# ELA: A Fully Distributed VPN System over Peer-to-Peer Network

Sadanori Aoyagi[1] Makoto Takizawa[2] Masato Saito[2] Hiroto Aida[2]
Hideyuki Tokuda[1,2]
[1]Faculty of Environmental Information
[2]Graduate School of Media and Governance
Keio University, 5322 Endo, Fujisawa, Kanagawa 252-8520, Japan
{sada, makoto, masato, haru, hxt}@ht.sfc.keio.ac.jp

## Abstract

*In this paper, we propose a fully distributed VPN system over peer-to-peer(P2P) network called Everywhere Local Area network (ELA). ELA enables to establish private overlay network for VPN among nodes of a group without any servers. As opposed to the existing VPN systems, nodes of a group can build VPN without setting up a VPN server, and there is no problem of a single-source bottleneck and a single point of failure. Though it is known that VPN system using TCP as tunneling protocol does not work well [6],there are some nodes which can use only TCP because of NAT or Firewall. Therefore each node uses both UDP and TCP appropriately depending on the situation in ELA. The topology of ELA-VPN mitigates performance deterioration. We implemented a proto-type of ELA on Linux, and show result of experimental latency between two nodes.*

## 1 Introduction

Local Area Networks (LANs) are constructed at many places such as corporations or universities to support a sharing of knowledge and cooperative work smoothly. A Virtual Private Network (VPN) enables a private connection to a LAN through a public network such as the Internet. With a VPN, data is sent between two nodes across a public network in a manner that emulates a dial-link. There are two types of VPN systems, one is used for connecting LANs across the Internet, and the other is used to connect a remote node to a LAN across the Internet.

These VPN systems, however, are not suitable for groups that do not own their own LAN or when nodes of the group are geographically dispersed. Because these VPN systems require a remote access VPN server, it takes a lot of trouble with setup and management of the server. All traffic go through the server, and they have problem with traffic congestion and a single point of failure. Therefore, the system that establishes the dedicated VPN without a remote access VPN server among geographically dispersed nodes is absolutely necessary.

We propose Everywhere Local Area network (ELA) that is a fully distributed VPN system over peer-to-peer (P2P) network. ELA enables to establish a fully distributed VPN called ELA-VPN without a remote access VPN server when nodes communicate with each other. The P2P network of ELA is classified into Core-Group and Edge-Group in consideration of transport protocol, which realizes the efficient routing over P2P network. ELA enables to use existing network softwares for LAN without any modification of them.

This paper is organized as follows. In Section 2, we discuss existing VPN systems and define the goal of ELA. In Section 3, we illustrate the design of ELA and the network established by ELA. In Section 4, the detailed implementation of ELA is described. In Section 5, we experiment with a proto-type of ELA, and show the result of it. Finally, we state our conclusions and discuss further work in Section 6.

## 2 Related Work

SoftEther [3] and TinyVPN [7] emulate Ethernet hub and network interface card (NIC), and each node with a virtual NIC installed connects to the node with a virtual hub like Figure 1 (a). Both systems require a node which provides the functionality of virtual Ethernet hub, thus there is the problem with a single-source bottleneck and a single point of failure. In addition, both systems always use TCP to carry encapsulated Ethernet frame, and there are always the problem of performance deterioration because of TCP over TCP [6]. In contrast, ELA does not have the problem with a single-source bottleneck and a single point of failure because there is no server. Each node of ELA-VPN uses UDP and TCP appropriately depending on the situation, thus the performance deterioration because of TCP over TCP is mitigated.

Proceedings of the 2005 Symposium on Applications and the Internet (SAINT'05)
0-7695-2262-9/05 $ 20.00 IEEE

