A ROBUST VPN IMPLEMENTATION BASED ON RESILIENT OVERLAY NETWORK CAPABILITIES

By

AZHAR JAVED

BURHAN UL HAQ

MUHAMMAD RIZWAN

(BESE 10)

SUBMITTED TO THE FACULTY OF COMPUTER SCIENCE DEPARTMENT MILITARY COLLEGE OF SIGNALS, NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY, RAWALPINDI, IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF BE IN COMPUTER SOFTWARE ENGINEERING

MARCH 2008

ABSTRACT

A ROBUST VPN IMPLEMENTATION BASED ON RESILIENT OVERLAY NETWORK CAPABILITIES

BY

AZHAR JAVED
BURHAN UL HAQ
MUHAMMAD RIZWAN

Typical Virtual Private Network (VPN) is a private data network over the public network i.e. Internet and inherits all the underlying network problems like slow link failure recovery, inability to detect path or performance failures and enforcement of blunt policy expression.

Resilient Overlay Network (RON) seeks to quickly detect and respond to network failures by aggressively probing the paths connecting its nodes. Routing based on application specific metrics ensures tighter integration with applications and an expressive policy routing. We have implemented a software application that runs as a service on each RON node to provide the functionalities of tunneling amongst participating RON nodes.

COPYRIGHT NOTICE

No portion of the work presented in this dissertation has been submitted in support of any other award or qualification either at this institution or elsewhere.

DEDICATION

In the name of Allah, the Most Merciful, the Most Beneficent

To our parents, without whose unflinching support and unstinting cooperation, a work of this magnitude would not have been possible

ACKNOWLEDGEMENTS

First of all, we are grateful to Allah Almighty for granting us the courage and wisdom to understand this project, complete it, though we were not able enough.

A Special thanks to Major Faisal Amjad for his encouragement and guidance in the beginning when this project looked like an uphill task. Our heartfelt thanks to him for his educational and administrative support. He was always available to us with his expert opinion, kind advice and encouraging support, whenever we faced any problem or got stuck during the course of this project.

We are really grateful to Lecturer Bilal Bashir; no words are yet made which can completely describe his help to us in completing this project. We are thankful to him for explaining complex details of Resilient Overlay Networks, networking APIs like NDIS API, network programming in .NET framework (Visual C#) and giving us his valuable advises and opinions during the entire course of project.

And last but not the least, these acknowledgements would be incomplete without mentioning the names of the following persons, who remained with us and answered our emails and questions; Dr. David Andersen (Utah State University, USA), Lecturer Umar Kalim (NIIT, Rawalpindi).

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

This project provides an implementation of resilient overlay network (RON) based nodes that are deployed at various Autonomous Systems (ASs), at the application layer of OSI Model over the existing Internet. These RON nodes keep probing the neighboring RON nodes as well as the Internet and based on certain parameters of latency, throughput and loss rate forward data either via RON nodes or by the way of Internet. Because ASs are independently configured and operated, they generally fail independent of each other [1]. Due to this reason there is a physical path redundancy and it is possible for RON nodes to find alternate paths to other RON nodes even if the existing Internet protocol cannot.

1.1. VIRTUAL PRIVATE NETWORK AND OVERLAY NETWORKS

Virtual Private Networks (VPNs) are discrete network entities, configured and operated, over a shared network infrastructure [1]. For example in an intranet all the sites belong to a single organization where as in extranet VPN entity two or more organizations may share the information. Layer 2 and layer 3 are two broad arenas of VPN. Internet is composed of independently operating ASs. In AS detailed information is maintained with in itself and the information that is shared with other ASs over the Internet is heavily filtered by a router existing at the border of each AS. These routers use Border Gateway Protocol (BGP) to filter the information to share between ASs.

Internet is highly scalable but at the same time less reliable. This reduced reliability comes because the BGP limit the number of network links and hiding the

information about these links. BGP's slow fault recovery mechanism also makes it unreliable and there is a chance that certain path outage may cause a big disruption in communication. Due to this reason existing VPN over the Internet gets interrupted in link failures, routing faults, causing serious problems for the Internet service providers (ISPs) to manage their services to the customers.

An overlay network is a "virtual" network created on top of existing network [1]. Each RON node uses other RON nodes to send data. These RON nodes are deployed at various locations over the underlying network i.e. Internet or VPN. The path exploring mechanism of Internet is less aggressive in contrast RON can be more aggressive path exploration and maintenance. Application of RON can be in multimedia conferencing program that may link directly with RON library and hence forming an overlay network between all participants of conference. In the concept of overlay VPN, an administrator may wish to use RON nodes to form overlay network between multiple VPNs. Using RON, traditional ISPs can provide more reliable and error resistant Internet services to its customers.

An N node RON has $N (N - 1)$ unidirectional virtual links available among its nodes. RON when detects a packet, reads its parameters in the RON header and detects if it is for the particular RON node if not it is routed over the Internet or via other RON nodes. This process continuous until packet reaches to intended RON destination.

RON nodes in an overlay network observe and monitor quality of virtual paths based on certain parameters like latency, throughput and loss rate. These nodes probe Internet and their neighboring RON nodes and maintain routing table. These RON nodes route packets to best available link either via Internet or RON nodes. Each RON node checks for the availability of its neighboring RON nodes.

As compared to existing VPN, the RON architecture meets the following significant goals that existing VPN in the Internet substrate cannot: (i) the limited number of RON nodes helps to aggressively probe for path performance metrics. RON nodes exchange values of metrics quickly and it helps RON nodes to keep up to date information about its neighboring nodes and Internet. The result is better performance. (ii) RON has better reliability as each RON node has an independent application specific definition of what constitutes a fault. Thirdly, RON can efficiently detect alternate path even if the underlying Internet layer incorrectly believe that all is fine.

1.2. CONCLUSION

In this thesis we examined the design and architecture of RON and Internet and RON approach in their response. Evaluation of the results includes the deployment of RON nodes at various locations over the Internet. Two main contribution of this thesis are: (i) a flexible architecture for application level routing. Implementation provides a set of libraries that programmers can use to add application level routing capabilities to their programs, further networking facility for the researchers and better reliability in communication for end user and, (ii) validation of the effectiveness of the indirect routing. Through the deployed RON nodes over the Internet it is seen that even through a single intermediate RON node has improved overall reliability and performance of the Internet communication.

CHAPTER 2

OVERLAY NETWORKS

2.1. INTRODUCTION

The concept of overlay networks is quite mature in the field of networking. Overlay networks are deployed on top of an existing network. The Internet itself, in the beginning, was deployed as an overlay on top of the telephone network, using long-distance telephone links to attach Internet routers. Overlays in these days operate similarly but using the Internet paths as links upon which the overlay directs data and hence building a network on top of the network. Networks commonly overlaid on the Internet include the Multicast backbone [2] for extending multicast functionality across areas where the Internet did not natively support multicast and the IPv6 backbone [3] for testing the deployment of IPv6 [4].

2.2. TYPES OF OVERLAY NETWORKS

Initially Internet was deployed as a data network over the telephone lines and today a large number of networks are appearing based on modem lines. There are overlay networks deployed for multicasting, for example Multicast Backbone, and a network deployed for the testing of deployed IPV6 is IPv6 backbone. Here is a brief account of the existing overlay networks.

2.2.1. MULTICAST BACKBONE

Multicast Backbone [2] is composed of networks that support multicast. On each of these networks, there is a host that is running a multicast thread. The multicast routers are connected with each other via unicast tunnels. Tunnels are set up along

links of the underlying network. Each tunnel has a metric and a threshold. The metric is used for routing and the threshold to limit the distribution scope for multicast packets.

2.2.2. AUTOMATED OVERLAY DEPLOYMENT SYSTEM

A system for deploying and managing the Internet overlays is Automated Overlay Deployment System. Automated Overlay Deployment System creates IP tunnel-based Internet overlays consistent with a wide-ranging approach for network virtualization of the Internet. The Automated Overlay Deployment System allows different applications on the same end host or router to be associated with different overlay networks through its application deployment mechanism.

2.2.3. YOUR OWN INTERNET DISTRIBUTION

Your Own Internet Distribution [6] is a set of protocols that allows all of the replication and forwarding required for distribution for a given application to be done in the end hosts that are running the application itself. In other words, Your Own Internet Distribution works in the case where the only forwarders or distributors of the content are, the consumers of that content themselves.

The key attribute of Your Own Internet Distribution , and indeed the only thing that makes it different from other services already out there, is that it auto-configures tunneled shared-tree and mesh topologies among a group of hosts using ubiquitous Internet protocols only. This is an extremely powerful tool. Virtually every form of distribution in the Internet can be built upon this single capability.

2.2.4. IPv6 BACKBONE

The IPv6 backbone [3] is a virtual network layered on top of portions of the physical IPv4-based Internet to support routing of IPv6 packets [4], as that function has not yet been integrated into many production routers. The 6bone is thus focused on providing the early policy and procedures necessary to provide IPv6 transport in a reasonable fashion so testing and experience can be carried out. It would not attempt to provide new network interconnect architectures, procedures and policies that are clearly the purview of ISP and user network operators. In fact, it is the desire to include as many ISP and user network operators in the 6bone process as possible to guarantee a seamless transition to IPv6.

2.2.5. APPLICATION LEVEL MULTICAST INFRASTRUCTURE

Application Level Multicast Infrastructure [7] is tailored toward support of multicast groups of relatively small size with many to many semantics. Participants of a multicast session are connected via a virtual multicast tree, i.e., a tree that consists of unicast connections between end hosts. Application Level Multicast Infrastructure takes the centralized control approach to maintain tree consistency and efficiency.

2.3. OVERLAY NETWORKS STANDARDS

Overlays are in use for many years and a protocol named as multicast protocol evolved with the evolution of an overlay called Multicast Backbone. Although much research work was not done in overlay networks but it is emerged as a distinct area of research. In late 90’s two overlay standards came in, named as routing overlay and structured overlay.

2.3.1. OVERLAY ROUTING STANDARD

Main focus of the routing overlay was to enhance the existing underlying Internet or to replace it with new one due to its performance, accessibility, security and high scalability features. There are overlay networks that provide unidentified connection, censorship-resistant publication [8] and many extra features. Overlay networks like Tarzan [9], Tor [10] and FreeHaven [11] are designed to restrict viewers from determining the identity of corresponding hosts. This principle can be used directly in a system to provide greater protection against certain identity threats.

2.3.2. STRUCTURED OVERLAYS

There are certain structured Overlays like Chord [12], Pastry [13] and Tapestry [14] that focus on procedure to manage the power of large distributed collections of machines. Today many of the large system projects are based on Storage and lookup overlays. For example the systems presented in this thesis, main focus of many of these projects is to create an overlay network that is highly flexible to the failure of individual links and nodes.

Such overlays characteristically reproduce contents across a number of individual nodes. This reproduction of contents is similar to those performed by a content distribution network, but data in a structured overlay is often inconsistent, hence not just copy of original object. Today research is made to use structured overlays to implement RON-like services [14], high bandwidth data dissemination [15], and distributed file systems [16].

2.4. ADVANTAGES OF USING OVERLAY NETWORKS

Overlay networks are in use for many years due to its proficiency over the existing Internet. Although there are many networks in market these days however overlay network take advantage due to many prominent features and advantages. Few are discussed here.

The first advantage of using an overlay network compared to changing Internet protocols or routers is that the overlay offers a fast and easy deployment path that is deficient in many of the industrial and biased obstacles of a router-level deployment. Internet researchers feel that Internet Protocol (IP) and the IP routing infrastructure have become inflexible because of its enormous success. Replacement of the thousands of at present installed IP-speaking devices can cause a considerable challenge. Researchers are focused to design new protocols for IP layer instead of completely changing it that run on top of IP in an overlay.

Second advantage of overlays is that they are away from loading the underlying network with features better performed at upper layers and error in the overlay network is much less likely to crash the underlying IP routing infrastructure. Functions such as content routing require that the content routers acquire good familiarity of the application protocols that run through them. These application protocols are likely to be changed frequently; supplement central routers with application-specific knowledge would burden them and as a result high processing would be required by a small part of the traffic that passes through them. Many routers hold IP options and other odd events on a slow path that has only a fraction of the forwarding speed of normal packet processing. This discrimination has traditionally made such routers at a risk to additional denial-of-service attacks,

requiring more mysterious processing requirements to routers that will add up performance as well as security issues to administrators.

Overlays offer access to resources miles away from the constrained environment of high-speed routers. An overlay can take advantage of the excess of processing, memory, and permanent storage available to perform tasks that would normally be more acceptable than a conventional router. The capability to execute these tasks makes possible the creation of powerful new facilities such as scalable, distributed publish-subscribe systems and content distribution networks; in such a way that slow, expensive and time consuming tasks through routers' shortest path available.

CHAPTER 3

VIRTUAL PRIVATE NETWORK (VPN)

3.1. INTRODUCTION

With the arrival of Internet, it is found as a cheap and a high speed medium for data communication. New technologies were discovered to make use of this new communication medium, recently available. Basic idea was to use Internet as a substrate for the new emerging technologies. Hence the concept of VPN arrived in the networking field. VPN is a collection of three words i.e. Virtual, Private and Networks. It is Virtual, for the reason that there is no factual direct network association between the two communication associates, but only a virtual association provided by VPN, taken in normally over public Internet. And private in the sense that only the associated members of the corporation connected by the VPN have the access to the available data. VPN is a technology that is in extensive use these days. A VPN network link is carried over a shared or public network, in this case the Internet, and encrypts the shared data so that only the VPN client and server can have access to this data. VPN connections cost much less than dedicated connections.

3.2. VPN OVERVIEW

A typical VPN can be illustrated as a layout of logical connections available by special software that maintains isolation by protecting the connection hosts i.e. clients and the server. Internet today is such a networking medium that is used to gain privacy by recent cryptographic techniques. To have an overview of a typical VPN, let’s consider a scenario that NUST is an organization and it has two campuses, MCS

and EME. If the NUST subdivision in MCS decides to get some data about some student, then the EME subdivision might need to know that at the same time. There are faculty members teaching in both the institutions and data of a faculty member might be hosted by the EME data server sitting at MCS at the time of query. Both the networks are deployed over the public Internet and border gateways provide Internet facility to the faculty members. VPN software is installed at the data accessing network hosts at MCS and EME. Then VPN Software is configured to establish the connection to the other side, as shown in Figure 3.1.

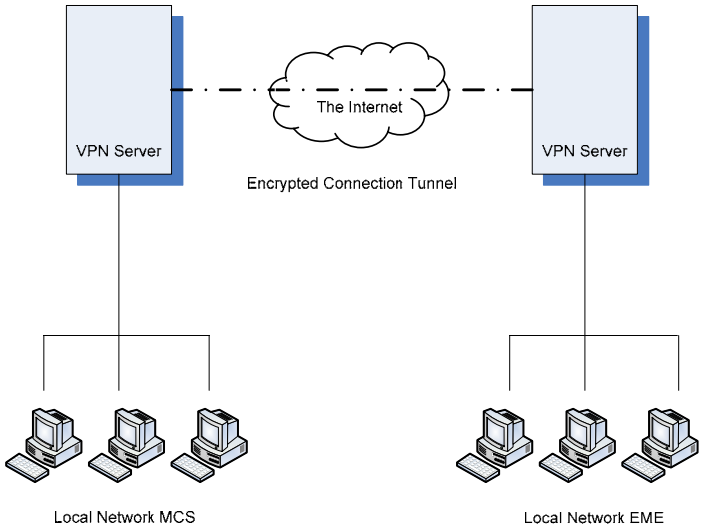

Figure 3.1 A typical VPN setup

MCS VPN server has to accept connections from the EME VPN server, and the MCS server must connect to EME and vice versa. Now each of the two campuses has a working Virtual Network.

The two VPNs are connected via the Internet and can work together like in a real network. This VPN is without privacy, because any Internet router between MCS and EME can read the data to be exchanged. To make this Virtual Network as Private we can use various encryption techniques. Now only computers or persons owning

the decryption key can open this encrypted data and look at the contents, this phenomenon is shown in Figure 3.2.

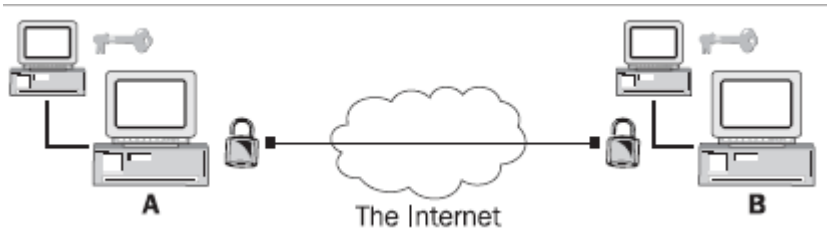

Figure 3.2 A typical VPN connection

A VPN connection is normally built between two Internet access routers equipped with VPN software. The software must be set up to connect to the VPN co-workers, the firewall must be set up to allow access, and the data exchanged between VPN partners must be secured (by encryption). The encryption key must be provided to all VPN partners, so that the data exchanged can only be read by authorized VPN partners.

