Technical Report

UCAM-CL-TR-978
ISSN 1476-2986

Number 978

UNIVERSITY OF CAMBRIDGE

Computer Laboratory

A Next Generation Internet Architecture

Alexander G. Fraser

February 2023

Edited by Elisabeth Fraser and
foreword by Anil Madhavapeddy and
David J. Scott.

15 JJ Thomson Avenue
Cambridge CB3 0FD
United Kingdom
phone +44 1223 763500

https://www.cl.cam.ac.uk/

Technical reports published by the University of Cambridge Computer Laboratory are freely available via the Internet:

https://www.cl.cam.ac.uk/techreports/

ISSN 1476-2986

A Next Generation Internet Architecture

Alexander G. Fraser (1937 - 2022)

Edited by Elisabeth Fraser

Foreword by Prof. Anil Madhavapeddy and Dr. David J. Scott

This report is the unmodified work-in-progress monograph written by Sandy Fraser from 2003 onwards to capture his vision for a global network architecture that would scale to levels of quality and resilience beyond that of anything existing at the time. There are multiple areas of the document that are marked as "to do" or are otherwise incomplete. We believe it to be a valuable historical record of the original research that occurred first at AT&T Labs Research and subsequently at Fraser Research, and have preserved it in its entirety in this technical report.

Queries may be directed to:
Anil Madhavapeddy <avsm2@cam.ac.uk>,
David J. Scott <dave@recoil.org>, or
Elisabeth Fraser <efraser@me.com>.

Contents

Acknowledgements

Fraser Research grew out of a 33 year career at Bell Labs and AT&T Labs, working in a community of extraordinarily talented engineers and scientists with in-depth knowledge on virtually every facet of technology applicable to telecommunications systems. Without that background the research project whose results are described in this report could not have taken place.

The project to define a new future for global networking started in 2002 and gained momentum in 2003 with funding from the National Science Foundation and other donations which enabled the creation of a research lab in Princeton, New Jersey. More recently the project has been supported by IARPA. Most of the work on this project has taken place in that lab with students who participated in a 3-month summer internship program. The following have contributed to the program.

Jaime Adeane, Mark Batty, Alastair Beresford, John Billings, Alex Bradbury, D.J.Capelis, Tom Craig, Somdip Datta, Nandita Dukkipati, Paulo Ferreira de Castro, Simon Hay, Frank Hoffmann, Stephen Kell, Alden King, Alan Lawrence, Anil Madhavapeddy, Ganesh Narayanaswamy, Robin Message, Aaron Patzer, Sandeep Sarat, David Scott, Michael Smith, Chris Smowton, Sriram Srinivasan, Chee Wei Tan, Alastair Tse

This group has been most supportive of the project and a delight to work with.

The work has also benefitted from discussions with other members of the project “100 Mb/sec for 100 million homes and small businesses” which includes groups led by Hui Zhang at Carnegie Mellon University, Ed Knightly at Rice University, Nick McKeown at Stanford University, John Chuang at the University of California, Berkeley, and Larry Landweber for Internet2.

I am particularly indebted to the Trustees of Fraser Research for their support and counsel - Bill Brinkman, Ed David, Jim Flanagan, Andy Hopper, Larry Landweber, David Roscoe, Jeff Walsh, Rich Wolf, and Dory Yochum, plus my wife Elisabeth Fraser who as Vice President and trustee of Fraser Research has worked tirelessly on this project and has provided unwavering support over many years.

Alexander G. Fraser

1 Introduction

It might be said that June 12, 2009 was the day when digital communication finally arrived in the United States. On that day television broadcasters were required to cease transmitting analog video signals. It was the culmination of a digital revolution which began in 1948 when stored program computers were first demonstrated in the USA and the UK. Those 61 years have been a period of extraordinary scientific discovery and engineering skill that have resulted in there being more personal online electronic devices in the world than there are people. Four billion cell phones, more than 1.4 billion computers and untold numbers of television sets are presently in use. Sadly, the same technologies which deliver the information cornucopia that characterizes the present era also deliver a daily plague of malicious, fraudulent and invasive attacks on the people of all nations. There is now a growing fear that the plague will lead to terrorism and cyber war. So we are presented with what may seem to be contradictory requirements - to rid society of that plague while preserving the Internet communication service which has been a benefit to us all.

Much has changed since the Internet was conceived. During this period lifestyles and business practices have been transformed. Microelectronics, fiber-optics, wireless communication and computers have redefined everything from consumer products to national infrastructures. We understand better what a global communication network can do for us, and the prospects are profound. Society is moving from an industrial market-based economy to a liberal system in which individuals have a much greater say. Along with those changes we find that there are new risks, new invasions of our privacy, new threats to our security, and potentially new forms of war. Already the United States Air Force has created a Cyber Command[1], recognizing the importance of the network in future conflicts.

1.1 Security is the primary concern

Widespread concern about the present condition of the Internet is driven by several factors, the most prominent being frustration with a high volume of SPAM which undermines the value of e-mail, viruses and other malware which causes great inconvenience to users who are ill-equipped to handle the resulting problems, and various types of fraud which too often result in financial loss. While it is not appropriate to blame all loss of security and privacy on the network, it is reasonable to ask whether a network might be designed which substantially protects ordinary citizens against these abusive behaviors.

At a national level, there is great concern for the security of the country's networked infrastructures such as electric power distribution systems, banking systems, and air traffic control systems[2] [3]. Reduced operating cost and increased quality of service cause medical service providers to put medical records online and even to put physicians online when treating their remote patients. These critical systems cannot be allowed to fail, even for a moment, and they must be defended against terrorist attack.

Much research has been devoted to improving the present Internet architecture. Responses have been developed for individual methods of attack yet concerns about security continue to grow. So Fraser Research has chosen a more radical approach: a "clean slate" search for an architecture which meets current and anticipated needs and leverages current and

---

[1] http://www.afcyber.af.mil/

[2] The National Strategy to Secure Cyberspace. (2003)

[3] Vulnerability Assessment Of The Transportation Infrastructure Relying On The Global Positioning System. John A. Volpe National Transportation Systems Center (2001)

emerging technologies. The work program focused first on architectural components and technologies that permit a fundamental improvement in network security. This process benefitted from knowledge that was not available twenty-plus years ago when the Internet was founded. When a sufficient list of alternatives had been identified, work started on a preliminary network design that allows a practical strategy for network evolution. It was critical that the first stage of this process should identify promising design concepts while avoiding unnecessary commitment to specific design details. That left room for the second stage to converge on a specific design with relatively low cost of network evolution.

There are two overlapping aspects of the security problem. On the one hand there is the network infrastructure, much of which is operated by professionals among whom there is a certain level of trust. It employs equipment that meets prescribed standards of design and quality. On the other hand there are the computer systems which house networked applications. They are operated by people with a wide range of skills and training, and they contain software which is abundant but variable in quality. Accordingly, our strategy has been to identify a system design for the core of a professionally operated global infrastructure, and allow for a less disciplined periphery which is deployed by network users with widely varying skills, needs and interests.

The threat model used in this design study, assumes that attacks are launched by way of the network from user premises, where the means of attack may involve customized hardware and/or software. It is recognized that there may be issues of trust between network operators, and that nations may wage war by manipulating the network. It is assumed that international issues will be dealt with through bilateral agreements, monitoring systems and international gateways. Physical attacks on a network's outside plant and switching centers are not included in our considerations.

1.2 Internet vulnerability

The Internet is open to attack in some very serious ways. Its management and control systems are reachable by sending payload packets directly to them [router attacks]. There is no inbuilt structure which assures the integrity of an information flow, nor can a recipient be sure where the information came from [spoofing attacks]. A secure connection in today's Internet is a network overlay that typically uses encryption. Unfortunately, many network users find cryptographic systems difficult to deploy [Bellovin p318]. During the past 30 years there has developed a breed of network hacker who is as clever as the best network designers. They seek out and penetrate computers owned by people who are weak defenders even of their own interests. Unfortunately, a founding principle of the Internet's design [the end to end principle] put critical network functions in host software. A weakly defended host is not the right place for software that needs to be trustworthy [ARP attack, TCP congestion control].

The Internet's infrastructure is itself vulnerable to attack. Since 1990 it has been known that there are serious vulnerabilities in the Domain Name System, although the original researcher who discovered the problems waited five years before he published his findings because of concerns for the security of the infrastructure[4] [5]. A 1993 project was launched to solve the problem. The results was DNSSEC, an encryption-based protocol that offers a better but not perfect defense. However, even now (2009), DNSSEC is not widely used. In a review published in 2004 [RFC 3833] it was noted that aspects of DNSSEC are difficult to use, and it is not a complete solution to the known set of problems. A major new exploit revealed in 2008 by Kaminsky showed that the majority of name servers on the Internet remain vulnerable to attack.

Internet routing is an adaptive process that relies upon the BGP protocol. It is also vulnerable to attack [Kim Zetter, "Revealed: The Internet's Biggest Security Hole", Wired Magazine, August 2008]. In August 2008 it was demonstrated that it is possible for any person with sufficient knowledge and skill to cause BGP to assign an improper route for certain traffic thereby allowing the attacker to redirect selected traffic anywhere in the world. This attack abuses BGP's normal mode of operation, which means that the protocol will be very hard to repair. A Pakistan ISP that was ordered to

---

[4] Bellovin. Using the Domain Name System for System Break-ins. Proceedings of the 5th conference on USENIX UNIX Security Symposium, Salt Lake City, Utah (1995)

[5] P. Vixie. DNS and BIND security issues. Proceedings of the Fifth USENIX UNIX Security Symposium Salt Lake City, Utah, June 1995, 1995.

censor YouTube accidentally managed to take down the video site worldwide for several hours as a result of this protocol design[6].

There are common themes within these attacks. Among them are packet interception, eavesdropping on other people's conversations, improperly identifying the source of a packet, and inserting extraneous packets into targeted conversations. Hackers have developed tricks that exploit specific weaknesses in the Internet's administrative and control protocols. With time, no doubt, all of these vulnerabilities will be addressed and countermeasures will be found. However, time is not a luxury we can afford because crimes can be committed even as solutions are being sought. Also during that interval of exposure, other vulnerabilities will surely be uncovered and exploited. The Internet is a large and complex software project that is constantly evolving. It is in the nature of such things that human error slips in and must be found after the software has been put into service. Therefore one must conclude that there is a natural and perpetual cycle: install a software update, attack a new vulnerability, create a new software update to fix the vulnerability, and install that update. The network is already large enough that each cycle takes many years to complete.

1.3 Host vulnerability

In the book "Firewalls and Internet Security", Cheswick et.al. identify many vulnerabilities in software and system design that have made ordinary network users vulnerable to attack from hackers and other rogues. Anderson, in his book "Security Engineering", substantially enlarges that list of attack methods. The number of vulnerabilities which these authors describe and the unrelenting hacker activity has become a concern for virtually all network users. Of particular concern is the pessimism expressed by experts in the field. Cheswick, Bellovin and Rubin end their book with the following remark: "We are losing ground. We can't afford to, and must do better."

Hacker exploits on the Internet were once more of a sport than a threat. Today the situation is very different. For-profit and politically motivated attacks are being launched against individuals, institutions and even nations. These antisocial behaviors are amplified by the presence of bot-nets. Bot-nets are networks of host computers that have been compromised so that they can be remotely controlled by a hacker. Bot-nets run rampant on the backs of weakly defended hosts. For example, the Srizbi bot-net occupies 315,000 compromised hosts and is capable of sending 60 billion messages per day. These messages may be spam, transmitted as a for-profit service, or they may be a denial of service attack on a merchant who is threatened with extortion. A deeper concern is that terrorists will cause chaos or worse using the same techniques.

On 27 April 2007, Estonia, a small Eastern European country, was forced to disconnect itself from the worldwide Internet because banks, telephone companies, media outlets and name servers were simultaneously inundated in a denial of service attack launched from outside the country's borders. The majority of attack packets came from bot-nets in the United States, however it seems likely that these bot-nets were controlled from Russia and the attack was politically motivated. Wired Magazine described the event as an act of war. Since the bombing of the World Trade Center in September 2001 the United States has been concerned that there is at least a theoretical possibility that its national infrastructures could be attacked through the Internet. Estonia changes that theoretical possibility into a serious concern.

No single technique or architectural change will solve the problem of host software that can be compromised or misused. There are too many hosts that exhibit too many forms of vulnerability, and too many creative folk who are searching for more ways to attack. Nor are the vulnerabilities confined to application programs; Cheswick and Anderson point to vulnerabilities in host operating systems. The Internet finds itself firmly in the middle of this battle because it is the means by which many attacks are delivered. But what actions can a responsible network operator take to combat the hacker's menace?

1.4 A medical analogy

The Internet and its trouble with hackers can been compared with the human body and its problems with infection. Prior to the 19th century, plagues ran wild and accounted for many deaths with little that could be done to fight them. Then Fleming discovered the first antibiotic and Nightingale promoted a clean environment. After that, progress was rapid; viruses and bacteria came under control.

---

[6] http://www.wired.com/threatlevel/2008/02/pakistans-accid/

Software viruses spread quickly through the Internet and in recent years there has been much progress on filters that scan for the signatures of messages which install viruses in unsuspecting hosts. This “deep packet inspection” is the Internet’s antibiotic. Today (2010), clever pattern recognition makes it possible to inspect backbone traffic at full line speed. But there is a lesson to be learned from medical practice - extensive use of antibiotics prompts the bacteria to evolve, thereby becoming resistant to the antibiotic. Prudence suggests the use of antibiotics only when other defensive practices fail.

The computer network equivalent of Nightingale’s cleanliness is a collection of practices and architectural features that create a practical level of security for network users with ordinary skills. The goal is to dramatically reduce a network user’s exposure to attack, thereby creating a largely trouble-free environment in which to operate. That is the focus of Interlan network architecture.

1.5 Securing the environment

The required action is multifaceted and the practical result will probably be as it is today for real (as opposed to virtual) property: The law defines property rights and penalties for violation of those rights, property owners put up a modest defense that at least makes it obvious when the law is being violated, and a police force provides a deterrence by monitoring behavior and apprehending those who are found to have violated the law. With that in mind we set the following goals for the new network.

First, put significant barriers in the way of those who would attack the network itself, on the basis that network operators can be suitably trained to maintain a high level of discipline. This is an essential step for if the control system is vulnerable then little else can be relied upon. Critical network control functions such as name service, routing control, and network management should be, as far as possible, isolated from the payload network thereby removing the opportunity to launch a direct attack.

Second, define network service at a high functional level, removing as much as possible of the network implementation from host computers. A new host/network interface will enable this restructuring if it reduces to a minimum the amount of information that the host has about how the network operates. This diminishes the opportunity for host software to mount a low-level attack on the network.

Third, equip the network user with easy to use tools for basic security. Access control for a service or network might be as simple to use and understand as the locks that are commonly used on the doors and windows of a home.

Fourth, create a framework in which users and networked devices are identified and authenticated. This will enable an online culture in which users are held responsible for their actions, and it will become the basis for access controls which distinguish between those folk who are invited into a home network and those who should not be there.

Fifth, provide a system of traffic monitoring which enables an effective online police force. It is recognized that not every user can mount a strong defense of their own interests, however the impact of their diligence can be significant when it is backed up by effective police work.

1.6 Comparison with operating system security

Since the 1960s when computers were first shared among many concurrent users, much thought has gone into the aspects of computer system design which would provide privacy and security for those who use the machine. The circumstances for users that share a computer network are similar enough that lessons can be learned by making a comparison between the Internet and a host operating system.

Protecting the network operating system

Computers today have two operating modes: kernel mode allows full access to the resources of the computer while user mode restricts the functions and memory space that are accessible to application programs. This protects the system itself from its users. The network equivalent would be to isolate the network’s core functions, such as routing and name service,

from the payload network which is the only part of the network to which the user is allowed access. Today there is no such mechanism. Hosts providing core internet functions are directly addressable by any other host on the network; this means that any virus-infected host can make a direct attack on the network's most critical functions.

Protecting one user from another

All modern computers have a memory management unit which prevents application processes from writing into each other's memory. The network equivalent is machinery which maintains separation between one information flow and another. This will prevent one application from intercepting another's data (snooping) and will stop an intruder from inserting packets into another's conversations (spoofing). Today, internet traffic flows are not protected as packets travel through the network, so interception and insertion are common practice for hackers.

User authentication

All computers provide the means for a user to login and thereby authenticate himself to the system. In contrast, the Internet has no system wide authenticated identity for network users. There is a libertarian tradition which has been associated with the Internet as it has developed. Those who hold this view are uncomfortable with the concept of authenticated identification for network users; in the Internet today, authentication takes place at the application layer rather than lower down. However, there is another community which wants to lead a life that is protected from malevolent and abusive behaviors, and that can probably only come about when users are held responsible for their actions. That requires a user identity. Technologies which implement and authenticate that identity are available and can be integrated into a new network design; the challenge is to find a compromise system that both communities will accept. The matter is gathering urgency and difficulty as the risk and consequence of cyber terrorism and warfare grow.

Unified namespace

What was once a name space for files held by an operating system has become a name space for all but one of the resources that are available to a computer's user. The network is the one exception. Name space evolution was central to the advantages which Unix brought to computing systems. A well designed name system brings ease of use and simplified design to the operating system - including those aspects related to security. By contrast, the Internet has its Domain Name System where the only resource type is a host. Services, which arguably are the most important of the Internet's resources, do not have domain names - instead, references must be made to a host name and a "well known" port number. In a global system that is expected to have a long life, the assignment of a single range of integer identifiers is unrealistic. Networks are another important resource type for the Internet. They also lack names and rely upon a numeric system derived from an IP address. The Internet could make better use of its name space and that could lead to a more convenient user interface.

Access controls

Each named resource in a host has a resource record, an inode, in which is stored an encoded list of access controls. The fact that Unix has access controls is of critical importance for the privacy and versatility of the file system. Whether the range of access control techniques is adequate for the purpose is a topic for debate. The Internet has no such mechanism associated with its named resources. Yet that network allows every computer to send packets to any other computer without constraint. To make matters worse, those packets can be sent with zero incremental cost to the sender. As a result the owner and operator of a networked host is directly exposed to, and must reliably reject, the threats which are delivered daily to his computer. One unfortunate mistake, clicking on an e-mail message, can convert his host into a bot, expose his family to financial risk, and cause much difficulty as he tries to extricate himself from the trap. The situation with e-mail today is tolerable only because of filters that everyone must use. As network service becomes more pervasive and the number of applications grows, households will need some more fundamental and easy to use tools for their own protection.

1.7 Other design objectives

In addition to setting a high standard for public and personal safety, our network design project targets the following objectives.

Quality multimedia service

Multimedia services, primarily audio and video, were not economically important to the Internet during the first 30 years of its existence. Today that situation is changing. Soon there will be more bytes of video flowing through the Internet than there are bytes of any other service type. Also telephones are rapidly being transformed into a mobile computing platform, potentially growing to become the largest block of networked appliances. As these packetized services mature, their users will demand quality service, the parameters for which are well known even if it is not so well understood how to deliver quality results in a packet multiplexed network. Congestion control was an afterthought for the Internet; it cannot be so for the new network.

Reliable service

Networks today touch on almost every aspect of daily life many of which involve significant financial exposure or are life threatening when failure occurs. It is unacceptable to do anything less than one's best to deliver a service with the highest level of reliability and availability. That is most readily achieved when automatic recovery from failure is part of the initial design.

Intrinsic mobility

Advances in technology have dramatically increased the functionality and convenience of portable and mobile devices. Cellular telephones with extended functionality are growing in popularity. There are 4 billion cell phones - twice the number of internet hosts. But mobility is an unnatural fit with the Internet's datagram technology, requiring in effect the creation of an overlay network that adds substantially to the complexity of the system. The technical challenge is to maintain the integrity of a conversation while endpoints move about. For this to be a success the network must be aware of each conversation. Mobile devices must be fully integrated into network service, and support for mobility must be a fundamental and natural capability of the network.

Global scale

The network must serve a world population that is expected to reach a peak of 9.2 billion people by 2075, although some experts anticipate continued growth reaching 34 billion by the year 2300 [www.unpopulation.org]. In either case, the Internet will be the world's largest machine and the world will be totally dependent upon it. The first and most obvious problem for the Internet is that the IPv4 address space is too small, and in 13 years IPv6 has made only limited progress towards solving the problem. Network Address Translation is more widely used but it distorts the Internet's basic design and causes application designers to engage in tricks that in the long run will surely be an embarrassment and/or inconvenience for everyone. A new network design must accommodate large scale much more gracefully.

Independent evolution

Continuous evolution of the world's network is not possible if each evolutionary step requires cooperation on the part of every network user. Slow progress in the evolution of the present Internet can in part be attributed to host software that is so closely linked to the internal operation of the network that network evolution depends upon software upgrades being installed in billions of host computers and cell phones. The host/network interface must decouple the two systems much more effectively in future than it has in the past.

Internet evolution

With regard to this clean slate design project, our goal is to circumvent the above-mentioned constraints imposed by the Internet's present architecture and thereby chart a course for change that places the least burden on network users. Transition from today's internet to a next generation network based upon the new design must not force individuals to purchase new hardware, and host software should continue to operate as it does today. Of course something has to change. Our intention is that the burden will be felt by network operators. However, by basing the design on Ethernet we hope that the necessary investment will be in line with current plans for network evolution, probably driven by the need for higher speed and better support for converged services.

1.8 A vision for the United States

We envision an infrastructure to serve the United States when every person carries a mobile computer and the conduct of every-day life rests firmly on networked computing. A large majority of the country's 100 million households will be located in approximately 100 regions, each with a population density exceeding 3 households per square mile, and each offering a propagation delay that is less than 2 msec for the round-trip between a home and its regional center. This low round-trip time is important because human response time is on the order of 10 msec. The typical radius of a region, 100 miles, is practical now that fiber covers most of the distance in modern aggregation networks.

The regional center will become both a switching center and a hub for data storage and computing. These centers will be the information-age equivalent of the goods distribution centers that today are to be found on the outskirts of the large cities in the United States. Even today, in these early days of network evolution, this design makes sense because it makes the best possible use of aggregation network bandwidth - the single most challenging investment that must be made. In most regions today the downstream bandwidth is between 2 Mb/sec and 10 Mb/sec per home but the upstream rate per home is only 1/5 of the downstream rate. A regional center gives the local hub parallel access to all branches of the aggregation network.

We are now seeing a transition in consumer computing from a computer or two in each home to a hand-held computer per person backed up with computing service in a cloud. Centralized storage and computing is convenient when each family has more than one computer, and it relieves the user of responsibilities that have been quite difficult for many people to handle. Keeping one's computer up-to-date and free of viruses is quite a burden. For business, Amazon has demonstrated the same sort of convenience for small and medium size businesses, and Google has demonstrated the same logic for consumer services. Today they and others are building massive computing centers in places with good power but well away from population centers. However, once the technology and the momentum have built up, I believe that we shall see the computing centers move closer to the customer. Fast interaction and maximum bandwidth to the customer will turn out to be important. Akamai understands this. They have invested in regional systems that will deliver HD video on a large scale.

This regional design has an unexpected further advantage: significant increase in security for the individual and the nation. Management of regional computing centers can deliver better security than personal computers managed by consumers, and massive storage in or near a regional center can help in a circuitous way with one of the most difficult security problems: controlling the bot-nets.

1.9 Structure of the report

The structure of a network designed to meet these objective has two main parts:

(a) a payload network that is in effect a high capacity global Ethernet, and

(b) a network operating system which controls activities in the payload network.

The report describes two protocol layers of the payload network. The flow layer provides private but raw end-to-end connections between application processes, and beneath that layer the path layer supports device mobility and service restoration in case of hardware failure. The network operating system (sometimes called the control plane) brings open-ended support for 21st Century applications and services, including support for consumer grade privacy and security.

The report is divided into 3 sections. This introductory chapter and Chapter 2 comprise the first section. Chapter 2 sets the stage for the rest of the report. It provides an overview of the network's structure, identifies its principle components and introduces the terminology that is used in this report. It is important to read this chapter before moving on to the rest of the report.

Chapters 3 through 5 describe the new network architecture. Chapter 3 describes the network operating system, chapters 4 describes the payload network, and chapter 5 goes into more detail, illustrating one way in which a network may be implemented.

Any attempt to complete a clean slate design of a national (possibly global) network must inevitably cover a lot of ground. The work described here is but a start on such a task. In order not to overwhelm the network description, discussions about critical design issues and alternatives has been held over to chapter 6.

This report uses the name “Interlan” in reference to the architecture which is described here. There is as yet no network implemented according to the Interlan architecture.

1.10 Matters of public policy

There are aspects of network design that have a clear relationship to matters of public interest and policy. For example, it would be of considerable help to law enforcement if every person who uses the network can be held responsible for their actions. But there are many people who fear a loss of privacy or excessive government control. As a result there is no global or even national agreement on whether a network user should be required to prove their identity at the start of each session of network use. The authors of this Interlan document are certainly not in a position to make a judgement on such matters. On the other hand, there are engineering issues raised when user identity is to be authenticated that should be considered at an early stage in network design. Therefore we take the position that it is not our role to set policy but we should consider network capabilities that allow network operators to respond to public policy when and where it is decided. For brevity in the pages of this document we will discuss potential security systems without repeating on each occasion the qualification that any particular security system may or many not be used in any specific realization of Interlan architecture. The reader is asked to understand that we are not by these discussions and statements taking a position on matters of public policy.

2 Overview

The network design project has taken a broad look at communication service to serve a generation of users for whom the Internet and integrated communication services were a fact of life from the moment they were born. This generation has a comfort level with computer communication that gives them the freedom to envision and implement new roles for technology applicable to each phase of their lives. Accordingly, our work looks forward to the integration of computing with communications, and particularly the compatible evolution of computer and communications operating software. While addressing problems of today's network, the design creates a framework for communications control that we hope will give the next generation of system designers and application developers the flexibility which they will need.

2.1 Network structure

The network infrastructure has two main parts:

A global payload network which carries user data.

A global control network which is host to a network operating system.

These parts are illustrated in Figure 2.1.

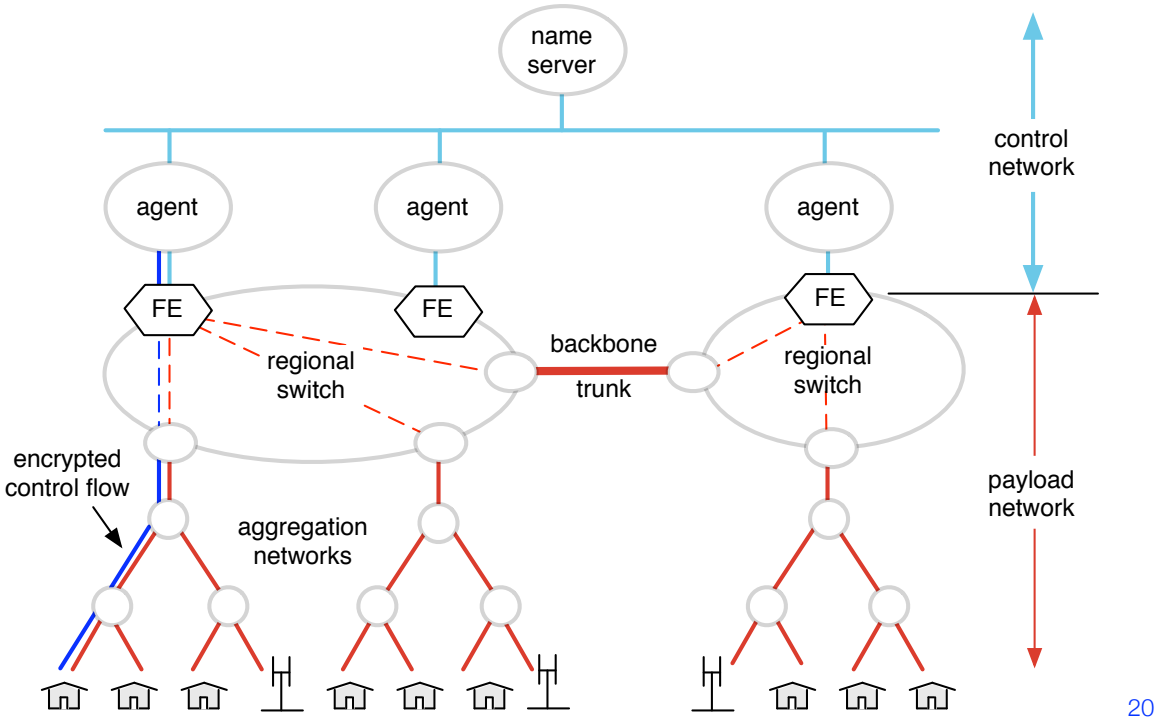


Figure 2.1. Payload and control networks

The payload network (lower half of the figure) carries host-to-host traffic within flows that are protected from intrusion and inspection. Forwarding engines (FE) steer payload packets along their routes. This network supports static and mobile hosts, indicated by the homes and radio antennas at the bottom of the figure.

The payload network is the high performance part of this infrastructure. The components illustrated in Figure 2.1 carry packets from one host application to another. Red lines represent the transmission paths over which the traffic is carried. Dashed lines represent packets that are moving within a switching center, and solid lines represent packet traveling on a transmission line. Forwarding engines receive specific instructions from agents concerning the route that the packets of each flow must follow. The forwarding engines are also responsible for keeping the packets flowing smoothly, so that hosts do not transmit too quickly and queues do not grow out of control.

Buildings at the bottom of Figure 2.1 represent homes and business premises in which network users live and work. The antennas represent wireless base stations. Aggregation networks carry traffic between these premises and the nearest regional switching center. Trunk transmission lines carry traffic between one switching center and another.

