ELA: 分散型VPNの設計と実装

青柳　禎矩[1]　　滝沢　允[2]　　斉藤　匡人[2]　　間　博人[2]
徳田　英幸[1,2]
[1] 慶應義塾大学　環境情報学部
[2] 慶應義塾大学大学院　政策・メディア研究科
{sada, makoto, masato, haru, hxt}@ht.sfc.keio.ac.jp

概 要

近年、公衆通信網上に仮想の専用通信経路を構築可能な VPN (Virtual Private Network) が注目されている。VPN は通信経路としてインターネットなどの公衆通信網を利用するため、専用回線などと比べてコストが低く、専用回線に匹敵するセキュリティの高さを持つ。そのため LAN に対して遠隔地から接続するという本来の目的だけでなく、ノード同士がインターネットを介した通信時の通信基盤として VPN を利用することも有用である。しかしノードが多数存在する場合、従来の VPN 機構では即興的な VPN は構築不可能である。

そこで本論文では、異種ネットワークセグメントに属するノード同士によって自律的に構築された P2P (Peer-to-Peer) ネットワーク上で VPN を実現し、即興的に仮想的な LAN を実現する ELA (Everywhere Local Area network) を提案する。ELA を用いることによって多数のユーザノードの通信基盤として VPN を即興的に構築でき、ユーザ間のセキュアな通信を実現する。

1　はじめに

近年、常時広域接続サービスの普及によりインターネット利用人口は増加している。一般的なユーザのインターネットの主な利用目的の一つとして他のユーザとのコミュニケーションが挙げられる。現在は電子メールや WWW における掲示板システムが主流であるが、近年の端末の高性能化およびネットワークの広帯域化により IM (Instant Messenger) やビデオチャットなどのアプリケーションが登場した。これらはユーザのノード同士が直接通信することによりリアルタイムなコミュニケーションを実現できる。

しかし、インターネットを介した通信にはセキュリティリスクが存在する。例えば途中経路における通信の傍受により、パスワードなどの重要な情報が悪意ある人物に取得されることが例として挙げられる。コミュニケーションなどのサービスにおいてセキュリティは重要である。しかし、セキュリティリスクを回避するには、コミュニケーションツールごとにセキュリティの設定をユーザが行う必要があり、この設定は手間がかかることが多い。

ユーザ間のセキュアな通信を透過的に実現する手法として VPN が挙げられる。VPN を用いると、ユーザ間通信の傍受や改竄などのリスクを大幅に抑えることができる。しかし従来の VPN 機構では、多数の VPN が存在する場合は設定が面倒だったり、高性能かつ高信頼の端末を常時設置する必要があったりなど、多数のユーザ間に VPN を形成することが困難である。

そこで、本論文ではインターネットにおけるセキュリティと手間の削減を両立する通信基盤の構築を目的とする ELA (Everywhere Local Area network) を提案する。ELA はユーザ間のセキュアな通信基盤として、VPN の構築を行う。VPN の即興的な構築のために、ノードが多数存在する場合でも VPN を自動形成することにより、ユーザは意識せず VPN による通信が可能となる。また、ELA は単一ノードの故障により VPN 全体が利用不可にならないよう、規模性および耐故障性について考慮する。本論文では ELA の設計、実装および評価を行う。

本論文は次のように構成される。本章である第 1 章では本研究の動機および目的について述べた。第 2 章では背景として VPN を概説した後、従来の VPN 機構では多数のユーザ間で即興的に VPN を構築できないことを指摘する。そしてその解決策を提案し、関連研究について述べる。第 3 章では ELA の設計について述べ、第 4 章で実装手法を述べる。第 5 章においてプロトタイプ実装した ELA の評価を行い、オーバヘッドを計測する。第 6 章で本論文をまとめ、今後の課題について展望する。

2　背景

本章では本研究の背景を述べる。まず本研究の目的の通信基盤として構築する VPN について概説する。次にその VPN を構築する機構を分類し、その特徴を述べる。最後に VPN を構築する通信プロトコルの説明を行う。

2.1 VPN の概要

従来、企業の本社と支社の拠点間で通信する場合などでは、専用回線などの物理的回線を拠点間に敷設していた。これらの手法は LAN 間通信にインターネットを経由しないため、途中経路における通信の盗聴や改竄などのセキュリティリスクがなく、プライベート IP アドレス同士の通信も可能である。しかし、これらの物理的回線を用いた手法は敷設するコストがかかる。また回線を敷設した場所のみでしか利用できず、拠点の新設や移設の場合、それに伴い物理的回線の新設や移設の必要がある。