3.3. SCENARIOS FOR VPN TECHNOLOGY

With the passage of time more and more organizations offer their customers or business partners a confined access to available data for their business needs and requirements, for example categorizing formulas or collection of data. In this sense we have three typical scenarios for VPN technology in the modern projects: (i) An intranet extended over multiple locations of a large organization, (ii) A dial-up access is dedicated for domestic purposes or company workers in the field, (iii) An extranet is dedicated particularly by an organization for their customers or business associates.

3.3.1. INTRANET VPN

Intranet VPN services help to interconnect local area networks located at multiple geographic areas over the shared network infrastructure. Typically, this service is used to connect multiple geographic locations of a single company. Several small offices can be connected with their regional and main offices. This service provides a replacement for the expensive dedicated links [17].

In this type of VPN technology it is quite easy to increase the capability of any of the links depending on the applications running on the VPN. As variations of the applications can occur with time, this architecture can be adapted to meet the varying needs. Secondly, new geographical sites can be connected to the VPN without any trouble. These efforts can reduce the overall cost however dedicated private links can be much expensive.

3.3.2. DIAL-UP VPN

The Dial-up VPN service supports mobile and telecommuting employees in accessing the company's Intranet from remote locations [17]. Employee or domestic user, who wants to access the data, dials into the nearest Remote Access Server (RAS). Using layer 2 [25] tunneling protocol RAS recognizes the user and establishes a secure VPN connection to the company after his successful authentication from the VPN authentication server.

The use of dial-up VPN technology helps in considerable cost reduction to the organization. It also eradicates the requirement of managing large modem collection and uses RASes that belong to the local Internet Service Providers (ISPs). This also reduces cost in a sense that in most of the cases connection is local so no remote

dialing costs are there in linking to the company using RAS, this adds to the company's financial advantage as well.

3.3.3. EXTRANET VPN

This infrastructure enables external vendors, suppliers and customers to access specific areas of the company's Intranet [17]. Whenever some company representative needs to access the company's Intranet, the firewall and authentication procedure of RAS, same as in above case, guarantees that the connection is directed to the part of intranet that is accessible to that user or representative. However in most of the organizations restrictions are not applied to all the people accessing the intranet, for example employees of the company may be given full access to intranet as compared to the vendors, suppliers or customers.

The flexibility of the extranet services helps to provide connectivity to new external suppliers and customers within a short period of time. The fast communications facilitated by the extranet helps in several e-commerce areas including efficient inventory management and electronic data interchange (EDI) [17]. These results in cost reduction and to effectively compete in the speedily changing market trends.

3.3.4. EXISTING VPN

A VPN is a private network created through a public network, mostly Internet. As all of the packets between two points are encrypted that is why a VPN is called private, so even though the packets are broadcast over a public network, their information remains protected. As the Internet is much cheaper than dedicated Wide Area Network (WAN) connections, and VPNs regularly make use of existing Internet connections for two (or more) locations. VPN clarifications range from easy ones that

can be applied on Windows, to dedicated specialized VPN applications that can sustain many of the customers.

Today VPN connections are used in two important ways: (i) To form WAN connections using VPN technology between two networks or groups that may be hundreds of miles apart, but each one of them has some way of accessing the underlying public Internet, (ii) To form remote access connections that enable distant users to enter a private network through a public network like the Internet

Both sides of a VPN connection must be running compatible VPN software using compatible protocols. For a remote access VPN solution, the software installed depends on the VPN itself. Dedicated VPN solutions also sell client software that can be distributed to many users as shown in Figure 3.3.

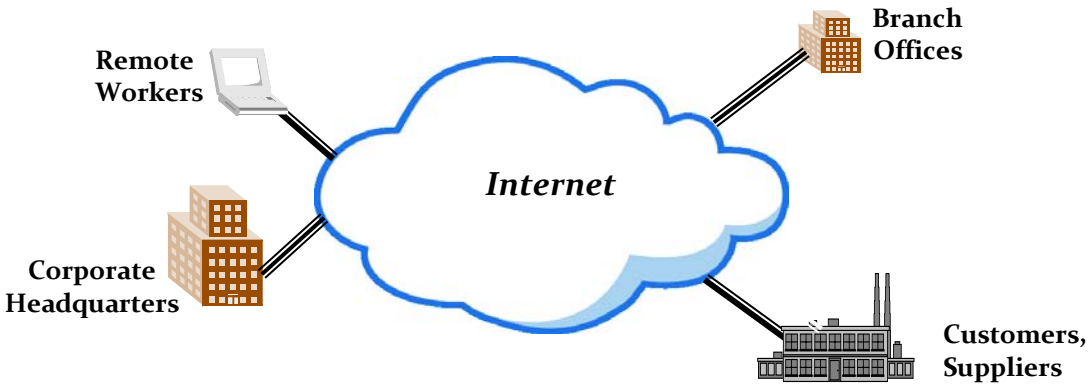

Figure 3.3 A typical Virtual Private Network

VPNs are becoming an increasingly important source of revenue for ISPs. Informally, a VPN establishes connectivity between a set of geographically dispersed endpoints over a shared network infrastructure. The goal is to provide VPN endpoints with a service comparable to a private dedicated network established with leased lines. Thus, providers of VPN services need to address the quality of service (QoS) and security issues associated with deploying a VPN over a shared IP network.

There are two popular models for providing QoS [18] in the context of VPNs: The Pipe Model [19] and The Hose Model [20].

In the Pipe Model, the VPN customer specifies QoS requirements between every pair of VPN endpoints. Thus, the Pipe Model requires the customer to know the complete traffic matrix, that is, the load between every pair of endpoints. However, the number of endpoints per VPN is constantly increasing and the communication patterns between endpoints are becoming increasingly complex. As a result, it is almost impossible to predict traffic characteristics between pairs of endpoints required by the Pipe Model.

The Hose Model alleviates the above-mentioned shortcomings of the Pipe Model. In the Hose Model, the VPN customer specifies QoS requirements per VPN endpoint and not every pair of endpoints. Specifically, associated with each endpoint, is a pair of bandwidths—an *ingress* bandwidth and an *egress* bandwidth. The ingress bandwidth for an endpoint specifies the incoming traffic from all the other VPN endpoints into the endpoint, while the egress bandwidth is the amount of traffic the endpoint can send to the other VPN endpoints. Thus in a Hose Model, VPN service provider supplies the customer with certain guarantees for the traffic that each endpoint sends to and receives from other endpoints of the same VPN. The customer does not have to specify how this traffic is distributed among the other endpoints. As a result, in contrast to the Pipe Model, the Hose Model does not require a customer to know its traffic matrix, which, in turn, places fewer burdens on a customer that wants to use the VPN service.

3.4. VPN TUNNELS

We can define VPN as an overlay network built with tunnels in which the tunnel payloads are encrypted and authenticated [17]. VPN technology often is called tunneling, because the data in a VPN connection is protected from the Internet as the walls of the road or rail tunnel protect the traffic in the tunnel from the masses of stone of the mountain above. Let's now have a closer look at how VPN Software does this:

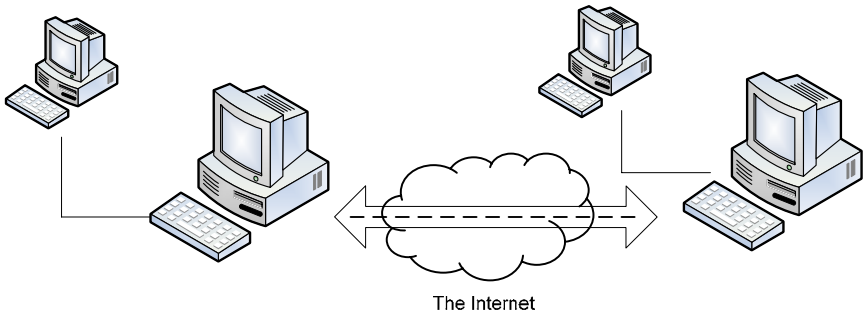

Figure 3.4 Tunnel in a Virtual Private Network

The VPN software in the locations A and B encrypts (lock) and decrypts (unlock) the data and sends it through the tunnel. Like cars or trains in a tunnel, the data cannot go anywhere else but the other tunnel endpoint. The following are put together and wrapped into one new package: Tunnel information (like the address of the other endpoint), Encryption data and methods, and the original IP packet (or network frame).

The new package is then sent to the other tunnel endpoint. The payload of this package now holds the complete IP packet (or network frame), but in encrypted form and thus not readable for anyone not possessing the right key. The new header of the packet simply contains the addresses of sender and recipient and other metadata necessary for and provided by the VPN software used.

It is noticed that the amount of data sent grows during the process of "wrapping" depending on the VPN software used; this overhead can become a very important factor. The overhead is the difference between net data sent to the tunnel software and gross data sent through the tunnel by the VPN software, as shown in Figure 3.5. If a file of 1 MB is sent from user A to user B, and this file causes 1.5 MB traffic in the tunnel, then the overhead would be 50%, a very high level. The overhead caused by the VPN Software depends on the amount of organizational data and the encryption used. Whereas the first depends only on the VPN Software used, the latter is simply a matter of choice between security and speed. Hence if better encryption techniques are used, the more overhead will be produced.

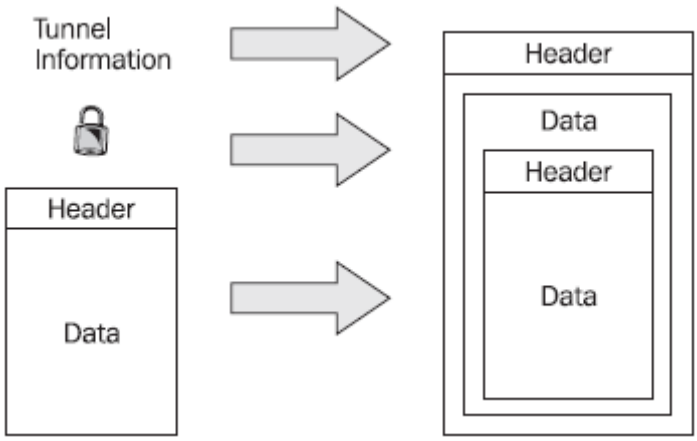

Figure 3.5 The overhead caused by the VPN Software

3.5. STANDARD FOR TUNNELING DATA

The General Routing Encapsulation (GRE) provides a standard for tunneling data. The concept of GRE is pretty simple. A protocol header and a delivery header are added to the original packet and its payload is encapsulated in the new packet. No encryption is done. The advantage of this model is simplicity that offers many possibilities, the transparency enables administrators and routers to look inside the packets and pass decisions based on the type of payload sent.

3.5.1. LEVEL 2 TUNNELING PROTOCOLS

Four well-known Layer 2 VPN technologies, which are defined by Requests for Comments (RFC) [21], use encryption methods and provide user authentication:

Point to Point Tunneling Protocol (PPTP*),* which was developed with the help of Microsoft, is an expansion of the PPP and is integrated in all newer Microsoft Operating Systems. PPTP uses GRE for encapsulation and can tunnel IP, IPX, and other packages over the Internet. The main disadvantage is the restriction that there can only be one tunnel at a time between communication partners.

Layer 2 Forwarding (L2F) was developed almost at the same time by companies like Cisco and others and offers more possibilities than PPTP, especially regarding tunneling of network frames and multiple simultaneous tunnels.

Layer 2 Tunneling Protocol (L2TP) is accepted as an industry standard and is being used widely by Cisco and other manufacturers. Its success is based on the fact that it combines the advantages of L2F and PPTP without suffering from their disadvantages. Even though it provides no own security mechanisms, it can be combined with technologies offering such mechanisms like IPsec [21].

Layer 2 Security Protocol (L2Sec) was developed to provide a solution to the security flaws of IPsec. Even though its overhead is rather big, the security mechanisms used are secure, because mainly is used SSL/TLS.

3.5.2. LEVEL 3 TUNNELING PROTOCOLS

IPSec is probably the most wide-spread tunneling technology. It is a set of protocols, standards, and mechanisms than a single technology. IPSec consists of three major protocols: AH (A protocol that provides data origin authentication, data integrity, and relay protection), ESP (A protocol that provides the same services as

AH but also offers data privacy through the use of encryption), IKE (A protocol that provides the all-important key-management function [17]. AH and ESP can operate in one of two modes [17]. The two modes are:

Transport mode: A technique that can provide security to the upper-layer protocol of an IP datagram

Tunnel mode: A technique of providing security to an IP datagram

3.5.3. LEVEL 4 TUNNELING PROTOCOL

It is possible to establish VPN tunnels only on the application layer. Secure Sockets Layer (SSL) and Transport Layer Security (TLS) are the protocols that follow this approach. Although SSL is most often thought of as a way of securing Web transactions it is a versatile protocol with many uses [17]. Security is achieved by encrypting traffic using SSL/TLS mechanisms, which have proven to be very reliable and are permanently improved.

3.6. SUMMARY

This chapter gives us an overview of VPN and scenarios for VPN technology i.e. Intranet, Extranet and Dial-up VPN. VPN is a private network created through public network. It is private due to the fact that data is encrypted between the two points. Most ISPs use Pipe Model and Hose Model to address QoS and security issues associated with the deployment of VPN. VPN technology is often called tunneling because the data in VPN connection is protected from the Internet and the General Routing Encapsulation (GRE) provides a standard for tunneling data.

CHAPTER 4

INTERNET AND RELATED PROBLEMS

4.1. THE INTERNET

Internet, since the day of its arrival in 1960's has changed a lot and the Internet today does not have a simple hierarchal structure. It is a collection of many wide and local area networks, joined by connecting devices or switching stations. It is very difficult to represent accurate Internet as it is continuously changing new networks are being added existing networks are being modified and networks of obsolete companies are being removed from the view. Today most of the end users use ISP to get an Internet connection. There are international service providers, national service provider's regional service providers and local service providers. Today instead of government, most private companies run the Internet. A conceptual view of Internet today [22] is shown in Figure 4.1.

Today's Internet infrastructure is a move from a core network to a more distributed architecture operated by commercial providers connected via major network exchange points, as well as direct network interconnections.

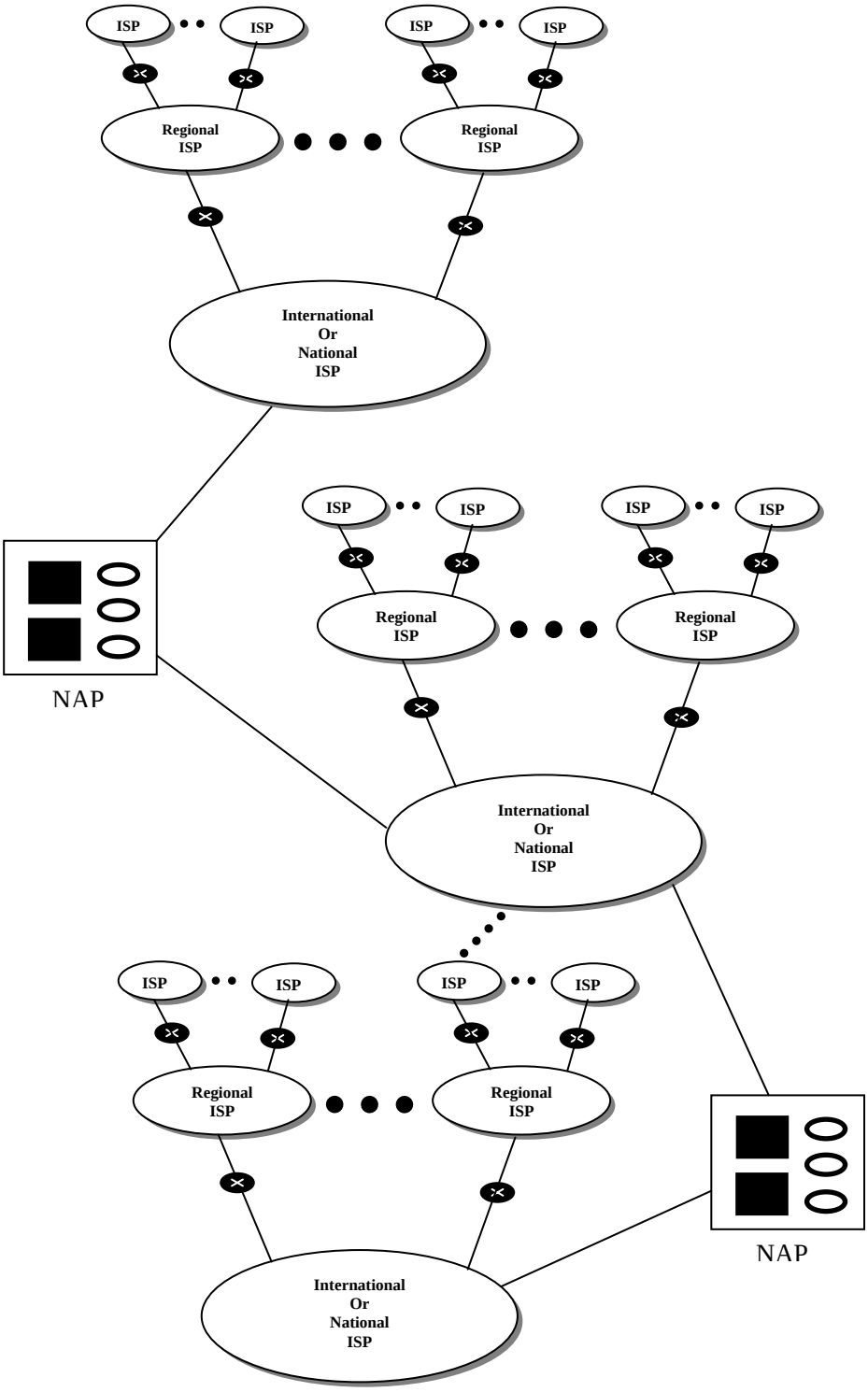

Figure 4.1 Existing Internet

4.2. ROUTING

The Internet today has a sense of hierarchy. The Internet is divided into international and national ISPs. National ISPs are divided into regional ISPs and regional ISPs are divided into local ISPs. If the routing table has a sense of hierarchy like the Internet architecture, the routing table can decrease in size.