The control network (upper half of the figure) is the secure operating environment for a network operating system which manages services delivered through the payload network. Transmission paths in this network are colored light blue. Agents and name servers collaborate to implement the user-facing services of the operating system. The agents listen to service request messages transmitted from hosts by way of control flows which are colored dark blue in the figure.

The control and payload networks are physically separate in order to be sure that payload packets cannot jump from one network to the other. This precaution is designed to substantially reduce the network's vulnerability to attack from network hackers. If transmission paths of the control and payload networks share a transmission facility, such as an optical fiber, then each one uses a different optical wavelength. The only bridge between these two networks is the highly constrained flow of encrypted control messages that are transmitted between host and agent by way of a forwarding engine.

The supervision network, not shown in this diagram, visits every part of the communications infrastructure. Supervision is a monitoring system that provides a constant and independent source of information about how the network is configured and how well it is performing. When an anomaly is detected the supervision system takes pre-programmed action according to the nature of the problem. That may involve sending an alarm signal or initiating a service restoration procedure. Some people have compared this network with the human nervous system which is constantly alert, sounds the alarm when necessary, and in serious emergencies takes quick action, leaving the brain to make a more considered decision.

2.2 Payload network

Essential elements of the payload network are

network interface units	sit between user premises and the wide area network
aggregation networks	first mile networks connect premises to a switching center
switching center	contains a switch and part of the network operating system
switch	moves packets within the switching center
forwarding engine	directs packets towards their destinations
trunks	transmission lines that interconnect switching centers

The network topology is hierarchical. Host computers on local area networks connect through aggregation networks to regional (switching) centers. Mobile devices connect to wireless base stations which are of two types. A private base station connects to a premises LAN while a public base station connects directly to an aggregation network. Regional centers are interconnected by the trunks of one or more national backbone network. International gateways connect the national networks to one or more international backbones. An international gateway is a special-purpose switching center which uses technology similar to that found in a regional center.

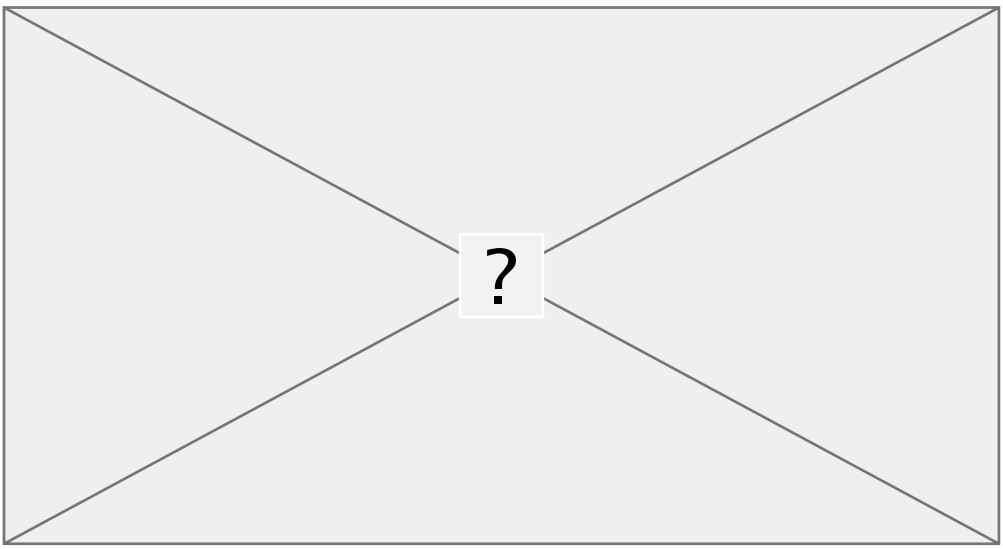


Figure 2.2. Hierarchical network structure

By default each client-server connection is a point-to-point “flow” that is protected from every other flow. Packet forwarding is handled in hardware and is controlled from forwarding tables which are set up when a flow is established. The hardware makes sure that information within a flow does not leak out into another flow, and no information enters the flow except from one of its two participants. Therefore each flow operates with strict privacy. Packet insertion and packet inspection, two of the fundamental tools that Internet hackers use to conduct their exploits, are not available in an Interlan network. (See xx for a defense against man-in-the-middle attacks.)

A rendezvous protocol is used for flow setup. The host interface is compatible with the socket interface used today by hosts connected to the Internet Each new service, such as a web site, is given a name; the service becomes a resource; Server processes in one or more hosts (servers) “listen” for clients of the new service; Clients “connect” to the service and thereby obtain a flow which links them to one of the listening servers. This method is compatible with current practice in the Internet but it has some additional advantages. Hosts that do not wish to offer a service do not listen for clients seeking connection to a service name. These hosts are not required to have a (domain) name and so are invisible to other hosts on the network. Only those host processes that listen for client connections are vulnerable; other processes in the same host remain invisible and are not subject to direct attack.

Today, the bulk of internet traffic either uses TCP or it uses a UDP protocol for multi-media streaming. TCP is used when flow control and packet retransmission are required, otherwise the use of UDP allows alternative ways of controlling a flow and/or coping with transmission errors. For all of these cases the point-to-point flow service described here is appropriate. Otherwise, a virtual datagram network should be used (see ref).

The route taken by a flow is chosen when the flow is set up. All packets follow that route until the flow terminates, or until one of the participating hosts is mobile and moves to a different wireless base station. If the host is mobile or there is a failure within the network, the flow is quickly rerouted.

Traffic is routed in zones. Each aggregation network (including the homes and base stations that it serves) is a single “aggregation zone”. Each backbone network, national or international, is a “backbone zone”. Switching centers interconnect the zones. The high-level view of the route taken by packet flow is expressed as a series of zones and the switching centers through which the packets will pass. Routing within each zone is more dynamic and is implemented by switches that each route traffic within their assigned zone. Zone boundary crossings are implemented by forwarding engines located in the switching centers.

2.3 Aggregation network

An aggregation network is the complex pattern of driveways and city streets which bring traffic to the highways which make up the network's backbone. It is often referred to as the 'first mile' by the networking community, though in practice and certainly in our vision, an aggregation network may span tens of miles. Without an aggregation network, the backbone would be a lonely racetrack that goes nowhere. In a world where connectivity is taken for granted, and broadband aggregation is increasingly on the uptake, the aggregation network is without a doubt the largest and most difficult to manage part of the network. Certainly, end users should not need to know about its structure; they see a logical point-to-point connection from their network interface unit (NIU) to the backbone of the network. This logical connection, which we call a path, is an abstraction that the aggregation network creates for the purpose of giving each household a unique interface to the nearest switching center.

The difference between this abstraction and reality, is that it is too expensive to have a single fiber to every home, so at some stage we need to multiplex several paths onto a single physical fiber. In the optical world, the traditional view has been to use strictly passive components out in the field, namely optical splitters, since they do not require electric power. These passive optical networks (PONs) have a number of disadvantages, many of which are characteristic of shared media in general. In particular, upstream packets (those which travel from a home into the network) must contend for access to the shared optical channel but without power there can be no local intelligence to coordinate that sharing. Bandwidth use is administered by a control system located some distance away and is therefore hampered by propagation delays.

The absence of local intelligence and switching also frustrates fault isolation and automatic service restoration. Restoration is the action taken when there is a hardware failure, caused perhaps by a tree that falls and breaks the cable. Redundant transmission lines provide alternate routes for traffic flow and small switches automatically reroute the paths when necessary. A restoration system recognizes when a failure has occurred, causes the required switching to take place, and reports on the event so that the network operator can respond quickly and efficiently. Today, automatic service restoration is found in relatively few aggregation networks, but as society's dependence on communication intensifies so will the demand for a highly reliable service.

An intelligent aggregation network can also assist with network security. As the value of intellectual property transmitted over the network increases, so one should expect an increase in the amount of fraud and interference that is perpetrated in the aggregation network[7]. The aggregation network is the most exposed and least easily monitored part of the infrastructure. A small amount of electronics in support of switching for service restoration can also monitor the plant and report when significant changes take place.

2.4 Global Ethernet

Ethernet is the dominant technology used in local area networks. Each premises network has a certain amount of "local" traffic that stays within that network. Other traffic uses the aggregation network to reach a regional switching center. Within the switching center the traffic for a given host is handled by a specific forwarding engine. Each engine can handle many hosts, and the hosts in one premises may use different engines. For this purpose, each forwarding engine has an Ethernet address. This allows for the possibility the some engines may be customized to provide special services.

Figure 2.3, shows one aggregation network that implements a separate path for each premises network. Each path forms an Ethernet bridge to a "root" which is the point where an aggregation network connects to a switching center.

[7] Cable television operators have reported many cases where consumers have modified the cable system in their neighborhood in order to obtain services that have not been paid for.

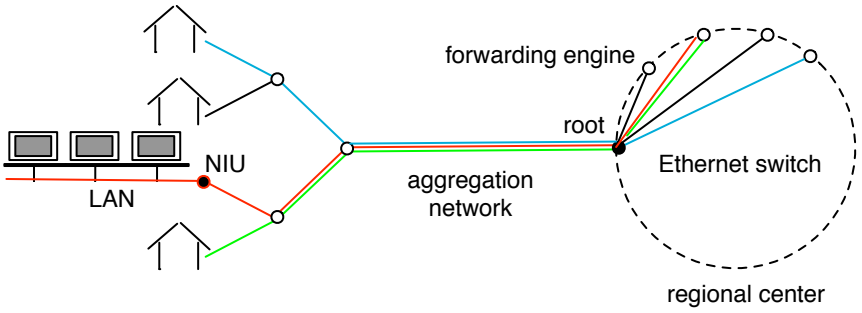


Figure 2.3. Each path is a bridge from a LAN to forwarding engine

The switch that is in the core of each switching center is an Ethernet switch, and premises networks are presumed to be Ethernets. The packet header for use by hosts that are aware of the flow system is illustrated in Figure 2.4. This is an Ethernet header with an extension header to allow the identification of flows. Legacy hosts that use TCP or UDP but do not know about flows transmit packets in the customary way, i.e. without the extension header.

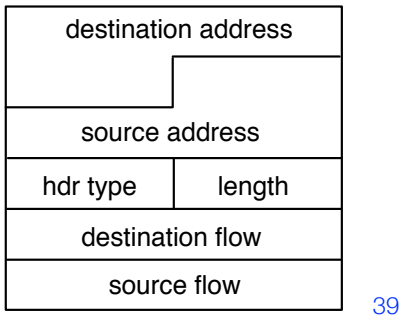


Figure 2.4. Ethernet header with flow extension

A backbone network also implements multiple paths, each one carrying traffic between one switching center and another. Each backbone path defines a route within the backbone network. Assignment of traffic flows to paths balances the load and provides each flow with the type of service that has been requested. For example, rate controlled video traffic is assigned to one type of path while best effort file transfers are carried in a different path type. The path structure allows a high traffic volume to be routed and controlled according to the quality of service objectives that users have selected for their data.

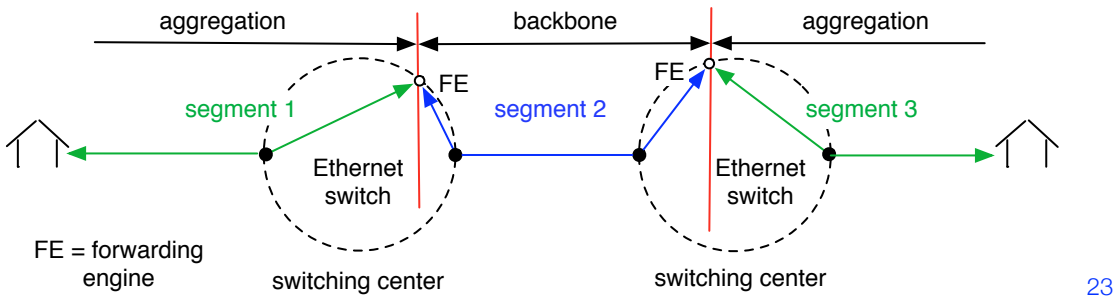


Figure 2.5. A three-segment flow

The set of all aggregation networks that converge on one switching center comprise that center's aggregation zone. A "zone" is an administrative partition of the global network. Each backbone network is also a zone. That part of a flow which spans one zone is called a "flow segment". At each segment end there is either a host or a forwarding engine. That segment endpoint has a unique "flow label" which consists of the Ethernet address of the endpoint and a flow number which is unique among flows at that endpoint. Fields comprising source and destination flow labels can be seen in Figure 2.4. For

compatibility with Ethernet equipment already installed, the addresses appear at the top of the header and the flow numbers appear in the header extension. Figure 2.5 illustrates a three segment flow which passes through two forwarding engines.

As a packet passes through the forwarding engine its routing header is checked, discarded and replaced with a new header. Thus, each zone is regarded as a separate Ethernet, and only by going through a forwarding engine can a host packet acquire a header that is appropriate for travel through the next zone. Header values are stored in forwarding tables, contained within forwarding engines. They are written there by the agents. Therefore packet headers in all segments of a flow except the first have values that were approved by the agents. No packet can follow a route that was not authorized by an agent.

2.5 Mobility

All devices, including those that are Mobile, have Ethernet addresses which are globally unique and do not change as the device moves from one place to another. The association between a mobile device and a forwarding engine is dynamic. As the device moves from one base station to another its traffic may be routed by successively different forwarding engines. The transition from one forwarding engine to another is accelerated by application specific hardware in each engine and by the structure of the switching center. That hardware also supports reconfiguration of the aggregation network in case an equipment failure should make that necessary. In this network, mobility and service restoration are supported in the lowest protocol layer, and for most purposes there is no practical difference between mobile and static devices other than that which the transmission medium would imply.

One of Ethernet's strengths is the ease with which local area Ethernets can be configured. Each Ethernet switch learns about the network's topology by observing the source addresses carried in Ethernet packets. This method is well suited to a LAN but does not work in a large wide area network. Interlan uses the agents to solve this problem. Every device that wishes to communicate requires the services of an agent. When the device moves it seeks a new agent from which to obtain service. In this way the network knows when a mobile device moves and where it went. Collaboration between agents maintains connectivity for active flows, and one agent, the "home agent", puts new clients in touch with the local agent that is currently in touch with a mobile device.

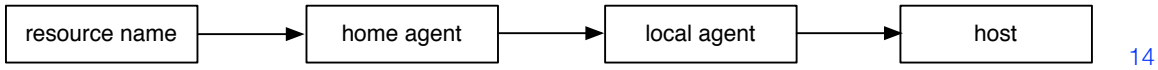


Figure 2.6. Locating a mobile host

The "agent" concept is borrowed from mobile telephony and from IP mobile service. However, in this network every host, static or mobile, is assigned to an agent, and all hosts participate in the same connection setup protocol. Thus mobility is a fundamental network capability that is available to all hosts.

A personal computer today is a laptop. Most come with wireless and wired network capability. It is not that the user works on the machine as he moves, but it is simply more convenient to not be tied by wire to a wall socket. By providing universal mobility support for all devices, Interlan reinforces the freedom to move without concern that a connection may be dropped.

The next stage in this evolution towards location-independent computing is mobility enabled by virtual machine technology. A process encapsulated by a virtual machine can carry with it the connections to its environment. Thus application processes will be able to move from one host to another.

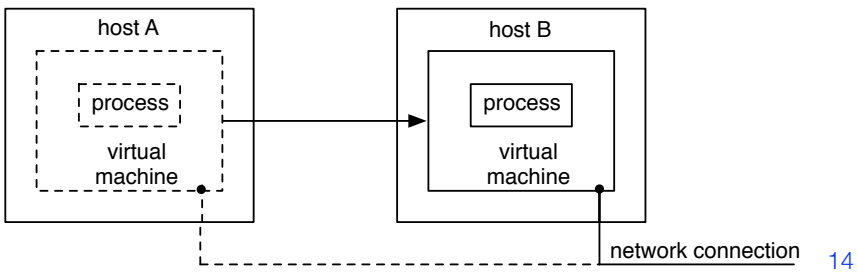


Figure 2.7. Process migration from one host to another

From the network's perspective, each virtual machine is an independent and potentially mobile host on the network. The logic which supports host mobility also supports the movements of virtual machines, and therefore active processes, from host to another.

Computing and communication have become integrated. Large computing systems are networks. An iPhone is a computer that hops from one base station to another. The distinction between mobility and portability is becoming blurred. In this report we use the word "mobility" in reference to networked computing where the movement of a machine (virtual or real) is supported by agent services which track hosts as they move from one network connection point to another, and reconfigure active flows in order to maintain the integrity of the executing applications.

Mobility and service restoration are closely related. Technology which allows a network to cope with unexpected link failures while it provides an essentially uninterrupted communication service can also provide good service to a mobile device that is constantly moving out of range for one wireless base station and repairs the damage by connecting to another. Accordingly, mobility is a core aspect of the new network design, rather than an overlay as with the Internet. Mobility is enabled by globally unique device addressing and, in the aggregation network, a path system which bundles traffic according the mobile device with which it is associated.

When a mobile host or other device type registers with the network, a path is assigned to carry all conversations in which the host participates. The path extends all the way from the host to a switching center. Movement of the host from one NIU or wireless base station to another results in the host hopping from one path to another. Path hopping is supported in the switching centers so that all flows attached to the host are moved in a single switch action. This greatly facilitates mobility for complex mobile systems, such as trains and buses.

The same technique which supports fast service restoration is also used in switching systems that support fast mobility. Transport and application protocols used by mobile devices are the same as those used by all other devices. There is no additional overhead. Any client or server can be mobile, including any which participates in or is the source of a multicast. Even networks can be mobile.

Further details are provided in section (ref).

2.6 Network operating system

The network operating system is a user interface and control system for Interlan networks. It is a distributed system which is implemented by the agents. Agents respond to service requests submitted by users, they exercise control over the delivery of those services and they carry out security checks on behalf of network users. A balance is sought between efficient operation on a large scale and features that can be customized to meet individual needs. Fiber optic communications and high performance clustered computing makes this practical. Fiber enables high performance regional networks that have larger than usual diameter. This allows the national networks to have a relatively simple structure with good consequences for traffic management. Clustered computing delivers the computational horsepower to reliably provide customized services to individual users and application processes.

Current experience indicates that consumers have a great deal of difficulty with the diligence and practice required to ensure their own security and privacy. So the network operating system is designed for a consumer-oriented world where the default network configuration creates a safe environment and minimizes user exposure to direct attack. The majority of process to process flows are established by a rendezvous protocol that is implemented in the agents. During the course of this protocol the agents can act as an independent and discreet third party who can, for example, authenticate a client without revealing the client's identity to the server. The agents may have other information, such as client location, which, if authorized by the client, can be used to establish his credentials with a server. The naming system, used to identify networks and service, includes controls on the use of names in a manner reminiscent of a host file system. Services can configure access controls appropriate for the applications involved with greater flexibility than is usually included in a file

system. The strategy is to include a default set of controls and then provide the means for network users to add their own checks as appropriate for the application and the clientele that they are serving.

The operating system has three main parts:

> The service control system is that which implements communications service according to requests received from network users. This system is implemented in the agents and includes a name service.
>
> The traffic management system is responsible for managing the flow of packets in the payload network including routing and control of congestion.
>
> The supervision system monitors all aspects of the infrastructure and databases, looking for unscheduled changes in state and signaling an alarm when something goes wrong. In certain instances the supervision system takes pre-planned preemptive action to deal with an urgent matter.

Each host communicates with the operating system through its encrypted control flow. The messages transmitted on this flow are known as “syscalls”. Reminiscent of traditional operating system system calls, these syscall messages convey application process service requests and responses to/from the operating system. They are written in this report as they would be in an application program, i.e. in terms of library functions calls such as connect(“twitter.com”).

Payload is transmitted in flows that each connect one application process to another. Each process is presumed to be under the control of a network user.

The following sections of this chapter provide an overview of the service provided by the network operating system from the point of view of a network user. The final two sections are on operating system portability and on special arrangements that are provided for existing applications designed for use with the Internet.

2.7 Resource names

The Internet’s Domain Name System (DNS) is a global infrastructure which gives network users around the world access to the names of computers and other networked resources worldwide. Its hierarchical structure breaks the global name space into meaningful parts and employs a distributed management system which has allowed the set of domain names to grow large in an orderly way. Today there are in excess of 1 billion names registered in that system. It is a considerable challenge to create a system of that size which gives fast access to any name from anywhere in the world. The problem is made further complicated by the precautions that must be taken to make sure that names are not “stolen” or their meanings corrupted. So, by design, the information which DNS contains is relatively static, recording primarily the Internet addresses that are assigned to domain names.

By contrast, an Interlan operating system has a dynamic and detailed view of network activity so that it can support universal mobility and can create a more secure working environment for networked applications. That shift in emphasis relieves some of the difficulty of ensuring host security and protecting access to private resources. It also provides greater flexibility in dealing with special cases such as multicast information flows and Ethernet bridging networks.

Interlan uses DNS to provide global access to slowly changing information, and it uses regional data records to hold time-sensitive information about matters that are important locally. For example, the name

//FraserResearch.org/emulator

might be the DNS name for the Fraser Research laboratory, and “emulator” which would be the name of an experimental system within that laboratory. A regional agent holds the resource record for //FraserResearch. A field within that record refers to the attributes of “emulator” which are held either by the agent or by a network gateway within //FraserResearch.org.

Application developers choose names for the services that they create, such as “www.twitter.com” or “//FraserResearch.org/emulator/”. Each name leads ultimately to a resource record that is held by an agent, or may be several agents if the application is widely deployed. The agents are clustered in regional switching centers where they manage the flow of information through that region. Resource records are where application-related state is stored. Each record has a type such as ”directory”, “service”, “network”, “host” or “gateway”. “www.twitter.com” is a resource of type “service”. “//

FraserResearch.org/emulator/big-apple" is a host called "big-apple", which is connected to the emulator network at Fraser Research.

This Interlan naming system is comparable with the naming system commonly employed in a host operating system. While the files in a host file system have types which are typically associated with the applications that created them, directories in a host file system typically have no application related type; they are just lists of names. By contrast, some Interlan directories have a type associated with them. For example "emulator" is the name of a network, so "//FraserResearch.org/emulator" is the name of a directory which lists the resources attached to that network.

2.8 Access control

A primary goal of the Interlan network architecture has been to enrich the interface between network and user so that the network can provide security in accordance with the wishes of users and resource owners. It is not the role of a network architect to define policy for its users but it is necessary that the network provide strong tools that a user can employ for his own protection and for the protection of resources that are in his charge.

Interlan achieves this goal by associating access controls with names as used in the syscall language. Interlan names are much more than human readable addresses; they identify resources which are of various types and have attributes that are recorded online in resource records. There is one such record for each name. Names also identify users. The operating system responds to syscalls in the context of the user which issues them. That makes it possible for the network to protect each user's interest, to allow collaboration between users when that is mutually agreeable, and otherwise to prevent one user from interfering with another.

In a system which uses names to access everything, the mere fact that something does not have a name grants to it a strong measure of security. For almost 40 years programming languages have leveraged this observation. A computer program is divided into parts which have a substantial degree of privacy. Only the names of those objects which are relevant to a given part are useable there. In this way program errors caused by accessing an object that has nothing to do with the matter in hand is ruled out by the simple fact that the object's name is unknown in this part of the program.

The Interlan architecture borrows this idea. The syscall language is name-based; addresses are not accepted as a means of referring to network resources. The dictionary of names is structured in such a way that meaningful groups of names are collected into directories and then access controls on those directories dictate in which circumstances each group of names is available. When access to the names of services and networks is restricted to a certain group of users, those services and networks cannot be used by anyone else. For example, a family can put its photographs in a directory which is accessible only by family members and selected other people. Those photographs cannot be seen by anyone else.

The name "//FraserResearch.org/emulator/big-apple" has three parts, each separated from the next by '/'. FraserResearch.org refers to a directory. Within that there is a network called "emulator" which contains "big-apple", a computer. Rules of access control for //FraserResearch.org/ are to be found in the document //FraserResearch.org/.users. A simple rule might specify the names of authorized users. Other rules might refer to the host that a user is using, the location of that user, a password that the user must give, or an encrypted certificate that the user must present. If there is no rule which authorizes a person to access the names listed in //FraserResearch.org/, the resources which those names represent cannot be accessed by that user. A physical barrier placed around these resources would not make them any less available.

It is convenient to compare //FraserResearch.org to a network where a gateway controls access to the resources within. The access control rules in //FraserResearch/.users are equivalent to program logic contained in that gateway. A directory which lists the resources in a virtual network defines that network in a more versatile way than is common for today's VPNs because there is no physical equipment to be configured. A virtual network for the exclusive use of family members living many miles from each other can be formed merely by defining a directory and by listing the family members as the only permitted users of that directory.

When a user makes reference to a resource the operating system investigates what rights of access that user has for that resource. If there are no rights the reference must be disallowed. A network operator has the option of requiring that users log into the network; in this situation the user ID itself is considered to be a resource. For example, suppose that a user Tom proposes to use the ID "tom@FraserResearch.org", the login process wants to know if Tom has the right to use that ID. To answer that question, the access control system looks in the document named "//FraserResearch.org/.users" where rules of access for each authorized user of an ID that ends in "@//FraserResearch.org" are listed. If "tom" is in that list the associated access control rules are applied to Tom. One rule might specify that the user be physically within the Fraser Research laboratory, another might require that a certain password be given. A more demanding rule might require that the user insert a particular hardware key into a USB port on his computer. The hardware "key" implements a challenge-response protocol using public key encryption. If the terms of any one of these rules is met, the user will have proved his right to use the identity "tom@FraserResearch.org".

2.9 Registration

Network users and hosts are autonomous entities; they can move independently of each other. Furthermore, a host may be connected to Interlan by more than one network interface, and a user may have more than one identity. Therefore each has a separate relationship with the network.

Host identity

A host has a registration number and each network interface has an Ethernet address. A host must register when it comes online, and Interlan will from time to time confirm that registration while the host remains online. Registration takes place through a particular network interface. It enables that interface. The host may register more than one network interface, in which case it may have access to multiple networks, or multiple points of access on one network.

When a network interface is registered it acquires a control flow which gives the host a way of speaking with the network operating system. That control flow may only be used to carry syscall messages. The control flow is encrypted even though Interlan flows are well protected, because the control flows are user accessible places outside the core network where packets can be transmitted into the core. Network safety requires that that flow not be abused. Encryption also helps to reassure the agent that it continues to talk to the host which it registered on that flow; in other words, the flow has not been taken over by another host.

If the host has multiple network interfaces, each one of them registered, then the host has multiple control flows, each one representing a separate relationship with the network. One may become disconnected while the others continue to operate. Each flow is associated with the same host interface as the control flow that was used to create it, but once a flow is established it can if necessary be rerouted through a different interface. These arrangements leave considerable flexibility with regard to the definition of what a host is. Today, that is not an idle question. A host may have more than one processor on a chip, there may be more than one chip in a box, or it may be a cluster of computers held together by nothing more that a dedicated network. So far as the network is concerned, the host is a registration number and a set of interfaces that have been registered with that number.

Virtual machine technology adds another dimension to this picture. That technology allows a single host, or "machine", to behave as if it has one or more "virtual machine" inside it. Each virtual machine executes some number of processes. A virtual machine has its own network interface through which those processes communicate. From an application perspective, this complex system is a set of applications running in one host, where each application has its own network interface and therefore its own control flow. In this way, virtual machine technology can support application mobility, i.e. a virtual machine with an application running inside it is able to move from one host to another. When the virtual machine moves and acquires a new network connection it must renew its registration with the network operating system.

User identity

There are diverse opinions about the lengths to which a network should go in order to identify and authenticate each user. Policy on this topic is implicitly defined by the access controls which users want for the protection of their networked

resources and the risk that they perceive in having their own identity made public. In order to release some of the tension between these two requirements, the Interlan operating system provides a system of access controls which is able to identity and authenticate users without making the user's identity public and without revealing the credentials that are used for authentication. By this means it is hoped that users will come to see the benefit of meaningful access controls and will not be fearful of employing user identity for that purpose.

It is not the network architect's role to decide policy, but the network architecture should include such tools and capabilities as are necessary to support the policy decisions that others will make. Therefore, Interlan is equipped with a login procedure during which a user identity and/or account number can be provided and authenticated. The network operating system associates a user with each host process and keeps track of which host process initiates each network activity.

2.10 Basic service

Basic Interlan service is a low delay service between a client and a server, both of which are application processes. Stream or message traffic travels in a flow that provides a full-duplex connection between one process and another.

A flow is a sequence of payload packets and a route along which those packets travel. Endpoints of the flow are usually processes running in host computers. When setting up a new flow, it is the network operating system's task to identify the endpoints, make sure that access to these endpoints is permitted, identify a route which the flow will follow, and configure the forwarding engines that will make the connection real.

While all flows have some basic properties in common, there is also some variation between them. Parameters which relate to the type of payload that is being carried and the style of communications service that has been selected are specified in the syscalls that an application process uses to launch the flow.

Within a flow two types of payload are recognized: A "stream" has a steady rate of flow, and a "message" is an amount of data which is not considered to have arrived until all of it has been delivered. Streams have priority and their transmission rate is controlled so that when used to carry audio or video the perceived quality of reception is as high as possible. Messages share network bandwidth in a way that minimizes delivery time.

A client-server protocol is used for flow setup. Client and server are application processes; their roles are best understood by an example. In this example the server implements a web site for playing poker and the client uses a browser to join in a poker game. The web site is "www.poker.com" which is in effect a brand name or trade mark for the poker service. The service is provided by a server, a computer that runs an online poker table. At the start of day the server sends a syscall message, listen("www.poker.com"), to the network operating system. This message makes the web site come alive for anyone who wants to use it. The client enters the service name "www.poker.com" in his browser and the browser sends a syscall message, connect("www.poker.com") to the network operating system. The result is a flow which connects the client's browser to the server's poker application. The poker game can then begin. Neither client nor server needs to have direct access to the other's computer, which has a substantial security advantage relative to Internet service. Premises networks can effectively isolate themselves from public view without detrimental effect.