しかし、インターネットなどの公衆通信網上において仮想の専用通信経路を構築可能な VPN が登場した。VPN は拠点間の通信回線としてインターネットを利用するため、通信回線の新規導入する必要がなくコストが小さい。また、VPN はインターネットに接続可能な環境であれば利用可能で、拠点の移動や新設にも柔軟に対応できる。

VPN は運用形態で Point-to-Point 型と Client/Server 型に分類でき、それぞれの概要と長所・短所について簡単に述べる。

- Point-to-Point 型

  拠点間を 1 対 1 で接続する VPN であり、少数の拠点を接続する場合に利用される。例えば、企業の本社と一つの支社間を接続するケース、家庭のホームネットワークへリモートアクセスするケースなどが想定される。またルーティングを行わず処理が高速であることから、拠点間接続の基幹部分のみに利用されることもある。

  しかし、多数の拠点間をフルメッシュ型のトポロジで接続する場合、拠点数を N とすると構築する VPN の数は計算式 1 によって求まる。

$$X = \frac{N(N-1)}{2} \tag{1}$$

  例えば 100 の拠点同士をフルメッシュ型で接続すると、4950 もの VPN を構築する必要がある。拠点間接続する場合、VPN をフルメッシュに接続する必要はないが、部分的にメッシュを構築するだけでも多くの VPN を構築する必要がある。VPN 一つ一つに対して IP アドレスを設定し、ルーティングテーブルに追加する必要があるため手間が増大する。このため多数の拠点を相互接続する目的には利用しにくい。

- Client/Server 型

  拠点間を 1 対多で接続する VPN であり、多数の拠点で接続する場合に利用される。例えば、企業の本社と多数の支社間を接続するケース、企業の LAN に多数の社員がリモートアクセスするケースなどが想定される。ルーティング処理は VPN 機構自身が備えているため、拠点が増加して VPN を新設する場合に手動でルーティング設定を行う必要がない。

  しかし Client 同士の通信を中継する Server が問題になる場合がある。Client 同士のトラフィックは全て Server を経由するため、Server の帯域や処理性能が十分でない場合は Server がボトルネックになる可能性がある。また Server は単一故障点であるため、もし Server が不慮の事故などにより動作しなくなると Client 間は全く通信できなくなる。このように Client/Server 型 VPN は Server に高性能と信頼性が要求されるため、Server の導入や維持が必要であるという欠点が存在する。

表 1: トンネリングプロトコルによる分類

プロトコル	VPN 機構
IPSec	L2TP [10]
GRE	PPTP [6]
TCP	SoftEther [4], TinyVPN [11], Emotion Link [1], OpenVPN [12], VTun [7]
UDP	CIPE [8], OpenVPN [12], VTun [7]

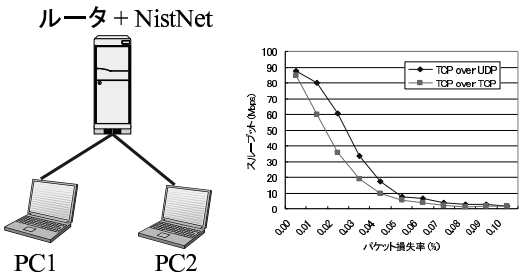
2.2 トンネリングプロトコル

VPN によって拠点間通信を行う場合、VPN は拠点間に仮想的なトンネルを形成する。そのトンネルを形成するためにトンネリングプロトコルが用いられる。VPN 機構の設計方針に応じてトンネリングプロトコルは異なる。従来の VPN 機構をトンネリングプロトコルによって分類した結果を表 1 に示す。

表 1 の中で TCP と UDP は他の一般的なネットワークアプリケーションで使用される汎用的なトランスポート層のプロトコルである。最近の Windows や Linux などの OS はネットワークアプリケーションの使用を前提に開発されており、TCP と UDP のプロトコルスタックが標準で存在する。そのため転送プロトコルに TCP や UDP を用いる VPN 機構を導入する場合、ユーザがノードやルータに新規のプロトコルスタックの導入する必要がない。

TCP はコネクション型プロトコルであるため、一度コネクションを構築すれば Firewall や NAT 環境においてもデータの送受信が可能である。UDP はコネクションレス型プロトコルであるため、Firewall や NAT 環境においては UDP データグラムを送信可能だが受信不可能という場合がある。UDP は簡素なプロトコルであるため処理が高速である。TCP は再送処理や輻輳制御などの処理が多く、また TCP over TCP [9] の問題によって利用可能帯域が減少する。

TCP と UDP のスループットの差を検証するため、VPN 機構実装の一つである OpenVPN [12] を用いてトンネリングプロトコルの違いによるスループットの違いを検証した。OpenVPN はトンネリングプロトコルに TCP と UDP を明示的に選択して