Let us take the case of a local ISP. A local ISP can be assigned a single, but larger block of addresses with a certain mask. The local ISP can divide this block into smaller blocks of different sizes and can assign these to individual users and organizations, both large and small. If the block assigned to the local ISP is A.B.C.D/n, the ISP can create the blocks of E.F.G.H/m, where m may vary for each customer and is greater than n.

The rest of the Internet does not have to be aware of this division. All customers of the local ISP are defined as A.B.C.D/n to the rest of the Internet. Every packet destined for one of the addresses in this large block is routed to the local ISP. There is only one entry in every router in the local ISP, the router must recognize the sub blocks and route the packet to the destined customer. If one of the customers is a large organization, it also can create another level of hierarchy by sub netting and dividing its sub block into smaller sub blocks or sub-sub blocks.

4.3. INTERNET ROUTING

IP addresses and packet switching provide the technical infrastructure which routing protocols use to transmit packets across the Internet. The Internet Protocol transfers packets between networks and provides the software bridge that knits the whole thing together

4.3.1. INTERNET PROTOCOL (IP)

Each computer on the Internet has a unique numerical address, called an Internet Protocol (IP) address, used to route packets to it across the Internet. Just as postal address enables the postal system to send mail to a house from anywhere around the world, computer's IP address gives the Internet routing protocols the unique information they need to route packets of information to the desktop from anywhere across the Internet. If a machine needs to contact another by a domain name, it first looks up the corresponding IP address with the domain name service. The IP address is the geographical descriptor of the virtual world, and the addresses of both source and destination systems are stored in the header of every packet that flows across the Internet.

4.3.2. DOMAIN NAME SERVERS (DNS)

Domain Name System (DNS) servers distribute the job of mapping domain names to IP addresses among servers allocated to each domain. Each second-level domain must have at least one domain name server responsible for maintenance of information about that domain and all subsidiary domains, and response to queries about those domains from other computers on the Internet. For example, management of domain name information and queries for the *.com* domain are handled by a specific DNS server. This distributed architecture was designed to enable the Internet to grow, as the number of domains grew, the number of DNS servers can grow to keep pace with the load.

Today, everyone who registers a second-level domain name must at the same time designate two DNS servers to manage queries and return the current IP address for addresses in that domain. The primary domain name server is always consulted

first, and the secondary domain name server is queried if the primary doesn't answer, providing a backup and important support to overall Internet reliability.

4.3.3. POINT-TO-POINT COMMUNICATION

Reliable point-to-point communication is one of the main utilizations of the Internet, where over the last few decades TCP has served as the dominant protocol. Over the Internet, reliable communication is performed end-to-end in order to address the severe scalability and interoperability requirements of a network in which potentially every computer on the planet could participate. Thus, all the work required in a reliable connection is distributed only to the two end nodes of that connection, while intermediate nodes route packets without keeping any information about the individual packets they transfer.

Overlay networks are opening new ways to Internet usability, mainly by adding new services (e.g. built-in security) that are not available or cannot be implemented in the current Internet, and also by providing improved services such as higher availability [23]. However, the usage of overlay networks may come with a price, usually in added latency that is incurred due to longer paths created by overlay routing, and by the need to process the messages in the application level by every overlay node on the path.

4.4. BORDER GATEWAY PROTOCOL (BGP)

Border Gateway Protocol (BGP) is an inter-autonomous system routing protocol. It first appeared in 1989 and has gone through four versions. BGP is based on a routing method called path vector routing. Distance vector is not a good candidate because there are occasions in which the route with the smallest hop count

is not the preferred route. For example, we may not want a packet to pass through an autonomous system that is not secure, even though it is the shortest route, and distance vector routing is unstable due to the fact that the routers announce only the number of hop counts to the destination without actually defining the path that leads to that destination.

4.4.1. ROUTING MECHANISM OF BGP

BGP is a path vector protocol used to carry routing information between autonomous systems. The term path vector comes from the fact that BGP routing information carries a sequence of AS numbers that identifies the path of ASs that a network prefix has traversed. The path information associated with the prefix is used to enable loop prevention. BGP uses TCP as its transport protocol. This ensures that all the transport reliability is taken care of by TCP and does not need to be implemented in BGP, thereby simplifying the complexity associated with designing reliability into the protocol itself. Two BGP routers that form a TCP connection between one another for the purpose of exchanging routing information are referred to as neighbors or peers as shown in Figure 4.2.

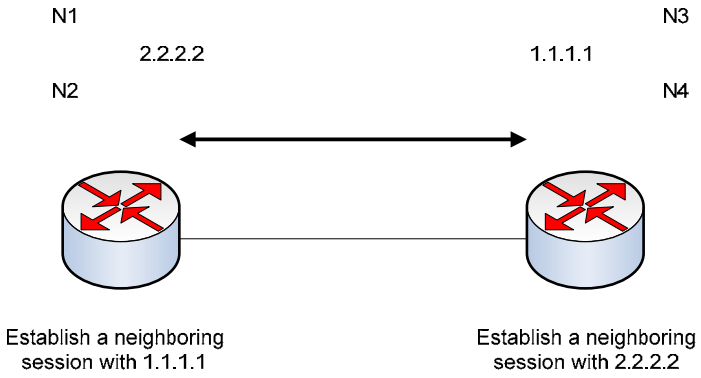

Figure 4.2 Two BGP routers as neighbors.

Peer routers exchange open messages to determine the connection parameters. BGP also provides a mechanism to gracefully close a connection with a peer. Initially, when a BGP session is established between a set of BGP routers, all candidate BGP routes are exchanged, as illustrated in Figure 4.3. After the session has been established and the initial route exchange has occurred, only incremental updates are sent as network information changes.

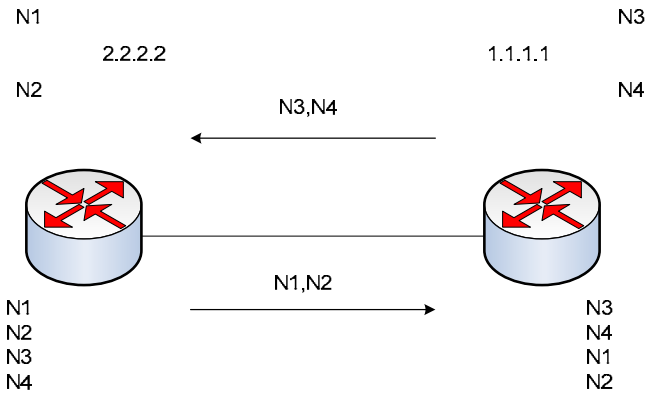

Figure 4.3 Exchange of routing updates

Routes are advertised between a pair of BGP routers in UPDATE messages. The UPDATE message contains, among other things, a list of <length, prefix> tuples that indicate the list of destinations that can be reached via a BGP speaker. The UPDATE message also contains the path attributes, which include such information as the degree of preference for a particular route and the list of ASs that the route has traversed. In the event that a route becomes unreachable, a BGP speaker informs its neighbors by withdrawing the invalid route. As illustrated in Figure 4.4, withdrawn routes are part of the UPDATE message. These routes are no longer available for use.

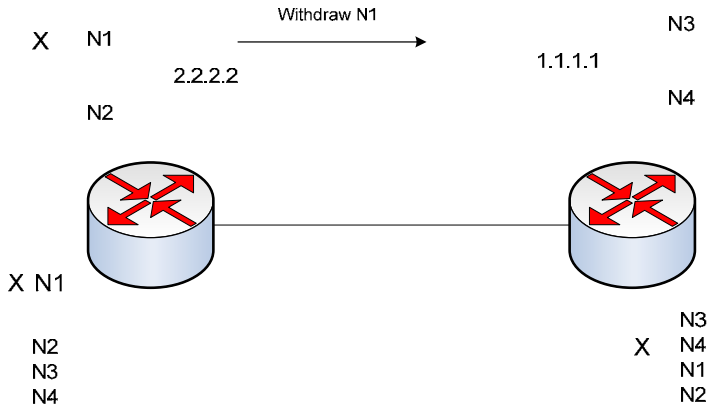

Figure 4.4 Withdrawn routing update

Figure 4.5 illustrates a steady state situation. If no routing changes occur, the routers exchange only KEEPALIVE packets.

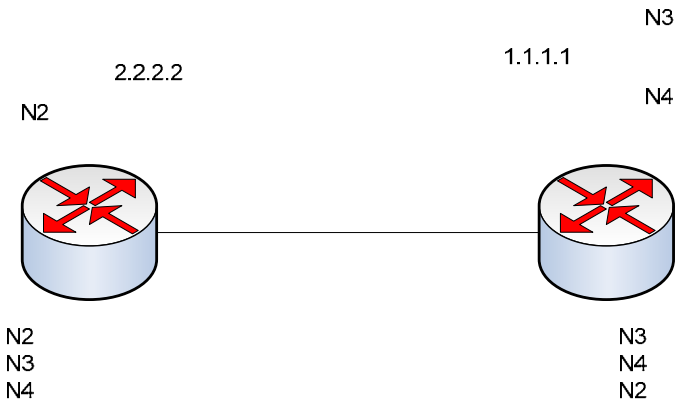

Figure 4.5 Steady state situation

KEEPALIVE messages are sent periodically between BGP neighbors to ensure that the connection is kept alive.

4.4.2. BGP CAPABILITIES NEGOTIATION

If a BGP router supports capabilities negotiation, when it sends an OPEN message to a BGP peer, the message may include an Optional Capabilities parameter. A BGP router examines the information contained in the Capabilities parameter of an OPEN message in order to determine which capabilities the peer supports. If a BGP

router determines that a peer supports a given capability, the router can use the capability with the peer.

A BGP router determines that a peer doesn't support capabilities negotiation if, in response to an OPEN message that carries the new Optional parameter, the router receives a NOTIFICATION message that contains an Error Sub code, set to "Unsupported Optional Parameter." If this occurs, the BGP router should attempt to reestablish the connection without sending the Capabilities Optional parameter to the peer.

4.4.3. BGP INSTABILITY

BGP instability correlates with poor path performance and path failures. If BGP instability precedes path failures, reactive routing might even proactively route around some failures before they occur. Using BGP information might therefore reduce the reaction time of reactive routing systems. An analysis [24] carried on one year of data collected on geographically and topologically diverse test bed of 31 hosts. The paths between these hosts traverse more than 50% of the well-connected autonomous systems (ASs) on the Internet. The data includes correlated probes, where active probes between hosts discover one-way path failures lasting longer than 2 minutes and trigger trace routes along these paths when a failure is discovered. Then they correlated these observed failures with BGP routing information collected at eight monitoring hosts at the same time. Finally, correlations between path failures and routing instability, as observed from collocated BGP route monitors. Table 4.1 summarizes the major results in this paper [24].

Table 4.1 Summary of BGP instability

While a few paths are much more failure-prone than other, failures appear spread out over many different links, not just a few “bad” links.
Failures appear more often inside AS’s than on links between them.
90% of failures last less than 15 minutes, and 70% of failures last less than 5 minutes.
BGP messages coincide with only half of the failures that reactive routing could potentially avoid, suggesting that these were failures that not even a “perfect” BGP could avoid.
Reactive routing is potentially more effective at correcting failures for hosts with multiple Internet connections.
BGP traffic is a good indicator that a failure has recently occurred or is about to occur. When BGP messages and failures coincide, BGP messages most often follow failures by 4 minutes.

4.5. INTERNET ROUTING PROBLEMS AND BGP

Today’s Internet is organized as independently operating autonomous systems. Detailed routing information is maintained within a single AS. Routing information is shared with other AS in the network using a router running at the border of each AS. These routers use BGP to filter information to be shared. But BGP based Internet does not handle failure quite efficiently. Two broad kinds of failures include, link failures and path failures. Link failures occur because of the failure of a router or link connecting two routers, due to software hardware or link disconnection problem. Path failures can occur due to congestion of traffic that causes packet loss, low throughput and variable latencies.

Outages and performance failures, associated with application failures are of two types: link failure and path failure. Link and path failures cause outages which in turn cause reduced throughput, increased latency rates and increased packet loss rate.

Hence degrading the performance of most of the protocols such as BGP and TCP etc. Four significant drawbacks in BGP based Internet routing are; (i) End to end communication failure, (ii) Overall influence over path performance, (iii) Inability to effectively multi-home multiple ISPs and, (iv) Blunt policy expression.

4.5.1. END-TO-END COMMUNICATION FAILURE

BGP-4 [25] link failure recovery mechanism is not robust enough, usually it takes much longer time to recover link failure and then discover a new route that is better than last one. In most of the cases these link failures cause path outages. Craig [26] examines the latency in the Internet path failures and several techniques of convergence to inter-domain routing. In such situation BGP's path selection mechanism fails and may result in loss of connectivity, increase in packet loss and latency as well. Studies of several thousand inter-domain routing faults concluded that measured upper bound on the Internet inter-domain routing convergence delay is an order of magnitude slower than previously thought. Measurements of the experiments performed show that fraction of the convergence hindrance can be fixed with some modification to the implementations of BGP, extensive hindrances and temporary fluctuation are a primary consequence of the BGP path vector routing protocol.

By using explore techniques discussed by Vern Paxson [27] that routing pathology avoids chosen Internet hosts from data exchange up to 3.3% of the time that was calculated over a long period of path exploration process, and this percentage has not improved with time [27]. By having a close look at the routing table logs at Internet backbones, that 10% [23] of all considered routes were available less than 95% of the time, and that less than 35% of all routes had an accessibility of higher than 99.99% [28]. In addition, they find that about 40% of all path outages take

more than 30 minutes to repair and are heavy-tailed in their duration. More recently, Chandra *et al.* finds using active probing that 5% of all detected failures last more than 10,000 seconds (2 hours, 45 minutes), and that failure durations are heavy tailed and can last for as long as 100,000 seconds before being repaired [29]. These findings do not foretell well for mission-critical services that require a higher degree of end-to-end communication availability. Summary of these results is put into Table 4.2.

Table 4.2 Review of the results of Internet link failure studies

RESEARCH WORK	RESULTS OF RESEARCH
Paxson	Serious routing pathology rate, 1995: 3.3%
Labovitz	10% of routes available less than 95% of the time
Labovitz	Less than 35% of routes available 99.99% of the time
Labovitz	40% of path outages take 30+ minutes to repair
Chandra	5% of faults last more than 2 hours, 45 minutes

4.5.2. OVERALL INFLUENCE OVER PATH PERFORMANCE

Internet's routing techniques don't efficiently handle the performance failures and due to this reason the quality of end-to-end communication between applications is degraded. Consider a link that is terribly loaded because of reasonable traffic or a denial-of-service attack. BGP mostly remain unaware of this situation, and even if an alternate path exists, it will not use it. Due to this problem of path diversity it is hard to handle performance failures and such failures may result in serious troubles.

Today's inter-domain IP-layer routing is not application dependent; it does not by and large permit route selection based on the nature of the application. While some research projects like FIRE [30] provide flexible intra-domain routing, the problem is by a long way harder for inter-domain routing. This is one of the significant reasons due to which applications running over the Internet are not able to influence the

choice of paths to best suit their latency, loss, or bandwidth parameters. BGP uses open shortest path first technique and look for a shortest path from source IP and destination without realizing the alarming traffic running through the very same path. Hence BGP path selection algorithm results in significant performance failures.