Flow establishment proceeds as follows: an Interlan agent receives a listen("www.poker.com") syscall from the server, and another agent receives a connect("www.poker.com") syscall from the client. The client's agent sends a connection request to the server's agent that then checks that the client satisfies access control rules which were previously defined for the www.poker.com service. Then the server has an opportunity to accept, redirect or reject the connection request. Finally, a route for the connection is chosen and appropriate instructions are written into appropriate forwarding engines. Meanwhile, the client and server have initialized the payload transport protocol that they are about to use.

The rate of packet transmission on a flow in the backbone network is controlled by the source's forwarding engine. A rate control protocol responds to traffic conditions along the route of a flow and is constrained by the recipient host. The source host must respond to flow control signals from its forwarding engine, or it risks losing data. A legacy host can use TCP but that is not necessary. By controlling flow in this way the backbone network can give explicit support for stream and best-effort traffic types while limiting the opportunity for attack on backbone service by way of the congestion control system. In

principle, TCP congestion control as implemented for the Internet can be attacked and this can result in a denial of service attack on the backbone network; that is not the case for Interlan.

Services in an Interlan wide area network are referred to by name. By contrast, the Internet uses host name and port number as a means of service identification, but is regarded as a security risk and its use will be discouraged in an Interlan backbone. (This practice may continue within a LAN.) Clients are invisible on the wide-area network and so are not vulnerable to direct attack. The only exposure which the server has is via the service resource, which is protected by access controls. Agents track the movement of mobile hosts and make necessary changes in the forwarding engines. No host needs to know when another one moves.

2.11 Virtual Networks

Most people learn to avoid contact with people whose motives they do not understand and behaviors they do not trust. They avoid that part of town and they train their children to do the same. In order to protect against criminal activity they lock their houses when they leave home, they lock their cars when parked away from home, and they keep their hand-bags close at hand when in public places. Apparently a community is capable of learning precautionary measures of this type.

The community also creates laws which define behaviors that violate this orderly framework of daily life. Theft, bodily attack, extortion and many other practices are recognized in law and those people that act this way can expect to be punished if they get caught. A police force attempts to make the probability of being caught sufficiently high that the amount of crime is not so great as to prevent ordinary citizens from leading a largely trouble-free life.

When a plausible activity exposes others to serious risk, society provides an isolated environment for its pursuit so that the danger can be contained. Car racing is an example. Racing cars on the highway is a serious danger for other users of this public resource. The solution is to race cars on a track that is separated from the highway.

Virtual networks have comparable roles to play in supporting these aspects of personal security within an Interlan network.

Virtual networks for privacy

One interpretation of a virtual network is that it is an online “space” where a set of people can lead private lives without interference from others. Think of it as a club or a family environment where there are rules of access,. The community that owns the virtual network decides who can enter. One can see from ordinary, non-networked, life that households and managed communities are an important part of society. They are constantly being formed, they evolve, they die away. Some are more exclusionary than others; some are considered to be hostile to others. These things are part of the pattern of life and we have learned to value them.

Virtual Private Networks are widely used among corporations and public institutions. Usually they involve network equipment that is more elaborate than that which is sold to consumers, and typically some skill is required for successful deployment. As networking becomes an integral part of daily life, virtual networking will grow in its importance for consumers. Therefore, the Interlan architecture includes a simpler and more versatile version of this concept. Virtual networks are defined in the name space and are not necessarily linked to any contiguous part of the underlying physical plant. So a family virtual network can include all family members, wherever they live, and can include the mobile devices which they carry. They can share files, talk privately and engage in joint online activities without exposure to other network users.

A virtual network is represented in the name space by a directory. The virtual network has a name, the name of the directory, and has access controls which dictate who can access resources named within that network directory. The virtual network owner is responsible for specifying those access controls. For convenience of expression we say that a user is “in” a virtual network when what we mean is that the user has satisfied the access controls which govern access to the network directory.

The enabling factor is that communication between networked resources uses a flow switched network which, by the nature of its implementation, protects one flow from interfering with another. An inquisitive network user cannot inspect packets

being carried in another user's flow. Neither can he delete or inject packets into that flow. (This claim for privacy applies only to attacks which are launched through the network; physical attacks on network plant are excluded. Also wireless communication relies upon the security of encryption for its security.)

A user that confines his activities to resources located within a virtual network may feel relatively secure, providing that the access controls are sufficient to fend off attacks from outside. The same applies to a person's home; door locks must be sufficiently strong. Of course, a closed community cannot be totally divorced from the rest of the world. Members can unknowingly bring viruses and other diseases into the community, but at least the protection afforded by a virtual network reduces the risk of infection. It is also observable that a person can be both a respected member of one community and, without disclosing the fact, belong to another community which is alien to the first community. So it is today for non-networked life. Just as in everyday life, we are learning to manage these situations.

Virtual networks for containment

Present Internet use is dominated by client-server communication. Even email, which might be regarded as a peer-to-peer service, employs mail handlers which have a client-server relationship with the email user. In these cases a server advertises its availability and clients connect to the service. The Interlan architecture leverages that asymmetry by asking that service names be published while a clients may not even have a name. In that way, only the servers are open to direct attack; clients without names cannot be reached. Besides being less numerous than clients, servers are probably managed by owners who are better educated about the network and are therefore more able to defend themselves.

Telephony, which is rapidly becoming an Internet service, is most naturally a peer-to-peer service where direct contact is made between one user and another - we shall identify those users as "initiator" and "responder". Email and telephony are both personal communication services; it is just that telephony requires that initiator and responder be simultaneously involved. However, life is more complicated than that. Voice recognition technology allows an answering machine to convert a spoken message into email text, so personal communications by phone and email are converging on a single "personal communication service".

Interlan basic service emphasizes these two styles of communication: client-server and personal communication, not only because they are pervasive but also because they can be implemented in a network design which protects network users against certain forms of attack. While these two styles are by no means all inclusive, they do offer the prospect of a world in which communication is much safer than it is today.

There are other styles of communication, datagram communication for example, which are useful or even necessary for certain purposes. However, those forms of communication can be abused at the expense of security for other network users. It is the connectionless nature of datagram communication that has made it difficult to secure some Internet protocols. For example, packet insertion is a common technique employed by network hackers on the Internet.

Styles of communication, which have features of interest to network users but which also carry significant security risk, are permitted within the Interlan architecture by the addition of special services which are implemented within virtual networks. By that means, we know that only applications which visit those managed networks are directly exposed to those risks. It does not necessarily mean that these applications carry a virus or have been compromised, it is just that they have been exposed to what may have been a risky situation. For example, some multi-person online games involve group activity where person-to-person interactions happen quickly without following any particular pattern. The problem is that we do not yet understand how to distinguish between these activities and patterns of behavior that are blatantly offensive. File sharing is in another category where some of its applications are on the border-line between acceptable and non-acceptable behaviors, depending upon who is making the judgement. It also employs techniques that allow bot-nets to thrive. Thus we find that wherever file sharing is successful bot networks can flourish and can evade detection.

So, a private network is an online "space" where a group of people can work or play together according to rules of their own making. Their style of network activity may be sufficiently unstructured that it is hard to distinguish between good and bad behavior. It may just be that patterns of behavior are changing rapidly so rules are hard to define. The one constraint for all

those who enter a private network is that the special network features, available to members of the private network, are not available to those same people when operating outside the private network.

So, while the Interlan architecture recognizes three “normal” styles of service, client-server, personal communication and multicast, it also includes private networks within which other behaviors are expected to develop. Anyone can form a private network and define its rules of behavior. The safeguard is that membership in a private network is explicit and entry to a private network is controlled. While membership is not public knowledge, it is knowledge that can be obtained by law enforcement. The hope is that by making it clear who belongs and who does not belong to a private network, society will develop practices which support risk while providing reasonable protection against anti-social activity.

There are those who will reject this concept. For example, those engaged in the exchange and distribution of copyrighted material contrary to the laws of copyright. That activity can take place in a private network, but clarity about network membership will mean that there is a risk that law enforcement will be able to penalize the entire group. That we see as the price for protecting innocent people and nations from being attacked by bot-nets.

2.12 Special services

This section reviews the way that the operating system architecture supports special communications services to meet specific user needs which can only be addressed in a wide area network. The capability to provide special services has been a necessary part of public communications networking for many generations. No doubt that pattern will continue.

Private networks provide the operating framework for Interlan special services described below. As described in (ref) an Interlan private network is defined in software. It corresponds to a distinct physical structure. In some cases specific hardware is a critical component, required in order to meet a particular need. One feature of this framework is the means by which physical and virtual infrastructures are correlated.

The special service “**.**config” is a web site that is used to observe status and set preferences for the network in which the **.**config name is defined. So, “mynetwork/.config” provides administration for mynetwork. (By convention, names which relate to built-in functions of the network operating system begin with a dot.) One service provided by **.**config relates network concepts to the components which make up the physical network. For example, a virtual network with a particular name may correspond to a specific configuration of wires and switches in the physical plant. In that case every host that appears in the virtual network should have a counterpart in the physical network. Automatic configuration will maintain that correlation and report on discrepancies.

The syscall language is based upon an abstract view of the infrastructure, partly because that infrastructure is much more complicated than users need to know. Furthermore, the operating system is used by people who are focused on the application that the network is being used for rather than the means by which it does its work. So, for example, the operating system presents a virtual network as a list of resources which exist within a specific security domain and are accessible to a certain list of users. The resources include real property such as host computers and services which are offered by those computers. For most purposes where the user interacts with the operating system it is the service that a user is concerned with rather than the physical hardware of the computer. Some computers have names because there are people that log into them or share files with them; most other computers do not need a name. Cell phones typically do not have a name, the relevant name for making contact is the user ID or, more likely, the abbreviated user name as written in a personal address book.

For many people “networking” is about staying in touch with family, friends and business associates. Social networks are causing a revolution in the way that this communication takes place. There has been a significant increase in the number of people that an individual stays in touch with on a regular basis. In this context, a virtual network can be a place where the group meets. Technically, the virtual network is a directory which contains a list of those friends along with rules which restrict others from breaking into the group. The virtual network provides privacy, keeping others out of the conversation and allows the group to share information and photographs etc. without fear that they will be exposed to the rest of the world. The virtual network with access controls and a membership list is created and managed by the group. Groups may have

overlapping memberships, and the members of a group can be spread wide apart on the network, yet each group's privacy is not compromised.

Virtual network structure and support is implemented by the network operating system; it does not add overhead to packet transport because, unlike the Virtual Private Networks used by corporations, there is no firewall involved. Decisions about whether a particular person can have access to a service are made when the use is attempted not when the data is being transferred. That is much more efficient in terms of the load that it puts on the network. Resources have access controls which are specified along with the list of users that may use the resource. If access controls do not permit access then an agent will reject the connection setup request. Once a flow has been established, payload packets follow the route that has been approved, packets cannot stray onto any other route, and there is no need to keep checking that the conversation is permissible. Thus there is little or no overhead caused by the use of a virtual network.

Four special services are briefly described below.

Traffic monitor

A traffic monitor is an application program that monitors and participates in the control of traffic within a private network. It can learn about every service request that enters or leaves the private network and the network may be configured to expose the monitor to all service requests generated within that network. The monitor has the opportunity to block requests or influence their routing. The monitor cannot alter the nature of a service request nor can it enable a request that the network has blocked. The monitor cannot force the use of a route that the network does not permit, but it can propose a route that the network may not have considered. Payload packets are out of the monitor's reach but it can learn about traffic volume and the status and configuration of the private network's equipment. The monitor feature is intended for installation in a free-standing host or network management product, and may be used by a local network administrator with sufficient privilege to administer all of the resources installed within the private network.

UDP datagrams

Most IP traffic uses the TCP protocol and can use Interlan basic service. That service can also be used by much of the remaining UDP traffic but there remains a set of UDP applications that rely upon datagram routing. An Interlan network can carry that traffic by feeding it into a private network which does UDP datagram forwarding. For that purpose, one or more of the forwarding engines in an Interlan switching center may be replaced with a UDP forwarding engine. When travelling between switching centers this traffic is encapsulated in Interlan flows. By transporting UDP in a private network, regular Interlan traffic is not exposed to the attacks which can be launched in a datagram network. Interlan's security arrangements prevent interaction with Interlan standard traffic and systems. One UDP application may interfere with another in the same private network, that is a long-standing feature of UDP service, but they will not interfere with users of other services.

Ethernet bridge

The Interlan packet format conforms to the Ethernet 802.3 standard, but Ethernet uses datagram packet forwarding while Interlan networks use flow forwarding. The Ethernet standard 802.3ah allows Ethernet packets to be encapsulated within Ethernet. This means that premises Ethernets can be bridged together with Interlan flows. Corporations can interconnect their Ethernet LANs, retain their privacy and enjoy the economic benefit of a shared backbone network by using an Interlan virtual network equipped with an Ethernet forwarding engine. This forwarding engine supports Ethernet packet forwarding and Ethernet multicast within the virtual network.

Multicast

Multicast is used today in at least two contexts: video distribution and coordination of asynchronous systems.

Today, CATV networks transmit scheduled digital video broadcasts efficiently in their multi-channel aggregation networks. However, there is a growing volume of unscheduled or independently scheduled digital transmission which could use a versatile high quality delivery system that is compatible with computer systems and mobile devices. An Interlan backbone is flow-based and rate controlled; it can deliver the transmission quality that is required for these flows. Multicast switching is

available in the context of a multicast virtual network. Multicast flows across a wide area are configured and controlled by hosts using syscalls and the .config network management system.

Distributed computer systems use multicast to coordinate their activities. An Ethernet bridge network, which incorporates multicast, can support this Ethernet application over a wide area. (But, of course it cannot reduce the propagation delays which are inherent in wide area communication.)

3 Network operating system

Computing practice is going through a transition which will surely have an impact on network design. Hosts which were once static are now mobile and mobile telephones are morphing into computers. Applications which were at one time executing in a single host are now distributed across multiple hosts. Google, Amazon and others have built massive computer systems with hundreds or thousands of processors in one large cluster, a "cloud". By connecting the cluster to the Internet they are able to offer its computing capacity to users across the country, perhaps across the world. With "cloud" computing it is no longer clear to an application user where his computing is taking place. The next step in this evolution merely increases that uncertainty. Virtual machine technology simulates one computing environment within another. The technology has simplified the task of moving an application program from one computer to another. Application mobility which takes place while the application is working on a user's problem is surely not far away. A task started in one machine may finish in another.

Computer hardware design is also changing. There are multiple processor cores in a single chip, multiple processor chips in a computer, and multiple computers in a rack-mounted cluster. Networks are embedded within these complex systems, and networks interconnect one such system with another. Thus the identity of a host is eroding and the distinction between networking and computing is becoming blurred. Network architecture and host operating systems must surely follow suit.

It is time to think of the network as part of the overall computing system. The name space for computing and communication is global, implemented today by a distributed name service (DNS) that has servers in every country. Host file systems are becoming local extensions of that global context. Instead of launching an application in his own computer, a client can now with equal ease use DNS to locate and execute an equivalent application in a host many miles away. A process in one machine can mount and refer to files stored in another. While these host file systems employ access controls which attempt to provide individual privacy within this global context, DNS operates like a global telephone directory which anyone can use without constraint to acquire the address of any one of a billion or more hosts. Thus, Internet networking undermines the security which host operating systems worked so hard to achieve.

There was a time when application programs used disk addresses of files when transferring data to and from disk. But time-sharing and the need for application privacy changed that. Applications today use only file names, and leave it to the operating system to obtain the relevant location information, to check that the user is authorized to access the file, and then ensure that the data that is transferred is actually that which was authorized. That evolutionary step was a great success. It led ultimately to device independent application software, file sharing administered by the host operating system, and access controls for individual files. Today, application programs executable from the Internet need that same ease of use, control over resource sharing and protection for intellectual property.

There need to be some fundamental assurances which derive from the integrated architecture of hosts and networks. Time sharing was founded on memory management, a secure boundary between application program and operating system kernel, and a system call interface by means of which operating system services can be obtained without serious risk that the kernel will be compromised. Are there equivalent fundamentals that can underpin worry-free computing in a global network? More fundamentally, is it possible to define a realistic goal for security and safety when the infrastructure spans the globe, is operated by communities with widely varying interests, and is seen by many as an avenue for terrorism, even war?

The time has come for a transformation in network architecture, from one which sought simplicity of operation to one which puts responsibility where it needs to be. Instead of a totally open architecture, there must be a kernel that administers

shared use and protects one customer's information from tampering by another. A framework is required which detects and discourages anti-social and criminal behaviors. The Internet is increasingly a mobile world, numerically dominated by consumers who have great difficulty understanding the complexities of the systems that they are using. They need help and support so that they can lead safe and prosperous lives. In short, it is time to create a network operating system.

3.1 The operating context

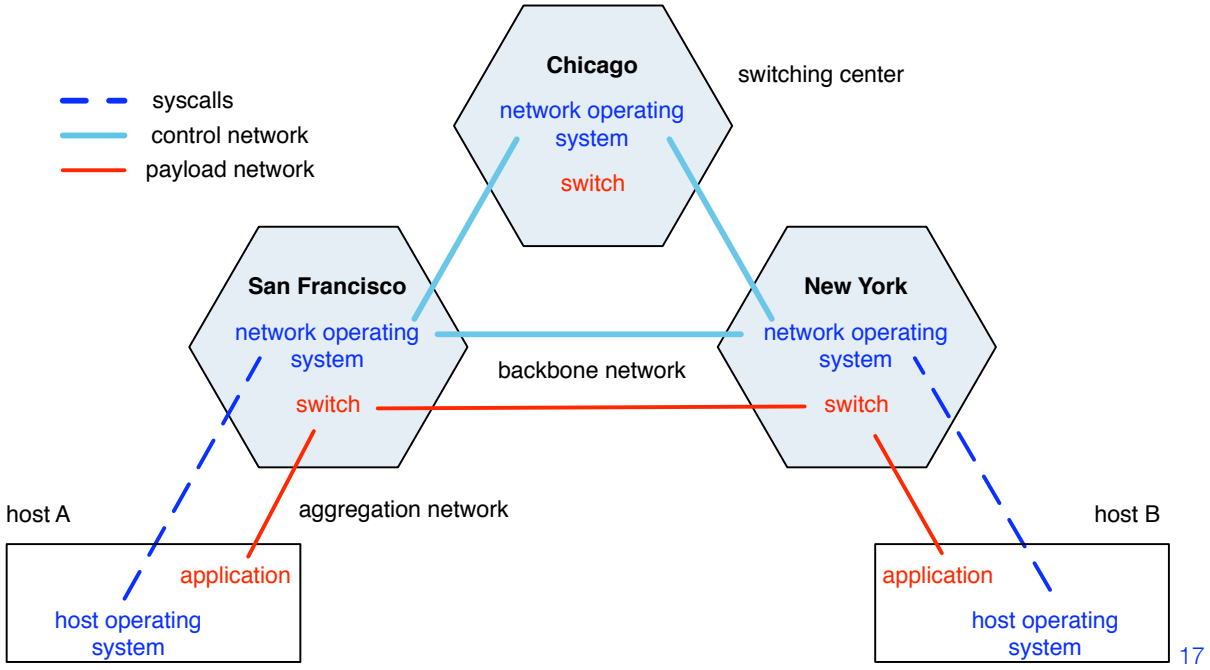


Figure 3.1. Elements of a global network - operating system view

Figure 3.1 illustrates network architecture at the level of the network operating system. Three switching centers are shown, each one is presumed to be at the center of a region. Homes within a region are served by aggregation networks which provide connection to the nearest switching center. For the present discussion of operating system design we assume that the infrastructure consists of many regions with switching centers that vary in size over a wide range; the largest being sufficient to serve a metropolitan area with millions of households. The switching centers are interconnected by the trunks of a backbone network. In Figure 3.1 red lines represent payload information flows within the aggregation and backbone networks. The term "payload" indicates that the information is generated by a user application and is being carried to another user application. A payload flow is equivalent to a TCP session, or a UDP session that involves just two application processes. (Support for other UDP applications is described in (ref)).

The network operating system resides in the switching centers. It is a distributed system consisting of computers that are interconnected by a control network. The control network is colored light blue in Figure 3.2. Elements of the network operating system use the control network to coordinate their work. That work is in response to syscalls which are generated by host operating systems. A syscall message is the Interlan network's equivalent of the system call which a host application uses when interacting with its host operating system.

The control flows upon which syscall messages are transmitted are represented as dashed blue lines in Figure 3.1. Each control flow implements an encrypted session between a host operating system and the network operating system. Encryption is used so that the network operating system can have some confidence that it knows which host it is talking to. Encryption also complicates the task for another host that would attack the control flow.

The global backbone network is virtually two separate networks, a payload network that carries user data and a control network that transports operating system messages between the switching centers. These two networks are physically

disjoint, or if that is not possible it must be that no payload packet, however formatted, is able to cross over between one of these two networks and the other. This does not necessarily require that separate transmission cables must be used; it could be that the distinction between payload and control is that they travel with different wavelengths in one optical fiber. These requirements are made so that there is the least possible risk that an attacker connected to the payload network can launch an attack on the control system. Experience with the Internet, its name servers and routers, has shown that attacks launched on these components of the network control system can be both devastating and difficult to prevent. These attacks are launched from host computers which can reach their target because Internet name servers and routers are addressable by packets launched by network users. Therefore, Interlan name service and routing functions are implemented in the control network by the network operating system. They can only be reached from other network operating system components attached to the same control network.

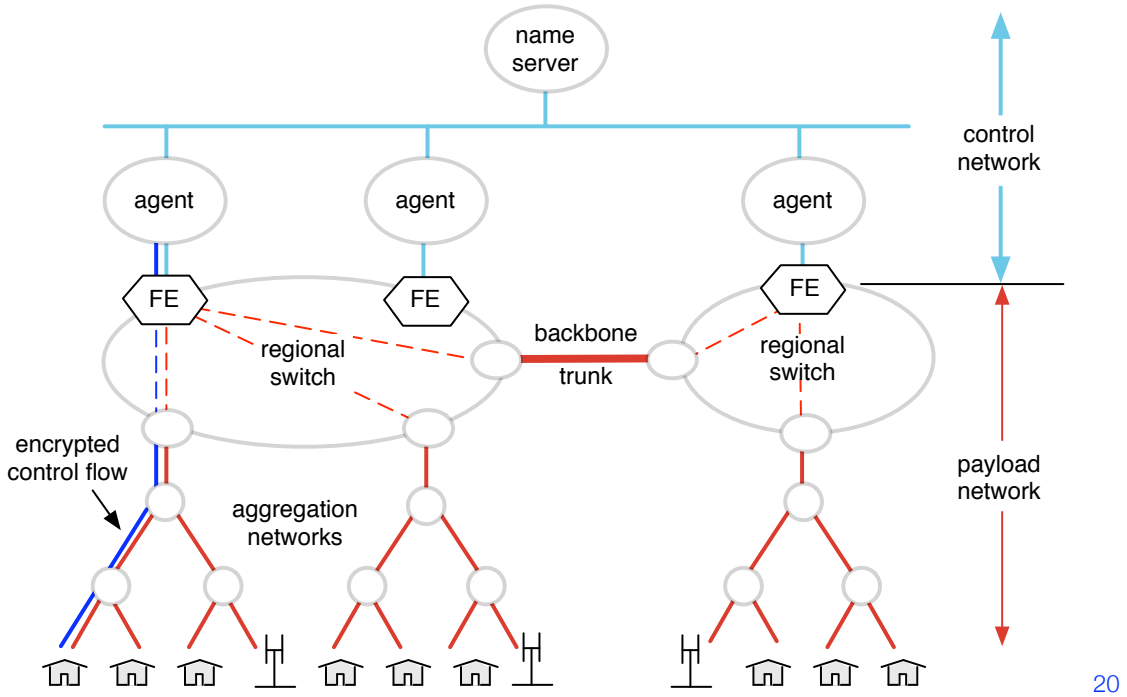


Figure 3.2. Two switching centers interconnected by a backbone trunk

Figure 3.2 illustrates switching center equipment for two interconnected regions. Agents implement the user-facing, traffic control and resource management features of the operating system. It is the agent to which each control flow connects. Payload flow is illustrated with red lines; dashed red lines show how these packets flow within the two switching systems. Forwarding engines, “FE” in the diagram, are the means whereby the packets of a flow are steered towards their destination. Homes, businesses and wireless base stations are illustrated by the icons at the bottom of the diagram.

3.2 Agents

The agents of one switching center are implemented in a computer cluster with an associated disk array. This report refers to agents as if they are individual computers because it is easier to visualize the system that way, but in fact the agents are part of a highly reliable computing complex with processes that can migrate from one processor to another. System operation does not restrict an agent to be a single process or reside in a particular processor, but each agent’s capacity for work can grow and shrink according to the work load that it must handle. Thus, an agent is a variable set of processes which share a single Ethernet address, and have disk space enough to keep resource records for a certain set of resources.

Each agent is connected to a forwarding engine through which a control flow connects the agent to its users. It is sometimes said (for convenience) that a host is connected to its agent. However, the host and forwarding engine are attached to the payload network whereas the agent is attached to the control network. Therefore, when a host uses an Ethernet address to reach its agent it is actually using the Ethernet address of the forwarding engine. It is that engine’s job to pass the message to the agent.

3.3 Switching center design

The switching center illustrated in Figure 3.2 has an unusual traffic flow that some would consider extravagant. Packets that enter from an aggregation network or trunk pass twice through the switch that is in the center of this design. Packets entering the system transit the switch to reach a forwarding engine where they are processed, and then they transit the switch a second time to reach the trunk or aggregation network on which they will depart. In a traditional large switching system packets are processed at the point where they enter the system, and then pass once through the switch to reach the point of their departure. Some might say that the Interlan design is extravagant because the switch is a critical and expensive part of the equipment, and to send every packet twice through the switch at least doubles its cost. However, there are both strategic and technical reasons for this design. Three strategic aspects will be mentioned here; the technical aspects are addressed in (ref).

More than half of the devices that use the Interlan network are expected to be mobile. Interlan architecture treats mobility as a core requirement rather than as a special feature added onto what would otherwise be a network designed for static devices. Also, security is a priority of the architecture, which means that packet processing in a forwarding engine needs to reflect the state and configuration of each flow. Some of this information is provided by the agent and some is collected by the forwarding engine. By routing the flow through the switch from the point of entry to the forwarding engine, and then through the switch to reach the point of departure, the system does not need to disturb the forwarding process when the hosts at the ends of a flow move about. In this way the design reduces the complexity of flow management. It avoids placing constraints on the functions that the forwarding engine can perform, and it makes it more likely that packet forwarding will work correctly at high speed. A similar situation arises when equipment failure forces rapid rerouting of trunk traffic. While reconfigurations due to failure are much less frequent than the movements of a mobile device, the single failure of a backbone trunk may suddenly demand that many flows have to be rerouted quickly.

The third strategic consideration is communication service versatility. Input switching provides the ability to have a choice of forwarding engines and to route flows individually to engines according to their traffic types. It is no longer necessary that one communications service must fit all needs and all situations. Incompatible service types can be supported in different forwarding engines. For example, some packet insertion attacks experienced on the Internet are linked to the integration of datagram and connection based services in one framework. By keeping those two traffic types separate hackers can no longer use datagram switching to launch packet insertion attacks on flow-based protocols. Multicast can also benefit from forwarding engines designed to meet its particular needs. As other traffic classes proliferate a switching center can introduce new forwarding methods and hostile or incompatible traffic types can be isolated from each other without creating separate physical networks.

3.4 Critical interfaces

Interlan security depends upon a clean and secure boundary between the network operating system and the payload network. Forwarding engines are the only equipment with access to both the payload network and the control network. The engine is focused on queueing and moving packets in the payload network. It is indirectly connected to the control network because it takes instructions from an agent which dictates how the packets of each flow are to be forwarded. Control flows also use the forwarding engine to cross the protection boundary between control and payload networks. Clearly, this is a critical point in the Interlan architecture; it could pose a risk to the security of the operating system.

To maximize performance of this critical device and to minimize the opportunity for attack, the engine is a hardware device with separate control and payload sections. This form of construction is designed to minimize any possibility that a payload packet will cross into the control network, and to make sure that packets in the control network cannot be inspected from the payload network.

Control flows necessarily cross the boundary between payload and control, so the forwarding engine's task is to inspect each message and do what it can to ensure that only genuine syscall messages from the appropriate hosts pass through. The engine decrypts the host transmissions, rewrites the syscall messages contained therein, and performs a first check on message format. To assist in this task, the syscall language used on control flows has a minimum number of message types and these have restricted properties. One might say that the forwarding engine does for an Interlan operating system what a

memory management unit does for a host operating system. It keeps one user from interfering with another's flows, and it allows syscall messages to pass between user and operating system while preventing user interference with the operating system.

The control network is used for internal communications within the operating system. Agents, name servers and other elements of the network operating system collaborate by exchanging messages on this network. These messages, known as signals, are confined within the control network. However, the global network of agents is large and spans many administrations. Attention must therefore be paid to interfaces between network administrations, particularly at international boundaries. If one administration cannot fully trust another, that may lead to a pair of gateway interfaces between two parts of the control network. Each gateway will need to do what it must to preserve the integrity of the control network to which it has allegiance.