(a) 実験環境　(b) 結果

図 1: OpenVPN によるスループットの測定

利用できる。検証のため図 1(a) のような環境を用意した。ルータ (CPU: Celeron 433MHz, Memory 192MBytes, OS: Red Hat Linux 9) に NISTNet [3] を導入し、ルータで任意の割合でパケットを破棄できるようにした。ルータに 2 台の PC (CPU: Pentium M 1.7GHz, Memory 1GByte, OS: Red Hat Linux 9) を 100Base-TX で接続し、それぞれ異なるネットワークセグメントを割り当て、2 台の PC 間で OpenVPN を用いて VPN を構築した。そして Netperf を利用して TCP セグメントを発生させ、VPN 経由における 2 台の PC 間スループットを計測した。

実験結果を図 1(b) に示す。ほとんどのパケット損失率において、トンネリングプロトコルに UDP を利用した方がスループットが大きくなった。パケット損失率が 0%の場合には両者のスループットは大きく変わらない。しかしルータにおいてパケットを損失させると TCP の場合はスループットが急激に減少した。特にパケット損失率が 1, 2%の時はスループットの差が 20Mbps 以上になった。パケット損失が起きると TCP は再送処理や輻輳制御を行うため、スループットが減少するものと考えられる。

2.3 問題意識

現在多くの VPN 機構が提案され、一部の VPN 機構は実際に拠点間接続に利用されている。その目的は企業や学術機関などの LAN に存在するファイルサーバやメールサーバなどの既存のネットワークリソースを遠隔地から活用することである。しかし VPN によって構築される通信経路はセキュリティの面において優れており、その他の目的に利用することも可能である。そこで VPN によって異種ネットワークセグメントに所属するユーザのノード間の通信基盤として VPN の利用も有用だと考えられる。従来の VPN のような既存の LAN に存在するネットワークリソースの利用が目的ではなく、インターネットを介したノード同士の直接通信のセキュリティをより強化することが目的である。そのためユーザが他のユーザと通信したいと思ったときに即興的に VPN を構築できる必要がある。

しかし現在提案されている VPN 機構はユーザのノード間で即興的に VPN を構築するのは困難である。Point-to-Point 型 VPN 機構はユーザのノードを 1 対 1 で接続する。そのためノードが多数の場合は多くの VPN を構築する必要がある。もし N 台のノードをフルメッシュ型トポロジで VPN 構築する場合、$N(N-1)/2$ の VPN を構築する必要がある。また VPN セッションごとにルーティングを設定する必要がある。Client/Server 型 VPN 機構は通信を中継するサーバが必要であるため、サーバによる運用および性能の問題が発生する。

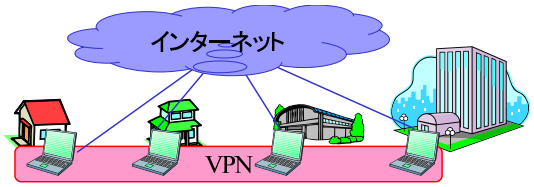


図 2: ユーザノード間の通信基盤としての VPN

そこで本論文では、P2P ネットワークトポロジの VPN 機構を各ノードが自律的に形成することにより VPN を実現する機構である ELA (Everywhere Local Area network) を提案する。ELA はインターネットを介したユーザ間の通信基盤としての利用を目的とした VPN 機構である。ELA のネットワークトポロジは単一ノードに通信が集中しない P2P 型であるため、単一ノードの故障により全てのノードが VPN による通信が不可能にならない。また、その P2P ネットワークは ELA によって自動形成される。

2.4 関連研究

ノード間通信の VPN を即興的に構築する機構として IVGMP (Internet VPN Group Management Protocol) [2] が挙げられる。ユーザの通信要求に応じて、VNOC (Virtual Network Operation Center) に IVGMP を用いて接続先および通信ポリシーを照会し、拠点間で IPsec による VPN を形成する。この機構はユーザが多数存在する場合でも VPN を自動形成し、かつ通信が単一ノードに集中しない。しかし VPN 通信開始時に必ず VNOC への照会が必要であるため、VNOC が単一故障点となる。

また、本論文では P2P トポロジによるオーバレイネットワーク上に VPN を構築することを目的としているが、オーバレイネットワーク上に VPN を構築する機構として [4], [11] が挙げられる。[4], [11] はスイッチングハブの役割をする Server に仮想的な NIC を持つ Client が接続しすることにより VPN を構築する。Server を中心としてオーバレイネットワークを形成することにより、多数のユーザ間で VPN を経由した通信が可能である。[4] は SSL、[11] は AES による通信の暗号化を行い、セキュリティを確保する。これらの機構は Client 間の通信に必ず Server を必要とするため、前節で述べた Server における性能と運用の問題が発生する。

