad hoc networks. MobiCom '00 Proceedings of the 6th annual international conference on Mobile computing and networking, pp. 255-265, 2000

[7] Erik Tews, Ralf-Philipp Weinmann, and Andrei Pyshkin. Breaking 104 bit WEP in less than 60 seconds. WISA'07 Proceedings of the 8th international conference on Information security applications,2007

[8] Snort. http://www.snort.org/

[9] tripwire. http://www.tripwire.org/

[10] IEEE 802.1X Standard (IEEE Computer Society LAN MAN Standards Committee). IEEE Standard for Local and metropolitan area networks: Port-Based Network Access Control, February 2010.

[11] Lashkari. A.H, Danesh. M.M.S, Samadi. B. A survey on wireless security protocols (WEP, WPA and WPA2/802.11i). 2009 the 2nd IEEE International Conference on Computer Science and Information Technology (ICCSIT 2009), pages 48 -52, 2009.

[12] IEEE 802.11 Standard (IEEE Computer Society LAN MAN Standards Committee). Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Medium Access Control (MAC) Security Enhancements,July 2004.

[13] Wi-Fi Alliance: Glossary (PSK). http://www.wi-fi.org/knowledge_center_overview.php?type=3

[14] Y. Hu, A. Perrig, and D. B. Johnson. Ariadne: A secure On-Demand Routing Protocol for Ad hoc Networks Proceedings of the 8th An-nual International Conference on Mobile Computing and Networking (MobiCom'02), pp. 12-23, September 2002.

[15] D. Sterne, P. Balasubramanyam, D. Carman, B. Wilson, R. Talpade, C. Ko, R. Balupari, C. Tseng, T. Bowen, M. Res, et al. A General Cooperative Intrusion Detection Architecture for MANETs. Third IEEE International Workshop on Information Assurance, pages 57 -70, 2005.

[16] Fabian Hugelshofer, Paul Smith, David Hutchison, Nicholas J.P. Race. OpenLIDS: A Lightweight Intrusion Detection System for Wireless Mesh Networks. ACM MobiCom, September 2009.

[17] Y. Huang and W. Lee. A Cooperative Intrusion Detection System for Ad Hoc Networks. Proceedings of the ACM Workshop on Security in Ad Hoc and Sensor Networks, pp. 135-147, October 2003.

[18] IEEE 802.11 Standard (IEEE Computer Society LAN MAN Standards Committee). Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Frame Formats, February 2000.

[19] Cisco Systems Cisco Wireless Control System Configuration Guide, Release 5.0, February 2008. `https://www.cisco.com/en/US/docs/wireless/wcs/5.0/configuration/guide/WCS50sc.pdf`

[20] John Bellardo, Stefan Savage. 802.11 Denial-of-Service Attacks: Real Vulnerabilities and Practical Solutions 12th USENIX Security Symposium, August 2003.

[21] MADWifi. `http://madwifi-project.org/`

[22] libpcap `http://www.tcpdump.org/`

[23] PacketFu `http://code.google.com/p/packetfu/`

[24] Aircrack-ng `http://www.aircrack-ng.org/`

[25] Intel Wireless WiFi Link drivers for Linux `http://intellinuxwireless.org/`

[26] Android `http://www.android.com/`