3.5 Payload network

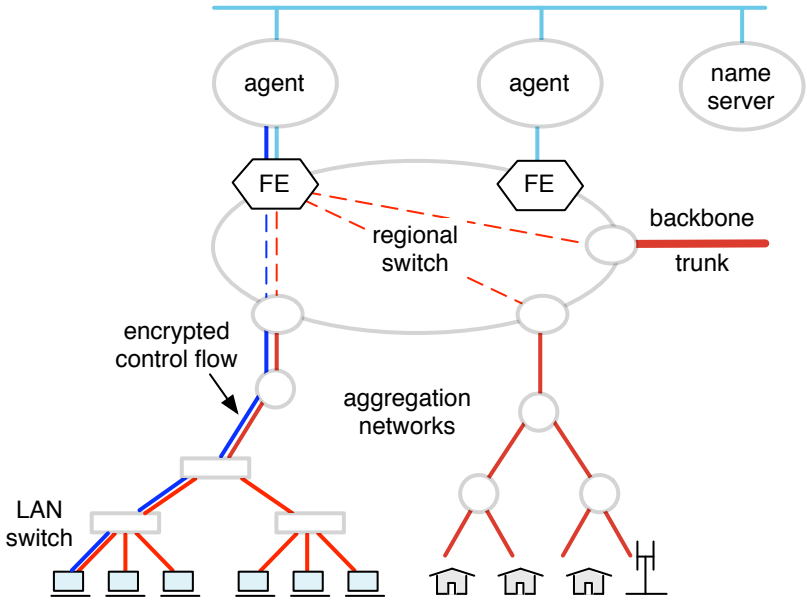
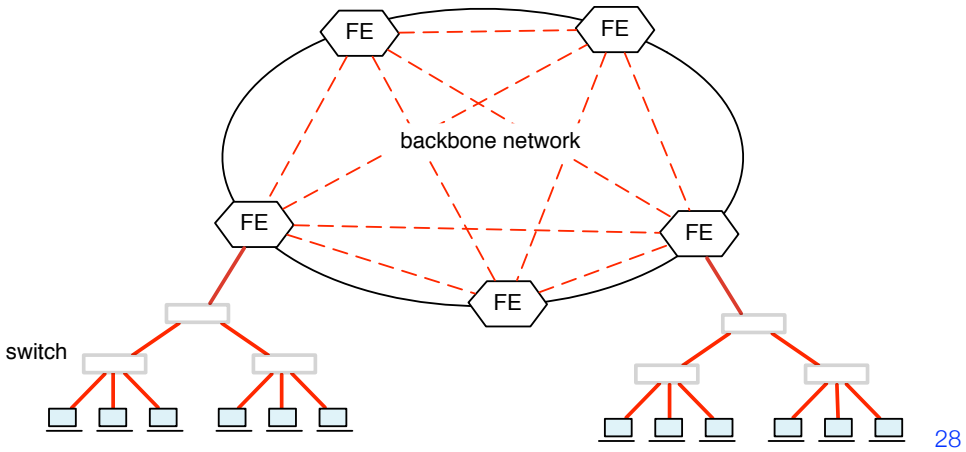


Figure 3.3. A premises network connected to Interlan

Figure 3.3 shows a premises network connected to a regional switching center. The keyboard and display icons at the bottom of the figure illustrate host computers and other equipment attached to a local area network. Blue lines represent a control flow that is encrypted and is in-band on the Ethernet. A separate control flow connects each host that is using the wide area network to an agent in the switching center. Payload traffic from each LAN has a logically separate connection to the forwarding engine which its agent controls. Even when an aggregation network is shared by many homes, the traffic to and from an individual home has its own identity which is checked in the forwarding engine.

The forwarding engines create a wall of protection around the backbone network. The concept is illustrated in Figure 3.4.



28

Figure 3.4. Conceptual view of backbone network isolation

The ellipse in the middle of Figure 3.4 represents the conceptual wall which surrounds the backbone payload network(s). The separate control network which operates in parallel with the payload network is not shown. For the purpose of this discussion the combination of all backbone payload networks is represented here as a single network. All payload traffic which enters and leaves the backbone must pass through a forwarding engine. The engines check that incoming packets conform to specifications prepared by the agents when each flow is established, and the engines apply a packet header for backbone travel which uniquely identifies each flow in the backbone. By this means there is no opportunity for a rogue host to inject packets into the flows of other hosts. The content of every end-to-end flow is protected from interference and inspection by other than the one to which that packet is addressed.

3.6 Operations and administration

In addition to control and supervision networks, a public service network requires a substantial data processing complex that is connected in various ways to data feeds from the operating network. That is evident from experience with the world's telephone systems. The terms "operations, administration, maintenance and provisioning (OAM&P)" have been used to summarize those ancillary functions. Whereas the agents and name servers are the means by which communications take place, the OAM&P systems provide the necessary framework in which the network can operate.

Network operations have to do with support for customers that use the network, administration is the busy work of running the network service business, maintenance refers to keeping the wheels oiled and turning, and provisioning makes sure that sufficient equipment is in place and suitably configured to deliver that service. While all of this may seem pretty ordinary business, when it applies to a national network with hundreds of millions of customers and when the customers are all online, expect a fast response and continuous service without error, the result is a large and complex data processing system that is linked to the communications infrastructure. Such a system cannot be without error and loopholes. So the traffic which the OAM&P systems generate should be kept away from the control network upon which the agents depend. In effect there is another service interface, not shown in Figure 3.2, which protects the network operating system from attacks launched through vulnerabilities in the OAM&P systems.

One more system is not shown in Figure 3.2. It is the supervision system which is present in virtually every part of the payload and control networks. Supervision is mostly a passive activity, monitoring events in the network, looking for unusual traffic patterns, checking on security check points, and looking for indications that trouble is either in the making or has occurred. For most of the time the supervision system reports its observations to relevant parts of the control system and it is the control system that initiates whatever action is necessary. However, there are circumstances, such as failure of a major trunk transmission line, which call for immediate action. In these cases the supervision system may make a pre-planned response even as it propagates an alarm signal into the control system. For example, a standby transmission line might have been assigned and the supervision system is programmed to trigger immediate cut-over to that alternative route. This design has been compared to the human autonomic control system where the central nervous system triggers an

immediate response to an emergency even as an alarm signal is transported to the brain where a more considered response may be initiated a short time later.

3.7 The network system

Just as a host operating system hides the diversity of a host's resources behind a uniform file system interface, so Interlan integrates the assets of a global network within a compatible "network system" of named resources and information flows.

Network system = Application process -> Information flow -> Named resource

Over time, application programmers will come to see the combination of host and network operating systems as a single system which manages application processes and three resource types:

(a) File - a document or other source of information which can be accessed by clients.

(b) Service - an autonomous process that does work for clients.

(c) Network - a global communications facility by which clients can reach remote resources.

Aspects of a host's architecture, including processor cycles, memory space and file system are typically bundled by host operating systems so that each user has the impression that a host is an infrastructure dedicated entirely to his purpose. This abstraction, with access controls to ensure a user's privacy, has been key to the successful sharing of computer system resources. Unix has been one of the more influential operating systems to use this technique. Local area networks and the Internet have broadened a user's perspective so that the resources employed go beyond the computer that he or she is using. Today, the Unix socket system provides access to a global name service and supports compatible transport protocols that allow hosts of disparate manufacture to exchange information. The challenge now is to Integrate the global communications capability with host operating systems so that users can safely share in the use of networked resources. The Interlan operating system augments individual host operating systems so that their users can have controlled access to shared services and networks world-wide. For example, controls specify who can see a resource name, what they can do with the resource, and which networks can be used to reach it.

Time sharing has taught us that application processes must be protected from one another. For that purpose, an Interlan service is, in effect, a protective front for the processes which individually execute the application on behalf of each client. The service is a resource, it has a name which is recognized by the network's name service, and there is an agent-administered procedure which authenticates the service request and connects the client to a server process. It is not necessary in this scenario for processes and host to have publicly disclosed names. As a result the opportunity for direct attack on a host is reduced.

The global infrastructure is a network of networks, some of which provide a protective operating context for clients and servers, and some offer special network features. For example, an Interlan network may protect the several homes of a family, or it may protect students in a school. In either case the network provides a certain degree of privacy for the people that it serves and can block connections that fall outside a certain profile. Multicast is implemented as a special-purpose network which might, for example, distribute educational programs to applications in city schools. An Interlan virtual network is a resource type, not to be confused with the physical infrastructure which underpins Interlan service. Just as a host operating system presents an idealized view of a file system, not a view composed of disk blocks and drives but one that contains named files, so the Interlan operating system presents an idealized view of the network system, wherein services and clients communicate directly with one another. For a host operating system, processes are the infrastructure components which provide computation; for users of the Interlan network operating system routers are the barely visible components of the network infrastructure. We know, of course, that the real equipment includes forwarding engines and transmission lines, but the user interface to the network operating system says nothing about that. (.config is an administrator interface for use when it is necessary to reach into the infrastructure (ref).)

resources infrastructure

	systems	components	systems	components
computing	service	process	Unix	processor
communication	network	router	Interlan	agent

Table xx summarizes the terminology and design perspective that is used in this report. Distinction is drawn between the infrastructure which underpins distributed applications and the resources which are created and managed by the network system at the request of those applications.

3.8 Resource directories

Resource names are recorded in a hierarchical system of directories, two of which are shown in Figure 3.5.

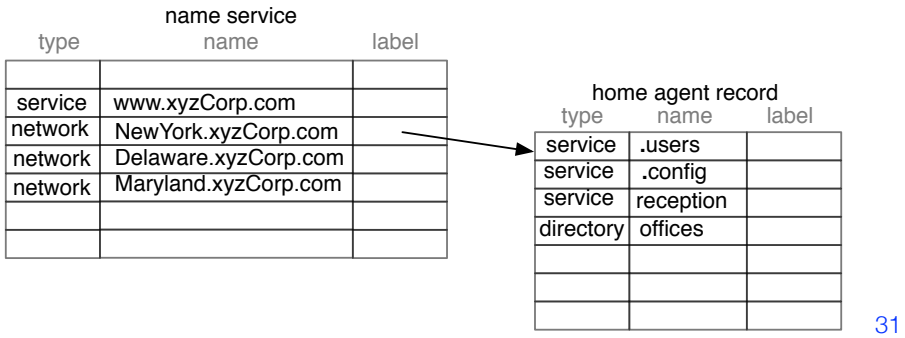


Figure 3.5. Directory for //xyzCorp.com/NewYork

Directory entry has three fields:

type	resource type = directory, service, network, host or router
name	component of the resource name.
	The reception desk service name is //xyzCorp.com/NewYork/reception
label	Ethernet address and resource number of resource record.

Special service names begin with dot.

.users	access control policy determines who can use the New York network (ref).
.config	network management system for the New York network (ref).

A service is a dynamically assembled set of “server” processes that are configured to do a certain type of work for “client” processes that connect to the service. A service advertises its availability by contacting an agent who creates a resource record for the service and asks the name service to make a suitable directory entry for the service name. The resource record can then be accessed by any suitably qualified client (or server) by connecting to (or listening to) the service name. In effect, the resource record becomes a meeting place where servers can group while waiting for clients to show up, and clients can visit in order to find a server.

The network operating system verifies that the clients and servers who visit the service have the required qualifications. Rules which govern access to a resource are administered by the service named “.users”. Approved service requests are then handed to an available server which may also check the client’s request before accepting the assignment.

The word “network” when used in the context of the network operating system is a collection of resources that are grouped together for mutual benefit and protection. This grouping may involve dedicated switches and transmission lines or it may be achieved entirely with operating system software. Resources that comprise the network are listed in the network’s directory. Network implementation allows customized connection administration, and may be supported by certain special services such as multicast and LAN bridging (ref).

From a network operating system perspective, the distinction between a host and a network is steadily diminishing. A network of hosts is sometimes seen as just a large host. Cloud computing illustrates the point. On the other hand several potentially mobile virtual machines running within a host may be regarded by Interlan as a network of hosts.

Interlan does not publish host names in the wide area network; service names provide a more easily defended public interface to host resources, and it is safer to keep host names out of reach where they are less likely to be attacked. In effect, hosts are invisible to users located outside that local context. Host names defined within the local context give a local user full access to all of his equipment with much reduced fear of attack from outside.

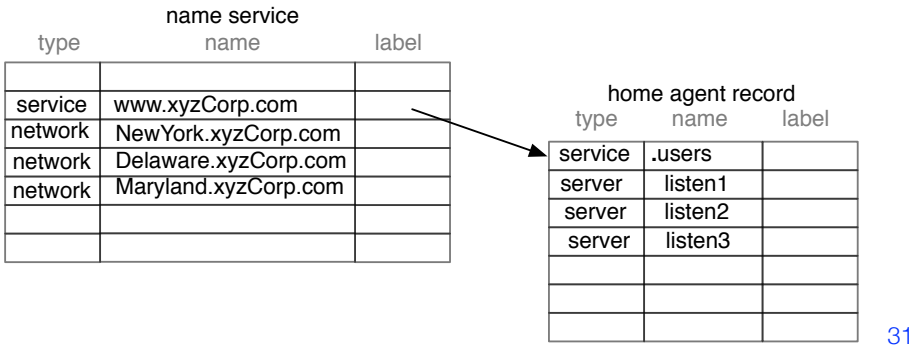


Figure 3.6. Directory for //www.xyzCorp.com

The information about a service is in two parts: the “name service record” and the “home agent record”. The left-hand directory in Figure 3.6 illustrates a name service record; it contains the service name “www.xyzCorp.com”. The home agent record has the structure of a name service directory and it is held by the home agent for “www.xyzCorp.com”. The distinction between the name service record and the home agent record is that the home agent record can support frequent change in the information that it holds, while the name service record is expected to be static for most of the time.

The home agent record has the structure of a directory so that name service can translate “www.xyzCorp.com/**.**users” correctly. However, “www.xyzCorp.com” is identified as a service, so the name service when acting for a network user will not consider it to be a directory. Administrators and operating system processes can obtain access to the names in a service directory. As a further defense of operating system integrity, name translation enforces the rules contained in “**.**users” even when the user is an operating system process.

The service is implemented by servers (hosts) that have connected to the service using the syscall listen(“www.xyzCorp.com”). The names “listen1”, “listen2” etc. are the agent’s names for these servers. One of these listeners will be chosen to handle each service request. A globally popular and/or mobile service may have multiple foreign agents for each of which there is an agent record. Home and foreign agents each handle service management for their area of responsibility. In this way distribution of the work load is coordinated, mobility is well covered, and major network outages have only limited impact (see (ref) for more about mobility).

3.9 Name service

Interlan operates a name service which is an extension of the Internet’s Domain Name System. When a domain name record contains a resource label, this label refers to the resource’s agent. Otherwise, the DNS directory entry refers to a resource that is available only on the Internet, so Interlan uses the Inernet to reach the resource.

Even though Interlan interprets names differently than Internet, the practical difference is small because Internet users perceive many domain names to be the names of services. For example, “weather.com” is a domain name that the Internet translates into a host address, then the application connects to that host using a particular port number and thereby obtains a service. When Interlan translates that same name it expects to obtain a “resource label”, which in effect is a reference to a record, which describes the weather.com service.

Access controls dictate the way that the name can be used. For example, access controls can specify conditions under which a user may list the contents of a directory, and which users can make changes to the directory. A user's perception of the Domain Name System is that it too is hierarchical, but the rules of access are not specified by network users. For example, the owner of a DNS directory name cannot make the directory invisible to all but a few friends. See (ref) for more information about access control.

A significant part of the difficulty involved in providing what on the face of it seems like a rather simple service, comes from the size of the network - the distance that it covers and the billions of users that rely upon it. Propagation delay is a factor. It takes light about 1/6 second to travel half the way around the world and back again. While that may not seem much to be concerned about, consider that a human reacts 10 times faster than that. So, many copies of the names database are distributed around the world. While adding complexity, database replication provides redundancy and ensures that a required name lookup can be completed quickly. Names in that database are structured so that management for billions of names can be delegated in an orderly manner. More challenging perhaps is the task of obtaining international agreement over the way in which all of this is done. The Internet community addressed that problem and today we have a Domain Name System in which there are more than a billion names already recorded. Many of those names are, in effect, trade marks and have significant financial value. Clearly, the Interlan must be built upon this base.

3.10 Resource names

As is common practice among host operating systems, an Interlan resource name is the name of a resource preceded by the resource name of the directory in which the resource name is held. A '/' separates one segment in a resource name from the next, and a leading '//' or '/' signifies that the resource name starts at the root of a hierarchical name space. The following example has three segments.

//FraserResearch.org/lab/printer

The first segment in this resource name is a Domain Name; it is translated in the DNS system. //FraserResearch.org refers to a directory which contains names specific to Fraser Research. The directory is a resource managed by an agent in the regional center that serves Fraser Research. The final two segments of the resource name are to be found in agents or in a name server on the FraserResearch premises.

It is perplexing to those who anticipate integration of computers and networks that domain names use a different syntax than names used in host operating systems. Both name systems are hierarchical but they use different punctuation. One reads from the left and the other from the right. For Interlan, the root of the tree of names is the root of the Domain Name system. The objects which are named by DNS are said to be "resources" and their attributes are described in resource records. The resources which are the focus of Interlan's attention are more mobile and have more active state than can comfortably be managed by DNS. For example, Interlan provides resource owners with support for host, process and network mobility, and it tracks network use in sufficient detail as is necessary to underpin the security of distributed and dynamic systems. So Interlan Integrates the DNS and host naming systems into a single design that is efficient at both ends of the hierarchical name space.

Reading from the left, the first segment of a name is a domain name and the records associated with it are designed for globally administered names. The resource name as a whole conforms to the syntax of a name in a computer system. It is also compatible with web Universal Resource Locators. The second and subsequent segments of the name identify records that are designed to support a resource's active state and the supervision necessary for security and privacy.

It is anticipated that in future port numbers will cease to be used as service identifiers and that the service name which is used to reach a service will become a Universal Resource Locator (URL). In this way resource identification is by name and when connecting with a service a client has the opportunity to say more about what he actually wants the service to do. That information will help the service scheduler choose the most suitable server to handle the task. The syntax chosen for an Interlan resource name is intended to be compatible with that outcome.

It is also anticipated that the boundary between computing and networking will continue to erode. This will occur not only as a result of cloud computing and services such as Akamai which are distributed in the network, but also it will be a natural

outcome from the increasingly widespread deployment of micro-controllers and sensors. These developments blur the boundary between networking and computation.

Finally, some flexibility in the interpretation of a name seems to be the best way to allow the naming system to evolve. The proposed rule for translating an Interlan name is as follows:

> Now translate the name, one segment at a time, from left to right. Translation stops when the end of the name is reached or a name segment has no translation. Record the length of the part of the name which was successfully translated.

As name translation proceeds, directories may be encountered which have access controls. Those controls must be evaluated during translation, and translation must terminate if access control on a name segment prevents access to the resource that it names. When name translation is part of a connection request, the entire name and the length of the translated part should be handed to the server which is the target of the request. In this way the translation process determines where the name ends and the additional text begins. The destination can use that additional information to obtain a better understanding of the client's intent. This method allows a connection to be routed through a multi-stage network structure such as may result from the further integration of computers and networks. Each stage translates a little more of the name until the final destination is reached.

3.11 Local names

There is a significant practical difference between circumstances in a home or office, and the circumstances in a wide area network. Many of the security problems which have plagued the Internet derive from attacks on personal computers that are installed on local area networks. The attacker's exploits convert those computers into platforms that assist in denial of service attacks and other offensive activities. Interlan must assist in tightening up security in premises networks if the overall goal of a safe computing and communications environment is to be achieved. A contributing factor to the difficulty faced is the significant degree of freedom that users need and want in their local area networks. For example, there are services in every computer for the analysis and repair of problems with the machine, its software and the network to which it is connected. Other host software provides a valuable service but is not robustly constructed. Many network users are not familiar with the technology which they have purchased and/or do not have the will or the discipline to defend it properly. Probably that should not be necessary; a home is, after all, the one private place that each person has and it is where he should feel able to let his guard down. Some aspects of this situation are to be found in offices and other small business premises. The priority there is on getting the job done rather than fending off attacks from outside. It is all too easy to allow that priority to override the caution that security would dictate. So it is necessary that there be a natural boundary around the premises network. Corporations create such a boundary with virtual private networks. At the end of this chapter we shall say more about that approach.

The manner in which host services and personal communications are implemented on Interlan make it unnecessary to compromise the privacy of a local area network even when offering a public service. When a host on a premises network advertises a network service on Interlan, the name of that service is defined in the Interlan name service and the server in effect connects to that name. A client of that service is also expected to connect to that service name rather than connect directly to the server. (See (ref) for more details.) Thus it is not necessary for the client to know where the server is or how to reach its location. In this way Interlan provides an opportunity to make the premises network and the hosts on that network invisible to all Internet users, even though some hosts on those premises may offer a publicly accessible service.

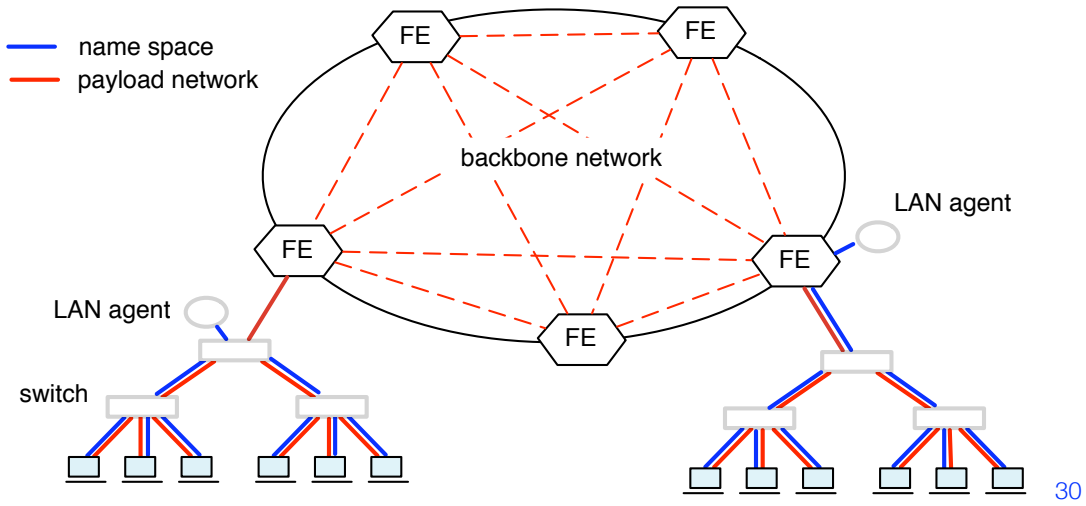


Figure 3.7. Local name spaces in a global network

All connections on Interlan start out with the name of a resource, service or network. If a resource does not have a name then it cannot be reached. Therefore, if the name space available within a LAN is not available to users in the wide area network, the LAN has good protection against direct attack. There are three ways of obtaining that result:

(b) The LAN may contain its own name server which serves only hosts within the LAN and the name server is invisible to the outside world. That is illustrated on the left-hand side of Figure 3.7.

(c) Hosts on the LAN can use a "local area name service" which is operated by an agent in the regional switching center. That service has the same properties as a name server dedicated to one LAN. Even though that name service operates in a regional center, the local names which it contains are available only on the user's LAN. That is illustrated on the right-hand side of figure 3.7.

(d) For LANs that are not too large, a zero configuration protocol such as Apple's Bonjour (ref) can be used. A client uses LAN multicast to search for a host with a particular name and the host responds with its address. This method can also be used to search for a resource the offers a specific type of service.

Whichever of these methods is used, hosts and other resources which are available to users on a LAN are not available for users in the wide area network.

3.12 Making connections

An Interlan flow is comparable to a TCP session with the added benefit that the flow is recognized and protected by the network infrastructure. An Internet client process uses the name system to obtain the address of a server and an end-to-end handshake with the server takes place before payload packets start to flow. All of this takes place in the Internet's payload network where other users can launch an attack. An Interlan network on the other hand, is structured more like a host operating system. The handshake is conducted within the protected space of the operating system and the content of each flow is as isolated from all other flows as are the contents of open files in a host operating system. An attack by one application on another's communications is therefore much more difficult to make. The network operating system knows which application processes participated in creating a flow and makes sure that other application processes do not get involved. Given that it is network operating system code that assembled the structures that protect the flow, it is less likely that the flow is not what it appears to be. For a network that cannot trust its hosts to behave themselves, this difference in protocol is profound. The network operating system establishes rules of good behavior and insists that they be followed. Those rules include identifying each packet with the flow in which it is traveling, and preventing others from transmitting packets in a flow which they did not create. The network operating system also provides an opportunity for client and server to authenticate the identity of each other.

An Interlan application needs a versatile interface with the network operating system. Traditional host operating systems have successfully used a set of procedure calls, known as “system calls” for that purpose. Interlan networks do essentially the same thing. However, in spite of the versatility of that interface it is not what Internet applications are programmed to do. Therefore, the Interlan operating system supports a second style of interface which is compatible with today’s users of IP network applications. (See (ref list)). Flows created by either of these two means operate in the same protected environment and are compatible with one another.

The native Interlan interface is a control flow on which “syscalls” are transmitted to the network operating system. A syscall is remote procedure call. Syscall messages pass back and forth between an application and its agent by way of the control flow.

The IP interface establishes an Interlan flow whenever a TCP or a UDP unicast connection is launched using current day operating system functions. A flow created in this way is compatible with a flow launched by using syscalls. However this remark only applies to point-to-point UDP applications. For security, multipoint UDP applications are required to operate within the framework of a virtual network. This report first describes Interlan networks in terms of syscalls, Then the IP interface is described in section (ref). Virtual networks are described in (ref).

The network operating system coordinates the availability and use of networked resources. Operating system agents carry out this task. The word “resource” has a special meaning here. It refers to a named entity that is managed by the operating system. The most common resource type is a “service” which, for example, might be a web site. A process which uses this service is a “client”, and the process which responds to the client’s requests for web pages is a “server”. The service, identified by the resource name, is an abstract entity, whereas the server and client are processes that do not have names. Agents of the network operating system coordinate the linkage between server, service and client, they create a flow between client and server, and they track the locations of these two processes if they are mobile.

A control flow, is a secure (encrypted) flow which connects an Interlan host to its network agent. Control flow establishment takes place when a host first connects to the network. (See (ref) for more on that topic). While it is convenient to say that a host process exchanges syscall messages with its agent, the more accurate statement is that a host process interacts with the host operating system, and it is the host operating system that sends the requisite syscalls to the network operating system. By this means it is intended that there should be a relationship of trust between a host operating system and the network agent that receives the host’s syscall messages. That trust allows the network operating system to respect the privacy of individual network users. For example, syscall messages are used to define the name of a new service and to associate access controls with that service. Ownership of the service (by default) goes to the user of the process which created it. The network operating system relies upon the host operating system to make that association.

The control flow is private, and for security it is encrypted. No other host has access to that flow. That is in contrast to the Internet where connection setup protocols have no security protection, the control information travels in the same channels as every other message type, and hackers have discovered that they can attack by inserting rogue packets into the conversations of unwitting hosts. Once an Interlan flow has been established, the forwarding engines which implement the flow prevent one flow from attacking or listening to another flow. Encryption of the control flow gives extra protection, particularly when a laptop is unplugged from a wall socket without first closing its network connections.

The following is an informal description of the procedure for service creation and client connection. The procedure is expressed here in terms of system calls that are representative of those used by an application process. These system calls reflect the language of the interface between an application program and the host operating system but have been simplified to avoid distracting detail.

First the owner of a service defines the service name.

```
create(SERVICE, “//weather.com”);
```

Then the server announces that it is ready to receive clients.

```
fr = listen(“//weather.com”);
```


In due course a client requests access to the service.

```
fc = connect("//weather.com");
```

The server receives a copy of the connection request message on fr, and agrees to accept the connection.

```
getmesg(fr);  fs = accept(fr);
```

Finally, the client sends the first payload data to the server.

```
write(fc, data);
```

The effect of this call sequence is to enable a 2-way **flow** of information in the payload network between client and server. The variables fr, fs and fc are flow handles which are functionally similar to Unix file handles. Information transfer occurs when client and server do read( ) and write( ) using their respective flow handles, fc and fs. The flow comes to an end when either fc or fs is closed. The server can stop listening by closing fr.

The similarity between the above system calls and those associated with the Unix socket system is intentional so that existing Internet hosts might easily transition to Interlan. See (ref). There are also some advantages over the way that connections are made today.

It is the service which is named, not the host. That is in contrast to the Internet where port numbers are a (semi-)managed global vocabulary of service identifiers. For a network that is intended to serve billions of people over many lifetimes, the use of numeric identifiers is surely a problem. Port numbers do not make a good name space, and if they were such they would need both more structure and more administration. Service names remove the need for all of that by making better use of the name space.

By giving each service its own name there is greater flexibility in the way that the service can be provided. Different computers may host the service at different points in time, and the client program will not need to know about it. When one server substitutes for another, the name of the service does not change. A server can be mobile, moving from one location to another while serving a single client. Multiple servers may concurrently provide the service to many customers, in which case successive clients may obtain connections to different servers. Each client can use the server that is nearest to him. Servers may be positioned in different parts of the network so that clients can avoid making long-distance connections. In this way the load is distributed and the manner in which this is done can be specified or managed by the service owner. When multiple servers concurrently offer the same service that service continues to be available when one of the servers fails.

Service names also bring greater security for networked hosts. Only hosts that have chosen to listen( ) for clients of a resource are exposed to other users of the network, and that exposure lasts only for as long as the host is listening. Once the listen( ) flow handle has been closed, the host is no longer visible on the network and cannot then be attacked.

Hosts that do not contain servers and therefore have no need to do listen( ), are not exposed to direct attack. These hosts are invisible to all other hosts on the network. It is common for an Internet hacker to scan the network address space for vulnerable hosts. For each address the hacker scans through the port numbers for one that is not well defended. Address scanning is not possible on an Interlan network because host addresses are not available for use when making connections; the connect( ) syscall accepts only names. Scanning through all service names is not so easy because there is no readily available list.