4.5.3. INABILITY TO EFFECTIVELY MULTI-HOME MULTIPLE ISPS

The client's connection to the Internet is a common source of both bandwidth blockage and breakdown. Many smaller client networks are only connected to the Internet through a single link. This lack of redundancy makes smaller clients particularly vulnerable to longer term interruptions of Internet connectivity. Networks with multiple upstream Internet providers are said to be multi-homed. In order to use multiple connections, these multi-homed networks typically participate in Internet routing to choose between upstream links and to inform the rest of the network that they can be reached via several paths.

The IP routing design depends on heavy aggregation of the addresses announced into the system. A small or medium-sized organization that wishes to obtain better Internet service might purchase service from two different ISPs, hoping that an outage to one would leave it connected via the other. Unfortunately, to limit the size of their routing tables, many ISPs will not accept routing announcements for fewer than 4096 contiguous addresses (subnet block of "/20") to Small companies, regardless of their reliability needs, may not even require 256 addresses, and cannot effectively multi-home. One alternative may be "provider-based addressing," where an organization gets addresses from multiple providers, but this requires handling two distinct sets of addresses on its hosts.

While provider-based addressing can provide long-term fail-over; for example if one ISP connection fails, it is unclear how on-going connections on one address set can seamlessly switch on a failure in this model.

4.5.4. BLUNT POLICY EXPRESSION

BGP is incapable of expressing fine-grained policies aimed at users or remote hosts; it can only express policies at the granularity of entire remote networks. This reduces the set of paths available in the case of a failure.

4.6. SUMMARY

This chapter gives us a brief understanding of the structure of existing Internet and its exponential growth is last two to three decades. Internet is a collection of Autonomous Systems (ASs), individually configured and maintained. Border Gateway routers, at the end of each AS controls the flow of information across the boundary of each AS using BGP. BGP has slow link and path failure detection mechanism which makes its less efficient and its inability to handle multiple ISPs for better performance and to increase available bandwidth. It is incapable to express policies as the user level rather than entire network.

CHAPTER 5

RESILIENT OVERLAY NETWORK (RON)

5.1. INTRODUCTION

A *Resilient Overlay Network* (RON) is an overlay network that allows distributed Internet applications to detect and recover from path outages and periods of degraded performance within several seconds [31]. RON is proposed to get better accessibility for the Internet applications distinguished by small collection of hosts that share information in a distributed environment. For example video conferencing, distant login and administration, links between data servers and to establish virtual private networks. The RON software can be used detached or stand-alone within an application, or on the other hand, impervious application traffic can be summarized and forwarded within a RON using the RON IP Tunnel application.

RON nodes cooperate with neighboring nodes to forward packets for each other to keep away from failures in the underlying Internet and as a result, these RON nodes form an application-layer overlay network. RON nodes are set up at various locations in the Internet. Each RON node observes the quality of the underlying Internet paths as well as paths to its neighboring nodes and uses this information to route the packets on the selected paths. Virtual link is a direct path between two RON nodes in the Internet. In order to get information about virtual links in the topology, to which it is not directly connected, each RON node contributes in a routing protocol to exchange related information about a range of quality metrics and based on these metrics virtual links are distinguished individually.

5.2. RON DESIGN

RON is described as a set of classes and structures such that different applications and programs can be linked against them, as shown in Figure 5.1. Here these applications or programs RON clients. These programs cooperate with each other in an overlay network to provide a distributed service. RON clients use service-oriented routing metrics to decide how to forward packets in the group.

The proposed RON design contains a range of RON clients, ranging from a generic IP packet forwarder that provide better reliability of IP packet delivery, to an organizational video conferencing application that integrates variety of application specific routing metrics to select best available route.

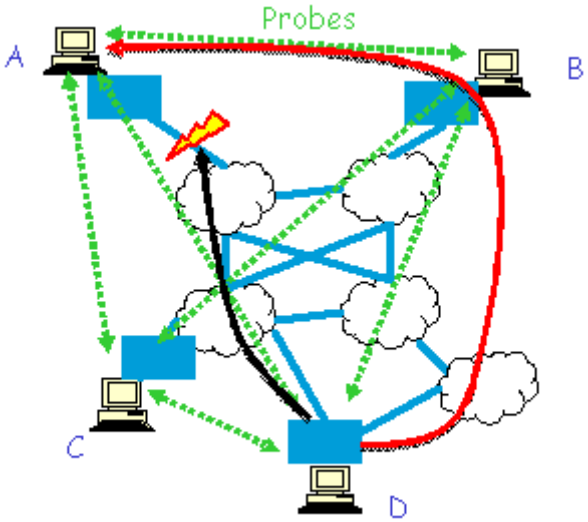

Figure 5.1 High Level RON design – RON Nodes probe network and determine network measuring characteristics between them. Using their knowledge of the network, they select robust available path to send data. In above case data is sent from D to A via RON node B and not via Internet.

Figure 5.1 shows a RON, comprising of four nodes, forming an Overlay Network over the underlying Internet. Four RON nodes are represented as A, B, C and D. Each RON node probes to determine the network characteristics at any instant of time. RON nodes monitor the quality of paths between neighboring RON node as well as with the Internet using certain metrics, like latency, through put and loss rate, to decide the route to forward packets. For example, RON node “D” has to send data to “A”. Node “D” probes and finds that due to certain path or link failure there is a block in the Internet, so it forwards packet to node “A” via another RON node “B” indirectly existing in this overlay network. The RON has very less number of nodes relative to the Internet this allows it to probe and explore paths more aggressively and to use a range of quality metrics, substituting this information using a link state routing protocol. It also makes possible the implementation of flexible policies that govern what types of packets may be forwarded, at what rates, and along which paths in the RON.

5.3. RON OBJECTIVES

RON is projected to achieve following principal design objectives: (i) Fast and constructive failure detection and recovery and, (ii) Application based efficient routing mechanism.

RON’s first objective is to detect the point where failure occurred and in the second step devise a mechanism to get rid of such problem in shortest possible time. These failures effect overall routing in the Internet resulting in performance failure. Today’s Internet routing often requires three minutes to recover from this type of failures. RON is more than application in which other nodes are willing to contribute their own network resources in order to ensure that other nodes in the network can

reach each other with minimum failure. These applications include both group communication applications such as audio and video conferencing, and applications such as Virtual Private Networks (VPNs) in which communication occurs between pairs, but the other members of the community have incentive to support their peers. The number of participants in such applications is often small, under 50, and RON takes advantage of these small sizes to more aggressively detect and mask failures.

Network conditions may affect different applications differently. For example network conditions that are fatal to one application may be acceptable and ever better to a more adaptive application. A UDP-based Internet audio application not using good packet-level error correction may not work at all at loss rates larger than 10%. However, at loss rates of 30% or more, TCP becomes essentially unusable because it times out for most packets [32]. In a RON applications are allowed to have independently defined *path metrics* that illustrate the quality of an Internet path. Path metrics include latency, loss rate, throughput etc.

Applications may prefer different metrics in their path selection e.g. latency may be preferred over throughput or bandwidth. A routing system may not be able to optimize all of these metrics simultaneously; for example, a path with one-second latency may appear to be the best throughput path, but this degree of latency may be unacceptable to an interactive application. RON allows application writers to easily construct their own metrics based upon their own measurements and to allow the routing system to make decisions based upon these metrics.

5.4. SUMMARY

RON is a software library such that programs and applications can be linked against it. RON nodes are termed as RON clients. These RON clients are deployed at

the application layer in the underlying Internet substrate. All RON clients are virtually linked. Prober probes the available paths over the parameters; latency, loss rate and throughput to its peers and Internet. When a data packet is to be forwarded then based on the performance measuring parameters best available path is selected either through the Internet or via another RON client.

CHAPTER 6

RON TUNNEL: DESIGN AND IMPLEMENTATION

6.1. INTRODUCTION

RON is a software library and programs are linked against it. These programs are termed as RON clients. RON is defined by a single group of clients that collaborate to provide a distributed service or application. This group of clients can use service-specific routing metrics to decide how to forward packets in the group. In this chapter we have discussed how RON software communicates with its peers, what are the modules in a RON node, how it will monitor the virtual links, formation of latency routing table as a result of probing and maintenance of performance database. Prober maintains the performance database; router retrieves required information from this database as required to forward packets.

6.2. ARCHITECTURE OF RON CLIENT

RON is an application layer overlay network. All the RON nodes are deployed at various locations over the Internet. Figure 6.2 shows our RON architecture. Data comes into the system from RON application that corresponds with the RON software on a node. Our RON architecture is defined by assembly of RON clients that collaborate to provide a distributed service or application. These clients can use application-specific routing metrics when deciding how to forward packets in the group.

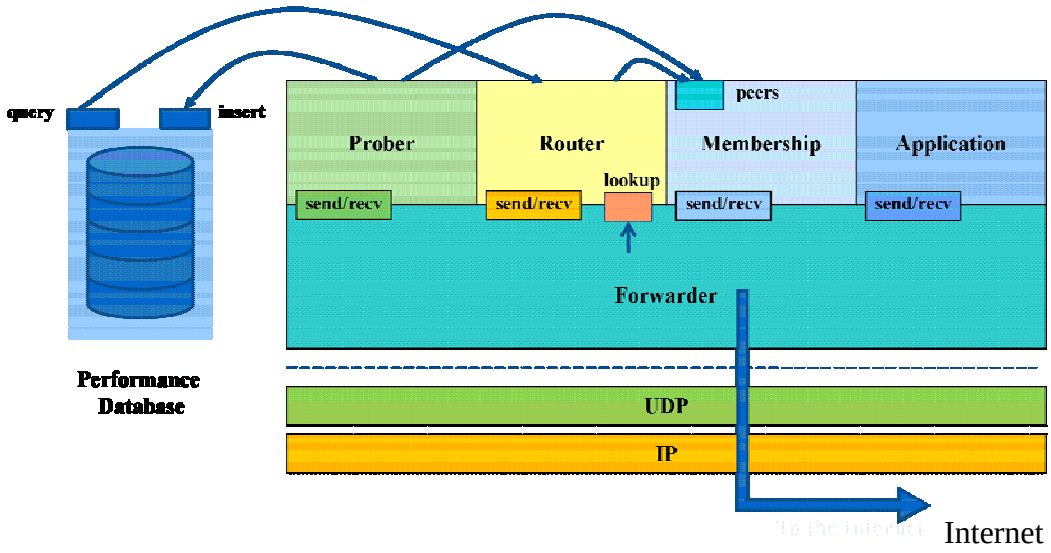

Figure 6.1 Architecture of RON client. Membership manager, router, prober, forwarder and performance database are the main modules.

6.3. RON SOFTWARE SYSTEM ARCHITECTURE

Our design accommodates a variety of RON applications, ranging from a generic IP packet forwarder that improves the reliability of IP packet delivery, to a multiparty conferencing application that incorporates application-specific metrics in its route selection. Figure 6.2 highlights the system structure of RON. A RON client interacts with the RON libraries across an *IP Conduit* built as an API, which it uses to send and receive packets. On the data forwarding path, the first node that receives a packet via the conduit classifies the packet to determine the type of path on which it should be forwarded (e.g., low-latency, high-throughput, etc.). This node is called the *entry node.* It determines a path from its latency routing table and encapsulates the packet into a RON header, tags it with some information that simplifies forwarding by downstream RON nodes, and forwards it on. Each subsequent RON node simply determines the next forwarding hop based on the destination address and the tag. The final RON node that gives the packet to the RON application is called the *exit node*.

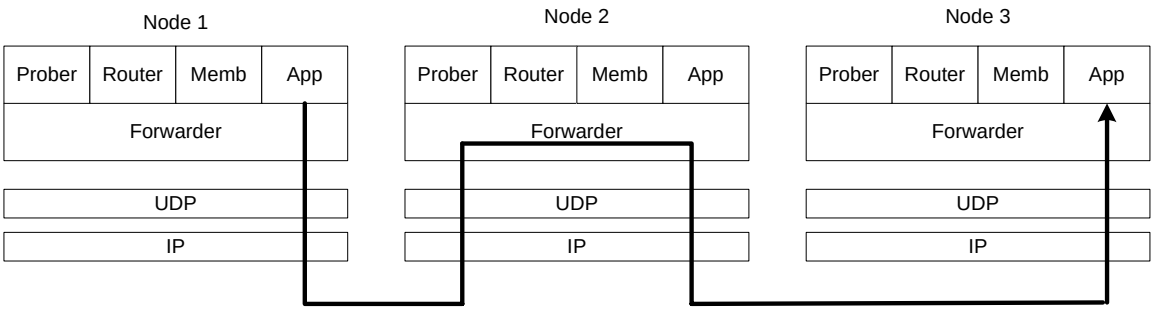

Figure 6.2 RON software system architecture.

6.4. RON BUILDING BLOCKS

Our RON architecture proposed above is composed of 7 major building blocks. These blocks are implemented as system modules. Here is the list of these modules:

1. Conduit
2. Membership Manager
3. Data forwarder
4. Prober
5. Router
6. Policy Classifier
7. Performance Database

Implementation of each one of the modules is given here.

6.4.1. CONDUIT

A RON application interacts with RON modules via conduit, which permits them to send and receive packets. On the data forwarding path, the first node receive the packet through conduit. This entry node' conduit first examines the packet and

asks for the necessary fragmentation to the initiating application if its size exceeds the max payload. Having done that it classifies the packet after having seen the type and hands I over to the data forwarder.

The conduit captures the desired packets as per the defined firewall rules via NDIS API, using *ipCap_IPCaptured(object, IPCapturedEventArgs).* It checks whether the packet contains some data, if not then the packet is discarded using *PassThrough( ).* Packet header is examined to determine the dst and a lookup *ip_find_dst(ip, dst)* is called to search the peers list, if not found the packet is discarded using *PassThrough( ).* Routing flag is determined using *pkt_classify(mypkt.data, mypkt.datalen).* If the data length exceeds the defined size (e.g 1500) then send a message to the sender application via ICMP protocol.

If the packet is in proper order and size then it is send to the forwarder after having set the routing flags and packet type using *dispatch(mypkt, dst, bool via_ron).*

6.4.2. MEMBERSHIP MANAGER

Membership manager is a module that helps the participating nodes to know the presence of each other. In each RON node there is separate independent membership manager looking after the particular needs of each node. All RON clients can have the independent mechanism to apply membership management procedure that is very much useful depending on the application to be processed. Our RON has a static membership mechanism and dynamic. Static mechanism allows selecting a member from the known group of nodes that wish to participate in the routing mechanism and a dynamic membership mechanism through which new nodes are allowed to join the RON they meet the membership criteria. In order to proceed for the routing it is necessary that each one of the clients in RON is connected to at one

another client participating in the routing. The new participating node uses this peer to broadcast its existence to the other nodes already in the RON.

We have implemented a static membership manager as a class with function *recv(pkt, node , bool via_RON)* to receive a message from the peers. Where peerslist has the list of peers of the particular node. When a new RON node wish to become member of an existing node the hosing node save its information to its *peerlist*. RON has the facility of flooding, implemented in *flood_protocol*, to broadcast the message to other members in RON. The flooder is simply another RON client. It registers with the forwarder to receive *TYPE_FLOOD* packets. Each flood packet has a unique 64-bit header; IDs are used to avoid duplication, client node forward packets to its neighbors and then de-capsulate the packet and hand it back to RON for further processing.

Membership manager is extended from Plumber class, which has registration policy for the participating node. The static membership manager is extended from membership class and, on initialization, reads its peers from a file and then simply returns this list upon request. Function named as *peers()* return the list of active RON nodes. When it receives a RON MEMBERSHIP data packet, it updates its list of known peers based upon the information in the packet header (Figure 5.3). This update is refreshed after every *UPDATE_FREQUENCY* time interval.

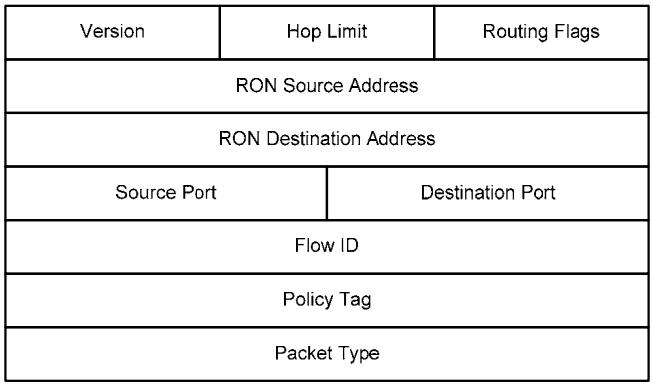

Figure 6.3 The RON packet header.