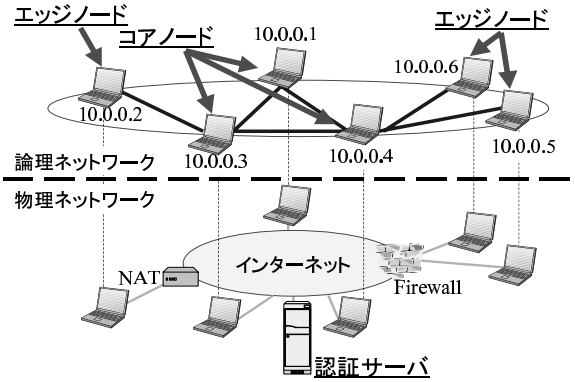


図 3: ELA によって構成される ELA-VPN のイメージ

3 ELA の設計

本章では ELA (Everywhre Local Area network) について述べる。最初に ELA の概要を述べ、次に ELA が形成するネットワークを説明する。

3.1 概要

本節では ELA (Everywhere Local Area network) を提案する。ELA は異種ネットワークセグメントに属するノード同士が P2P ネットワークを形成し、即興的に仮想的な LAN を構築する。その仮想的な LAN はノード同士のセキュアな通信基盤として利用可能で、本論文では ELA-VPN と定義する。ELA-VPN のイメージを図 3 に示す。TCP および UDP による通信の送受信可能なノードをコアノードとし、コアノード以外の NAT や Firewall などによって通信に制限があるノードをエッジノードとする。コアノードがメッシュネットワークを形成し、エッジノードがそれらのコアノードに接続するという規則に従い、P2P ネットワークトポロジを自動的に形成する。

ELA によって構築される VPN は既存のネットワークセグメントから独立した仮想的なプライベートネットワークである。他のノードによって ELA-VPN におけるプライベート IP アドレスが各ノードに割り当てられる。

ELA の想定シナリオとして以下があげられる。どちらのシナリオも利用ノードは 30 台までを想定する。

- ユーザ間通信の透過的な暗号化

  近年はサーバを介さずに、ユーザのノード同士が直接通信するアプリケーションが増加している。その例として IM (Instant Messenger) やビデオチャット、一部のネットワークゲームなどが挙げられる。これらのアプリケーションは通信経路の暗号化などを正しく設定しないと、途中経路で傍受される危険性がある。

  そこで ELA を利用してユーザのノード間で即興的に VPN を構築し、各アプリケーションによる通信を透過的に暗号化する。

- LAN 用アプリケーションの利用

  閉塞的なネットワークである LAN での使用を前提としているアプリケーションがが存在する。Windows Network や Doom [5] などの一部のネットワークゲームは、通信相手のノードの探索にブロードキャストを用いる。そのためこれらのアプリケーションは互いのネットワークセグメントが異なると使用できず、インターネットを介した利用は不可能である。NFS (Network File System) は同一ネットワークセグメントに所属するノードから利用され、通信経路は盗聴の危険がなく信頼可能という前提で設計が行われている。そのため通信経路における暗号化などが行われず、インターネット経由の利用はセキュリティの面から利用できない。

  そこで ELA を利用してユーザのノード間で即興的に VPN を構築し、LAN での利用を目的としたアプリケーションをインターネット経由で利用可能にする。

3.2 動作手順

本小節では P2P ネットワーク形成から VPN による通信に至るまでの、ELA の動作手順について説明する。

3.2.1 ノードの参加準備

1. ノードの発見

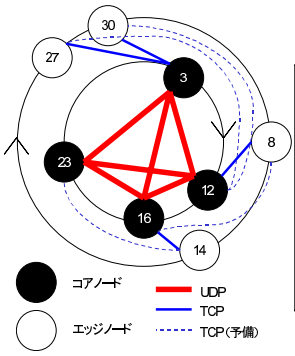
   まず既に ELA-VPN に参加しているノードを発見する。ノードの発見機能は ELA に組み込まれていない。ユーザは WWW やメールなどを利用してすでに ELA-VPN を構築しているユーザを発見し、インターネットにおけるノードの IP アドレスを訊ねる。まだ ELA-VPN を構築しているユーザがいなければ自身だけの ELA-VPN を構築し、他のノードから接続される準備を行う。**2.** を飛ばして **3.** にいく。

2. ユーザ認証

   ELA-VPN にノードが参加する場合、ELA-VPN に既に参加しているノードからユーザ認証をうける。ELA-VPN の参加ノードは参加が認可されたユーザのリストを共有しており、そのリストを用いて認証を行う。