3.13 Establishing a flow

The sequence of events which link a client to a service are summarized in Figure 3.8. Arrows in that figure correspond to messages transmitted. (To avoid clutter, Figure 3.8 does not show arrows for acknowledgement messages.) The lines are labelled with the type of message and they are numbered according the sequence in which the messages are transmitted. The arrows are colored according the the type of channel over which they are sent. It may be helpful for the reader to refer to the figures in Section 3.1. Syscall messages correspond to the system calls described in Section 3.3 above. Signaling

messages, which agents pass between one another by way of the control network have similar roles to these syscalls and are named accordingly, however in reality the agents pack more information into each signaling message than is shown here. The dashed blue line in Figure 3.8 represents the fact that the name service translates the name of the service into a reference to a data record within the service agent.

A flow cements a relationship in addition to carrying data between one host process and another. In the case of a server, the flow which is created when the server does listen( ) creates a bond between server process and service agent. For as long as that flow exists the service agent knows that this server is ready to receive clients. The service agent knows immediately if the server dies or its connection to the network is broken.

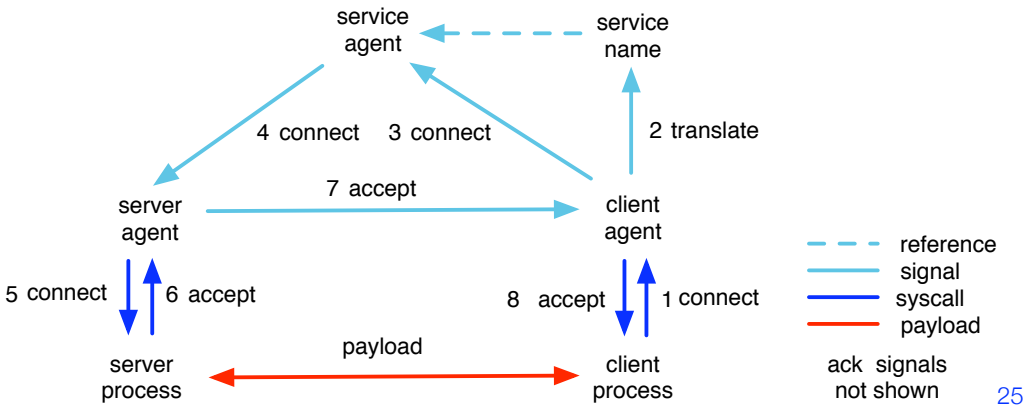


Figure 3.8. Message flow when connecting to a service

A client sends a syscall, connect(<service name>) [step 1], to its agent when requesting connection to a service. That message specifies the service by name and, for compatibility with Internet services, may also give a service port number. The first action by the client's agent is to obtain a translation of the name from the name service [step 2]. Successful translation yields one or more resource labels each of which contains the address of a service agent that keeps a resource record for the named service. A nearby service agent is chosen, or the Internet's statistical approach to wide area traffic distribution can be used. The service resource label is added to the connect( ) message and connect( ) is forwarded to that agent [step 3].

In a given region, the several agents for a service maintain a distributed but coordinated system for spreading the client load across the available servers. In the particular case that there is just one server for a service, the service agent and the server agent are the same thing. Once a server has been chosen the connect( ) message is forwarded to that server's agent [step 4].

The server agent now prepares to create a flow between server and client processes, and the connection request is forwarded to the server [step 5]. At this point the server may complete the translation of the name text contained in the connect( ) message. That may give the server more information about exactly what the client requires, and that may determine whether the request is to be accepted. Acceptance may also be determined from information about the client, provided in the connect( ) message by the client or his agent. If the request is not acceptable it may be rejected, or the request may be redirected to another server. Otherwise, the connection is accepted and a message to that effect is sent to the server's agent [step 6].

Up to this point the focus of activity has been on deciding which server, if any, will accept the connection request. The client and server agents will have made small initial steps preparing for connection establishment, but it is only now, when acceptance has been received and the location of the two ends of the connection are known, that a route for the flow will be selected and the forwarding engines which implement the flow will be configured as necessary. Thus, the route of the flow is as direct as it can be between the client and server processes. To the extent that route selection should rely upon information from the client and server hosts or their agents, that information is now available. These actions take place as the accept( ) message travels from server agent to the client [steps 7 and 8].

In spite of the fact that the route has been chosen and the forwarding engines have been instructed, there is one final decision to be made. The client's agent and the client process have the opportunity to confirm that the selected server is the one they expected to be working with. Either of them may have requested information from the server which assists in making that decision. Also, in case the flow should be rejected by the client, the forwarding engines have been instructed during step 8 to fully implement only one half of the full duplex flow, that is the half which carries payload data from client to server. The flow will be fully instantiated by the forwarding engines as the client's first payload packet travels towards the server. Until then, the server cannot use the flow.

When the client or the server is finished with the connection it sends a disconnect( ) syscall which travels through the control network, causing the connection to be dismantled. The server can also close the listen flow with the effect that the service no longer considers the server to be a candidate to serve new clients. However, closing the listen flow does not disconnect existing clients of the server.

The connection establishment protocol described above takes one round-trip from client to server and back before the first payload packet can be sent. The first payload packet, in effect, confirms that the flow is properly in place. This three-way handshake between client and server is sufficient to initialize the transport protocol. To the extent that flow establishment time is dominated by propagation delay between the two hosts, this flow establishment protocol takes no longer than would otherwise have been required by the SYN handshake which is the start of every TCP connection.

Rather than connecting a client directly to a server before negotiating the terms under which service is to be provided, Interlan defines an independent data structure, the service record, which is located within the service agent. This record is the logical interface between client and server. Being within the network operating system and controlled by the special purpose language of syscalls, this interface is harder to attack than are the ports on today's host computers. The result is greater security for Interlan servers. The service record interface also simplifies support for mobile servers because the service record is static, which leaves the service agent to carry the burden of tracking mobile servers when they move away from home.

3.14 IP connections

From an application perspective, an IP connection between an Interlan client and an Interlan server is the same as it is today on the Internet. The client sends a first packet that contains the server's IP address and service port number, and the server responds with the second packet which contains the port numbers to be used for the connection. When the second packet reaches the client the connection protocol is complete. IP payload packets transmitted between client and server have the same content as they had when the connection was made on the Internet. The one significant difference is that an Interlan network transports those payload packets within an Interlan flow. The flow numbers used are equal to the port numbers negotiated during the IP connection setup, and the packet length is as specified in the IP header.

From an Interlan operating system perspective, the first and second IP packets used for connection setup are alternative forms of the Interlan connect( ) and accept( ) syscalls. These packets are identified by the client and server forwarding engines respectively and are handed over to the local agents. There are several ways of identifying these packets depending upon circumstances; the most obvious is when the host uses an Ethernet header that does not contain an Interlan flow extension.

The client's agent requires that there be a resource record for every IP host operating as a server within the Interlan network, and there needs to be a registered name for that resource. The name is usually the host's DNS name, or it could be the host's IP address. If the DNS name is used, then the client must obtain the IP address from DNS prior to requesting connection to the resource (which of course is usual practice but it is not always followed.) The name service request is caught by the forwarding engine on behalf of the agent. The agent observes DNS requests made by a host and records the IP address and resource label from the name service response.

3.15 Traffic monitor

A traffic monitor is an application process that augments an agent's implementation of syscalls related to connection establishment. For a private network, the monitor reviews any attempt to create a flow that crosses the network boundary.

For a service, the monitor reviews each connection request prior to the service agent assigning a server to the task. Any private network or service can have a traffic monitor. It is installed by the resource owner doing

create(MONITOR, <network name>)

A service with the name ".monitor" is thereby installed in the private network or service directory. A server that listens on .monitor will receive copies of connect( ) and listen( ) syscalls and for each it is expected to respond with accept( ), reject( ) or redirect( ). Accept( ) allows the syscall to proceed, reject( ) blocks the request, and redirect( ) sends the request to a different place. In this way the monitor of a service can influence the way in which the load of incoming service requests is distributed across the available servers, and requests from unacceptable clients can be blocked. A monitor of a private network has the same control over connection requests that are entering or leaving that network. A connection request is said to be entering or leaving if one of the resource names involved is within the private network directory and the name of the other resource is external to that directory. In the case of requests made to or from a mobile device, it is the location of the wireless base station which indicates how the request will be classified.

Traffic monitoring provides flexibility in the way that services and private networks are controlled and protected. It can decide which server will handle a particular connection request, or which connection requests will be rejected and which will be accepted. Statistics collection is another possibility. The monitor process operates in the payload network, so it has access to application-specific data which it can use in its decision making. In general, monitoring should be dedicated to matters of policy and work-load coordination which can be handled expeditiously.

Each monitor server listens on a .monitor service. Multiple servers may support the .monitor for a heavily used service or network in order to avoid delay during connection setup. The monitor servers can communicate with one another as they coordinate the distribution of traffic and counteract temporary imbalances in traffic load.

Figure 3.9 illustrates the way in which a monitor process fits into the message flow for service requests. Let M be the service monitor for a service S. For each service request, M obtains a copy of the connect(S) syscall. M is not able to complete a connection to communicate with the client that launched the request, bur rather M provides guidance on how a request shall be handled. M may coordinate the assignment of clients to servers according to subject matter mentioned in the connect( ) message, or M may be an observer or filter that weeds out inappropriate requests. Having examined a request, M responds with accept( ), redirect( ) or reject( ). In each case the response is handed back to the service agent. M may also provide the agent with text (length restricted) which is to be appended to the request message. The agent will append that text after writing M's identity before it. The next server to receive the connect( ) message can read what the monitor had to say about it. If the response is accept( ), the request is scheduled in the default manner as specified in .config. If the response is redirect( ) the request is transferred to the server specified by M. To avoid abuse, redirect( ) can only refer to a resource within S's service directory. A reject( ) response causes the request to be rejected with the reason as given by M.

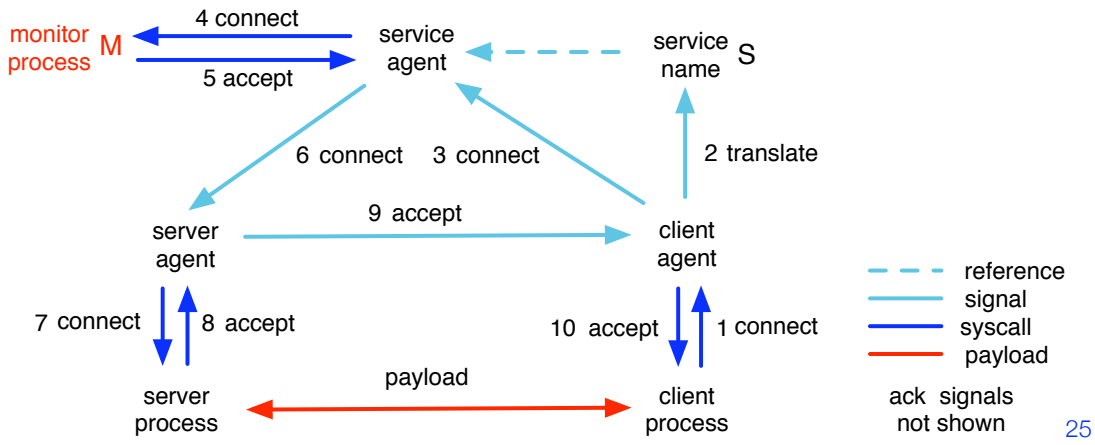


Figure 3.9. A monitor process intercepts connection requests for a service

M receives a copy of the connect( ) message and the agent retains the original. It is that original message that is forwarded according to the monitor's advice. In this way M can have only limited impact on the conduct of a service request. The service agent can share the load caused by the arrival of many service requests, and by using a sufficient number of monitors may be able to absorb the energy of a denial of service attack. For a large service with many service agents and servers, there can either be one monitor which coordinates the work load distribution among all servers, or a set of monitors that coordinate with each other for the same purpose. The terms under which the monitor operates are the same as for a server, so the monitor mechanism does not expose an Interlan network or service to an unusual security threat.

3.16 Mobility

The challenge of mobility is to maintain the perception of continuous service to a device that is moving from one radio cell to another. The concept implies a network that is aware of ongoing conversations so that it can coordinate route change with each move. That is one reason why flows are the fundamental component of Interlan service. It is also the reason why Interlan agents keep a record of the dynamic state for each active flow, and why Interlan name service refers not to the mobile device but to an agent which is tracking that device. This design is standard for all flows of all hosts on an Interlan network. While the design carries a little more overhead than would be required to support a static device, standardization on the one service model simplifies the overall system design and allows for common Internet practices such as moving a laptop computer from one place to another without closing the active connections.

Mobility is a property that is usually associated with a host. The agent which supports an Interlan service is always static even when the host which provides that service is mobile. That makes the service easy to find, and it is left to the service agent to keep track of the server. When a wired network is installed in a vehicle, it is the network that is regarded as being mobile and the agent tracks the location of that network's gateway.

"Home" for a mobile host is the region where it is normally located. An agent in that region is appointed to act as "home agent" for all resources implemented in that host. When a host is not at its home location it is managed by a "local agent" which is in or near the region that the host is visiting. The local agent keeps the current state of the host, the services which it contains, and the active flows which are involved. The local agent reports significant changes in state back to the home agent. In the case of a host that expects to be a long distance from home for an extended period, there may be a "foreign agent" which acts in place of the home agent. In this report, when discussing a mobile host, the term "local agent" is used in reference to whatever agent (local, home or foreign) is currently monitoring the host's state. Resources which are linked to a mobile host are included in the home agent's record.

When a host moves from one region to another, the network's first reaction is to maintain connectivity for all active flows. In particular the host's control flow will continue to be routed to the same agent that it was using before the move. That agent continues to be the host's local agent. To reach its agent the host's control flow now passes through an aggregation network in the new region and then is routed over a trunk to the region from which the host came. In principle this procedure can be repeated until the host has moved a substantial distance away from its current local agent. At some point, concerns about efficiency will cause the current local agent to transfer its end of the control flow to another agent which is closer to the host. That other agent becomes the host's new local agent. The previous local agent drops out of the picture unless it is the host's home agent. This procedure is known as "flow-layer mobility". See (ref) for a more detailed description of the mechanism.

The home agent has responsibility for tracking a host's movements. Each time the host obtains a new "local agent", that agent makes contact with the host's home agent and keeps the home agent aware of the host's status. There are circumstances when a host moves far from its home agent and stays there for an extended time, long enough that the overhead of the host's relationship with its home agent becomes a concern. In that case a foreign agent can be established by administrative request. A foreign agent has almost the same status as the home agent; it takes over responsibility for tracking and authenticating the host while it is in the foreign location.

With each host having multiple users all of which must be authenticated with an agent, one might be concerned that there will be a lot of busy work when a host moves from one agent to another. To avoid the delay that might arise, the Interlan operating system provides the host with a secret number before mobility occurs. Then, after the host has moved, when a

new control flow is established, the agent tests to make sure that the host still remembers the secret. That being the case, no further authentication is required.

Process mobility, migration from one host to another, is implemented as the mobility of a virtual machine. A virtual machine is not hardware, it is an operating system which emulates some features of a host. Those features include a separate network connection for each virtual machine. It is a virtual machine's network connection that carries the control flow which connects the virtual machine to an agent. Thus a host with several virtual machines within it has multiple control flows. Should one virtual machine, and the application processes that it contains, migrate to another host, the virtual machine's network interface also migrates, the control flow must be re-established, and active flows will be rerouted. (See (ref) for details of mobility.)

3.17 Host registration

Registration takes place each time that a host comes online to the network; login occurs when a user comes online and is probably a side-effect of the user logging into the host operating system. With these two procedures host and user identify themselves in preparation for the coming network session.

Registration is the means by which a host makes itself known and authenticates its identity to a network agent. The primary forms of host identification are an Ethernet address and a network account number. For their own security, hosts are advised to advertise their names on local area networks only, NOT in the wide are network. When acting as a server on a wide area network, the safe strategy for a host is to listen( ) on the domain name of any service that the host supports. Clients are expected to know the name of the service, not the name of the host(s) that implements that service. Clients do not need names for wide area communication and host name is not required in order to register. In that way hosts have the least possible exposure to attack from hackers operating in the wide area network. Traffic monitors and agents have primary responsibility for assigning clients to servers. They minimize service completion time, they balance the load, and they take responsibility for equipment failures.

Moore's law and the architectural changes which continue to take place are making it difficult in some circumstances to answer the question: what is a host? A computer cluster may be regarded as one host or many depending upon how it is configured and used. Cloud computing with virtual machine technology further complicates matters. Is the cloud a host, is each processor a host, or does virtual machine technology mean that the hosts are more numerous than that? Perhaps there is one virtual host per network interface. The Interlan perspective on this question is that a host is defined by an operating system. When an operating system registers it does so through a network interface that no other operating system is using. In that case a host consists of that operating system, its network interface and whatever computing resources and network interfaces the operating system claims as its jurisdiction. By this definition no two hosts share a network interface and, at any moment, every host process comes within the jurisdiction of just one operating system. How that is achieved in practice is not the network's province to know.

The host interface through which an operating system registers its presence on the network is the interface through which the host's primary control flow passes. If the host has other network interfaces, an operating system should register each one of them so that each has its own control flow. In this way a host can have multiple concurrently active interfaces to one or more networks. However, there can be no more than one control flow for each interface. Each application flow is assigned to the network interface which carried the syscall message that created the flow. The flow will continue to use that interface until host or process mobility causes the flow to pass through a different network interface.

Some mobile devices have two network interfaces, each one potentially communicating with a different base station. Ordinarily, one is the active interface through which active flows pass, while the other is looking for an alternate base station that may be needed when the active interface begins to lose signal. In that circumstance at the critical moment each flow hops from one interface to the other. From a higher layer perspective this interface pair constitutes one network interface.

Hosts are presumed to be long-lived. Each has an account number that represents the host's relationship with the Interlan network service that the host is presently using. The relationship between host and network service provider is established during registration. The host account number, which must be globally unique, is used on each subsequent occasion when

the host registers. Also at that time, a "home agent" is assigned from among the agents located within the host's regional switching center. The home agent is the host's representative within the Interlan network. It will persist in that role until such time that the host moves permanently to another region. There will be one home agent per host regardless of how many network interfaces the host has, and regardless of how many service provider networks the host visits. That home agent coordinates information about where the host is presently located and how it might be reached.

During the first registration the host sets up the procedure that will be used in each subsequent registration. That procedure must reliably establish the host's account number. Ideally, the host's account number and Ethernet address together with a secret should be stored in a trust module that is permanently attached to the host. Subsequent registrations involve a temporary agent which authenticates the host then hands over to the home agent or an appropriate local agent. The host may also establish a reciprocal relationship with the network, so that during registration the host can confirm that it is connected to the correct network.

After registration there is a periodic handshake between host and agent which allows each re-authenticate the other. Also at that time, they compare their respective views of the current state of their relationship. For example, this is an opportunity to make sure that they agree on which flows exist and which flow numbers are available for re-use. Robust recycling of flow numbers is fundamental to reliable service.

"Resumption" is an abbreviated procedure that expedites registration without loss of security after a brief break in the network connection. Short interruptions are a matter of course for mobile hosts, but in current Internet practice they also occur quite frequently when a laptop is relocated without terminating the flows that are at that moment in progress.

There is a dimension to this situation which will be new to Internet users. The Interlan operating system administers access controls and related checks which limit active flows to those that conform to a user-specified policy. Host mobility can change a prior relationship so that a flow which conformed to policy no long does. For example, a laptop might be moved from a public network to one that is private. In that case, the resumption protocol must consider whether the currently active flow is permitted by the access control rules in the new environment. (See (ref)).

3.18 User login

The Interlan access control system depends substantially upon user identity, which means that user identity plays a prominent role in an Interlan network. It is possible for a user to associate his or her identity with a host; in effect using the host to authenticate the user. In that case one must consider the possibility that the device may be stolen and its loss may undermine personal security. More usually, we expect, a network user will login to the network at the same time as he or she logs into a host. In either case, login should support authenticated identification of the user and the protocol should be as secure and convenient as possible.

A growing repertoire of technologies is available to support mechanized identity checks. SIM cards that contain the identities of mobile telephones, and bank cards with little more than a magnetically recorded stripe on the back are giving way to applications of low cost tamper-resistant processors and the cost-effective deployment of public key encryption. It is anticipated that within a few years processors capable of public key encryption will be available at a price that allows their use in bank cards, car keys and laptop computers. Therefore, there is a distinct possibility that every host and network gateway will one day use this technology in a security module which implements a challenge-response protocol. In this way technology can reduce the risk of identity theft. We shall assume that that technology is on the same time scale as any deployment of an Interlan network.

There is widespread concern that a strong form of identity for individuals will lead eventually to tyranny by a government that has assembled a dossier on every individual. On the other hand identity theft is a wide-spread and growing problem. Bank card numbers are stolen in large quantity from local area networks or on portable media. These thefts lead to personal bank accounts that are drained of their contents and, for the consumers involved, a long and painful period of recovery. In many cases the theft can be attributed to careless behavior by bank card users and retailers, by the owners of personal computers, and by the irresponsible design of business networks. It is not clear to what extent these are curable problems and to what extent they reflect the natural weaknesses of mankind.

A way forward seems to lie in a system which allows multiple albeit narrow-purpose ways of identifying an individual. When robust identification of that individual is required it may be necessary to assemble information relating to more than one name. With that in mind, the Interlan architecture supports more than one identity for each user, and each identity may have its own means of authentication. The goal for authentication is that it should be obtainable without releasing the underlying facts which prove the identity. Trust modules with public key encryption offer that prospect. Location information may be another helpful factor. Caller-ID for the United States telephone system allows a user to know who is calling before he takes the call. For some purposes that has been accepted as providing adequate support, particularly when it is used with another form of identification. A user can disable this feature for the calls that he makes, in which case the call recipient will know that the caller has refused to be identified. Interlan networks could use a similar technique based upon the location information that the agents possess.

3.19 User identity

The challenge is to administer user identity while avoiding a heavy bureaucracy. This challenge was nicely met for the global email network by building upon the administrative framework created for the Domain Name System. Any lessee of a DNS resource name can today administer email addresses based upon that name. The method decouples administration of the name space from the identity of the email service provider. It is suggested that this method be used for Interlan user identities. Therefore, each identity is of the form <user name>@<community name> where the community name is a domain name which may match the name of a resource. A person's email address can be one of his network identities.

It is usual for the community name to match the name of a corporate network or service. An employer such as "xyz corp" may contract with an email service, such as "yahoo", and employees will have the option of using <employee name>@xyz.com as one of their email addresses. In this case, "xyz.com" is both the domain name of the company's internal network and it is the community name for the company's employees. Two uses for the one name "xyz.com" is no problem for the name service because it associates a different service name with each translation.

Some people may be concerned that email addresses are widely circulated; they are published on web pages and they are used by many retailers and service providers as a form of client identity. Today, a person's email address is no secret and attacks launched by way of email are common. While that is a real concern it should be possible to provide enough capacity in the naming system to give users the flexibility where necessary to use different names for different purposes.

Looking forward, it seems likely that the two popular and successful naming systems: resource names based upon DNS and user identities in the style of email addresses are likely to become universal in their respective spheres of application: resource names for computer-communication systems and user identities for person-to-person communication.

3.20 Authentication

When an agent possesses secret information which allows robust authentication of a client's identity, the client may allow the agent to use that information when a server is seeking confirmation of the client's identity. The server may trust the agent sufficiently to carry out that authentication without revealing the secret information to the server. The following are examples of information that an agent may acquire for this purpose:

Host ID	This is information which the network gathers when a client's host is registered.
Token	The token is a small security device which contains hardware support for public key encryption and for the storage of a secret. This device can be plugged into a computer or cell phone, and then participates in a challenge-response protocol that is administered by an agent.
Certificate	A document which uses public key cryptography to link the name or identity of an individual with the public key that he or she uses.
Location	An Interlan network operator can be expected to have good information about the locations of the hosts that it is serving.
User identity	The identity provided and authenticated when the user logged into the network.

An agent's role as arbiter in an authentication process may vary depending upon the size and nature of the server's business. In some cases there will be an opportunity for something like a monitor where the agent can interact with a customer database or authentication service. This could be helpful for services that have large numbers of customers. It would allow the agent to keep the client's secrets while obtaining information from the server that allows a server-specified access control to be implemented.

3.21 Access control for services

One can recognize two broad categories of computer network application. The focus of this report so far has been on **users** that employ the network to reach **services**. A popular service may have thousands, even millions, of users most of whom are no threat to the service provider. However, there will be some who have bad intentions. The service provider is therefore knowingly taking some risk with every user that it serves, and that risk must be managed. The second category is focused on specific **applications** that are implemented by **systems** composed of computers and communication networks. Examples are air traffic control systems, electricity delivery networks and the Federal Reserve check clearing system. Commonly, the immediate users of these systems are employees of the system operator. The risk is not so much from those employees as it is from those who would attack the system from outside with intent to inflict hardship on the people that the application is constructed to serve, or to steal assets which the system is designed to protect.

Access controls therefore are of two different types. For the protection of services, access control means authenticating users, selecting those that are to be served, and restricting what they do. For the protection of application systems it is usual to focus on the system perimeter by constructing a private network. We begin with a description of controls directed towards users and the services which they use.

Many of the attacks which plague today's Internet are pitched at a higher level than the infrastructure for making connections. Hackers install daemons that imitate well intentioned users, viruses are found in national infrastructures, criminals bent on credit card fraud steal card numbers from local area networks, and extortion is a flourishing business for those who threaten to overload legitimate network services. There are other widespread concerns that are of a non-criminal nature - email users suffer a plague of unwelcome messages, and parents worry about children being exposed to pornography. To combat these situations network users need additional tools and services that help them distinguish between friend and foe, help them erect barriers against intrusion, and watch over the activities of their children. It is under the broad titles of "access control" and "virtual networks" that we address these matters.

Only the providers of a service know how to distinguish between legitimate and suspect client behavior. And it is only the parents who know what standard they want to set for their children. Therefore it is evident that these groups of network users need to be involved in the design of their own protection. The network can help by providing tools which are programmed in a language that users can understand. Interlan access controls are tools of this type. They are focused on circumstances which arise when a service is being used, and they leverage the client-server paradigm for creating a network connection. The goal is to identify well-intentioned clients and to constrain those who cannot be trusted. For the client's benefit it is necessary to distinguish between genuine offers of service and fraud that would trap an unwary client.

Access controls for files in a computer operating system have existed almost from the earliest days of those systems. The file mode in the pioneering operating system CTSS (ref) indicated which operations could be carried out on a file, and Unix to this day employs an access control list per file. These lists are implemented as bit-maps within the file system so that they might be quickly checked, and they are represented to the user in a language that is easier to use but no more flexible. However, this framework lacks generality and so has not evolved to become the versatile user interface that we anticipate is necessary for access control in a network operating system. Today, processor speed is many orders of magnitude greater than it was for the pioneering host operating systems and compiler technology allows language that is not so closely linked to the compiled form of an access control list.

Access control rules proposed for Interlan resource access control are expressed in policy documents that are attached to service and network directories. There are policy statements which apply to the different resource types. For example, these statements specify who can list the contents of a directory, and who can search for a name therein. Inability to search means that the user is not permitted to access any of the resources named in that directory. There is language that restricts

the network sites that a child might visit and other language which describes how a connection should be routed. Some of this language is described here, but in general it is expected that the scope and nature of the policy language will have to evolve. Therefore, a general framework for policy expression and implementation is described here. Users with appropriate privilege can edit policy statements contained in policy documents using standard browser software. Policy documents of different types have different names. The document which specifies access control is called ".users". Such a document can be installed in a resource directory by means of the syscall create(POLICY, < resource name >).

The policy-based security architecture described here is intended to be an open-ended framework for establishing confidence among network users and service operators. They need to have confidence that the network will defend their interests and implement the service options that they prefer. The .users documents are the means by which this is done. Each resource may have its own policy document which is where policy relating to that resource is written. By default it is the owner of a resource that controls access to the resource's policy document. That control can be delegated.

3.22 Policy description language

Policy statements are expressed in the Policy Description Language (PDL). These statements are edited by a special-purpose editor that expects to interact with a web browser. The editor is executed by clicking on the policy document's name. When the document is opened it displays the currently active policy. When editing is complete, the document is compiled into a form that is efficient for interpretation by the network operating system. If no errors have been detected, the compiled policy takes effect when the editor is closed.

A policy document describes the terms under which access to a particular resource is permitted. The document does not introduce new users into an Interlan network nor do they define new resources. User communities, services and networks all have records that are created by the create( ) syscall. These calls are contained in network administration software. Policy documents do not describe network topology. That is discovered by the Interlan supervision system. That system constantly monitors the infrastructure, detects its configuration and identifies each component. Policy documents refer directly to the names of services, networks, communities, users and network components which have been declared or discovered in this way.

Policy documents describe access controls and so are concerned with the authentication of users and systems. However, there are no secrets encoded in these policy documents. Secrets such as passwords and public keys are specified when a user logs into an Interlan network. These secret are stored in the user's account.

A policy document may describe additional authentication requirements that the owner of a particular network resource requires of its users. Those requirements may call for additional credentials that the client must make available. The server does not have to see the client's credentials in order to get the benefit of them. The client may provides thos credentials to his agent and may ask that the agents handle authentication on behalf of the server. In addition to privacy for the client, agent mediated authentication of client and server authentication can improve the speed and safety of a client/server interaction. When the operating system knows what information the server needs for client authentication, that client's agent can anticipate the need for an access control check and can forward the required information through the control network even before the server has asked for it. The quality and speed of authentication may also be improved simply because the client and server have already been authenticated by the network.

Access control policies apply in the following situations.