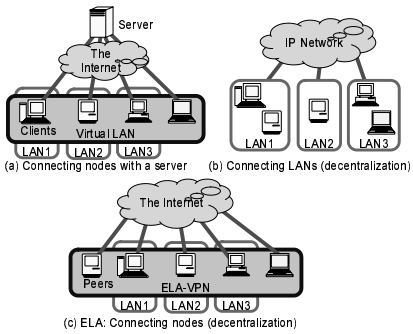


**Figure 1.** Topology of Existing VPNs and ELA

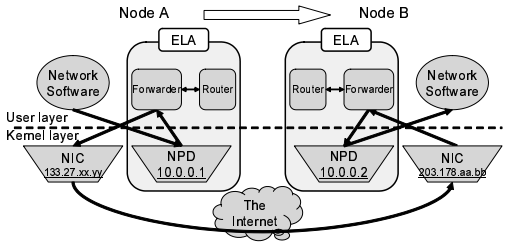


**Figure 2.** The image of ELA system architecture and the flow of an encapsulated IP packet on overlay network

IVGMP(Internet VPN Group Management Protocol) [1] is a protocol with which VPRN is constructed on the Internet. Two or more LAN and node are connected with IPsec mutually. The policy concerning security is managed collectively with the server that is called VNOC(Virtual Network Operation Center). However, the topology of VPN is not defined. Router of the network which the nodes belong to must correspond to IPsec.

## 3 Design of ELA

### 3.1 Overview

Exactly what tailscale topology looks like?

Nodes of ELA, deployed at various locations on the Internet, establish an application-layer P2P network for VPN. The VPN established by ELA is called ELA-VPN, and the image of ELA-VPN is shown at Figure 1 (c). IP packets from various network softwares are encrypted, and they are forwarded to the node of destination over ELA-VPN. The dedicated IP address for ELA-VPN is a private IP address like 10.0.0.1, and they are assigned randomly to every node within ELA-VPN. When a node communicates with other node over ELA-VPN, the node specifies the dedicated IP address of ELA-VPN as the destination IP address of any network software.

### 3.2 Software Architecture

The conceptual design of ELA, shown in Figure 2, is simple. ELA is composed of forwarder, router, and Network Pseudo Device (NPD).

Each node in ELA-VPN communicates through its own NPD as a node communicates through a network device on the Internet. And any network software can use NPD as a normal network device. Interface name of NPD is `ela0`, and it is possible to assign IP address for ELA-VPN like 10.0.0.1 to NPD. If a node has IP address that it wants to use, the node can use that address unless that address is already used by other node. If not, the node is assigned an IP address which is not used by other nodes. For IP packets within ELA-VPN that go through NPD, routing table is configured as follows.

```
Destination  Gateway       Genmask         Flags  Iface
10.0.0.0     *             255.255.255.0   U      ela0
133.27.xx.0  *             255.255.254.0   U      eth0
127.0.0.0    *             255.0.0.0       U      lo
default      133.27.xx.yy  0.0.0.0         UG     eth0
```

Forwarder and router are a kind of application software. Forwarder of each node consults with its router to determine the next node, and forwards encapsulated IP packets after encrypting it. Routing mechanism of ELA-VPN is described in the following subsection in more detail. To bootstrap ELA-VPN, a new node needs to know host name or IP address of at least one node in ELA-VPN. The new node is classified into Core-Node or Edge-Node which are described in the next subsection, and it makes its own NPD and assign unique IP address to it.

Figure 2 shows the image of IP packet flow.

**send** When a network software of Node A sends an IP packet whose destination is 10.0.0.2, the IP packet goes through NPD. NPD forwards it to forwarder. Forwarder asks router the node to forward (Node B in this case), encrypts the IP packet, and forwards it to Node B through network device.

**receive** When forwarder of Node B receives the IP packet, forwarder decrypts it, and sends it to NPD of Node B because the destination of IP packet equals to 10.0.0.2. NPD forwards it to the network software of Node B.

### 3.3 Network Topology and Routing

Nodes of ELA-VPN are classified into two groups, Core-Group and Edge-Group. Core-Group is internal side of ELA-VPN, and Edge-Group is external side of ELA-VPN. The criterion of which group nodes belong to is whether to use User Datagram Protocol (UDP) or Transmission Control Protocol (TCP) to communicate with other node. Gen-

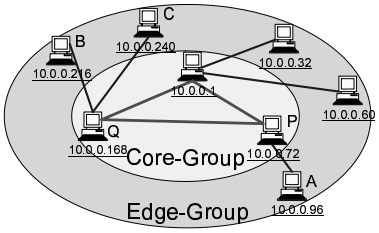


**Figure 3.** Network Topology of ELA-VPN. CNs are in Core-Group, and ENs are in Edge-Group

erally as transport protocol, UDP and TCP are used by various network softwares. It is known that VPN systems using TCP as tunneling protocol do not work well because of long delays and frequent connection aborts. OpenVPN [9] and VTun [4] support either TCP or UDP as tunneling protocol, but they are recommended to use UDP because of that reason. Some nodes, however, cannot use UDP to communicate because of Network Address Translation (NAT) or Firewall. Those nodes only use TCP to communicate.