6.4.3. DATA FORWARDER

Data Forwarder is the basic building block of RON clients. All other modules register themselves with it. Data forwarder has the mechanism of encapsulation and decapsulation to see the contents of the RON header for further decision making process. The encapsulation function has packet (to be sent), source IP address and intermediate RON node to reach the destination. The RON forwarder ties together the RON functions and implements versions of the above functions that send and receive packets to and from the network. It also provides a timer registration and callback mechanism to perform periodic operations, together with a similar service for network socket data availability.

localhost is a class implemented which has function *check_intfs()* to check that if the received packet is designed for itself or for further forwarding. Forwarder object is configured using function *configure(forw).*

All other modules register themselves with the forwarder to *recv()* aparticular king of packets they are interested in. The design of data forwarder is shown in Figure 6.4. Its operation is divided into two modes. Mode one is to *recv()* from conduit that means the forwarder of entry node. Mode two is when acting as a intermediate or exit node and receiving packet from socket via *MessageReceivedCallback(IAsyncResult).*

In mode one, packet type is verified from the *type_dst_table[ptype]* and respective *recv(packet, mynode, bool via_ron)* of type specific module is invoked which after having done the respective operations calls *dispatch(mypkt, dst, bool via_ron)* to send the packet to other nodes.

In mode two, on receiving the packets it first examines its RON packet header and determines whether it is predestined for the local client or a remote node. If it needs further delivery, the forwarder passes the RON packet header to the routing

table, the routing table lookup completes in three steps. The first step examines the policy tag, and locates the proper routing preference table. There is one routing preference table for each known policy tag. Then, the lookup procedure checks the routing preference flags to find a compatible route selection metric for the packet. There is dedicated routing table for each metric. The next hop is determined from the specific table and then based on that it is send to the next hop for further delivery.

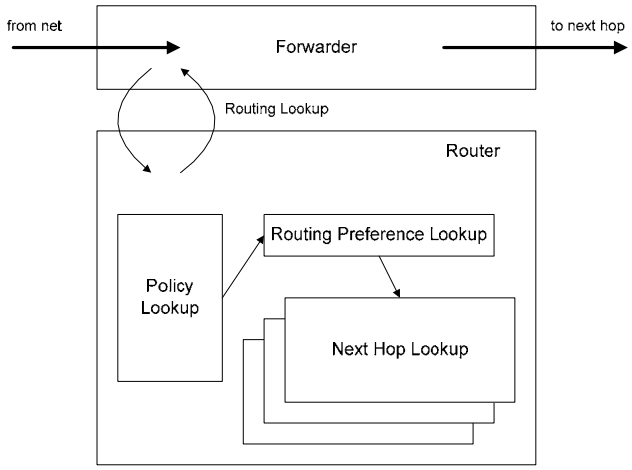

Figure 6.4 Data forwarder and routing mechanism

6.4.4. PROBER

Prober monitors the virtual links between participating RON nodes. Probes results are periodically inserted the performance database client. Each RON node in an N-node RON monitors its $N - 1$ virtual links using periodic probes. The Prober maintains a copy of the active prober component maintains a copy of a *peers* table with a time-decrementing *TIMER_INTERVAL* field per peer. When this field expires, the prober sends a small UDP probe packet to the remote peer. The process used by the probe protocols is shown in Figure 6.5. The prober relies upon a timer to perform the following periodic actions: (i) Check for outstanding packets that have expired,

(ii) Select the node that was probed the longest time ago and, (iii) If that time is longer than the probe interval, send a probe.

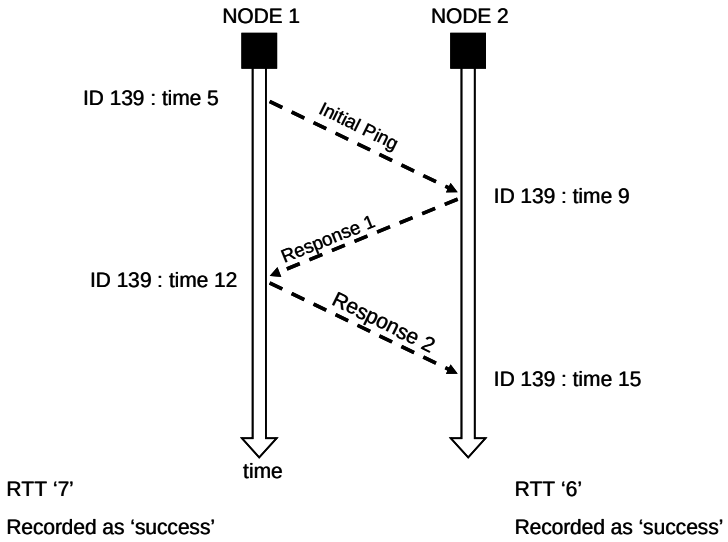

Figure 6.5 Probing mechanism. With three packets, both participants get an RTT sample.

The prober contacts each other host every *TIMER_INTERVAL* seconds, plus a random additional offset of *1/3*TIMER_INTERVAL* seconds. If the packet does not come back within *PING_TIMEOUT* seconds, then the prober deems it lost and records a loss event. To speed outage detection, when a packet is deemed lost, the prober will immediately send another, up to a maximum of four fast probes.

Three types of pings are sent, *LOSS, REACH and LAT* are sent using *send_ping( )* method and their results are inserted in the performance database as per their type in the appropriate performance table.

6.4.5. ROUTER

RON routers are aware of the basic system metrics of latency, loss, and throughput; this proposes a way for a public RON environment to support users who may also have client-defined metrics, and still provide them with good default metrics to use. A lookup in the routing preference table leads to a hash table of next-hops based upon the destination RON node. The entry in the next-hop table is returned to the forwarder, which places the packet on the network predestined for the subsequent node.

The *router* component of RON is composed of three sub-components. The function of these three sub-components is described below.

6.4.5.1. APPLICATION SPECIFIC ROUTING

By default, RON *router* maintains information about three specific metrics for each virtual link: (i) latency, (ii) packet loss rate, and (iii) throughput. RON clients can override these defaults with their own metrics, and the RON library constructs the appropriate forwarding table to pick good paths. The router maintains forwarding tables for each combination of a routing policy and a routing metric.

RON routing considers hops only through a single other node when making routing decisions. This decision allows RON to use non-linear routing metrics like "throughput," which standard shortest-paths algorithms cannot handle.

6.4.5.2. LINK STATE TABLE DISSEMINATION

The most of the switching nodes in a network being implemented today use link-state routing protocol to disseminate topology information between routers, which is in turn used to build the forwarding tables in our case these are latency routing tables. For example consider an N-node RON. Each RON node will have *N* -

1 virtual links. Each node's router periodically requests summary information of the different performance metrics to the *N - 1* other nodes from its local performance database and disseminates its view to the others. Performance database is kept up-to-date by prober, using the probing mechanism to explore routing paths. Router keeps sending updates every *ROUTING_INTERVAL*. The node then propagates these values to the neighboring nodes. This information is sent via the RON forwarding mesh itself, to ensure that routing information is propagated in the event of path outages and heavy loss periods. Thus, the RON routing protocol is itself a RON client, with a well-defined RON packet type. This approach ensures that the only time a RON router has incomplete information about another RON router is when all paths in the RON from one node to another are unavailable.

6.4.5.3. PATH EVALUATION AND SELECTION OF BEST AVAIL PATH

A set of algorithms are devised to select the potential paths for data forwarding. For this purpose several quantifying the reliability of the path, metric evaluators are devised for this purpose. Point to focus here is the method to select the path for data sending from one RON node to other and the path evaluation mechanism that will ensure the successful data transfer. Every RON router implements outage detection, which it uses to determine if the virtual link between it and another node is still working. It uses the active probing mechanism for this, if the last *OUTAGE THRESH* probe packets were all lost, then an outage is flagged. Paths experiencing outages are rated on their packet loss rate history; a path having an outage will always lose to a path not experiencing an outage.

By default, every RON router implements three different routing metrics: the latency-minimizer; the loss-minimizer, and the throughput optimizer. RON does not attempt to find optimal throughput paths, but strives to avoid paths of low throughput

when good alternatives are available. The performance of bulk TCP transfers is a function of the connection's round-trip latency and the packet loss rate it observes. Throughput optimization combines the latency and loss metrics using a simplified version of the TCP throughput equation [32]. The granularity of loss rate detection is 1%, and the throughput equation is more sensitive at lower loss rates. We set a minimum packet loss rate of 2% to prevent infinite bandwidth, and to prevent large oscillations from single packet losses. Calculation is based on a simplified TCP throughput Equation 6.1:

$$X - \frac{1}{rtt * sqrt(2 * \frac{p}{3})} \quad \text{..... 6.1}$$

Three types of router to evaluate the path for three kinds of metrics, latency, loss and throughput have been implemented with a main router class to receive a packet and build the forwarding table based on policy and metrics. The algorithm that builds the routing table is shown in Figure 6.6.

```
BuildForwardingTable (policy p, metric m, List<peers>)
{
foreach p  in policy
    foreach m  in metric
            foreach dst in peers
                    foreach hop in peers
                            if      p.permits(me;hop      &&
                    p.permits(hop;dest)
                            {
                                    score = m.eval(me;hop;dst);
                                    if ( score > best score)
                                    {
                                            best score = score;
                                            next hop = hop;
                                    }
                            }
                             table[p][m][dst] = next hop;
}
```

Figure 6.6 Algorithm to build forwarding table

6.4.6. POLICY ROUTING

Policy routing defines the type of traffic allowed on particular network links. On the Internet, “type” is typically defined only by its source and destination addresses. RON extends this notion to a more general notion of packet type. RON separates policy routing into two components: c*lassification* and *routing table formation*. Packets are given a policy tag when they enter the RON, and this policy tag is used to perform lookups in the proper set of routing tables. A separate set of routing tables is constructed for each policy by re-running the routing computation without links disallowed by the policy.

The *policy classifier* component provides a policy tag, a data classifier that answers the question, “Is this packet my type?”, and a *permits* function to tell the router if the policy permits the use of a particular link.

6.4.7. PERFORMANCE DATABASE

Performance database is a repository for pre-defined metrics i.e. latency, throughput and loss rate. It allows users to access and make an analysis to take certain decisions and to predict future. Over the Internet we don’t experience similar network conditions all the time when communicating with other hosts. The RON performance database may or may not be shared. And it is unrealistic to send large performance histories to all participants in the RON. For example an outage detector may want to know how many packets were successfully sent in the last 10 seconds, but a throughput analyzer may be interested in a longer-term packet loss average. So an efficient database summarization mechanism is needed to have better performance.

We have implemented RON performance database as an application running at local RON node. Router communicates with the performance database to evaluate

the best next hop based on specific policy and metric. First of all Probes insert data into the database by calling *insert(node , probetype, value).* To avoid maintaining hard state in the database, the caller specifies a type field that tells the database what kinds of values will be inserted. Clients (router) that want information from the database make a request via *query(node , probetype, summarizer , param, result).* Based on the information obtained as a result of probing a latency routing table is maintained along with other parameters like loss rate and throughput. Whenever a data is to be retrieved for the database a query is made as a request from the client and performance database make a summary of the queried data and replies as call back. This mechanism is described in the Figure 6.7.

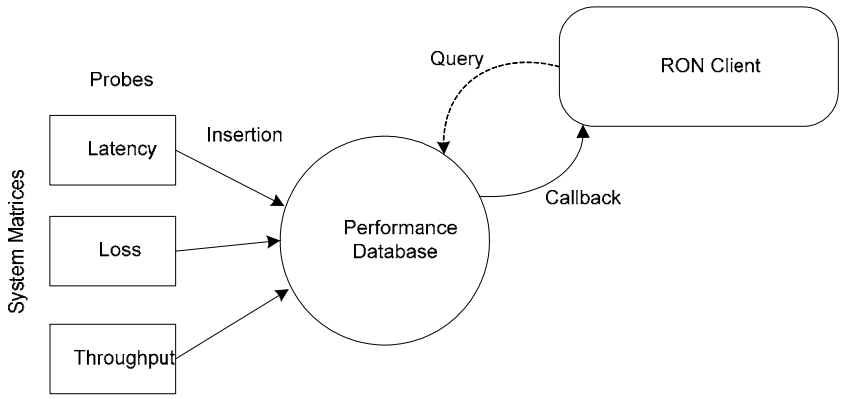

Figure 6.7 RON performance database

Table 6.1 Summarizer methods supported by PDB

Summarizer_Type	Description
SUMMARIZE_EWMA	Exponential weighted moving average with parameter of the entries.
SUMMARIZE_MAX	Largest entry of the last N entries
SUMMARIZE_MIN	Smallest entry of the last N entries
SUMMARIZE_FIRST	Earliest first entry of the last N entries
SUMMARIZE_RECENT	Most recent entry of the last N entries
SUMMARIZE_AVG	Average of last N entries
SUMMARIZE_MEANDEV	Mean deviation last N entries

6.5. RON TUNNEL

Once collaborating RON nodes starts communicating with each other and virtual links are monitored as per the implementation above it happens to establish RON Tunnel amongst RON Nodes.

The RON tunnel takes IP packets off of the wire, sends them via RON, and then transmits them from the exit RON node to their destination. This application improves IP packet delivery without any modification to the transport protocols and applications running at end-nodes. The detailed architecture of the RON tunnel is shown in Figure 6.8.

We have implemented the IP tunnel using NDIS API from high performance packet filtering Winpktfilter framework, to automatically receive IP traffic and divert it to RON, and emit it at the other end via UDP socket. NDIS API permits us to

capture raw IP packets that match our firewall rules, modify the packets by adding RON packet header, and retransmit them. The RON tunnel provides classification, encapsulation, and decapsulation of IP packets through a special module called the Conduit. The tunnel first configures the system's IP firewall to divert applicable packets to the tunnel application.

A selective set of policies implemented for the purpose of forwarding packet when data movement is within the participants of the RON. The result of this data movement is to minimize the amount of data processing. The RON tunnel then waits incoming packets using *MessageReceivedCallback (IAsyncResult).* On receiving the it wraps the incoming IP packet with a RON packet header and passes control of it to the RON forwarder; from this point, it is handled by the RON routing until it reaches the destination tunnel application. The destination tunnel separates the RON packet and sends the IP packet using a raw socket.

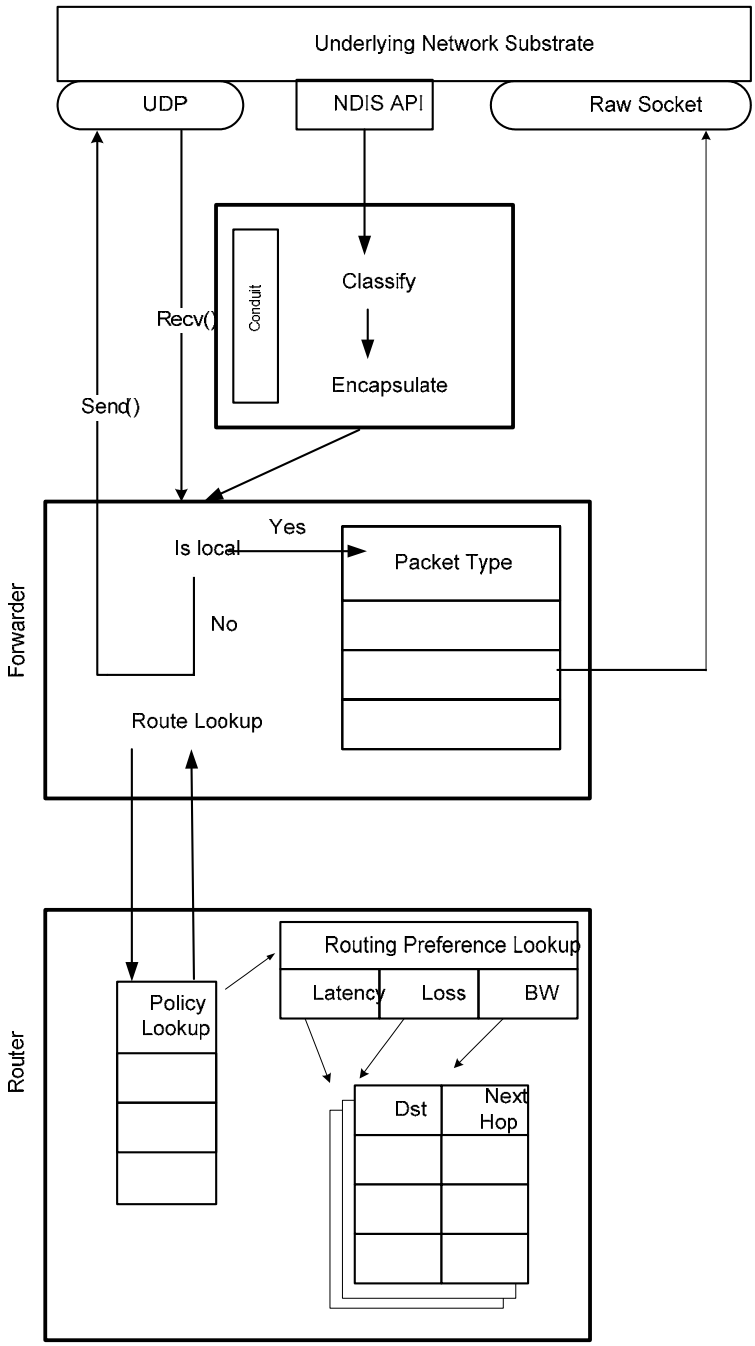

Figure 6.8 The implementation of the IP encapsulating RON client.