3. プライベート IP アドレスの割り当て

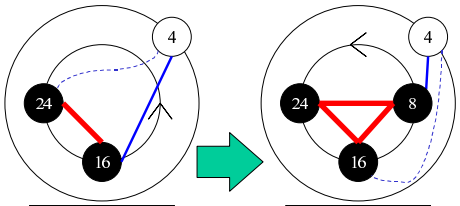
   ELA-VPN におけるプライベート IP アドレスを割り当てる。新規参加するノードが特定のプライベート IP アドレスを利用したい場合、ELA-VPN の既存のノードと IP アドレスが重複しなければその IP アドレスを使用できる。重複があった場合、あるいは利用したい IP アドレスがない場合、利用されていないプライベート IP アドレスの中からランダムに決定する。



ルーティングテーブル

Key	Value
3	3のIPアドレス
8	12のIPアドレス
12	12のIPアドレス
14	16のIPアドレス
16	16のIPアドレス
23	23のIPアドレス
27	3のIPアドレス
30	3のIPアドレス

図 4: ELA の P2P ネットワーク



Key	Value
4	16のIPアドレス
16	16のIPアドレス
24	24のIPアドレス

Key	Value
4	16のIPアドレス
8	8のIPアドレス
16	16のIPアドレス
24	24のIPアドレス

図 5: コアノード参加時

4. ノードの分類

   各ノードはコアノードとエッジノードに分類される。コアノードは他のノードから TCP セッションを確立可能で、UDP によるデータグラムの送受信が可能である。エッジノードはコアノード以外のノードである。

3.2.2 P2P ネットワークの形成

ELA-VPN の P2P ネットワークの例を図 4 に示す。ELA-VPN におけるプライベート IP アドレスのハッシュ値を ID とし、P2P ネットワークでは ID の小さい順に時計回りに配置される。P2P ネットワークの論理的に中央の位置にコアノードが配置され、その周りをエッジノードが囲む。

コアノード間は UDP で通信を行い、エッジノードはコアノードに TCP セッションを確立して通信を行う。各エッジノードが TCP セッションを確立する相手のコアノードを、そのエッジノードの「親」と呼ぶ。反対に各コアノードに TCP セッションを確立しているエッジノードを、そのコアノードの「子」と呼ぶ。例えば ID 30 の親は ID 3 のコアノード、ID 3 の子は ID 27 と ID 30 のエッジノードとなる。エッジノードが TCP セッションを確立するコアノードは、そのエッジノードの時計回りに ID が一番近いコアノードである。

コアノードが ELA-VPN から離脱した場合、そのコアノードの子も ELA-VPN から離脱してしまう。それを防ぐため、各エッジノードは親の次に ID が時計回りに近いコアノードに予備の TCP セッションを確立しておく。あるエッジノードの親が ELA-VPN から離脱した場合、今まで予備の TCP セッションを確立していたコアノードが親となり、エッジノードは新しく他のコアノードに対して予備の TCP セッションを確立する。コアノードが正規の終了手続きを経て ELA-VPN を離脱する場合は、エッジノードは即座に親のコアノードを切り替えることができる。コアノードがネットワークトラブルにより突然 ELA-VPN を離脱する場合は、TCP セッションが明示的に切断されないため、エッジノードは通信のタイムアウト等から親が離脱したと判断する。

3.2.3 データ転送

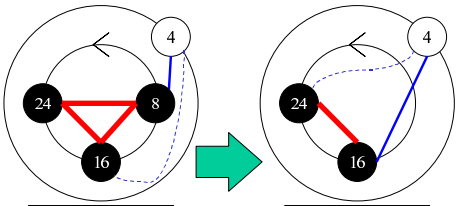
ELA の P2P ネットワークはフルメッシュ型ではないため、論理的に直接接続していないノード同士は他のノードを経由して通信する。そのため経由ノードを動的に決定するルーティング処理が必要である。

ルーティング機構によって決定される途中経路は発信元ノードと宛先ノードによって異なる。コアノード同士は UDP を用いて直接通信し (例：3 ⇔ 16)、コアノードと子のエッジノードは TCP セッションを用いて直接通信する (例：16 ⇔ 14)。これら以外の場合は、上記のパターンを組み合わせる。例えばコアノードと子ではないエッジノードと通信する場合、他のコアノードを経由して通信する (例：3 ⇔ 16 ⇔ 14)。エッジノード同士が通信する場合、コアノードを経由して通信する (例：27 ⇔ 3 ⇔ 16 ⇔ 14)。