Therefore, each node of ELA-VPN uses UDP and TCP appropriately depending on the situation. The node which can use both UDP and TCP to communicate with others is Core-Node (CN). The node which uses only TCP to communicate with others is Edge-Node (EN). Because CN can use UDP to communicate with other CNs, CN assumes the role of routing on ELA-VPN as shown in Figure 3. This topology mitigates performance deterioration because TCP over TCP does not affect entire ELA-VPN. Each EN connects to one CN whose IP address is nearest and smaller than the EN's one. Each EN also connects to another CN whose IP address is nearest and smaller than the CN's one as backup. If CN goes down, these ENs change the main CN and connect to another CN as backup. Each CN uses UDP to communicate with other CNs in one hop. The topology of Core-Group is the mesh [8].

There are four patterns of hop based on whether sender and receiver are CN or EN.

- $CN \rightarrow CN : CN \rightarrow CN$
  For instance, $P \rightarrow Q$.
- $CN \rightarrow EN : CN(\rightarrow CN) \rightarrow EN$
  For instance, $Q \rightarrow P \rightarrow A$ or $P \rightarrow A$.
- $EN \rightarrow CN : EN(\rightarrow CN) \rightarrow CN$
  For instance, $A \rightarrow P \rightarrow Q$ or $A \rightarrow P$.
- $EN \rightarrow EN : EN \rightarrow CN(\rightarrow CN) \rightarrow EN$
  For instance, $A \rightarrow P \rightarrow Q \rightarrow B$ or $B \rightarrow Q \rightarrow C$.

Routing of ELA-VPN is very simple. Each node changes the process of routing based on whether the node is CN or EN. In the case of EN, the process is very simple. Since EN has only one TCP connection to CN, EN always forwards encapsulated IP packets to CN. In the case of CN, the process is more complicated than the case of EN. **1)** CN looks for EN whose IP address on ELA-VPN equals to the destination of encapsulated IP packet from the list of ENs which belongs to that CN. **2)** If it is not found in the list of ENs, CN looks for the node from the list of CNs which includes itself. CN forwards to the other CN if found. **3)** If it is not found in the list of CNs and ENs, CN looks for CN whose IP address on ELA-VPN is nearest and smaller than IP address of the destination of encapsulated IP packet from the list of CNs. If the CN is itself, CN abandons the encapsulated IP packet because the node of destination does not exist. If not, CN forwards the packet to that CN.

## 4 Implementation of ELA

We have implemented a proto-type of ELA on Linux kernel 2.4.20. This proto-type assumes the static number of nodes. Forwarder and router are implemented as user-level application softwares with C language (approximately 1150 lines in length), and forwarder uses Blowfish [2] algorithm for encryption. NPD uses Universal TUN driver [5] which is software network device for IP tunneling.

### 4.1 Network Pseudo Device

NPD receives IP packets instead of a physical media. And NPD sends IP packets to forwarder instead of a physical media. NPD is implemented at kernel-level of UNIX OS, and it is possible to assign IP address, broadcast address, network mask, gateway address, and routing entry. Forwarder uses system call `read` when forwarder receives the whole IP packet from NPD. And forwarder uses system call `write` when forwarder sends the whole IP packet from NPD. NPD is shown at the left bottom of Figure 4.

### 4.2 Forwarder and Router

Forwarder, shown at the left top of Figure 4, carries encapsulated IP packets between NPD and other nodes' forwarder. For using UDP and TCP to communicate with other nodes, we have implemented the forwarder using socket API. Forwarder receives the encapsulated IP packets from NPD and other nodes' forwarder through the node's own network device (1). Forwarder encrypts the payload of encapsulated IP packets which are from NPD. Forwarder checks the destination field of an encapsulated IP packet. When it equals to the node's own IP address (2a), forwarder decrypts an encapsulated IP packet, and forwards it to NPD. When not (2b), forwarder asks the next node to forward of

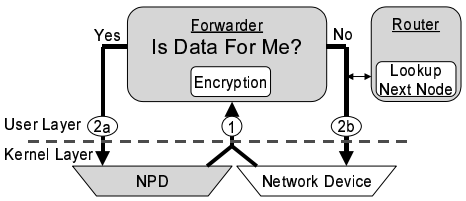


**Figure 4.** The image of implementation of ELA

router, and forwarder forwards an encapsulated IP packet to other node's forwarder by using UDP or TCP.