6.6. SUMMARY

This chapter provides detail infrastructure and implementation of the significant building blocks that make up a RON client in a RON. These Building blocks or modules that are essential for the deployment of RON over the Internet; are Conduit, Membership Manager, Data Forwarder, Prober, Router and Performance

Database. Implementation of these modules is described briefly but quite comprehensively, covering all the important aspects of different classes and structures. Chapter also covers how data is routed in RON over the existing Internet substrate, how routing tables are updated based on the performance data base best available path is selected and routed either via another RON node or through the Internet. Performance database keeps up-to-date information available for the clients. To retrieve data based on certain parameters like latency, throughput and loss rate, router client sends are in the form of requests and performance database replies them as callback to the query.

ANNEX A

ACTIVITY DIAGRAMS

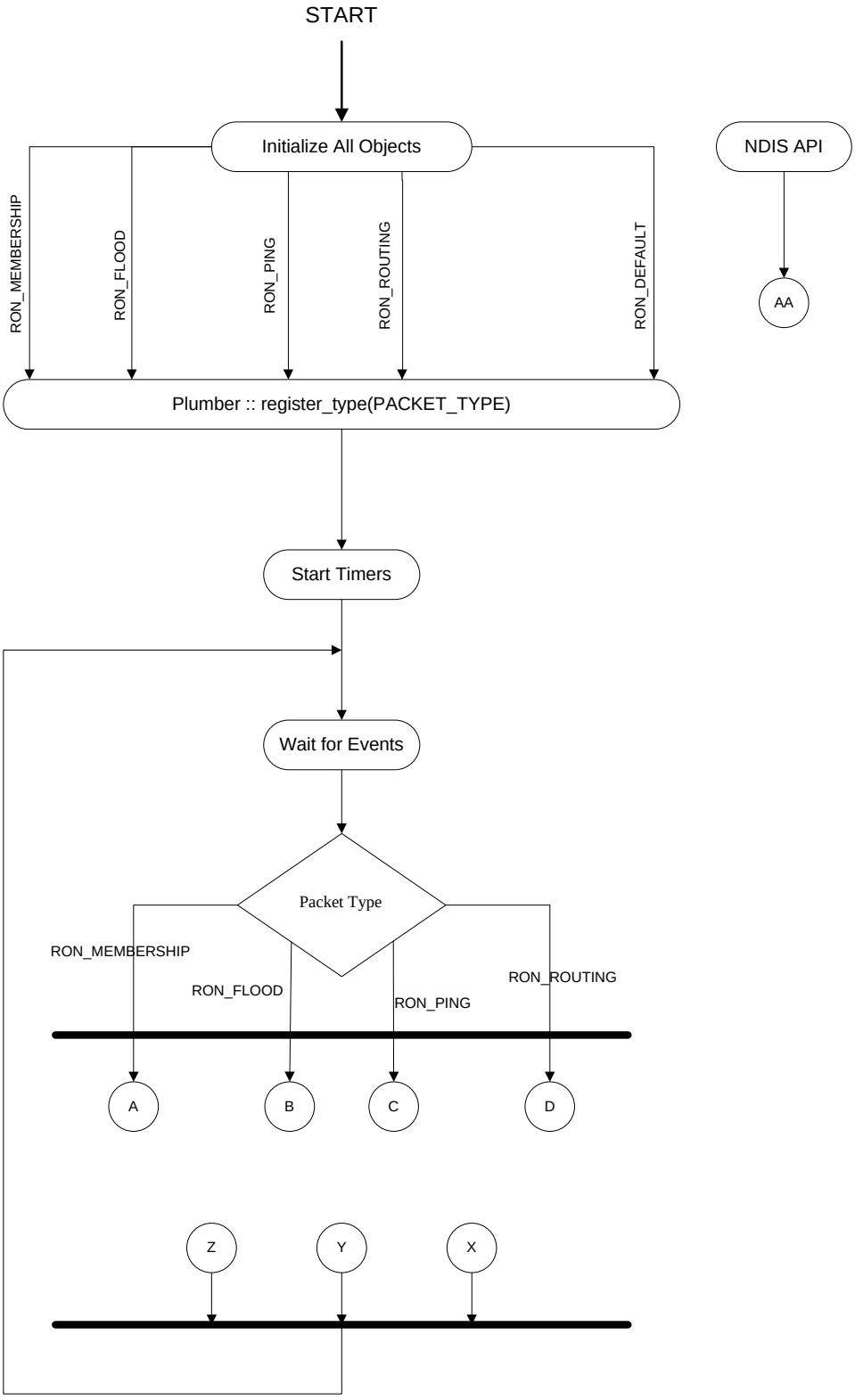
START
Initialize All Objects
NDIS API
RON_MEMBERSHIP
RON_FLOOD
RON_PING
RON_ROUTING
RON_DEFAULT
AA
Plumber :: register_type(PACKET_TYPE)
Start Timers
Wait for Events
Packet Type
RON_MEMBERSHIP
RON_FLOOD
RON_PING
RON_ROUTING
A
B
C
D
Z
Y
X

Figure A.1

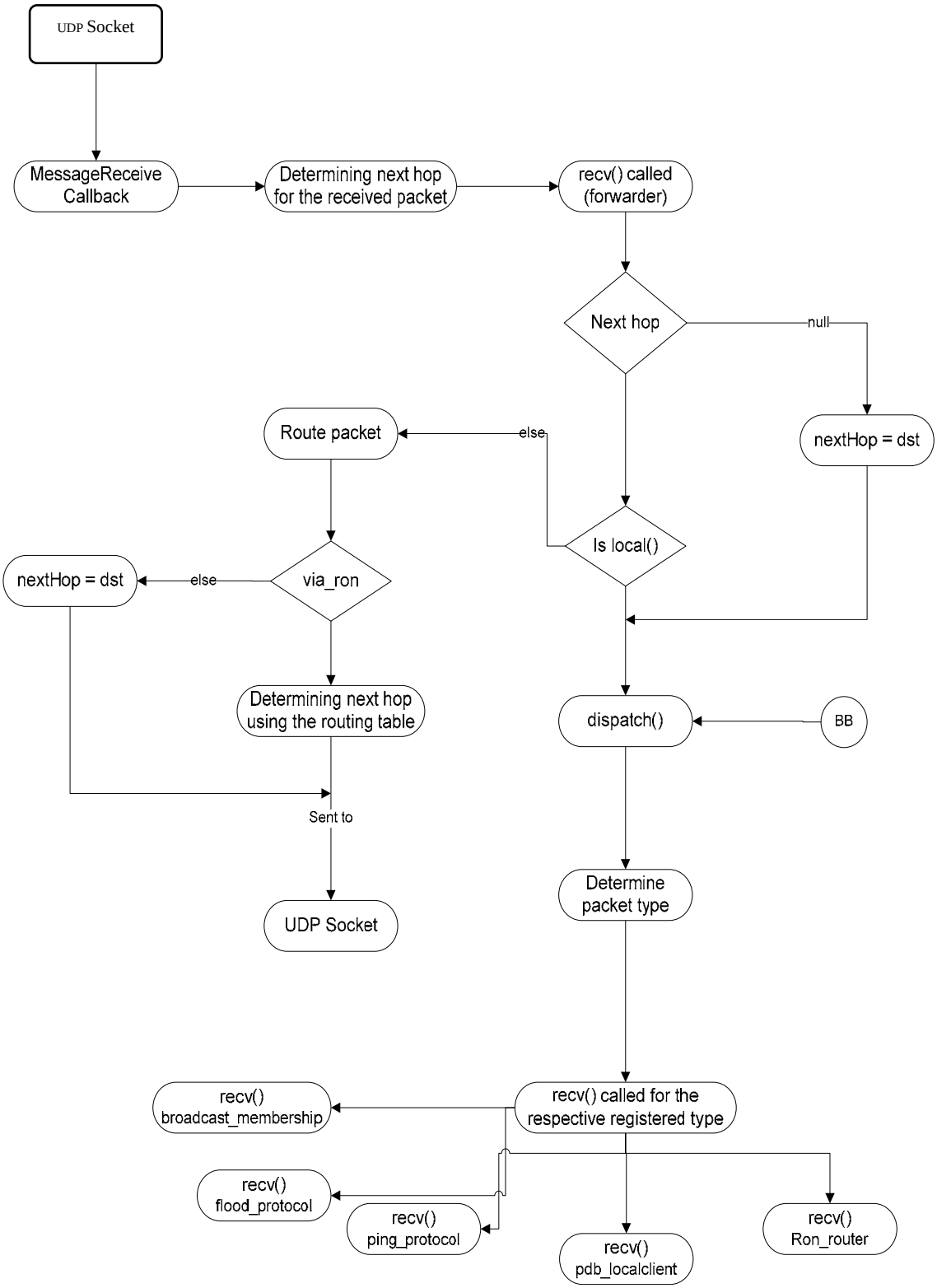
UDP Socket
MessageReceive Callback
Determining next hop for the received packet
recv() called (forwarder)
Next hop
null
nextHop = dst
Route packet
else
Is local()
via_ron
else
nextHop = dst
Determining next hop using the routing table
dispatch()
BB
Sent to
UDP Socket
Determine packet type
recv() called for the respective registered type
recv() broadcast_membership
recv() flood_protocol
recv() ping_protocol
recv() pdb_localclient
recv() Ron_router

Figure A.2

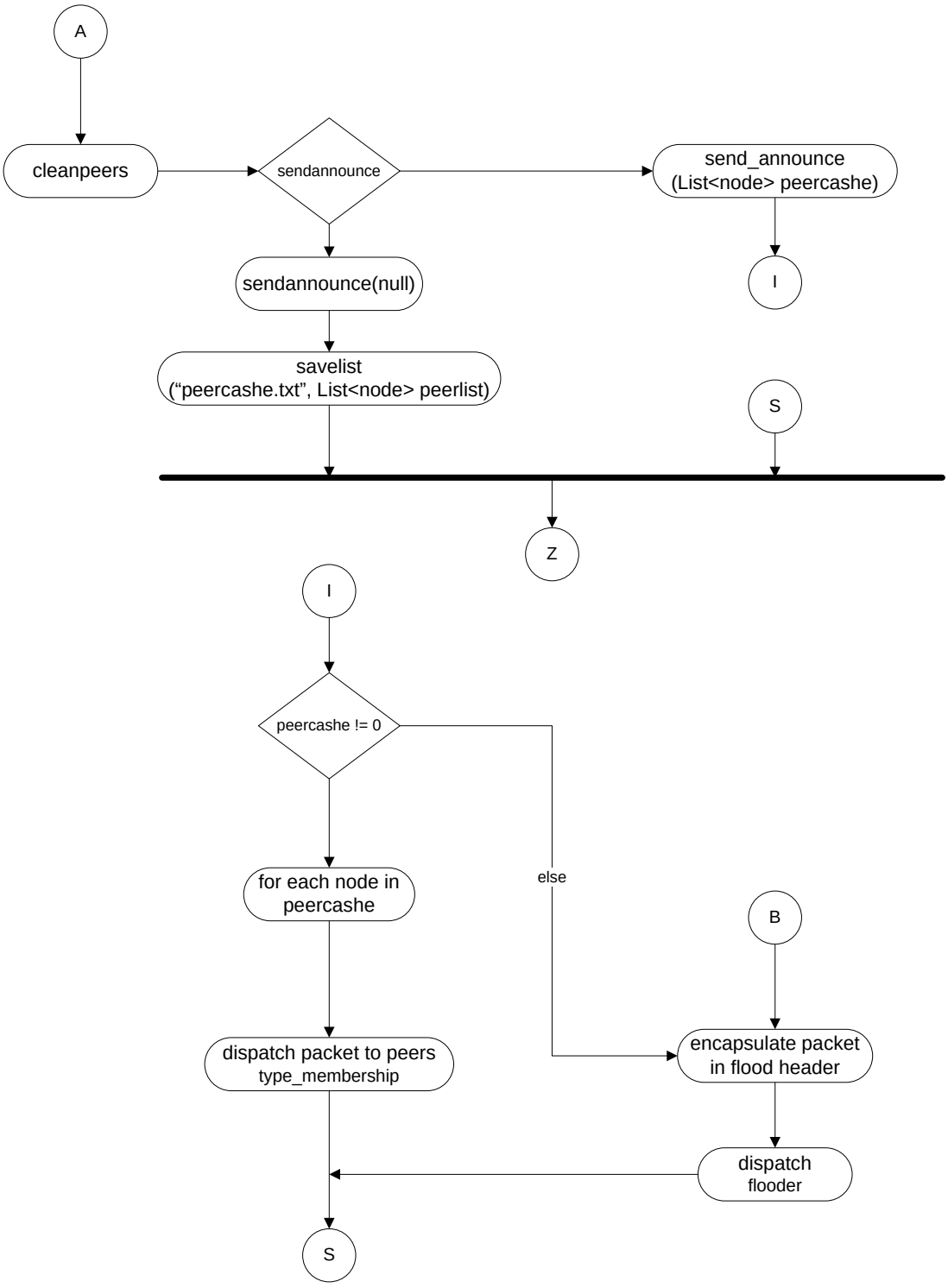
A
cleanpeers
sendannounce
send_announce
(List<node> peercashe)
sendannounce(null)
I
savelist
(“peercashe.txt”, List<node> peerlist)
S
Z
I
peercashe != 0
else
for each node in
peercashe
B
dispatch packet to peers
type_membership
encapsulate packet
in flood header
dispatch
flooder
S

Figure A.3

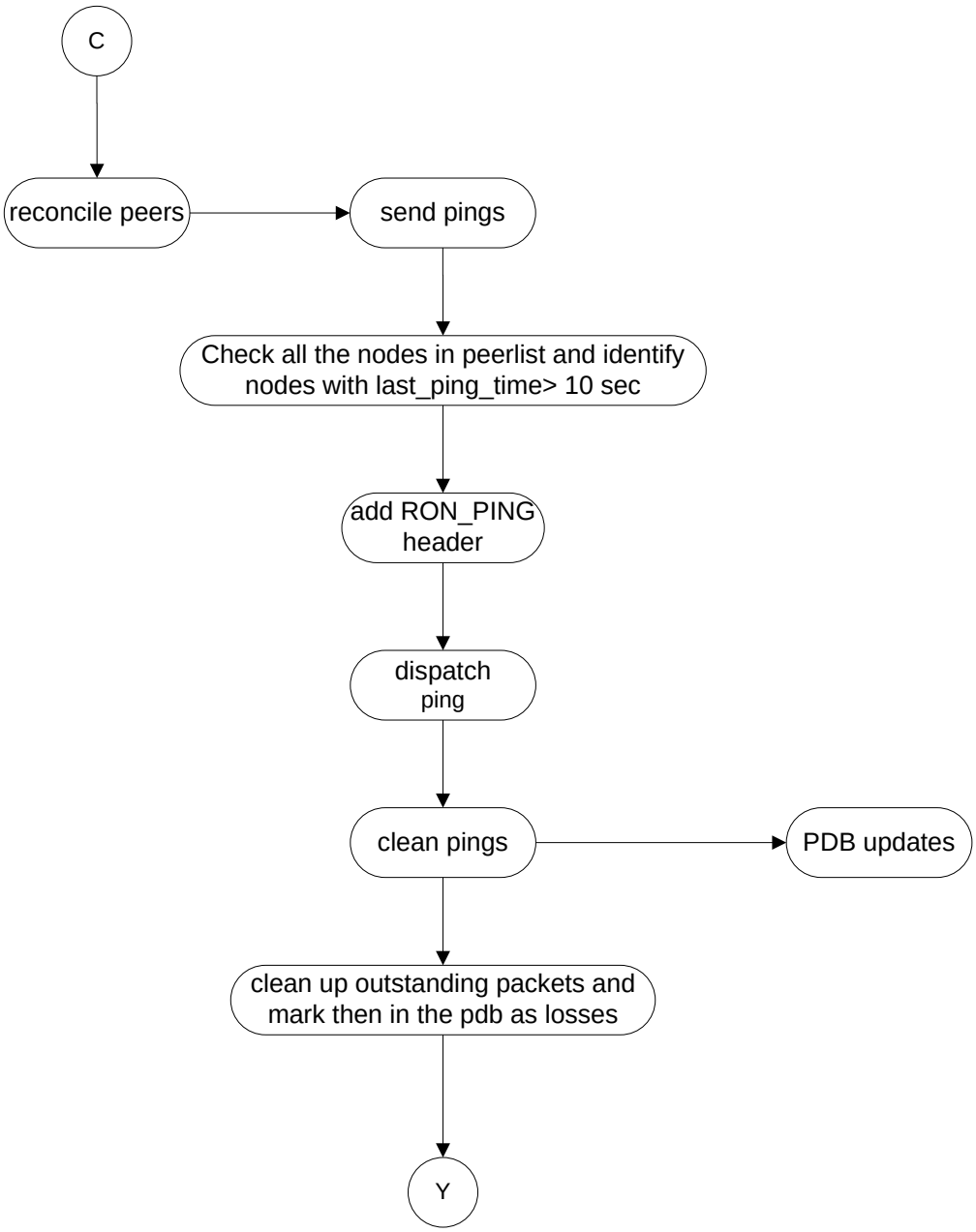
C
reconcile peers
send pings
Check all the nodes in peerlist and identify nodes with last_ping_time> 10 sec
add RON_PING header
dispatch ping
clean pings
PDB updates
clean up outstanding packets and mark then in the pdb as losses
Y