Admission	While every user is required to log into an Interlan network each time that he or she comes online, a virtual network owner may specify in policy rules additional requirements that must be met before that user can admitted into the virtual network.
Modes of use	When a user accesses a resource his actions may be restricted by policy rules associated with that resource. For example, a resource may be readable by everyone while writing is restricted. Access controls can implement that restriction if the user declares his intended mode of use in the text of the connect( ) syscall.

Perimeter control — Common carrier services have for many years supported virtual private networks (VPN) which encapsulate within a defined perimeter a part of the physical network that is of special interest to a single (corporate) customer. Traditionally, perimeter control regulates network access based upon a set of rules administered in gateway routers located on the perimeter of that network. Perimeter control for Interlan private networks can be implemented with policy rules.

Route control — Wide area networks that comprise the global network have diverse characteristics with non-uniform performance and cost characteristics. When making a connection request, a network user may have a direct interest in how that connection is routed, including the transit networks that are used, the service quality that is provided, and the cost of the connection.

Network and user can work together when the network knows enough to understand a user's needs. For example, based upon a policy document provided by a server, network agents can filter service requests and distribute them among a network of servers. When the load is particularly heavy, access control rules can be distributed to regional centers where there is the capacity to handle heavy loads. Filtering can then take place in the neighborhood of the client rather than be allowed to saturate facilities in the neighborhood of the server. Denial of service attacks, the phenomena of large crowds, and natural disasters can all benefit from this technique which has long been used in the telephone network.

Each resource that a user visits can potentially introduce its own requirement for user authentication. Therefore, one might be concerned that rigorous checking adds substantial overhead to network operation and burdens network users with repeated proof of identity. However, these checks take place under agent supervision during connection establishment, and in most cases repeated authentication can be avoided when agents remember what authentication has been performed. A server can be advised when repeating an authentication procedure would be redundant. The persistence of agent memory, and confidence in agent integrity, can allow servers to obtain good protection without placing a heavy burden on the network user.

It is also important that the network operating system has a continuously available global database which allows it to offer services that help network users stay synchronized. For example, capabilities (which are physical tokens that unlock private resources) can be issued and subsequently revoked at any time of the day or night, from anywhere in the world where the network has a presence.

Thus, there is a big difference between a network that consists of just wires and switches, and one which responds to a global operating system with hardware protection for a service model that is not easily compromised. Interlan forwarding engines and agents are the core components of such a system. Policy documents are the means by which network users tune the system to their needs.

3.23 Policy document

The following is a simple policy document which relates to a resource named "//sports.com".

```
//sports.com/.users = {
Jim: allow;
Tom: allow;
}
```

This document lists two users whose identities are "Jim@sports.com" and "Tom@sports.com". By compiling this policy document these two users are permitted access to the //sports.com resource. In other words, Jim and Tom are "sports.com users". The following is a contents list for the //sports.com directory.

```
//sports.com =
.users;
```


soccer;
tennis;

This directory contains two services and the policy document. The presence of the policy document means that access to //sports.com is controlled; if there was no such document, access to //sports.com would be uncontrolled and the two service names would be visible to everyone. Each service may have its own policy document within which authorized users of that service are identified.

These constraints are implemented by agents as they execute syscalls which refer to //sports.com. If a constraint is not met the syscall will not take the requested action. Thus, the policy is implemented in agent software and the payload network does not have to handle any traffic that is inhibited as a matter of policy. When an unrecognized user, Harry, does connect("//sports.com/tennis") the connection request is blocked by the agent for //sports.com and Harry's connect( ) syscall fails. Had the same request been made by Jim or Tom, it would have gone through without difficulty.

When a new user joins an Interlan network for the first time, an administrator uses the syscall create(USER, <user ID>) and thereby assigns a new account number. The user may acquire a domain name for his own use, or he can request permission to use the domain name of a service provider. Suppose that Harry and Pat Potter, living in Connecticut have acquired the domain name

"//Potter.Berkshire.ct.us"

for their home network. The two parents become users of the resource "//Potter.Berkshire.ct.us" and so are entitled to use the identities "Harry@Potter.Berkshire.ct.us" and "Pat@Potter.Berkshire.ct.us". When Harry logs into his computer he also logs into the network using this user ID.

Rules of access which govern network admission for members and friends of the Potter family are contained in a document named //Potter.Berkshire.ct.us/**.**users The minimal requirement is that this **.**users document has the following content:

//Potter.Berkshire.ct.us/**.**users = {
Harry: allow admin;
}

This allows Harry to login with the identity "Harry@ Potter.Berkshire.ct.us" and, when logged in, the word "admin" says that he can edit the .users policy document. This policy statement as written does not require that Harry authenticate his user ID when logging in. It would be more secure if he were to use a password, or even better, he could use a hardware token within which there is a challenge-response module that uses public key encryption. In that case the .users document must contain the word "challenge", as follows:

//Potter.Berkshire.ct.us/**.**users = {
Harry: if (challenge) allow admin;
}

The first time that Harry logs in and uses the trust module he will be asked to provide its public key. The network operating system will record that key and on subsequent occasions when Harry logs in he will be required to respond correctly to the network's challenge. This he does this by plugging his trust module into the computer that he is using.

Other members of Harry's family will become users of "//Potter.Berkshire.ct.us" when Harry adds them to the **.**users document.

//Potter.Berkshire.ct.us/**.**users = {
Harry: if (challenge) allow admin;
Pat: if (password) allow member;
}

Pat provides her password when she first logs in.

Each person in the Potter family has a collection of personal communication devices. Pat, for example has a baby minder, Its resource name is "//Potter.Berkshire.ct.us/Sam". She is the only person allowed to use it. To make sure of that, Pat creates a **.**users document.

//Potter.Berkshire.ct.us/Sam/**.**users = {Pat: allow admin;}

The baby minder is online and implements a simple web server. "//Potter.Berkshire.ct.us/Sam" is the service name. As with all Interlan services, //Potter.Berkshire.ct.us/Sam has a directory and that is where Pat puts her **.**users document.

We now describe in a little more detail what happens when Pat uses her baby minder while away from home: Pat goes online through her phone. Then she uses her browser to check the baby minder. Her local agent obtains a translation for the name
"//Potter.Berkshire.ct.us/Sam". The first step is to retrieve the resource record to which the first segment of the resource name refers. That first segment is "//Potter.Berkshire.ct.us". The resource with that name has a **.**users document. The next step is to see if that policy document allows the current user to access that resource. In this case, Pat is a member of the network
"//Potter.Berkshire.ct.us", so she can access that resource after she has typed the appropriate password. The next name segment to be translated is "Sam". A resource of this name is found in the directory for "//Potter.Berkshire.ct.us", so the resource record for "Sam" is fetched. That resource has a **.**users document which contains the line: "Pat: admin, allow". Therefore the resource name translates successfully and Pat can use the baby minder for which the resource record is now in hand.

Group names have the same syntax as user names.
//HPotter.Berkshire.ct.us/**.**users = {
Harry: group = parents;
Pat: group = parents;
Brenda, Charlie: group = children;
}

To prevent the children from watching the online TV a parent can add the following to the policy document.
//HPotter.Berkshire.ct.us/TV/.users = {
children: deny;
}

This last example shows how access to a resource can be controlled by a policy document when the resource (the TV) and the users (the children) all fall under the same administration (the parents). However the situation is a little different when the resource is in a different part of the network. For example, suppose that (as had at one time been proposed) all Internet pornographic sites are in the domain "//xxx". Brenda and Charlie's parents will no doubt want to prevent the children from connecting to any web sites in that domain. The following statement can be added to the
"//HPotter.Berkshire.ct.us" policy document.
children: if (connect //xxx) deny;

Each of the rules contained in the above examples is a "statement" in the policy description language. The general form of a PDL statement has four parts:
<group> : <condition> <action> <duration>;

The group is a set of people to whom the policy applies. A group name represents the members of the group. So "children: deny" is equivalent to "Brenda, Charlie : deny".

The set of all resources within the domain "//xxx" is represented by the domain name, so the expression
if (connect //xxx)
is true when a connection is being made to a resource in the domain "//xxx". The statement
children: if (connect //xxx) deny;
unconditionally prevents the children from connecting to any resource in the domain //xxx. The condition can specify any of the syscalls: connect, listen, create, delete, list and search.

A PDL statement is an expression of policy with regard to some action that a user may take in connection with a particular resource. The action can take place when either the condition is true or no condition has been specified. When an action is

requested the network operating system looks for policy statements that specify a group that contains the user. Among those statements the operating system then looks a statement where the condition is true for the resource to which the user seeks access. If such a statement is found the user's request is granted. Discrimination on the details of the action (such as which syscalls can be used) is contained in the condition part of the PDL statement.

Besides "allow" and "deny", an action can assign an attribute to the subject of the statement. For example,

Pat : group = parent;

assigns the attribute "group = parent" to Pat. If the group includes multiple people each one acquires that attribute.

The list of attributes is open ended. Attribute values can be transported in syscall messages and in signals that pass from one agent to another. In that way a client can provide information to a server which may help the server decide to accept the connection, or the server can pass information to the client as a connection is completed. For example, we note that the //xxx domain was vetoed by the President of the United States, so another way of protecting children from inappropriate movie content must be found. One possibility is to assign a rating to every movie viewed on the Internet. When a person requests download of a movie the supplier is required to tell the recipient the rating value before the download starts. An Interlan network could carry that rating as an attribute in the accept( ) syscall which sets up the connection to be used for download. A parent then uses the following statement to protect his child from all but the most harmless movies.

children: if (rating != 'G') deny;

Some communities have a coherent membership in which certain users play roles that give them special responsibilities and privileges. The following roles are recognized with respect to communities of that type.

owner	The person on whose account the community name is recorded.
admin	A person who can edit a policy document.
user	Any person that can access resource available to a group or community.
member	A community member that has the community name in one of his identities.
visitor	A user that is not a member.
friend	Another type of visitor.

A policy document has three parts: the title, a set of definitions, and a set of policy statements. The title is the default resource to which a policy relates. Policy documents can become large and the inclusion of full domain names can make them difficult to read. So, to simplify the document, PDL allows simple names to be defined for use within the policy document. For example see Figure 3.10.

When there is no policy document for a resource the policy is "allow" all requests. If there is a policy document the default policy is to "deny" all requests except those approved by statements within the document. If either the current user is not contained in the group identified in a policy statement, or the condition in that statement is false, that statement is not applicable. The action takes effect if among the applicable statements there is at least one allow and no deny. During connection setup, policy documents for the client and service (not the server) are both applied and both must agree to the request.

```
//Hamley.Brookside.ct.us/.users = {
#define
home = //Hamley.Brookside.ct.us;
work = //Sports.com;
school = //St.John.Brookside.ct.us;
Jim = Jim@home;
Pat = Pat@home;
Brenda = Brenda@home;
Charlie = Charlie@home;
parents = Jim, Pat;
children = Brenda, Charlie;
family = parents, children;

#constraints
```



```
        home/PC {
        family: allow;
        Charlie: deny;
        user: if (junk) deny;
        user: if (connect //xxx) deny;
        }

        home/TV {
        family: allow;
        children: if (time > 9:00 pm) deny;
        children: if (rating !=  'G') deny;
        }

        home/phone {
        family, Pat@school: allow;
        }
}
```

Figure 3.10. Sample policy document

3.24 Virtual networks

The directory "//sports.com" contains three resources:

.users;

soccer;

tennis;

Soccer and tennis are the names of services and the policy document describes who can use these resources. The directory itself may have access controls which determine who can visit or even see the resources contained within.

We say that sports.com is a "virtual network", not because of any wires and switches that might be involved but because it has network functionality. Access controls associated with a network directory prevent unwanted traffic from entering the network while traffic can move freely within the network. The configuration of this network is described by the list of resources which are the directory's content. Networks can be nested to exhibit a hierarchical structure.

If the policy document allows user Jim to login to //sports.com as a member of that network, he will acquire the identity "Jim@//sports.com" and we will say that Jim is now "**in** the virtual network //sports.com". As a member he probably has general access to the resources of the network. Of course, Jim has not moved anywhere, but the concept of logging in and entering a virtual network helps one to develop on intuition for the power of virtual networking.

Today, a virtual private network (VPN) uses a gateway or firewall to restrict access to a local network. Network privacy is thus obtained, but at some cost. When the network is defined in the name space, policy rules achieve the same effect. A local agent applies the access controls specified in .users to every incoming connection request. If a request is rejected no connection will be made. Thus, there is no need to spend energy blocking a flow of unwanted payload packets because no such flow will be launched.

A network that is implemented in the name space has one big advantage - the resources which comprise the network are not constrained to have any particular physical relationship to one another. Indeed, the servers for //sports.com/soccer might be located in Europe and those for tennis might be in the United States. Privacy of the network //sports.com would be no different than if it had been enclosed within one room. Independence from physical location opens up many opportunities for Interlan users. Any group of people who have a common shared interest can create a private network to regulate who can see their data, who can join in their conversations, and who can use their services. Members of the group can be in one place or spread around the world; it might even be that they are mobile. The group can with little effort invite visitors or turn them away. No new equipment is required and there is no need to install new wires.

The syscall language which a client uses to connect to a service is another way in which the Interlan architecture supports private networking. Syscalls listen( ) and connect( ) allow server and client to be separated from the service resource which

they use when making a connection. A server in a private name space can listen( ) on a service in the global name space and thereby connect to clients that are outside the private network. Yet other users that are outside the private network cannot enter, or even see into, that private name space. Therefore this method of connecting to a server allows a LAN which is home to a server to have a local name space that is entirely separate from the wide area network. Local hosts are not disadvantaged by this arrangement because they can reach into the wide area network by using names that are fully qualified, i.e. begin with a domain name.

Separation of local and wide area name spaces allows LAN users great flexibility in what they do. They can adopt a more liberal access control policy for intra-LAN communication than would be prudent if exposed to the wide area network. The Internet today demands that its users have an impractically high level of discipline and expertise in order to stay out of trouble. Too many users either cannot remember passwords or for other reasons choose passwords that are easy to crack. Most people do not understand enough about host software to know whether what they are doing is safe. Therefore it will be a big step forward when every LAN is a private network so that the names used at home and in the office are unknown on the wide area network.

Naming systems for local use are already available, (zeroconf for example). A naming system of that type should not be linked to the global name space, and the LAN should be defined in the global name system as a private virtual network with no access from the global network.

The syscall used to create an Interlan virtual network is create(NETWORK, < network name >). This defines a network name and installs an Interlan data structure for the network resource. A **.**users policy document can then be added by using the syscall create(POLICY, < resource name >). The network becomes private when this document is installed.

3.25 Private networks

A virtual network becomes a managed network when a "**.**config" document is installed within the network resource. That is achieved by means of the syscall create(CONFIG, < network name >); The .config document is accessed in the same way that a .users policy document is accessed; a web browser is used for browsing and editing.

Managed networks defined in the name space ordinarily know no physical bounds. They are comparable with programs written in a programming language. The language conceals details of the underlying network hardware from the ordinary network user. However it is possible to define a managed network which corresponds to some part of the underlying physical infrastructure. In that way a suitably privileged user can monitor and influence the way that the physical network operates. A network of this type is called a "private network."

Interlan networks are self-configuring. That capability is largely based upon periodic reports received from the supervision system as it scans the physical plant. Those reports includes the identity and status of every network node that has an Ethernet address and every transmission line that links one node with another. The routing system overlays a network of paths upon the physical network discovered by the supervision system. Those paths define routes that are not only physically possible but also reflect management policy and needs demonstrated by the ongoing traffic flow. Policy, set by network operators and users of the .config management system, also has an influence on path routing.

Linkage between the logical and physical views of a private network is achieved by matching Ethernet addresses. The goal is to obtain a one-to-one correlation between a resource name and a piece of equipment in the physical plant. Connectivity as perceived by the user is expressed in terms of flows. In the underlying system connectivity is based not only on the presence of physical transmission facilities but also on their status in the routing system. Some of the hardware connectivity may be excluded from the path network when engineers are working on the equipment.

When correlation between the logical and physical views has been achieved, there is a basis for meaningful communication between the network operating system and the user about events that are taking place in the underlying physical plant. Facilitating that communication is the role played by a .config document.

Supervision incorporates an alarm system. Alarms are generated when there is an equipment failure, when queues overflow, and when a network node fails an identity check. In some critical cases the supervision system itself will trigger corrective

action, but usually the alarm is directed to the specific management subsystem(s) that is in a position to take corrective action. The .config management system can also receive and display these alarms.

On some topics the configuration management system looks for guidance from the user. For example, if there is a change in the physical plant the operating system will try to correlate that change with the user's plans for configuration changes. Any discrepancy is reported and activation of any new plant may be held in abeyance until the user's approval has been given. Similarly for transmission lines. Reconfiguration of the network's connectivity is subject to guide-lines that have been approved by the user. Traffic may not be routed over a new transmission path if user approval has not been given. By these means the .config system adds strength to the network's defenses against configuration errors and attacks on the physical network.

Each network resource can have its own .config management system. It is also possible to integrate the management information feeds from several networks to create a coherent view that spans a larger territory. In this way the Interlan network management system has the modularity to meet a wide range of user needs. It is possible to distribute responsibility for the different parts of a network which a system operator leases to network users. At the same time the system operator can oversee all aspects of his network operation by monitoring the collective .config display.

3.26 Datagram networks

During the past two decades, TCP traffic volume has dominated all other traffic types carried by the Internet. But that was an era in which the three major traffic types were each carried in a separate network. Telephone, television and Internet networks existed side-by-side, even within the same cables. Looking forward, the next generation network must efficiently carry a heterogeneous load that is the combination of at least these traffic types. Not only is that technically feasible and economically advantageous, but also there is a compelling service opportunity at the application level where integration of computing with communication is transforming the way that networks are used.

In some cases the business or technical requirements for a service have in the past taken advantage of specialization in the manner that a network did its work. Television distribution networks are one example. They have provided high bandwidth multicast distribution for video content, and low speed upstream transmission for control signals transmitted from individual homes. That paradigm is now changing but it should be anticipated that other forms of specialized network configuration will be economically justified in future. That is one reason why an Interlan switching center can support more than one style of forwarding engine, and it the means by which some existing Internet practices are supported on Interlan networks.

Datagram packet forwarding is a minimally constrained communications service. Today, it is of particular interest to those engaged in file sharing and multi-person games. These applications involve multiple processes that send short messages to one another in what may seem to be a random pattern. A flow-specific network architecture is not well suited to their purpose. These applications make heavy use of UDP datagram service. Unfortunately, in this unrestricted form of networking it is hard to distinguish between well intentioned and malicious behaviors. So the question is how to provide the best possible service for these very interactive applications while providing protection for society against the seriously offensive behaviors of others.

The method proposed for the Interlan architecture is to support datagram and specialised packet forwarding services in private networks which use configurable forwarding engines. The degree of flexibility contemplated is illustrated by three examples that will be described below.

a) UDP datagram networks
b) Ethernet bridging networks
c) Multicast networks

Each of these private networks is a resource defined within the Interlan name space. It must be defined and its functionality must be configured before any application process connects to it.

UDP datagram network

An Interlan wide area network employs Ethernet in place of IP for wide area communication, but at least during a period of transition, end-to-end compatibility with non-upgraded hosts requires that the IP header still be included. So no additional header is required when a host is looking for UDP datagram networking across a wide area. In the regional switching center and in the backbone network, IP source and destination addresses refer to the endpoints of datagram travel, and the Ethernet header provides the necessary support for those UDP packets to move through a long-distance network comprised of Interlan flows. From an application's perspective this is UDP networking in a private network. Presumably there is no shortage of IP addresses here because the datagrams are entirely confined to the private network. Inerlan routing and Ethernet globally unique addressing will ensure that each packet finds its proper destination. Meanwhile, the host is in effect using UDP/IP in the traditional manner. Hosts that are individually connected to the private network may be located anywhere that receives Interlan communication service. They can even be mobile.

Ethernet bridging network

It is commonly required that local area networks installed in the several sites of a large corporation be combined so that they function as a single large network. In terms of Interlan network features, "behaving as a single network", means respecting access controls for services and networks. There is a separate level of concern which gives rise to the IEEE 802.3ah standard. That protocol allows for one Ethernet to be encapsulated within another so that one corporation is prevented from spoofing, or otherwise attacking, another when classic Ethernet is used to multiplex traffic in an aggregation network (i.e. there is no path protection). An Interlan private network with forwarding engine configured for datagram routing creates the framework for inter-site connectivity where each private network is identified in the primary header. When it comes time for the secondary header to be processed, the engine acts for the designated corporate network as it processes the primary addresses. Thus the primary header protects against a spoofing attack; it is not made redundant by the fact that Ethernet addresses used in the several corporations are (supposed to be) mutually distinct. From the point of view of a single corporation, the multi-site network operates as a single Interlan, meaning that services and virtual networks can be defined with access controls in the corporate name space and they will be implemented as the traffic is routed through the wide area network. This creates a much more versatile environment than is provided by Ethernet VPN technology.

Public safety

In order for an Interlan user to send and receive datagrams he must connect to a datagram private network. There is no means for transmitting datagrams except in such a network. Therefore, to launch an attack based upon the more unconstrained nature of datagram communication, the attacker must first persuade his quarry to join a datagram network. A user that has no intention of using a datagram service can set a policy in his account which prevents his host from connecting to a datagram network. By this means any virus that makes its way into the host is blocked from connecting the host to a datagam network. The block is implemented in the local agent and so is difficult for a virus to circumvent.

Datagram packets are routed through forwarding engines that keep each private network separate from all others. Each network employs its own virtual backbone, made up of flows that are dedicated to the private network. In this way substantial barriers are placed in the way of a hacker that tries to attack users in another private network, even when the two networks share the same transmission facilities. Never-the-less, a private network has access to the standard features that come with Interlan communication, including mobility and automatic service restoration.

There does remain the danger that a host may at one time be connected to a datagram network and at another to hosts in the public network. A successful attack launched in the datagram network can leave a virus which comes active in the public network. That problem is not solved by the mechanisms described here. However, a virus operating in the public network cannot use datagrams to launch its attack.

3.27 Multicast networks

There are two primary applications for wide area multicast:

a) Integration of local area networks that are connected by a wide area network, and
b) Distribution of content to many destinations in a wide area network.

LAN integration

Premises Ethernet networks offer a multicast service which is designed to work well in the local area. Each LAN has multiple multicast trees, each tree having a distinct Ethernet address. When interconnecting LANs that are spread over a wide area, the multicast trees must be interconnected in a way that forms a single multicast which does not deliver duplicate packets to any host. That mechanism should work reliably even after there has been an equipment failure. The difficulty is that providing reliable service requires the creation of alternate routes, but that should not mean that a multicast packet is replicated and sent by both routes.

Content distribution

Various technologies have been developed for distributing content over a wide area by way of an aggregation network. DOCSIS and EPON are two such systems. Other possibilities include a switched fiber-optic network that put switches in the neighborhood of user premises. These content distribution networks typically expect the source of content to be external to the distribution network. Cable television networks deliver content that is typically paid for by subscription, or the content is associated with an advertisement. In the case of subscription services, content distribution is controlled so that non-purchasers do not get free copy. Many advertisements are short messages distributed to every premises. However, there is increasing interest in selectively distributing advertisements according to the interests of those who will receive them. Thus content distribution networks are constantly being reconfigured as users select different content and distributors send out different advertisements.

Dynamic routing of content that is distributed through a tree can be accomplished by putting switches in the nodes of the tree or by filtering the content stream just before it enters a premises LAN. There is extra cost when node switches are used, and a risk of that the distribution will be tampered with when filtering is used. So, both methods are candidates of interest.

Virtual multicast networks

A backbone multicast distribution system has two substantial parts:

a) Replication of content
b) Flow switching

It is proposed that content replication is a function that is available in a virtual multicast network, and that network agents should provide a service for configuring content distribution systems by means of flow switching.

A multicast virtual network is characterized by the presence of the special service ".multicast" in its directory. Other services in that directory implement multicast trees; each named service implements a single tree. The name of the service is also the name of the multicast tree. Figure xx illustrates a virtual network directory with multicast capability.

Virtual multicast network name:	CityMedia
Multicast service manager:	**.**multicast
Multicast services:	CBS News
	Weather Channel

A "multicast client" is an application which receives content from one or more of the trees. So, for example, with respect to figure xx, a client that connects to "CityMedia/CBS News" will receive video from that news station.

The .multicast service is a manager configures the multicast trees. By connecting to .multicast a sufficiently privileged application can add a new multicast service to the virtual network. A "multicast source" is an application that feeds content into a multicast tree. The privileged person whose job it is to attach video feeds to broadcast services will connect the news feed that is received by satellite to the multicast service "CityMedia/CBS News".

The routing of tree content is controlled by .multicast. A single tree may have alternate sources available; several sources having connected been to the tree. The routing of output from a multicast service is also under the control of .multicast. That service manager responds to commands issued by applications which connect directly to .multicast.

LAN integration is most economically handled by means of one multicast service that can be configured with a list of the LAN Ethernet addresses used for wide-area multicast. Configuration is handled by .multicast. Each premises gateway connects to a backbone multicast tree as a client which is both source and receiver of wide area multicast packets. Additional multicast services can be added when, for example, a client wants to receive a public video service on one of the Ethernet multicast addresses.

Configuring aggregation networks

The virtual network abstraction can be used to configure a wide range of multicast services including those that involve special-purpose aggregation network technologies.

The type of virtual network service that **.**multicast installs in a multicast virtual network reflects the type of physical network that is being used. Different service process types are used for the different aggregation network and LAN network technologies that are available. The network interface used for configuring backbone multicast networks (described above) is generic enough that it can probably be used to configure multicast content distribution in a variety of network types. The service processes interpret connection of consumers and producers to multicast trees according to the technology of the network being used.

Suppose, for example, that the aggregation network is a broadcast system with video channel filters located in NIUs which connect a user's LAN to the aggregation network. A client on user premises connects to a multicast service in the virtual network that corresponds to the physical network that provides his service. The request to connect to a particular broadcast channel is handled by the appropriately named multicast service located in the virtual network directory. That service identifies the relevant NIU from the connection request message sent to the user's agent, and sends a message to the NIU asking for the appropriate change in channel filter.

3.28 Internet evolution

The strategy for Interlan design has been to minimize the impact of network evolution on premises networks, hosts and application software. Internet host and local area networks can connect to an Interlan wide area network without any change in premises hardware. Host application software does not need to change but each host needs to install an application which runs while the host is connected to the network. There are recommended changes in host operating systems.

Once connected, a host can use Internet software with added security for wide area communications. TCP and unicast UDP sessions are routed through Interlan flows, giving them protection against packet inspection and packet insertion. This result is achieved by treating certain IP packets as "IP syscalls". In particular, the first and last packets in TCP and UDP sessions are functionally equivalent to the Interlan syscalls connect( ) and disconnect( ) respectively. IP applications which refer to an Interlan service by name can connect to that service if they satisfy the service's access controls. Interlan applications that use an Internet gateway can connection to servers that operate on the Internet. The gateway converts syscalls into equivalent IP protocol sequences.

An IP host, which is attached to an Interlan network, must register as a server with its local agent before it can receive connection requests. The agent will create a resource record and will define a service name as appropriate. Names already registered with the Domain Name Service retain their ownership and are available for use with Interlan resources. An IP host that is registered on an Interlan network can also define access controls that will be applied whenever a client requests connection.

To go beyond the use of TCP and unicast UDP communication, an IP host must install operating system software which supports the use of Interlan syscalls.

4 Payload network

This description of payload network architecture begins with a brief review of technical and operational factors which have influenced its design.

4.1 Critical technologies

The characteristic parameters for wide area network performance are transmission speed, switching speed and memory size. The speed and capacity of production transmission facilities continues to improve. Today (2010) 10 Gb/sec. is the cost-effective rate for long-haul transmission and Infinera now offers 160 wavelengths per fiber. It is expected that 40 Gb/sec then 100 Gb/sec per wavelength will soon be cost effective and will become the preferred service rates for long distance systems. Not only is there an economy of scale in long haul transmission, there is also a reduction in delay when high speed allows an increased level of multiplexing. However, switching system electronics is expected to become a limiting factor. Optical switching can solve that problem in the core of a backbone network but at the backbone's edge where queuing is unavoidable, something close to 10 Gb/sec per switch port may be the realistic choice.

The past four decades have seen a persistent and rapid change in the technologies which enable computing and digital communications. Advances in material sciences and in manufacturing have been particularly influential. Moore's law characterized the era by capturing the exponential increase in switching speed and circuit density. Today, transistor switching speed has stalled leaving processor clock rate at about 3 GHz. For further improvements in transmission and processor performance we must look to system architecture - transmission channels that occupy more than one wavelength and processors that contain multiple cores. Device performance, it seems, can no longer be expected to follow an exponential track.

The size of the largest (SDRAM) memory chip has not yet stalled. Devices with 4 Gbit capacity and 3 ns cycle time are becoming available (Micron) and the International Technology Roadmap for Semiconductors 2007, (www.itrs.net) forecast continued exponential improvement (2x every 3 years) for the next 7 years. However, a static cycle time means that memory bandwidth will remain a critical factor for switching systems in the years to come. Here also, architecture rather than device technology is likely to be the key to future performance gains.

Steadily, optical transmission is taking over aggregation networks. That is even true where DSL and DOCSIS networks predominate. In these systems the bulk of the distance from a switching center to a home is already being replaced by fiber. Reducing the length of the copper segment in a DSL network has allowed impressive gains in signal processing to raise bandwidth beyond 100 Mb/sec. But complexity and power consumption will surely work against DSL's future. Not only does fiber have superior performance; it is also a more reliable medium for outside plant and passive optics reduces the need for power distribution. It is in the last 50 feet that installation costs are holding up progress (at least in the United States). One has to assume that eventually that financial problem will be overcome. IEEE 802.1 is working on 10 Gb/sec EPON which is most likely to set the future direction. Ethernet's dominance on user premises is steadily spreading into wide-area networks. If that trend is to continue Ethernet must find a practical way to route traffic over a wide area, and Ethernet packet switching must circumvent the performance limits of CAM technology. That is a challenge that Interlan architecture will address.