Router, shown at the right top of Figure 4, returns the next node to forwarder. Router changes the process based on whether the node is CN or EN. In the case of EN, router always returns the file descriptor of TCP connection to CN that the EN connects to. In the case of CN, router accommodates CN list and EN list. CN list includes mapping of IP address in ELA-VPN and IP address in the Internet of all CNs. EN list includes mapping of IP address in ELA-VPN and file descriptor of TCP connection of ENs which connects to the CN. Router looks for the next node from CN list and EN list by using the algorithm which is described in Section 3.3. And router returns IP address of other CN or the file descriptor of TCP connection to EN, or router suggests to abandon of the encapsulated IP packet.

## 5 Experimentation

To investigate latency caused by ELA, we have conducted the following experiment. We prepared two notebook PCs (CPU: Pentium M 1700MHz, Memory: 1024MB, Ethernet Card: 100Base-TX), and those two nodes are connected to the same Ethernet hub. We used `ping` 512 times to measure Round Trip Time (RTT) between two nodes, and evaluated the average and standard deviation of each in the three cases. **1)** Two nodes build ELA-VPN, and they run as CNs. **2)** Two nodes build ELA-VPN. One node runs as CN, and the other runs as EN. **3)** Measurement without ELA.

**Table 1. Latency between two nodes.**

| | Mean (msec) | Std Dev (msec) |
|---|---|---|
| 1) CNs | 3.74 | 2.91 |
| 2) CN and EN | 4.02 | 2.55 |
| 3) without ELA | 0.22 | 0.04 |

Table 1 shows that **1)** and **2)** have longer latency and varied more than **3)**, and there is not much difference between latencies of **1)** and **2)**. The cause of longer latency is the need for additional processing, which are transmitting IP packets via user layer software, routing over ELA-VPN, and encrypting.

## 6 Conclusions and Future Work

We have proposed ELA that is a fully distributed VPN system over P2P network. ELA enables to establish a fully distributed VPN called ELA-VPN without any VPN servers when nodes of a group communicate with each other for cooperative work. The node of ELA-VPN is classified into CN and EN by the presence of the limitation of communications such as NAT and Firewall. CNs are internal side of ELA-VPN, and CNs communicate with other CNs directly using UDP. Each EN connects to one CN using TCP, and each EN communicates with other nodes via the CN. As ELA do routing on ELA-VPN automatically, users don't need whether each node is CN or EN. Each nodes are assigned private IP address automatically,

Our future work includes an implementation of ELA which includes new routing mechanism to enhance scalability without increasing latency and overhead. Furthermore, we evaluate the performance and scalability of it.

## Acknowledgement

This work has been partially conducted in Ubila Project by Ministry of Internal Affairs and Communications. We thank the anonymous reviewers for their constructive comments which helped to improve the quality of this paper.

## References

[1] L. Alchaal, V. Roca, and M. Habert. Offering a Multicast Delivery Service in a Programmable Secure IP VPN Environment. *In proceedings NGC 2002*, October 2002.

[2] Bruce Schneier. *Applied Cryptography: Protocols, Algorithms, and Source Code in C, Second Edition*. John Wiley & Sons, Inc, 1996.

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

[4] M. Krasnyansky. Virtual Tunnel (VTun). http://vtun.sourceforge.net/.

[5] M. Y. Maxim Krasnyansky. Universal tun/tap driver. http://vtun.sourceforge.net/tun/, 1999.

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

[7] A. Yamamoto. TinyVPN. http://www.shimousa.com/tv/.

[8] S. S. H. Z. Yang-hua Chu, Sanjay G. Rao. Enabling conferencing applications on the internet using an overlay multicast architecture. *In Proceedings ACM SIGCOMM*, 2001.

[9] J. Yonan. OpenVPN. http://openvpn.sourceforge.net/.