Figure A.4

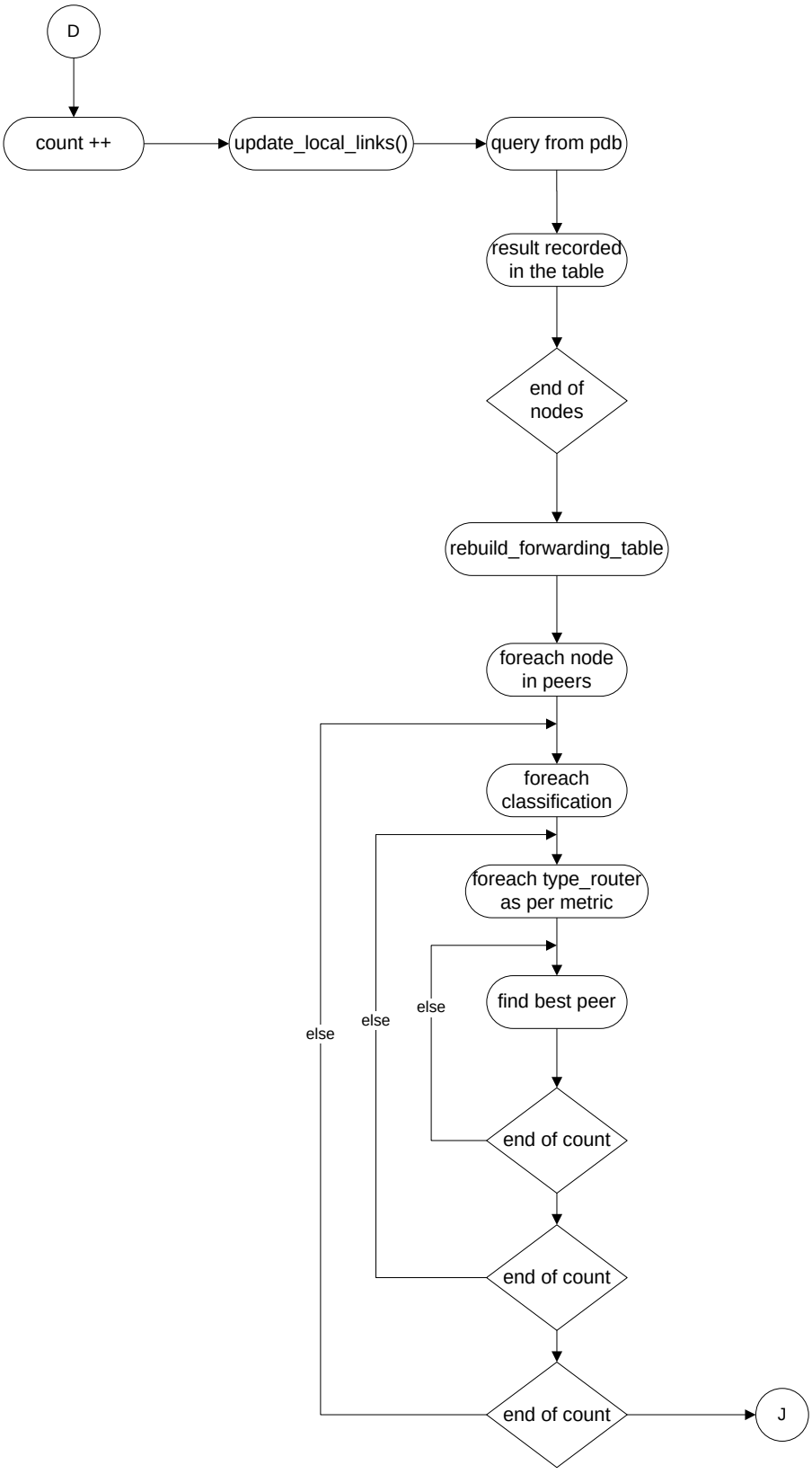
D
count ++
update_local_links()
query from pdb
result recorded in the table
end of nodes
rebuild_forwarding_table
foreach node in peers
foreach classification
foreach type_router as per metric
find best peer
end of count
end of count
end of count
J
else
else
else

Figure A.5

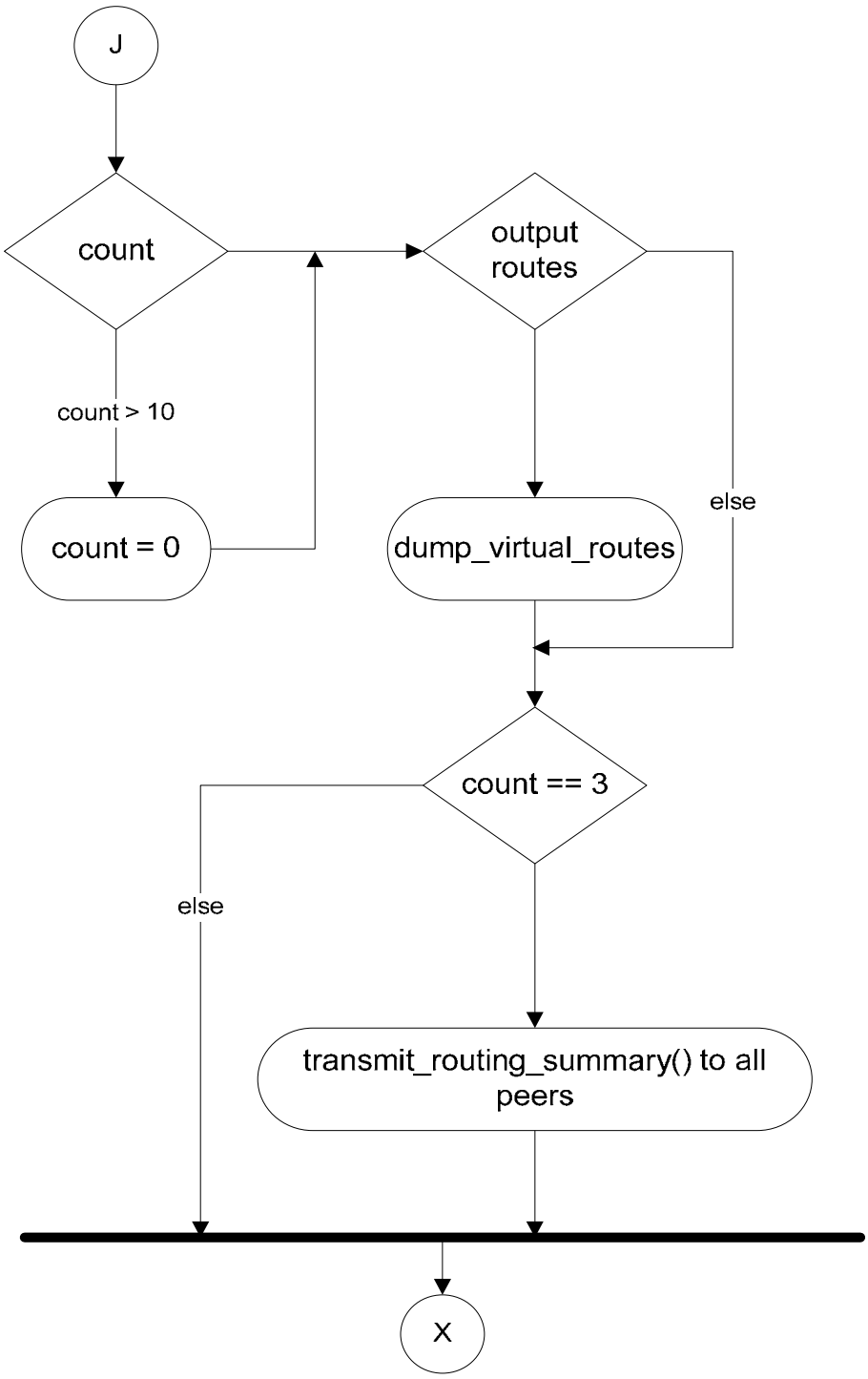
J
count
output routes
count > 10
count = 0
dump_virtual_routes
else
count == 3
else
transmit_routing_summary() to all peers
X

Figure A.6

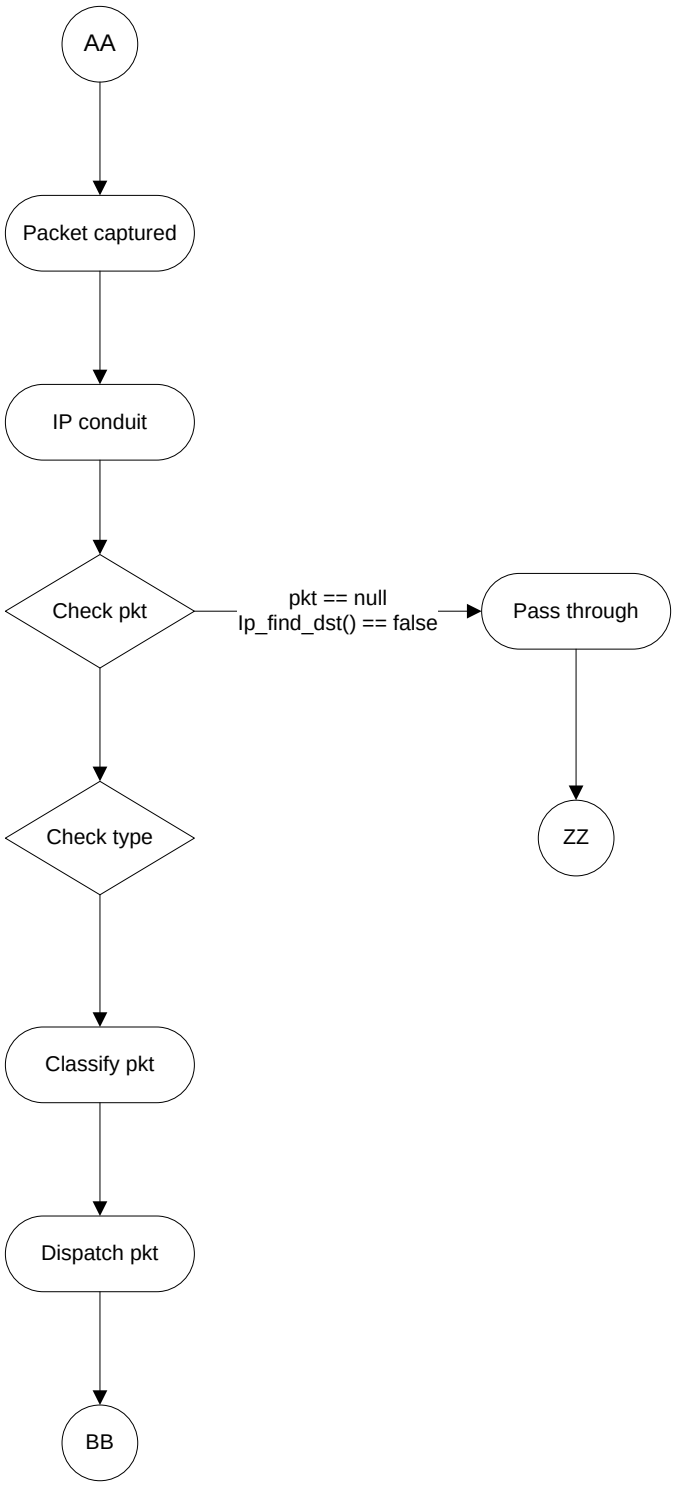
AA
Packet captured
IP conduit
Check pkt
pkt == null
Ip_find_dst() == false
Pass through
ZZ
Check type
Classify pkt
Dispatch pkt
BB