4.2 Demand for bandwidth

Estimates of future bandwidth demand during a period of rapid applications innovation is a risky matter. However, considering only what we now know, there is clear indication that video traffic will dominate bandwidth use in the years immediately ahead. A decent NTSC signal requires about 4 Mb/sec continuous service and large screen HDTV requires about 20 Mb/sec.

Measurement of Internet traffic in the early 1990s and in 2000 suggest that the mean dwell time between one page download and the next has remained constant at about 10 sec. It would appear that this time interval is governed by human "think time", a factor that does not change much over time. The volume of data delivered during a download varies greatly (with a long tailed distribution), and is presently constrained by available data rates in the aggregation network. However, suppose that everyone has 100 Mb/sec burst rate available for download, and suppose that a one page download occupies 1 megabyte. The expected delivery time for one page is 100 msec. The average data rate for that high quality experience is 1 Mb/sec, much less than even NTSC video.

A CATV industry observation is that in the busy hour about half of all homes are active and the reset appear to be "lights out". A few years ago the average active home had two televisions active during that busy hour. One may conjecture that some of the interest in TV is being replaced by an interest in content obtained by way of a computer, but the same number of eyeballs are likely to be involved and the availability of high definition video has, if anything, heightened interest in that type of content. So it seems that in future the average service rate into a home during the busy hour will be 20 Mb/sec and the demand will be for a peak rate which can deliver a high quality image in close to a human response time. Transmission at 100 Mb/sec comes close to achieving that result.

Fiber is the most realistic way of delivering reliable service at 100 Mb/sec or greater. Thus one has to believe that fiber deployment in aggregation networks will become widespread. However, it is expensive to install a new fiber cable in the street. Patriot Media, a one-time cable television operator that provided service in New Jersey, announced at the time of its purchase by Comcast that they had rewired their CATV serving areas on the east coast with fiber at a total cost of about $25,000 per mile. This Patriot Media cost includes all capital costs and expenses associated with the facility upgrade, but that figure does not include the short distance from telephone pole into the house. That last piece of cable, which goes through a the home-owner's front yard, can be quite expensive to install, particularly if local ordinances require that the cable be buried. It is even more expensive than that for a new service provider with no existing right of way. So in spite of the growing demand for a fiber aggregation network the incumbent network operators are likely to be the only large scale providers of fiber into each home. Among CATV operators there are those that believe that no one makes money when a second operator moves into town. The problem, at least in the past, has been that the cost of a second facility cannot be recovered out of the one revenue base. If that experience is repeated with the new digital services, it seems likely that in the long run there will not be many service providers with overlapping fiber aggregation networks.

The aggregate bandwidth required for the nation's backbone networks is hard to judge. Patterns of video consumption are changing rapidly now that movies can be downloaded with little effort. We anticipate that regional distribution centers will play a significant role in the years to come. Akamai's recently constructed HD network fits this pattern. The large regional switching centers are the natural places to put content distribution centers. In these centers there is the largest concentration of transmission lines that reach into individual premises. Regional distribution of content not only avoids the expense of long-distance transmission it also is a practical way of minimizing propagation delay between client and server. If this prediction is correct, it would mean that demand for backbone bandwidth is less than might be suggested by our estimates of consumer consumption.

The immediate challenge for the industry is to create a rewarding business plan for this market. Industry wisdom has it that one cannot make a profit by just moving bits. Hence the present scramble to offer video service on new facilities. While there is so much uncertainty about how things will evolve there is caution in making the large investments that are necessary to create new infrastructures. However, it seems that the destiny for computer communication is clear - to enable anyone to engage in any activity while in any place at any time. Only then will the full potential of the information revolution be realized.

The Interlan architecture anticipates that long-term view. Our goal is to describe a service and network architecture which will meet today's need and can evolve successfully to deliver service well into the future.

4.3 Regional design

"Buildings are expensive." That has been a common complaint of telephone system engineers. Buildings imply operating costs of many types - including real-estate, building and plant maintenance, operating staff, and the services which support these people.

At the close of the Bell System (in 1984) the AT&T network had about 10,000 central office buildings. These were located so that 80% of the homes were within 18,000 feet of a switching center. The properties of analog transmission on copper wire explains these numbers. However, it will not be many years before fiber aggregation networks become the dominant influence on network architecture (even if fiber does not make it all the way into a home). Therefore, the rationale for having so many switching centers will cease to exist.

How many switching centers should there be in a new network?

Denial of service attacks are an unfortunate fact of modern life which will be more easily overcome when the size of a switching center is increased. A larger switching center necessarily has greater total capacity both in terms of switch bandwidth and in the network operating system. As a result, a denial of service attack will need to generate more traffic in order to overwhelm the center. The network operating system will have more compute capacity, and there will be more time in which to respond with defensive moves.

Congestion control is another matter which speaks for large switching centers. This conclusion can be simply stated (but not really explained) by reference to the "law of large numbers." More traffic being processed by faster servers leads to shorter queues. Fewer switching centers means a simpler hierarchy for the network's backbone: Aggregation networks feed into switching centers which are interconnected by a relatively small backbone network. Control of ingress traffic flow in the aggregation network is relatively easy because the propagation delay is short and the topology is usually simple (a tree). The backbone network is a very different matter; round-trip time can be long and there is ingress traffic from many different directions. Precise control is much more difficult to achieve. However, queue length as a function of the number of flows has variously been observed to be proportional to 1/N for exponential flows, and 1/sqrt(N) for TCP flows. So, a large regional network feeding into a large switching center will produce a relatively smooth traffic flow in a simplified backbone, thereby reducing delays and making congestion easier to control.

To investigate the concept of large regional networks for the United States we overlaid a map of the country with hexagons, each about 100 miles radius. Using the United States census for the year 2000, we obtained population data by zip code and thereby estimated the number of households in each hexagon. Those hexagons with 100,000 households or more are considered large enough to have their own regional network and switching center. People in hexagons with less than 100,000 households were included as participants in the networks of adjacent regions. Figure x shows the 109 regions in solid color. The less populated hexagons are transparent. One hundred miles is seen as a plausible size for a regional network because the round-trip time from a home to the center and back is little more than 1 msec, and 100 mile range is practical when glass fiber is the transmission medium of choice.

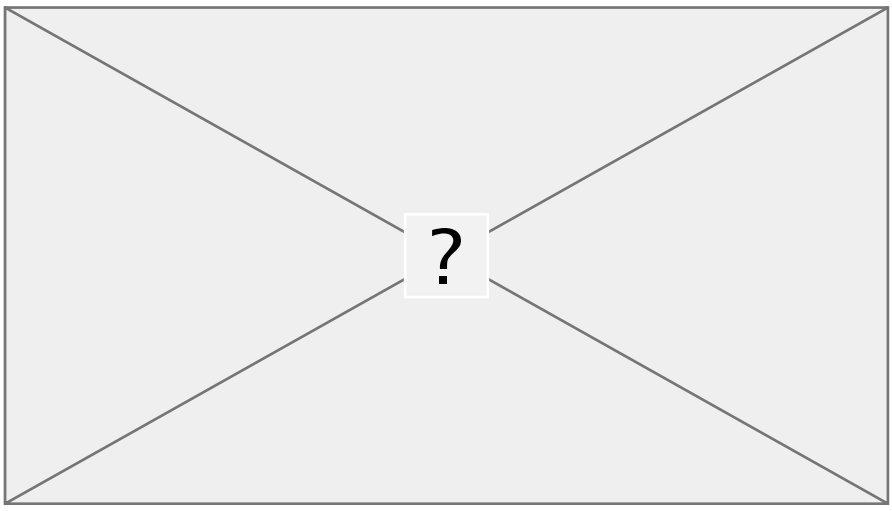


Figure xx. The United States with 109 regions

The color of each region indicates the number of households served, rounded up to an integer number of millions. The average population of a region is one million households. The largest region, New York, contains 9 million households.

For comparison, an initial study of China reveals a country with 1.3 billion people distributed across 10 million sq km. which we divided into 91 regions each 100 mile radius. The data was per province, a lower resolution than for the United States. Some data was as much as 10 years old. Also we did not have complete data about the number of people in each family. Twenty-one regions are estimated to contain 9 million households or more, the largest are the two regions which together house the 40 million households of Shandong Province. Given a switching machine that can serve 9 million households, some regions would require more than one switching center. The China backbone network would interconnect 111 regional switching centers.

The USA and China cover approximately the same land area. They both have large densely populated areas along with some large areas that have very little population. While the population of China is 4 or 5 times larger, its population is more highly concentrated. Whereas the United States has one major concentration, 9 million households in the New york region, China has 21 regions that each contain at least 9 million households. While much of the research that we have done for a USA Interlan network can be applied in China, there is a need in China for switching centers that are 3 times the size and a backbone that carries perhaps 4 times as much traffic.

The United Nations anticipates that the world population will reach 9 billion people by 2050 [8]. There are 192 countries that are members of the United Nations. The three largest are China, India and the United States. They account for 40% of the world population today (2010). If one excludes the land and people of China and India, the rest of the world has an average population density of 32 people per Km^2. The average density in the United States is 31 people per Km^2. Therefore, while this report focuses on the United states, it is anticipated that our conclusions will be appropriate for a large fraction of the world population.

4.4 Interconnecting the regions

For security, network operators do not publish detailed information about the routes followed by their networks. So, continuing in the spirit of exploration to gain an initial feel for the network design problem, we generated our own backbone maps for the United States based entirely upon the air miles between regional switching centers. (Real transmission line routes are strongly influenced by available rights of way, such as roads, railways and gas pipelines, and by prominent characteristics of traffic flow. Today there are multiple national backbones which in practice will strongly influence the form of any future network deployment.)

[8] United Nations 2003 population forecast

For this exercise, we are interested only in glass fiber transmission lines. Cost, reliability and performance are all in its favor. We assume that each of the 109 regions contains a switching system at its center and that backbone trunks interconnect at those points. Typically, the dominant cost of fiber deployment is the cost of construction rather than the glass that the cable contains. For that reason our cost estimates are based solely on route miles. No attempt is made to estimate the required bandwidth of the deployed facility. However, redundant (alternate) routes are important as a means of achieving service reliability.

Switch cost is a function of the required peak traffic throughput. So, rather than have long-distance traffic pass through the centers of many regions, we have explored a two-layer backbone design in which a small number of high capacity transmission lines carry long-distance traffic, and then direct interconnection of neighboring regions brings the traffic to where it is needed.

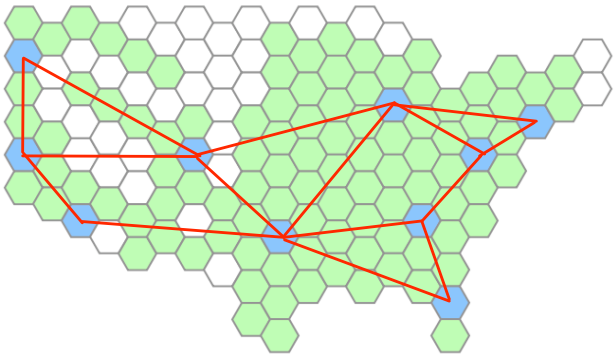


Figure xx. Exchange network

The long-distance transmission lines constitute an “exchange network”. Figure xx illustrates a 10-node design which is redundantly connected for service reliability. The ten nodes, colored blue in the figure, are named for the major cities which are nearby: Seattle, San Francisco, Los Angeles, Denver, Dallas, Chicago, Atlanta, Miami, Philadelphia and New York. We estimate the “cost” of this network to be 13,317 miles of cable.

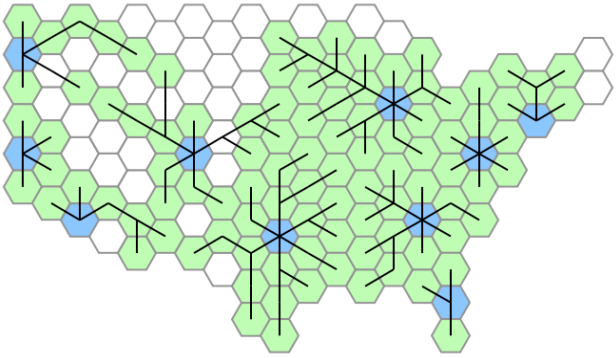


Figure xx. All regions connected to the exchange network

Figure xx shows a minimal design that gives each regional center a route to the exchange network. The longest of these routes passes through three “transit” switches before reaching the exchange network. The cost of this network is 19,469 airline miles. Propagation delay based only on airline miles, is 1.2 msec to transit a hexagon.

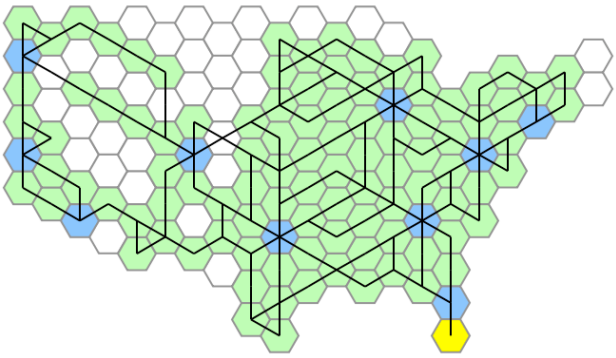


Figure xx. All regions redundantly connected to the exchange network

A more satisfactory network configuration will provide every region with at least two independent routes to reach the exchange network. Figure xx shows how that can be done[9]. The cost of this design is 28,814 miles, approximately 50% more than the design without complete redundancy.

The three examples illustrated here are “transmission line networks”. Each line represents a cable which contains a number of fibers and therefore provides a specific amount of bandwidth between one regional center and another. The word “path” is used to describe the more abstract concept of a route which payload may follow when moving from one region and another. Each path follows a specific sequence of transmission line segments contained in the available backbone network(s). Its route may include passage through an exchange network. For example, a path from Houston to Boston might enter the exchange network in Dallas and proceed via Chicago to New York. After leaving the exchange network in New York, it passes through the regional switch in New York before arriving at Boston. Another path between the same two cities might avoid the exchange network altogether, taking instead the “slow route” through a series of 10 regional switching centers which includes Atlanta, Philadelphia and New York.

4.5 Paths

The payload network is implemented as two sub-systems: The “flow subsystem” focuses on a user’s need for information transport between one application process and another. Flows are created by users and are directly linked to a user’s applications. By contrast, the “path subsystem” focuses on network performance. It strives to make best use of the communications infrastructure. A single path may carry many flows; it is a way of transporting flows in bulk. Therefore, Interlan paths play a role in packet routing which is somewhat like the Ethernet “spanning tree”; packets of a flow find their way to their destinations by following the available paths. Paths also help with network supervision and traffic management. These essential activities are generally invisible to network users.

Paths can be thought of as virtual transmission lines. They provide an efficient way of routing flows while concealing the vagaries that are associated with the physical transmission plant. Physical transmission lines have specific bandwidths, are tied to particular endpoints and are installed at significant expense. Paths are created by the network operating system as and when it needs them, and each has properties that are assigned by that system. A path need not have a defined bandwidth; paths which are following different routes can share a single transmission line for some part of the way, and each may be optimized to carry traffic of a different type. If transmission line failure makes it necessary, a path may be rerouted so that connectivity of the path network is preserved.

Paths are used in aggregation networks as a means of ensuring that the payload reaches the appropriate premises. A separate path is dedicated to each of the premises served by that network. When the packets of a flow arrive at a regional switching center, the identity of the path on which the packets travelled can be used to verify which premises they came from. Given that paths cannot be created or redirected by network users, this mechanism confirms the source of the traffic.

---

[9] Figure xx shows only a single connection to the southern tip of Florida. That is an artifact of a design system that only contemplated connections between regional centers and did not consider routing an under-sea cable.

Paths are not a new concept: an Interlan path number is equivalent to an EPON logical link identifier or a DOCSIS service identifier. The Internet backbone today also uses paths. Typically, backbone paths are implemented by a virtual circuit network (MPLS) and they enshrine routes that were chosen by a traffic engineering process; once the topology of the path network has been chosen it rarely changes. When all packets within a flow follow the same path, packets of the flow have a high probability of arriving at their destination in the same sequence with which they were transmitted. However, Internet experience has shown that confining a flow to a given path makes it more difficult to balance the network's traffic across multiple transmission lines, thereby reducing the efficiency with which bandwidth is used. That matter is addressed in section (ref).

For wireless networks a separate path is associated with each radio channel. Each device that is using a public wireless base-station transmits its packets on a path which is dedicated to that radio channel. When the device moves from one base station to another, the device switches from using one path to another. This minimizes the interruption to ongoing conversations, and allows multiple concurrent conversations to be switched simultaneously. That is particularly helpful for public transport where one vehicle may contain many customers actively using their computers and audio/video devices. A network gateway installed in the vehicle combines the packets on a single path (or a small number of paths) to the nearest base station.

In a similar way, paths facilitate security and restoration in a backbone network. Each path enshrines a route to a particular distant region. Assigning a flow to a path determines the route which the flow will follow. Checking on the path by which a packet arrives allows a regional switching center to verify that a flow has not been infiltrated by packets from an inappropriate source. The method also increases the efficiency of the backbone, because many flows can be routed and rerouted in a single act. In this way a backbone can handle a high volume of traffic without excessive busy work.

Flows may be assigned to paths according to traffic type or security classification. For example, live video and interactive data packets are carried in separate paths. That does not mean that these paths necessarily travel in different fibers, but the manner in which traffic in the two paths is scheduled ensures that each traffic type gets the quality of service most appropriate for the applications to which it applies.

4.6 Flows

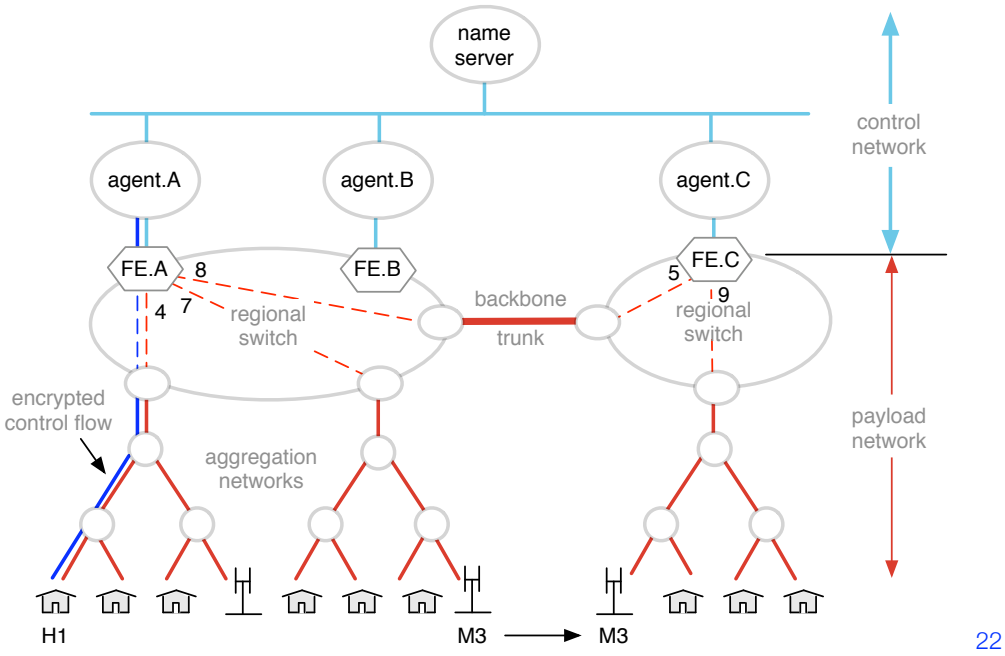


Figure xx. Flow between a static host H and mobile device M

Figure xx illustrates a flow between a static host H and a mobile host M. Initially H annd M are both connected to the same region and are served by the same agent A. The route for the flow passes through forwarding engine FE.A. Initially the flow

has two segments: the first goes from H through the aggregation network to engine FE.A; the second segment goes from engine FE.A to the second aggregation network and thereby to M. Each end of a segment is assigned a “flow number” which is used to label the flow. So H knows the flow by the label H1, and engine FE.A knows of it by the label A4. Labels for the two ends of the second segment are A7 and M3. The flow number at the end of a flow segment is chosen by the device at that end of the flow.

When mobile host M moves across to the region containing engine FE.C the flow is reorganized. Now it has three segments: H1 to A4 is as before, the segment A8 to C5 which passes through a trunk transmission line to reach engine FE.C, and the final segment C9 to M3. Notice that when M moves the label for its end of the flow does not change.

Flow labels are used only in the host and network operating systems. They are not used by application processes. As already described (ref), an application obtains a flow handle when executing system calls listen( ) or connect( ). Flow handles, like file handles, are integer values chosen by the host operating system for use by an application process when making system calls. The flow number which the host operating system chooses for its end of a flow is contained in every packet of the flow that the host sends and receives but is unknown to the application process. In Figure (ref) host H has chosen flow number 1 and host M has chosen flow number 3. Other flow numbers (4, 7, 8, 5 and 9) have been chosen by the network operating system.

The Interlan operating system uses Ethernet addresses to unambiguously identify each host/network interface. (These addresses are, for brevity, often referred to as “host addresses” but they are always assigned one per host interface.) An Interlan host may have more than one network interface, each connected to the same or a different network. Forwarding engines also have Ethernet addresses. From a user’s perspective there is one address per engine. So, within the network operating system a flow label consists of a flow number and an Ethernet address. Each label identifies an end-point of a flow segment which connects to a host interface or to a forwarding engine. For convenience of explanation in this report the Ethernet address is represented by a letter, such as H or A, and a label by the combination of that letter and a flow number.

A typical flow between two hosts within one country has just two or three segments. The first and last segments are carried by aggregation networks and the middle segment is carried on a backbone network (or passes through a series of backbone networks.) In the case that the two hosts are within one region there is no backbone segment.

In the following Figure D and E are hosts within homes W and Z. A flow between them passes through forwarding engines A and B in the New York and Texas regions respectively. Agents linked to A and B within those regions coordinate connection setup between the hosts. Then, as packets flow between D and E, the forwarding engines steer those packets along their route.

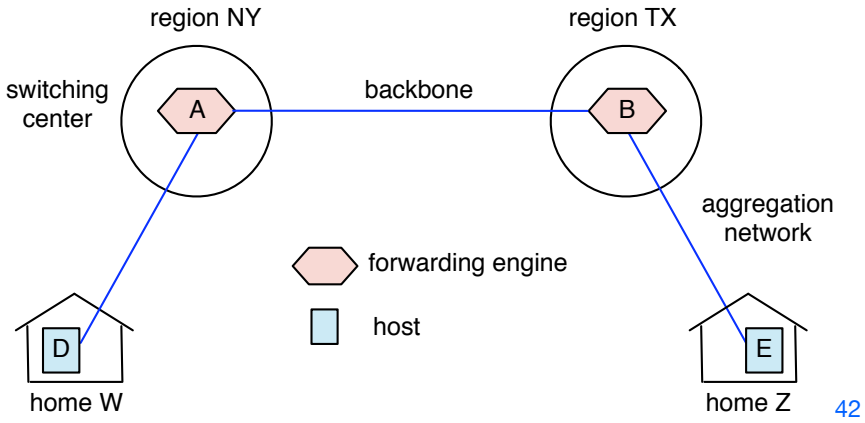


Figure xx. Flow routed through two regional centers

The paths that carry the traffic for route segments DA, AB and BE are recorded in the network interface units for the two homes and in the forwarding tables of engines through which the flow is routed. There is a path in the aggregation network between D and A, and a backbone path defines the route between A and B. More generally, an aggregation network path extends from a home or wireless base station to a regional switching center, and a backbone path extends from one regional

switching center to another. Thus, one may represent the typical route of a flow within a single wide area network by three sides of a trapezium. For long-distance connections, including International connections, a flow may pass through multiple switching centers. If, as seems likely, a country requires that there be a gateway between the national network and international backbones, then that gateway can use the technology of a regional switching center.

4.7 Flow switching

The following figure shows flow number assignments for a flow between hosts D and E.

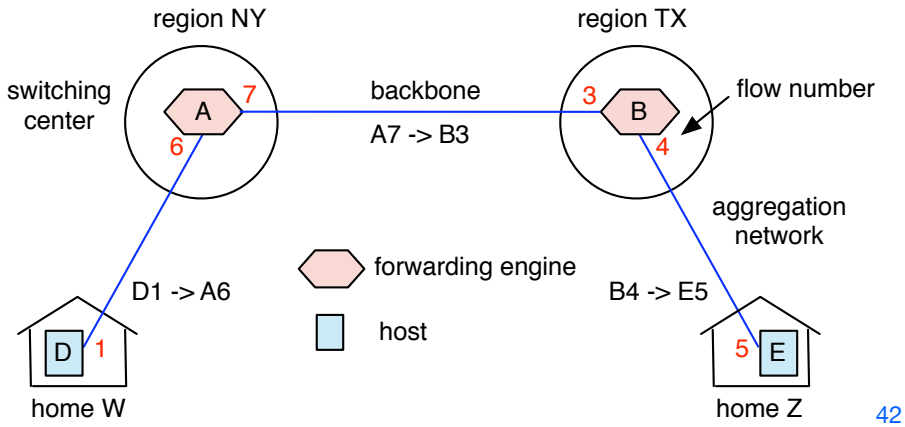


Figure xx. Flow labels for a flow between two regions

Each Interlan packet carries two flow labels in its header - a “source label” indicates where the packet came from and a “destination label” indicates where it is going. The label values refer to the two ends of the flow segment on which the packet is traveling. The flow from D to E is in three segments. Header values for these segments are D1-> A6, A7-> B3 and B4 -> E5.

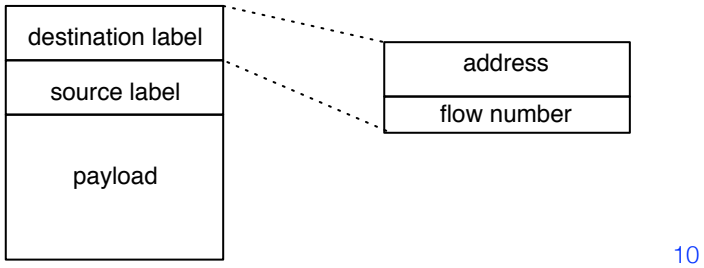


Figure xx. Conceptual view of payload packet and flow label

As a packet header passes through a forwarding engine, the source label is checked and the labels are replaced. The information which the forwarding engine needs when performing this task is held in its forwarding table, placed there by the agent when the flow was established. Table entries in engines A and B are as follows.

Table entries in forwarding engine A	Table entries in forwarding engine B
6: -> 7; D1	3: -> 4; A7
7: -> 6; B3	4: -> 3; E5

The number preceding each colon is the table entry number; to the right of the arrow is another table entry number; after the semicolon there is a flow label. An incoming packet from host D has D1 -> A6 in its header. The address and flow number in the destination label takes the packet to engine A, and to entry number 6 in A’s forwarding table. The engine compares the packet’s source label with the label, D1, in table entry 6. If the two values do not match the packet is either discarded or is set aside for analysis.

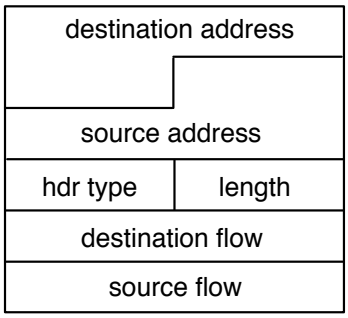
The next step for the engine is to generate the replacement packet header. It does this by following the reference to entry number 7 in engine A. That is where the label B3 is located. The packet header is replaced by the label pair A7 -> B3.

Packets traveling in the reverse direction arrive at A with destination label A7 and source label B3. In that case engine A will use table entry 7 to check the source label and will use table entry 6 to generate the replacement packet header, A6 -> D1.

Host and forwarding engine addresses are Ethernet addresses. Globally unique device addressing adds considerably to the versatility and simplicity of the design, and the choice of Ethernet addressing means that Interlan does not require network users to purchase new equipment. A globally unique address also means that every flow label is globally unique, which simplifies the routing of control messages and makes the network operating system robust in the presence of mobile hosts.

It is a fundamental requirement of network security that each host shall have a unique identity which is linked to a reliable and accurate means of packet delivery. IEEE facilitates the security requirement by administering a system in which Ethernet addresses are made available without duplication to manufacturers who then install them one in each host-network interface. (See Ethernet address administration in Chapter 7.1 for more discussion of this topic.) The packet delivery requirement is amply demonstrated on user premises by the millions of local area networks that employ Ethernet addresses. The method of header swapping illustrated in figure xx and described in the preceding paragraphs supports secure and reliable packet delivery in Interlan backbone networks.

A packet format that is consistent with the IEEE 802.3 standard is achieved by placing the two label addresses in the usual locations within an Ethernet packet, and the corresponding flow labels are carried in a header extension. Figure xx illustrates the header design. “hdr type” is an Ethernet type code which identifies this header format. The header extension also includes packet length so that the flow-layer packet is self-sufficient and does not rely upon a length field in a higher layer protocol header. This makes it possible for agents to process Interlan packets without knowing every one of the protocols that an Interlan packet may carry.



39

Figure xx. Interlan header with 48-bit addressing

Hosts that use IP while being connected to an Interlan network communicate with forwarding engines that are specialized for IP packet forwarding. These packet can be formatted exactly as they are today. No change is required in the TCP/IP and UDP/IP header structures. Instead the IP forwarding engine looks for packet length in the IP header, and the TCP/UDP port numbers are employed in lieu of destination and source flow numbers. (See (ref) for more discussion of IP protocol support.) This interpretation of IP frames applies only to packets traveling in an aggregation network. IP packets traveling in an Interlan backbone are encapsulated within an Interlan flow.