コアノードは他のコアノードか子のエッジノードに通信内容を転送するので、コアノードは通信内容の宛先によって転送先のノードを決定する必要がある。そのため各コアノードは二つのテーブルを保持する。一つは ELA-VPN に参加する全てのノードに関するルーティングテーブルで、*key* には ELA-VPN における IP アドレスのハッシュ値、*value* にはコアノードの場合はそのコアノードのインターネットにおける IP アドレス、エッジノードの場合は親のコアノードのインターネットにおける IP アドレスが格納される。このテーブルは全コアノードで共有する。もう一つは各コアノードの子に関するテーブルで *key* にはエッジノードの ID、*value* には TCP セッションの識別子が格納されている。コアノードは他のノードからデータを転送された場合、これらのテーブルを参照して他のノードへ転送する必要がある場合はそのノードへ転送する。エッジノードは P2P ネットワークの論理的に外側に位置するため、ルーティングに関して考慮する必要がない。

3.2.4 P2P ネットワークの維持

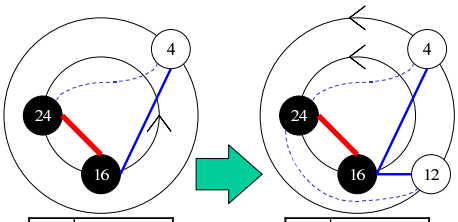
各ノードはユーザの都合に応じて ELA-VPN へ参加および離脱する。その度にルーティングテーブルを修正し、P2P ネットワークにおいて正しくデータ転送できよう維持する。



Key	Value
4	16のIPアドレス
8	8のIPアドレス
16	16のIPアドレス
24	24のIPアドレス

Key	Value
4	16のIPアドレス
16	16のIPアドレス
24	24のIPアドレス

図 6: コアノード離脱時



Key	Value
4	16のIPアドレス
16	16のIPアドレス
24	24のIPアドレス

Key	Value
4	16のIPアドレス
12	16のIPアドレス
16	16のIPアドレス
24	24のIPアドレス

図 7: エッジノード参加時

コアノードが参加する例を図 5 を用いて示す。ELA-VPN に新規に ID 8 のコアノードが参加するものとする。ID 8 のコアノードは他のコアノードからルーティングテーブルを取得し、自分自身の情報を追加する。ID 8 のコアノードはルーティングテーブルを参照して、本来自分に対して TCP セッションを確立すべき ID 4 のエッジノードを確認すると、親のコアノードを変更するよう ID 4 のエッジノードへ通達する。ID 16 のコアノードに TCP セッションを確立している ID 4 のエッジノードは、ID 8 のコアノードに対して TCP セッションを確立し、今まで ID 16 に対して確立していた TCP セッションは予備用となり、ID 24 に対して確立していた予備用 TCP セッションは終了する。

コアノードが離脱する例を図 6 を用いて示す。ID 8 のコアノードが離脱するものとする。ID 8 のコアノードは子の ID 4 のエッジノードに対して TCP セッションの確立先の変更を要請する。これにより ID 4 のエッジノードは今まで ID 16 に対して確立していた予備用 TCP セッションを本来の TCP セッションとし、ID 24 のコアノードに対して予備用 TCP セッションを確立し、今まで ID 8 に対して確立していた TCP セッションは終了する。そしてルーティングテーブルから自分の情報を削除し、ELA-VPN から離脱する。

エッジノードが参加する例を図 7 を用いて示す。ELA-VPN へ新規に ID 12 のエッジノードが参加するものとする。ID 12 のエッジノードは任意のコアノードに対して、自分はどのコアノードに TCP セッション、予備用 TCP セッションを確立すべきかを問い合わせ、実際に TCP セッションを確立する。そして親になったコアノードはエッジノードをルーティングテーブルに追加する。

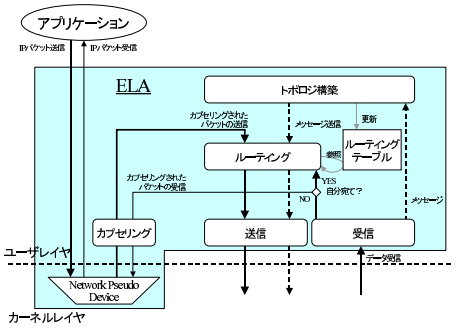


図 8: ELA のシステム構成

4 実装

ELA のモジュール構成図を図 8 に示す。影の付いたモジュールが ELA によって提供される機能で、点線より上がユーザレイヤのソフトウェアとして動作、下がカーネルレイヤで動作する。

あるアプリケーションが他のノードと ELA-VPN 経由で通信する時、モジュール間の連携は次のようになる。アプリケーションが ELA-VPN におけるノード宛ての IP パケットを送信すると、NPD (Network Pseudo Device) に IP パケットが渡される。カプセリングモジュールは NPD から IP パケットを取得する。カプセリングモジュールは ELA ヘッダを付加して IP パケットを暗号化を行い、ルーティングモジュールに渡す。ルーティングモジュールはルーティングテーブルを検索してデータの転送先ノードを決定し、送信モジュールに渡す。送信モジュールは受け取ったデータを、ルーティングモジュールからの指示に従って他のノードへ転送する。