Figure A.7

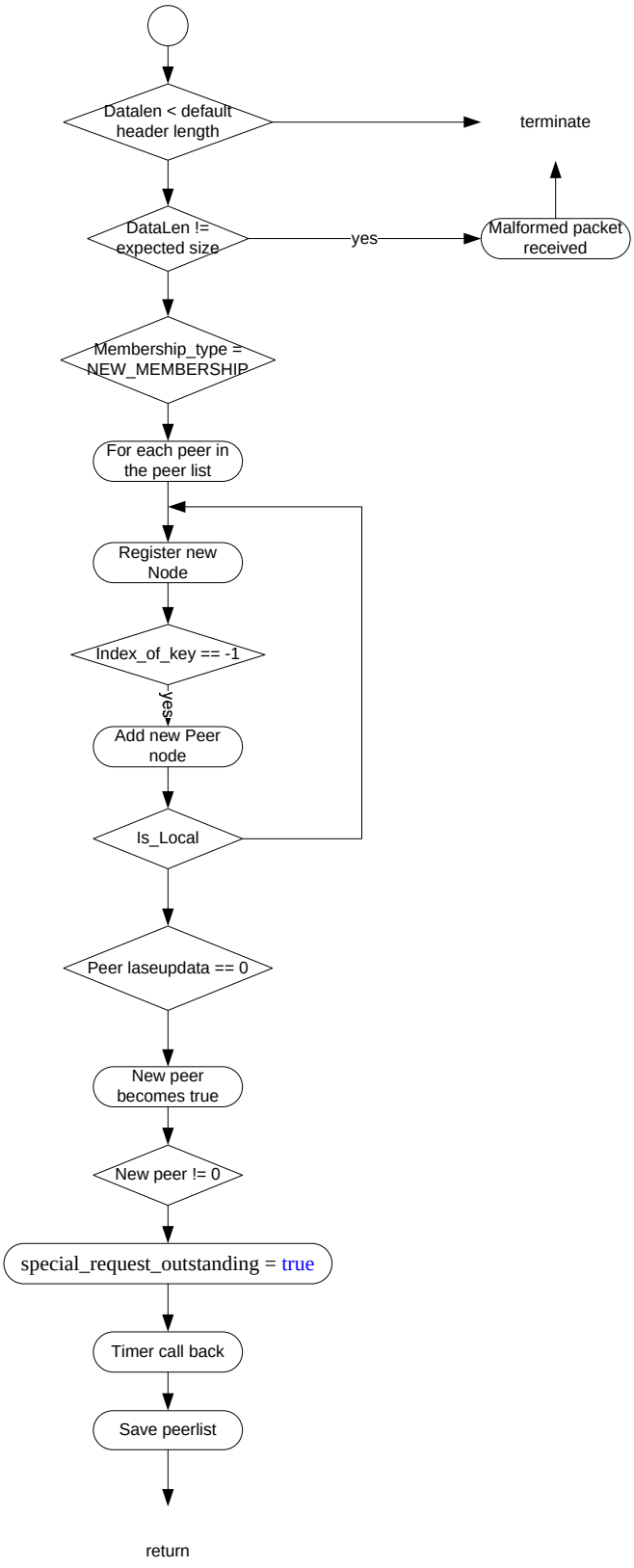

Figure A.8 broadcast_membership :: *recv()*

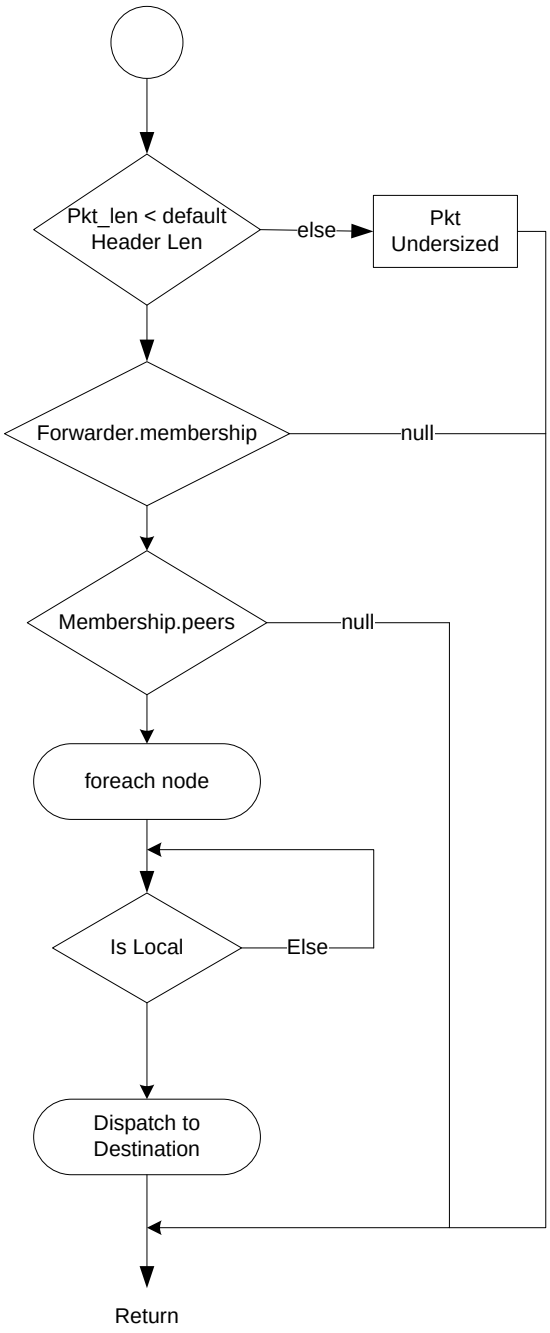

Figure A.9 flood_protocol :: *recv()*

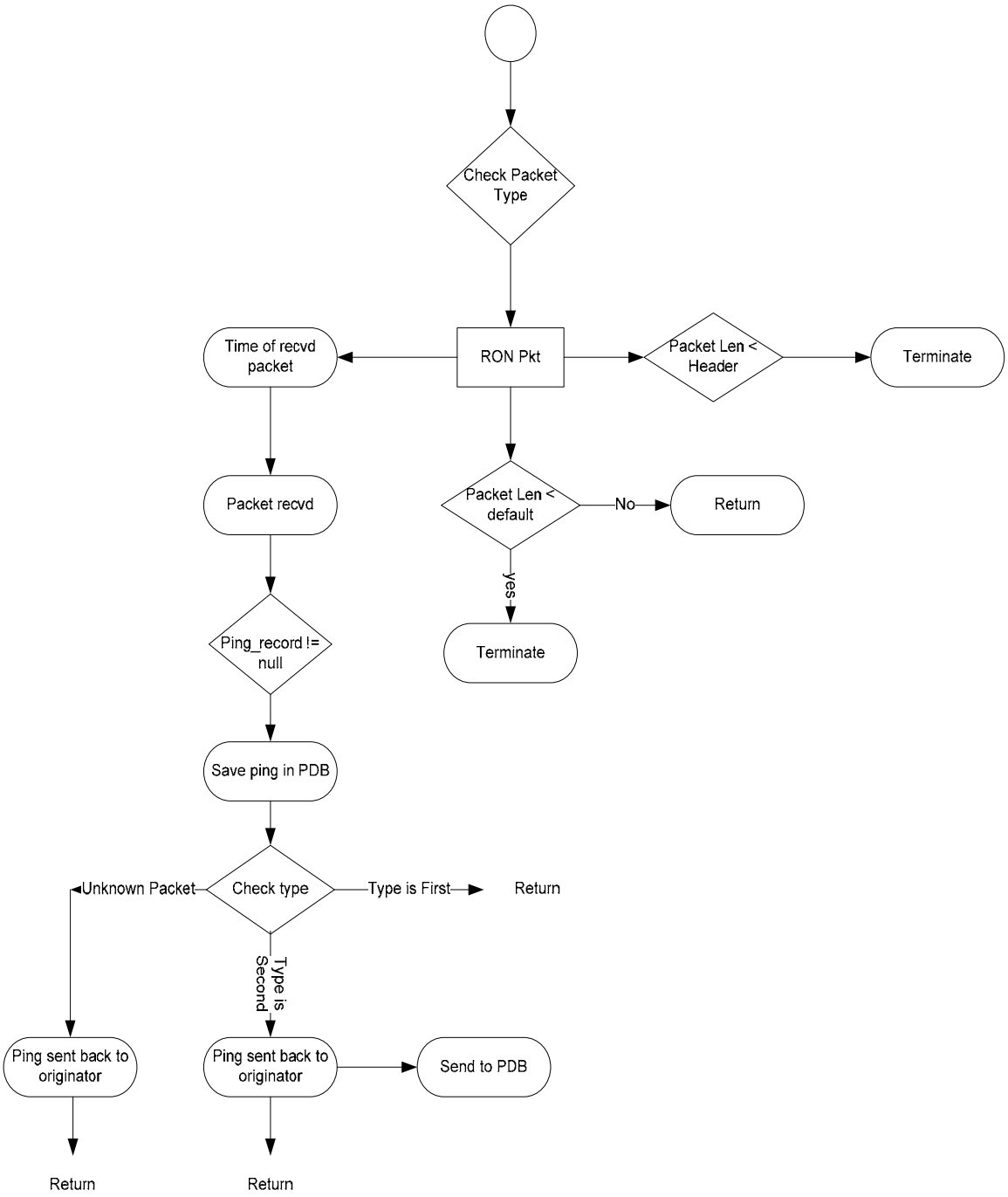

Figure A.10 ping_protocol :: *recv()*

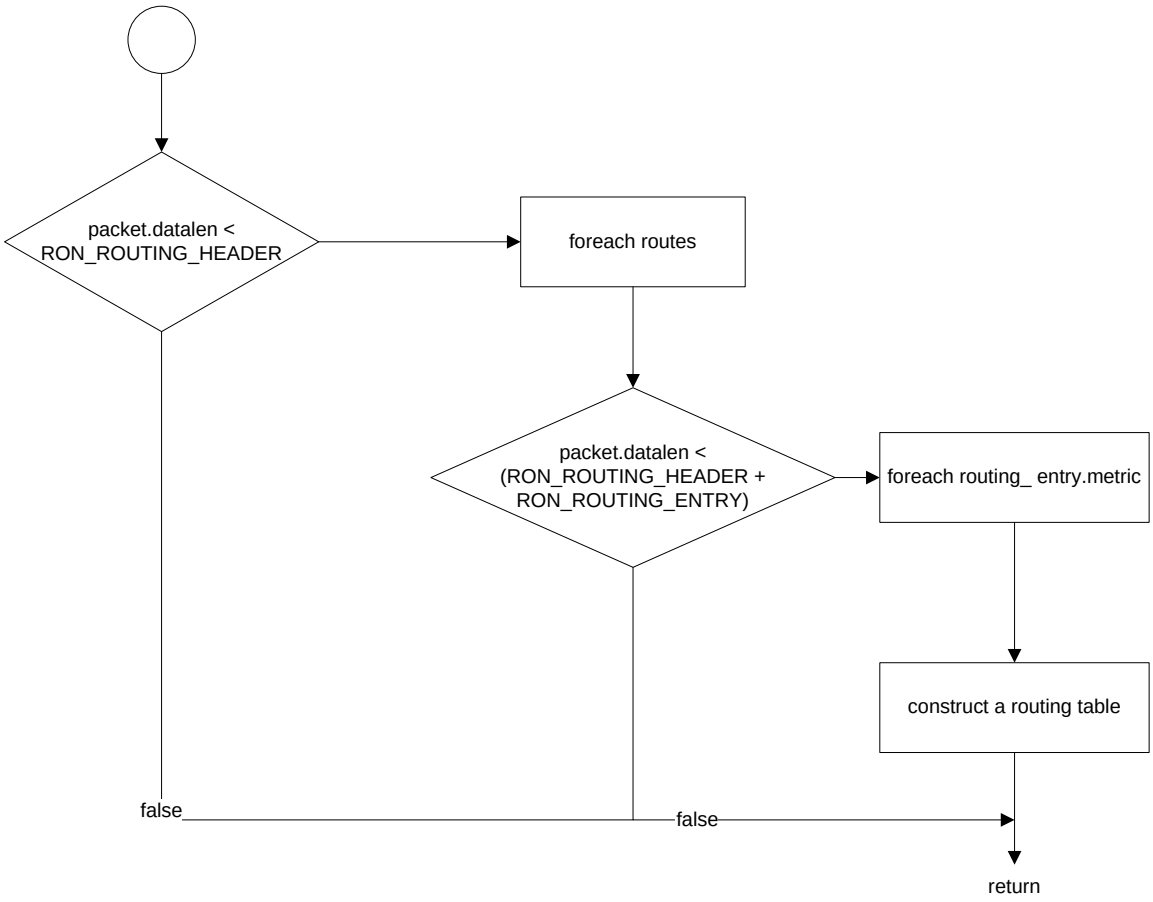

Figure A.11 ron_router :: *recv()*

ANNEX B

DEFINITIONS, ACRONYMS AND ABBREVIATIONS

VPN: Virtual Private Network

RON: Resilient Overlay Network

AS: Autonomous System

BGP: Border Gateway Protocol

ISP: Internet Service Provider

IPv6: Internet Protocol version 6

MCS: Military College of Signals

EME: College of Electrical and Mechanical Engineering

NUST: National University of Sciences and Technology

RAS: Remote Access Server

WAN: Wide Area Network

QoS: Quality of Service

GRE: General Routing Encapsulation

PPTP: Point to Point Tunneling Protocol

L2F: Layer 2 Forwarding

L2TP: Layer 2 Tunneling Protocol

L2Sec: Layer 2 Security Protocol

SSL: Secure Sockets Layer

TLS: Transport Layer Security

IPSec: Internet Protocol Security

NAP: Network Access Point

IP: Internet Protocol

DNS: Domain Name Service

ALMI: Application Level Multicast Infrastructure

TCP: Transmission Control Protocol

UDP: User Datagram Protocol

NDIS: Network Driver Interface Specification

API: Application Programming Interface

mypkt: my packet (Function)

forw: Forwarder (Function)

RTT: Round-trip Time

BIBLIOGRAPHY

[1] Improving End-to-End Availability Using Overlay Networks by David Godbe Andersen; Available at: http://www.cs.cmu.edu/~dga/papers/andersen-phd-thesis.pdf

[2] H. Eriksson. Multicast backbone: The Multicast Backbone. Communications of the ACM, 37(8):54–60, 1994.

[3] Hans Erikson M6ONE: The Multicast Backbone. August 1994/Vol.37, No.8 Communications of the ACM; Available at: http://netlab.cs.iitm.ernet.in/publications_files/pub/selva/seacomm96.ps

[4] I.Guardini, P. Fasano, and G. Girardi. Ipv6 operational experience within the 6bone. 2000. Available at: http://www.springerlink.com/index/RE6QJR12L3H9JWKL.pdf

[5] J. Touch and S. Hotz. The X-Bone. In Proc. Third Global Internet Mini-Conference in conjunction with Globecom ’98, Sydney, Australia, November 1998.

[6] Yoid?? Extending the Internet Multicast Architecture Paul Francis ACIRI francis@aciri.org, www.aciri.org April 2,2000. Available at: http://www.cs.cornell.edu/People/francis/yoidArch.pdf

[7] Dimitrios Pendarakis, Sherlia Shi, Dinesh Verma, and Marcel Waldvogel. Almi: An application level multicast infrastructure. In Proceedings of the 3rd USENIX Symposium on Internet Technologies and Systems (USITS), pages 49–60, 2001.

[8] I. Clarke, O. Sandbert, B.Wiley, and T. Hong. Freenet: A distributed anonymous information storage and retrieval system. In Proc. the Workshop on Design Issues in Anonymity and Unobservability, Berkeley, CA, July 2000.

[9] Michael J. Freedman and Robert Morris. Tarzan: A peer-to-peer anonymizing network layer. In Proc. 9th ACM Conference on Computer and Communications Security, Washington, D.C., November 2002.

[10] Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: The second-generation onion router. In Proc. 13th USENIX Security Symposium, San Diego, CA, August 2004.

[11] R. Dingledine, M. Freedman, and D. Molnar. The Free Haven Project: Distributed anonymous storage service. In Proc.Workshop on Design Issues in Anonymity and Unobservability, Berkeley, CA, July 2000.

[12] Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, and Hari Balakrishnan. Chord: A scalable peer-to-peer lookup service for Internet applications. In Proc. ACM SIGCOMM, San Diego, CA, August 2001

[13] Antony Rowstron and Peter Druschel. Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems. In Proc. 18th IFIP/ACM International Conference on Distributed Systems Platforms, November 2001.

[14] Ben Y. Zhao, Ling Huang, Jeremy Stribling, Sean C. Rhea, Anthony D. Joseph, and John D. Kubiatowicz. Tapestry: A resilient global-scale overlay for service deployment. IEEE Journal on Selected Areas in Communications (J-SAC), 22(1), January 2004.

[15] Dejan Kostic, Adolfo Rodriguez, Jeannie Albrecht, and Amin Vahdat. Bullet: High bandwidth data dissemination using an overlay mesh. In Proc. 19th

ACM Symposium on Operating Systems Principles (SOSP), Lake George, NY, October 2003.

[16] Frank Dabek, M. Frans Kaashoek, David Karger, Robert Morris, and Ion Stoica. Widearea cooperative storage with CFS. In Proc. 18th ACM Symposium on Operating Systems Principles (SOSP), Banff, Canada, October 2001.

[17] Virtual Pivate Network, Various Services and implementation Scenarios, By R. Venkate Swaran. February/March ieee 2001 0278-6648/01/.

[18] OpenVPN-Building and Integrating Virtual Private Networks, By Markus Feilner

[19] Algorithms for Provisioning Virtual Private Networks in the Hose Model Amit Kumar, Rajeev Rastogi, Avi Silberschatz, Fellow, IEEE, and Bulent Yener.

[20] N. G. Duffield, P. Goyal, A. Greenberg, P. Mishra, K. K. Ramakrishnan, and J. E. van der Merwe, “A flexible model for resource management in virtual private networks,” in Proc. ACM SIGCOMM, 1998, pp. 95–108.

[21] Virtual Private Networks, Second Edition. By Charlie Scott, Paul Wolfe, Mike Erwin. Publisher: O'Reilly

[22] Data communications and Networking By Behrouz A. Forouzan pages 15-18, 557-560; Third Edition.

[23] D. G. Andersen, H. Balakrishnan, and M. F. K. R. Morris. Resilient overlay networks. In Operating Systems Review, pages 131–145, December 2001.

[24] Measuring the Effects of Internet Path Faults on Reactive Routing Nick Feamster, David G. Andersen, Hari Balakrishnan, and M. Frans Kaashoek

MIT Laboratory for Computer Science 200 Technology Square, Cambridge, MA 02139 feamster,dga,hari,kaashoekg@lcs.mit.edu

[25] The Latest in Virtual Private Networks: Part I, By Chris Metz • Cisco Systems chmetz@cisco.com

[26] Dalayed Internet Routing Convergence. Craig Labovitz,Abha Ahuja, Abhijit Bose. Available at: http://delivery.acm.org/10.1145/350000/347428/p175-labovitz.pdf?key1=347428&key2=5674524021&coll=GUIDE&dl=GUIDE&CFID=57100669&CFTOKEN=65703785

[27] End-to-End Routing Behavior in the Internet by Vern Paxson: 1063 6692/97, 1997 IEEE. Available at: http://delivery.acm.org/10.1145/250000/248160/p25paxson.pdf?key1=248160&key2=8394524021&coll=GUIDE&dl=GUIDE&CFID=57100669&CFTOKEN=65703785

[28] Craig Labovitz, G. Robert Malan, and Farnam Jahanian. Origins of Internet routing instability. In Proceedings of the IEEE INFOCOM '99, New York, NY, March 1999. Available at: http://citeseer.ist.psu.edu/cache/papers/cs/1693/http:zSzzSzwww.eecs.umich.eduzSz~farnamzSzpaperszSzCSE-TR-368-98.pdf/labovitz99origins.pdf

[29] End-To-End WAN Service Availability by Michael Dahlin, Member, IEEE, Bharat Baddepudi V. Chandra, Lei Gao, and Amol Nayate. IEEE/ACM TRANSACTIONS ON NETWORKING, VOL. 11, NO. 2, APRIL 2003. Available at: http://ieeexplore.ieee.org/iel5/90/26877/01194825.pdf

[30] Craig Partridge, Alex C. Snoeren, W. Timothy Strayer, Beverly Schwartz, Matthew Condell, and Isidro Castineyra. FIRE: Flexible intra-as routing environment. 19, 2001.

[31] Resilient Overlay Networks by David G. Andersen: Master's Thesis; available at:http://nms.lcs.mit.edu/ron/http://ieeexplore.ieee.org/iel5/9524/30172/01386195.pdf?arnumber=1386195

[32] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose. Modeling TCP Throughput: A Simple Model and its Empirical Validation. In Proc. ACM SIGCOMM, pages 303–323, Vancouver, British Columbia, Canada, September 1998.