反対に ELA-VPN 経由でカプセリングされた IP パケットを受信する流れは次のようになる。他のノードからカプセル化されたパケットを受信した受信モジュールは ELA ヘッダを参照して自分宛だと判断するとカプセリングモジュールに転送する。カプセリングモジュールはヘッダの削除後に復号化を行い、NPD に渡す。NPD は IP ヘッダの宛先ポートを参照して適切なアプリケーションに転送する。

各モジュールの概要を以下に示す。

Network Pseudo Device

NPD はネットワーク仮想デバイスで、ノードが ELA-VPN を経由して通信する場合のネットワークデバイスとして利用される。NPD は OS から通常の

ネットワークデバイスと同じように扱われ、IP アドレスやネットマスクなどを割り当て可能である。各ノードに割り当てられる ELA-IP アドレスも、NPD に設定される。NPD は後述するカプセリングモジュールからパケットを受信し、カプセリングモジュールへパケットを送信する。

トポロジ構築モジュール

ELA-VPN における P2P ネットワークの形成および維持を行う。VPN 構築のために、以下の機能がある。

- ユーザ認証

  悪意あるノードが成りすましをして ELA-VPN に参加することを防止するため、ノードの ELA-VPN 参加時に行うユーザ認証の機能を提供する。

- ELA-VPN における IP アドレスの決定

  新規参加ノードに対し、ELA-VPN におけるプライベート IP アドレスを割り当てる。新規参加ノードが特定の IP アドレスを利用したい場合、その IP アドレスが既に利用されていないかルーティングテーブルを参照して調べる。もしその IP アドレスが利用されていなければ、新規参加するノードに対してその IP アドレスが利用できることを伝達する。すでにその IP アドレスが利用されていた場合、もしくは特に利用したい IP アドレスがない場合、ルーティングテーブルを参照して利用されていないプライベート IP アドレスの中からランダムに決定し、その IP アドレスを新規参加するノードに対して伝達する。

- ノードの分類

  IP アドレスが決定した新規参加ノードをコアノードかエッジノードに分類する。新規参加ノードに対して TCP セッションの確立、UDP データグラムの送信を行い、その試行が共に成功した場合はその新規参加ノードをコアノードとする。それ以外の場合はその新規参加ノードをエッジノードとする。

P2P ネットワークを維持するため、以下の機能を持つ。

- 他ノードとのメッセージの送受信

  他ノードと情報などを交換するため、メッセージを用いた情報の共有を行う。メッセージは XML (eXtended Markup Language) 形式でやりとりされる。

- ルーティングテーブルの更新

  ELA-VPN でのノードの参加および離脱が発生するたびに、トポロジ構築モジュールはルーティングテーブルの更新を行う。

ルーティングモジュール

カプセリングモジュールあるいは受信モジュールから渡されたデータの転送先を決定する。転送データの ELA ヘッダを参照し、転送先ノードをルーティングテーブルより決定する。このモジュールはコアノードのみ使用する。

ELA ヘッダには ELA-VPN における宛先 IP アドレスが格納されている。ルーティングテーブルには各ノードの情報が、ELA-VPN における IP アドレスを key として、インターネットにおける IP アドレスを $value$ として格納されている。ルーティングモジュールは ELA ヘッダの宛先 IP アドレスを key として、ルーティングテーブルから検索を行う。

検索の結果、宛先 IP アドレスが他のノードだった場合、その IP アドレスのノードへ転送するよう送信モジュールへ伝達する。宛先 IP アドレスが自身だった場合、その IP アドレスを使用している子のエッジノードへ転送するため送信モジュールへ伝達する。該当するエントリが存在しなかった場合、宛先不明のデータとして破棄する。

カプセリングモジュール

ELA ヘッダの処理と通信の暗号化・復号化を行う。

NPD から IP パケットを渡された場合、IP パケットの先頭に ELA ヘッダを付加し、ELA のバージョン情報や ELA-VPN における宛先 IP アドレスなどをヘッダに格納する。次に IP パケットをトポロジ構築モジュールから渡された共通鍵を用いて暗号化を行う。受信モジュールからカプセリングされた通信内容を渡された場合、ELA ヘッダの除去および復号化を行い、NPD へ IP パケットを渡す。

送信モジュール

カプセリングされたパケットを、ルーティングモジュールから指示されたノードへ転送する。他ノードへの転送プロトコルとしてコアノードは TCP あるいは UDP、エッジノードは TCP を用いる。コアノードは転送先ノードに応じて転送プロトコルの使い分けを行う。

受信モジュール

他ノードからカプセリングされた通信内容やメッセージを受信する。メッセージはトポロジ構築モジュールへ渡し、カプセリングされた通信内容は宛先 IP アドレスが自分であればカプセリングモジュールへ、そうでなければルーティングモジュールへ渡す。

コアノードは他のノードから TCP セッションを確立される。TCP セッション確立の待機も受信モジュールによって行われる。

5 評価実験

ELA によるオーバヘッドを評価するため、ELA による通信の遅延を測定した。そのために、次の環境を用意した。ルータ (CPU: Celeron 433MHz, Memory 192MBytes, OS: Red Hat Linux 9) 下に 2 台の PC (CPU: Pentium M 1.7GHz, Memory 1GByte, OS: Red Hat Linux 9) を 100Base-TX で接続し、それぞれ異なるネットワークセグメントを割り当てた。そして以下のケースを想定し、`ping` を用いて RTT (Round Trip Time) を 512 回測定した。

1. ELA-VPN を構築して 2 ノードともコアノードの場合 (UDP による転送)
2. ELA-VPN を構築してそれぞれコアノードとエッジノードの場合 (TCP による転送)
3. ELA-VPN を利用しない場合の RTT

表 2: 2 ノード間における遅延 (単位 msec)

	平均	標準偏差
1. コアノード同士	3.74	2.91
2. コアノードとエッジノード	4.02	2.55
3. ELA なし	0.22	0.04

表 2 は **1.** と **2.** が **3.** のケースよりも遅延の平均が大きく、かつ分散しており、**1.** と **2.** に大きな違いはないことを示している。遅延の増加の原因として ELA によるオーバヘッドが存在することが挙げられる。ELA はカーネルレイヤの通信内容を一度ユーザレイヤに引き上げ、ルーティングや暗号化処理等を施してから再びカーネルレイヤに戻すため、オーバヘッドが生じて遅延が発生する。

6 まとめと今後

本論文では異種ネットワークセグメントに属するノード間で P2P ネットワーク型の VPN の構築を実現する ELA を提案した。ELA はインターネットに接続するユーザのノード同士が直接通信する時における、セキュアな通信基盤としての VPN を構築することを目的とする。上記の目的の VPN を構築するには、従来の VPN 機構は多くの手間やコストを必要とするが、ELA は即興的に VPN を構築可能である。また ELA のプロトタイプ実装を行い、VPN によるオーバヘッドの測定を行った。

ELA を用いることにより、インターネットを介したユーザ同士のよりセキュアな通信を可能にする。昨今の技術革新によりノードの処理性能はさらに高速化し、ネットワークはさらに広帯域化することが予想される。それに伴い、ユーザのノード同士が直接通信するネットワークアプリケーションも、今後さらに多く登場すると考えられる。そのような状況下において ELA はますます重要性が高まるであろう。

今後の課題としてトポロジ構築モジュールとルーティングモジュールの実装、QoS の考慮、NAT 越え技術を用いたコアノードの増加などが挙げられる。

Acknowledgement

この研究は総務省「ユビキタスネットワーク制御・管理技術の研究開発 (ubila プロジェクト)」の一部として行なわれた。

参考文献

[1] Emotion link: http://www.freebit.com/solution/emotion.html.

[2] Lina Alchaal, Vincent Roca, and Michel Habert. Offering a Multicast Delivery Service in a Programmable Secure IP VPN Environment. *In proceedings NGC 2002*, October 2002.

[3] Mark Carson and Darrin Santay. Nist net: a linux-based network emulation tool. *In Proceedings SIGCOMM Comput. Commun. Rev.*, Vol. 33, No. 3, pp. 111–126, 2003.

[4] Daiyu Nobori. SoftEther. http://www.softether.com/.

[5] id Software. Doom. http://www.idsoftware.com/, 1993.

[6] K. Hamzeh, G. Pall, W. Verthein, J. Taarud, W. Little, G. Zorn. Point-to-Point Tunneling Protocol (PPTP), RFC 2637, July 1999.

[7] Maxim Krasnyansky. Virtual Tunnel (VTun). http://vtun.sourceforge.net/.

[8] Olaf Titz. Crypto IP Encapsulation (CIPE). http://sites.inka.de/sites/bigred/devel/cipe.html.

[9] Olaf Titz. Why TCP Over TCP Is A Bad Idea. http://sites.inka.de/bigred/devel/tcp-tcp.html.

[10] W. Townsley, A. Valencia, A. Rubens, G. Pall, G. Zorn, B. Palter. Layer Two Tunneling Protocol "L2TP", RFC 2661, August 1999.

[11] Akio Yamamoto. TinyVPN. http://www.shimousa.com/tv/.

[12] James Yonan. OpenVPN. http://openvpn.sourceforge.net/.