Flow numbers in the packet header illustrated in figure xx are comparable with, but not necessarily the same as, port numbers in TCP and UDP protocols. The distinction is necessary for two reasons. First, it is critical for communication security that the network ensure the integrity of each flow. There must be no confusion between one flow and another. That requires coordination between network and host when flow numbers are assigned and subsequently when they are released. Port numbers have no such protocol associated with them. Second, an Internet host uses “well known port numbers” as if they were part of the name of a service. It is expected that Interlan users will find that names alone are a sufficient and safer means of referring to services, and that well known port numbers have neither the structure nor the management to withstand the test of time. (See (ref) for more discussion of this topic.) Port numbers are of course an

essential part of the Internet protocol suite, and they are recognized as such when an IP host is connected to an Interlan network.

The 32-bit fields assigned to source and destination flows, each include a 24-bit flow number and an 8-bit field devoted to per-flow congestion control. See (ref) for discussion of congestion control.

Application processes, not hosts, are the logical endpoints for a flow. This has important implications for security, mobility and cluster computing. From a security perspective, it is useful (perhaps even essential) that each flow endpoint be associated with a specific user. Usually there is one user per host process and the host operating system does its best to establish that relationship. Within the host, per-user privileges are commonly employed in the protection of files and other host resources. So it is usual for the host to require each user to authenticate his identity when logging in. So it will often be convenient to associate the host's concept of "user" with that used in the network.

The link between a flow endpoint, an application process and a user is helpful as increased emphasis is placed on network security. The credentials which give an application the right to access a resource may be specific to a particular user. If there is within the host a concept of user location, then the host should have the opportunity to verify that a flow and its user do not become separated. In particular, a flow label may steer an incoming packet to the appropriate processor in a cluster.

Flow labels have a central role to play when a user becomes mobile. In due course, applications will become mobile and their network connections will be required to stay with them as they move from one computer to another. Inerlan agents therefore track the movement of hosts and relocate user connections appropriately.

4.8 Global Ethernet

Ethernet's popularity derives from its ease of use and low cost. These characteristics are due to its globally unique address space, an automatically configured spanning tree for routing, widespread deployment of mutually compatible components, investments in silicon for address translation, and a continuing program of technology investments that has kept the network at the leading edge of communications technology. However, Ethernet needs help if it is to be used as the basis for a public wide area network.

(a) The mechanisms that use and maintain the spanning tree will not work on a large scale.

(b) There is a concern that, as transmission rate continues to increase, the technique for address translation will not be able to keep up with the speeds required in the center of a large network.

(c) The Ethernet header carries no location information so network size is bounded by the practical size of a hash table. Today that size is about 16K entries.

In summary, wide area traffic flow in an Interlan network starts with an aggregation network that brings traffic from homes and small businesses into a regional switching center. The traffic flows are then distributed to forwarding engines located within that center. A forwarding engine checks the incoming packet header to make sure that the packet comes from the appropriate host, and then installs a new header which contains sufficient information to carry the packet through the backbone network. The backbone network operates on two protocol levels. At the flow level it protects flows from infiltration and inspection by network hackers, and at the path level the packets follow a route that is automatically configured.

Figures (ref) and (ref) illustrate this concept. 'NIU' is a generic term used in reference to the device which connects an aggregation network to a premises LAN, and a 'root' is the module which interfaces an aggregation network to a regional switch. In today's broadband networks the NIU may be an EPON ONU, a DSL modem or a cable modem. The heart of a regional center is an Ethernet switch that connects root modules and trunk modules to forwarding engines. In essence, the aggregation network is a bridge between two Ethernets - one on user premises and the other in the switching center.

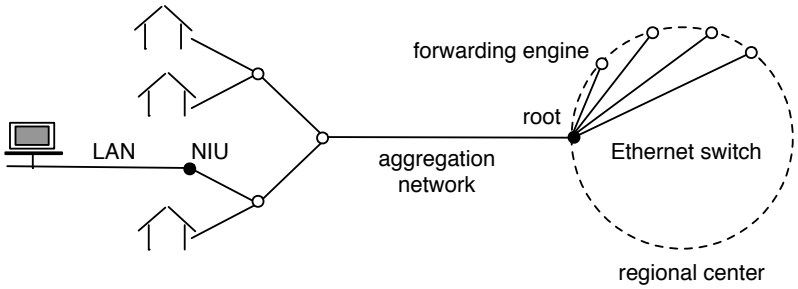


Figure xx. Homes connected to a regional switching center

Packets of a flow that enter a switching center are addressed to the forwarding engine that was assigned to the flow when that flow was established. The forwarding engine retains in its forwarding table information about each of the flows that are its responsibility so that it can check the route by which packets arrived there. Packets which arrived at the engine by an inappropriate route are erroneous, malicious or were generated by legacy software. They are either discarded immediately or are set aside for analysis. If legacy software is the problem, the agent may be able to help. (See (ref) for more discussion of of this case.) Acceptable packets are then prepared for the next stage of their journey.

If the route taken by an incoming packet is compatible with the information held in the engine's forwarding table the engine applies a new pair of labels to the packet and sends the packet to the backbone network. By discarding packets which arrive from unexpected sources the forwarding engine prevents source address spoofing. By installing new labels on a packet the forwarding engine ensures that every packet in the backbone network carries a header that was approved by an agent. This reduces the probability that cleverly constructed rogue packets can launch an attack on the backbone. Label checks are made and new labels are applied in each switching center through which the packet passes. The labels applied in the final switching center direct the packet towards the destination host.

Figure (ref) illustrates the three routing zones through which a packet traveling in a national network passes before reaching its destination. The first and last are the aggregation networks of the source and destination hosts. The central routing zone is the backbone network. Forwarding engines are positioned on the zone boundaries.

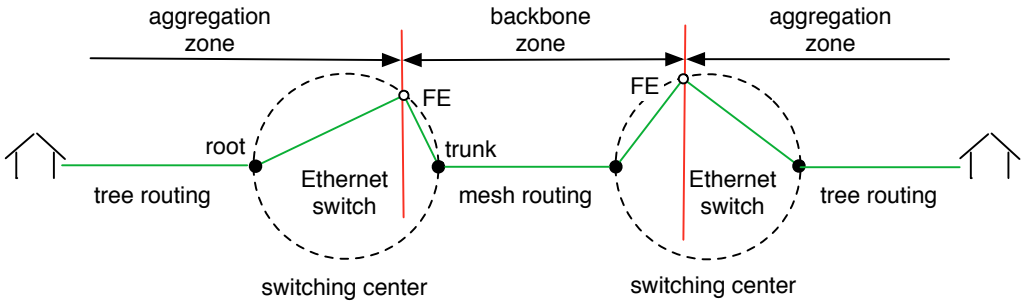


Figure xx. End-to-end flow through two switching centers

Within each zone there is a system of paths on which flows can be routed. Typically, the path topology is automatically configured and is relatively static. Load balancing, equipment failures and maintenance requirements are the most likely reasons for a change in path. In an Interlan backbone, every packet contains the number of a path along which the packets of the flow are routed. The path number is written into a packet header by a forwarding engine prior to sending the packet into a backbone network. (see (ref)) Each aggregation network has a way to identify the home to or from which a packet is traveling. For fiber-to-the home EPON networks a path number serves that purpose; for a cable modem the DOCSIS protocol uses a "service identifier" SID that serves the same purpose. In our discussion of aggregation networks we shall use the word "path" to identify this function.

In a backbone network there is at least one distinct path between each pair of regional switching centers. Multiple paths are usually available so that different traffic classes can be treated appropriately. Multiple paths also allow a large volume of traffic to be fragmented so that it can be spread across alternate routes.

A sequence of paths is assigned to a flow when it is established. The assignment is made as the server's accept( ) signal travels to the client. For a static (non-mobile) device, the assignment of its flow to a path usually persists for the life of the flow, although network technology permits flow rerouting when that is necessary. The most common reason for a change in flow assignment is when a mobile device moves from one wireless base station to another. As that move takes place the flow transitions from a path associated with the first base station to a path associated with the new base station. To expedite that action, all flows for one mobile device are collected together on one path. When the device moves to a new base station the flows in the first path "hop" over to the new path in a single action that takes place in a forwarding engine. This 'path hop' provides a quick response to device mobility.

The assignment of flows to paths is a way of simplifying traffic management in backbone networks. Rather than route the packets of a flow individually, a flow is assigned to a predefined path in a single action. For this purpose the routes of many backbone paths are chosen by an offline traffic engineering process with the expectation that the set of pre-routed paths will serve the needs of most traffic for an extended period of time. In this way the complex task of finding efficient routes is made compatible with the need to avoid substantial overhead in choosing a route for each flow.

Path numbers also play a role in flow security. A host cannot choose the path that its flows follow; instead that path is determined by host's location and the decisions made by agents. Thus, by checking on the path by which packets arrive at a forwarding engine, the engine can verify the location from which the packet came. This is an effective way of checking that a packet which claims to be part of a certain flow is legitimate.

The assignment of each flow to a specific path also ensures that packets will, with high probability, retain their initial sequence as they travel through the network.

4.9 Backbone flows

Large switching centers need to spread their traffic load among many forwarding engines. An Ethernet flow header provides the means of associating the packets of a flow entering a switching center with one particular forwarding engine where the state of that flow is kept. It is the Ethernet address of that engine that is contained in the destination flow label of these packets. Thus, from the perspective of an aggregation network, a switching center is an Ethernet on which forwarding engines are mounted. The aggregation network connects Ethernets in homes to the Ethernet in the switching center. (A mechanism in the switching center prevents traffic from one home from straying into another aggregation network.) In this way a regional switching center can grow to handle a large traffic volume on behalf of many homes and businesses. This result is achieved without demanding Herculean effort to create a forwarding engine with extraordinary traffic carrying capacity. The architecture also decouples transmission line interface function from the complexities of flow switching, making the interface modules less expensive and less power consuming than has become common in traditional large scale Internet packet switching systems. (See (ref) for more on this topic).

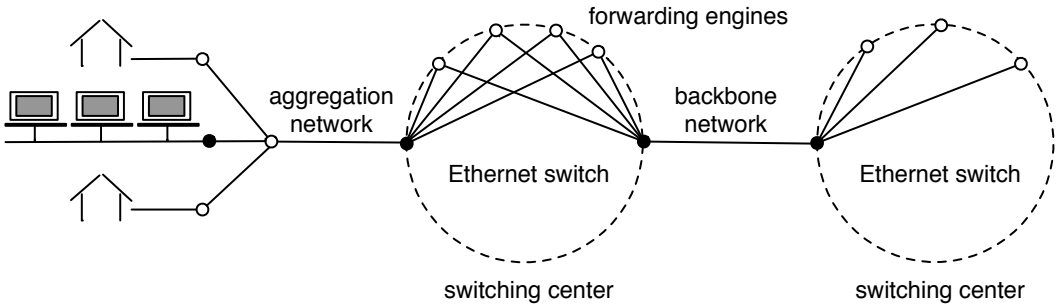


Figure xx. Networks are bridges between Ethernets

A backbone trunk may also be seen as a connection between Ethernets. Each packet contains the Ethernet addresses of forwarding engines in the source and destination switching centers. In effect, an Interlan network is a global Ethernet wherein forwarding engines are gateways between routing zones.

Each forwarding engine has two Ethernet addresses: one for use in aggregation networks and the other is for use in backbones. Packets in an aggregation network are not permitted to carry backbone Ethernet addresses; any such packet will be discarded by the root module when the packet attempts to enter the switch. Similarly, any packet traveling in a backbone is required to contain only backbone addresses. A packet containing any other type of address will be discarded when it reaches a trunk module. The checks are part of a security system the implements the following constraints:

a) A packet entering from an aggregation network must go directly to a forwarding engine.

b) A packet entering from a backbone network must go directly to a forwarding engine.

c) A packet leaving a forwarding engine may go wherever that engine sends it, including to another forwarding engine in the same switching center.

The intention is that every packet entering the network from a host must be checked and routed by a forwarding engine. This applies even when the flow is between two hosts on the same aggregation network. Forwarding engines maintain the privacy of each flow. They keep the flow's routing information and prevent hacker attacks by filtering out packets which do not follow the pre-approved route. Forwarding engines also implement a per-flow traffic control system. The rate of data flow is regulated so that each flow gets its fair share of network bandwidth and does not contribute to a build-up of congestion. Constraint (a) makes sure that these aspects of network service are not bypassed.

Constraint (b) arises because every packet destined for an aggregation network must be subject to a security check and must be given destination labels that are appropriate for the premises network towards which the packet are traveling. Label swapping is particularly important for packets destined for mobile and portable device. Agents keep track of these devices. Device movements are reflected in changes to the routing information held by forwarding engines. The engine closest to a mobile device will apply a destination label which reflects that last known location of that device.

Constraint (b) also applies to packets which arrive from one backbone and are about to transfer to another. The two backbones may use different conventions and it may be necessary to monitor traffic flow into and out of their territory. It is also possible that a time may come when some part of the global network has been compromised. Rogue packets have entered a backbone network and there is a risk that the problem will become widespread. One response to this threat is to install gateway switching centers between certain backbones. These centers filter packets and rewrite flow labels using the same technology as regional switching centers.

Item (c) in the above list is not so much a constraint as a degree of freedom which might not have been expected. The standard forwarding engine implements flow forwarding as described in the bulk of this report. However, there are other needs which can be met by installing special-purpose forwarding engines. These include engines that implement private networks (ref). These implement special services such as multicast (ref) and datagram forwarding for IP and/or Ethernet private networks (ref). The flexibility to include alternative forwarding engines within one switching center makes it economical to offer special services, and may provide an important advantage in future when the Interlan architecture must evolve into something new.

4.10 Path forwarding

Figure xx is an elaboration of the example flow shown in figure xx. Each aggregation network and backbone is identified with the path number that the packet of the illustrated flow will use. The numeric value of a path number has no particular meaning, it is chosen so that each path in a given transmission facility has a distinct number.

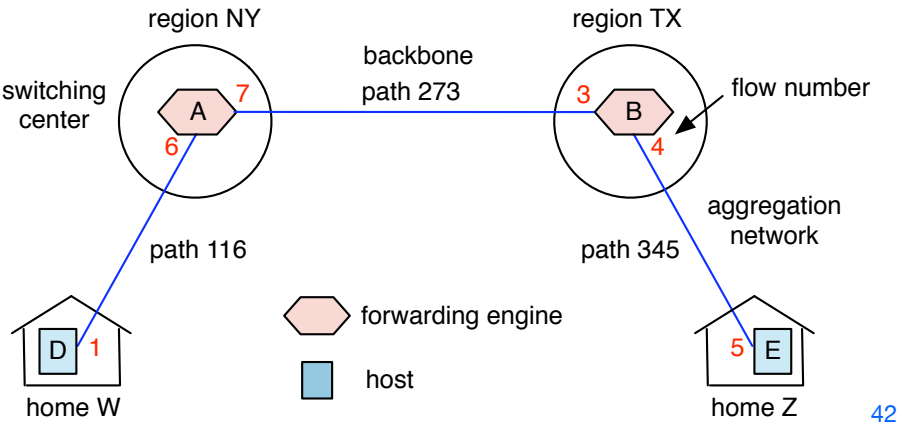


Figure xx. Flow forwarding with path assignment

The forwarding table for engine A contains the following two entries.

6: -> 7; (116) D1
7: -> 6; (273) B3

The table entry in location 6 specifies that packets having flow number 6 will leave the engine with flow number 7. Arriving packets must have source flow label D1 and must have arrived on path 116. These packets leave on path 273 with destination flow label B3. The significance of the parentheses will be explained below.

By checking the source path number contained in each incoming packet with the path number held in a forwarding table entry, the engine makes sure that all packets entering a regional center actually arrived on the appropriate path. By making this check in every forwarding engine through which the path is routed, the forwarding engines jointly provide a strong assurance that traffic from other sources has not penetrated the flows carried in this path. By the same means the engine makes it difficult for a hacker not located in the same home as D to generate a packet which looks as though it came from D.

A forwarding engine uses specialized hardware to check and move packets quickly through the network. It plays a critical role in the Interlan network design. Most obviously, the engine moves each incoming payload packet from the ingress path on which it arrived to the egress path on which it will be carried during the next segment of its journey. This and the other actions of the forwarding engine are guided by the content of a forwarding table as described in (ref). In addition to the table entries that refer to each flow, there is also a table with one entry for each path. That is illustrated in figure xx.

Each transmission facility, aggregation network or backbone trunk, terminates on a “line interface module. The line interface module for an aggregation network is known as a “root module”, and that for a trunk is a “trunk module”. Each module plugs into a switching system “slot”; thus the location of the module is referred to as a “slot number”. Entries in the path table are indexed by path numbers, and each contains the slot number for the trunk or root module through which the path is routed.

Interlan packets arriving from an aggregation network have Interlan headers as described in figure xx. The root module translates the packet’s destination Ethernet address to obtain the number of the slot in which the forwarding engine is located. Then a “switch header” is added to the packet for its journey through the switching system. That header contains the slot number for the forwarding engine, the slot number of the root module, and the number of the path upon which the packet travelled through the aggregation network. The combined packet header is now as illustrated in figure xx. Trunk modules operate in a similar way.

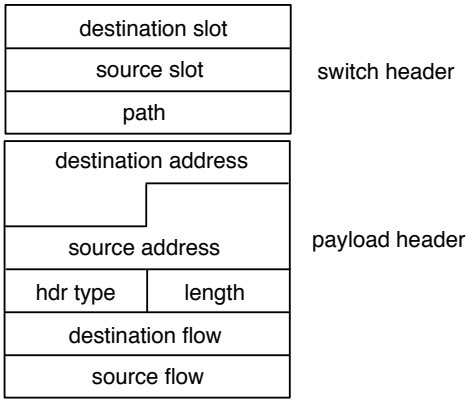


Figure xx. Packet headers used within regional switch

One may be surprised that figure xx shows the path number in the switch header, as opposed to the payload header. It is there because there is no agreement about size a location of a path number among existing transmission line frame structures. The forwarding engine's requirements are also unique. For example, EPON aggregation networks hide the path number in the Ethernet frame preamble, and the cable modem protocol DOCSIS assigns one or more "SID" to each home. As further reason for not including the path field in the payload header, we note that the path field is not required when the payload packet has entered user premises. So the path number is carried in the switch header, and each module is expect to copy that number to/from wherever it needs to be.

Connections between static devices are usually established during the exchange of connect( ) and accept( ) signaling messages. Thereafter, the sequence of paths which carry the flow remains constant until the flow is disconnected. For mobile devices the situation is different. As the device moves from one location to another a change in route is necessary. As each such move is made a new path is assigned, and when that assignment has been properly completed the flow is rerouted and the previous path is no longer involved.

Figure xx illustrates the forwarding process for a mobile device D. At the start of this scenario, D is communicating through antenna W using path 116, and after it has moved it will be communicating through antenna X on path number 416.

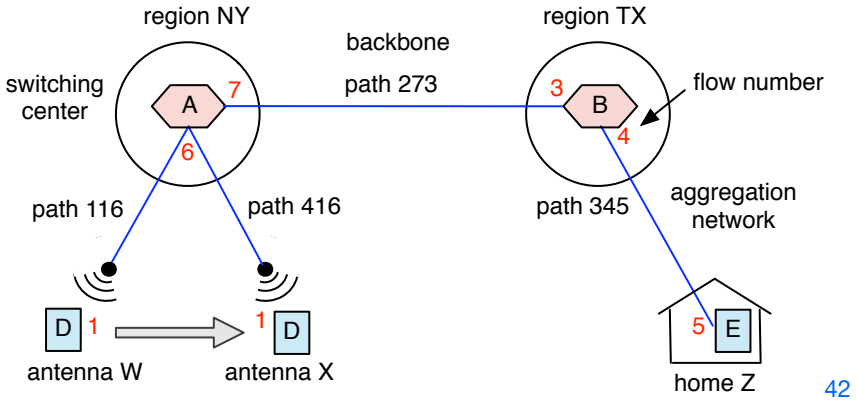


Figure xx. Flow forwarding for mobility hops from one path to another

Before and after the move, the forwarding table entries at agent A are as follows

Before the move	After the move
6: -> 7; (116) D1	6: -> 7; (416) D1
7: -> 6 (273) B3	7: -> 6; (273) B3

Only the path number in word 6 has changed, which means that it is not difficult to make the reroute an atomic event. That is desirable if packet sequence is to be preserved and packet loss is to be avoided, or at least minimized. This scenario presumes that D is capable of communicating with two base stations at one time. While host D continues talking to antenna

W it measures the signal strength from antenna X. When that signal strength is sufficient, D registers with the Interlan network via X before rerouting its payload through that antenna.

It could be that two flows from host D pass through A. These will be called D1 and D2. Each of these flows will need to be redirected when D moves from one wireless base station to another. So, rather than update several forwarding table entries when D moves, we separate the flow and path forwarding table information. Each flow forwarding table entry contains a reference to a path forwarding table entry. In this example, the two flows D1 and D2 share use of the one path forwarding table entry p1. Then, when there is a change in the path, only one table entry needs to be updated.

The following table entries in engine A are for two flows D1 and D2 both emanating from mobile host D.

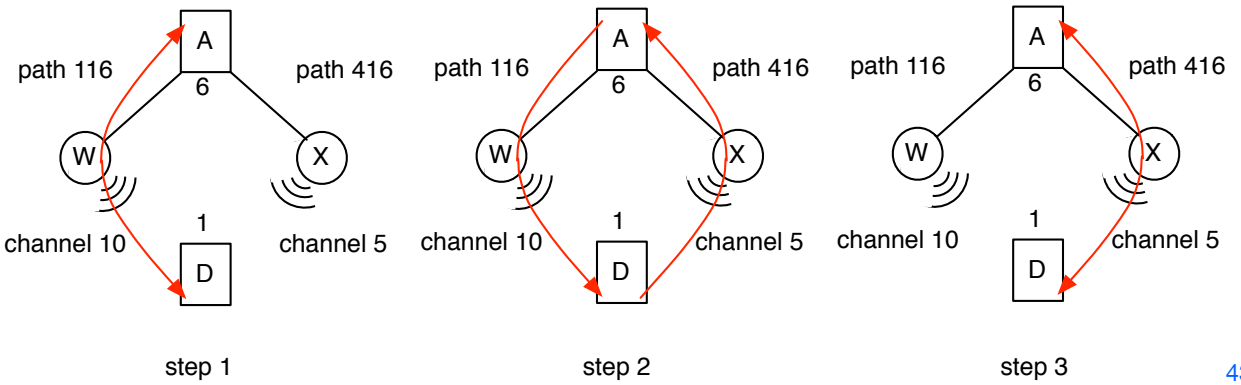
Flow forwarding table	Path forwarding table
4: -> 5; (p1) D2	p1: path 116, reserve 416
5: -> 4; (p2) B4	p2: path 273
6: -> 7; (p1) D1	
7: -> 6; (p2) B3	

Initially the two flows enter the forwarding engine for agent A by way of path 116 and leave on path 273. The forwarding process for a packet sent on flow D1 addressed to forwarding engine port A6 is as follows: When the packet arrives at A6, the source flow label, D1, is compared with the information in flow table entry location 6, and its incoming path and slot numbers are compared with the path table entry in location p1. When host D moves to base station X a new path, number 416, is assigned and that number is recorded as the “reserve” path number in table entry p1. Then, when D starts transmitting payload via X on path 416, forwarding engine A recognizes that packets from D no longer carry the path number 116 as recorded in register p1, but the incoming path number does match the reserve path number 416. So, the reserve number is copied into the primary path field for path table entry p1. Henceforth the forwarding engine only accepts packets on path p1 if they have path number 416. The effect of this action is to reroute all downstream flows traveling towards D. Whereas they had previously travelled on path 116, packets of the two flows D1 and D2 will now travel downstream to D on path 416.

This mechanism is available in the flow forwarding table as well as the path forwarding table. It allows flow and path privacy because each forwarding engine checks that incoming packets come from an approved source before permitting them to be forwarded. The checks apply even during the switch-over from one route to another. The agent must pre-authorize every reserve route but it is left to the forwarding engine to swap the table entries at just the right moment. Thus security is not compromised by the support for high performance mobility.

4.11 Flow restoration

When the reserved route mechanism is used on a full-duplex flow it provides a mobile device with some control over when flow redirection takes place, and it leaves the opportunity to minimize or avoid loss due to a change in route length. The mechanism is illustrated in figure xx3.



43

Figure xx3. Redirection of a full-duplex flow

Step 1 illustrates full-duplex flow between forwarding engine A and mobile device D before any redirection has taken place. The assumption is that D's radio allows simultaneous communication with both base stations, W and X. That allows D to login to the network via X and initialize its protocols for the new route before losing the connection through W. At this time D adjusts its signal and protocol receivers for the change that is about to come. In the case that the new outgoing route is shorter, D should pause its transmission at a convenient moment so that when, in step 2, it redirects its output, all of the data transmitted via W has passed through A before any of the new upstream transmission through X reaches A. The agent A can send a signal telling D the difference in path lengths. When the first of this redirected flow reaches A, the forwarding engine immediately redirects the downstream flow, step 3. Therefore, when the route through X is shorter than the route through W, D should be prepared for a brief period when traffic will be arriving from both X and W. Finally the flow from X stops and the transition is complete.

As a mobile host moves from one wireless base station to another it may gradually get farther and farther from its original location. Eventually the host may move so far away from where it started that routing all of its conversations through the original switching center leads to inefficient use of network resources. Considering where the two endpoints of a flow are, it may be possible to find a less expensive route between them. When such a route is found, the flow can be rerouted using a procedure very similar to that used for path rerouting.

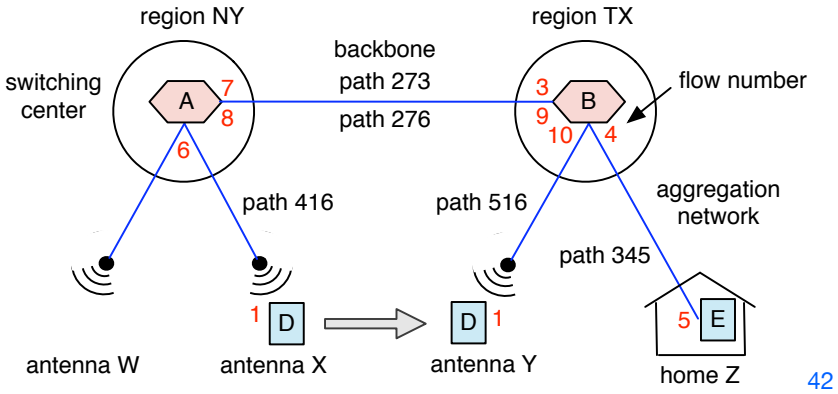


Figure xx1. Mobility can eventually lead to inefficient flow routes

Figure xx1 illustrates the situation. While D was transmitting through antenna X its flow was routed through A6, A7, B3 and B4. Then D registers with the agent B through antenna Y. In preparation for the coming transition, the agent for engine A is required to find new routes for all of the flows that D has established. The usual strategy is to initially extend the existing route(s). That is often a good choice because shortening a flow's route without warning can result in packet loss. Extending a route may be more quickly achieved by assigning a single path that serves the needs of all of D's flows that are impacted by the move. The more routes that have to be found the longer it will take to complete the move. A quick response can be important for a mobile device that does not have strong wireless coverage. So, for the present example, agent A chooses to extend all of D's flows through path 276 to B where path 516 leads to antenna Y. (Mobility also has an impact on the protocols used for congestion control. See (ref) for more on that topic.)

That reroute may have provided the least interruption of service for user D, but now the route of the flow between D and E is inefficient; the trip from B to A and back again is unnecessary. So agent B reroutes the flow as shown in figure xx2.

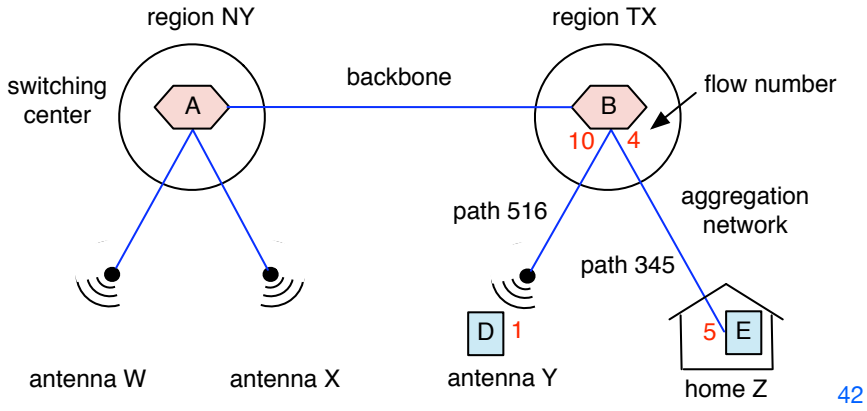


Figure xx2. Flow D-E after route optimization

Restoration procedures are also required in other parts of the network. It is necessary for load balancing, to circumvent failed transmission equipment, and to provide time for engineers to work on the plant. In these circumstances it is convenient, even necessary, to draw distinction between the two simplex parts of what the user generally regards as a single full-duplex flow or path. That is particularly true for the Internet and other modern networks that integrate voice, video and data in a single service. While telephony is for most practical purposes considered to be be two continuous and similar streams, that is not true for video and data communications. TCP file transfers are heavy in one direction and light in the other. Video streams are even more asymmetric, with some flows traveling in one direction only. So, for the purposes of traffic routing, to achieve optimal use of backbone transmission bandwidth, it is sensible to regard a flow as being composed of two simplex information streams moving in opposite directions. We say that such a flow is “dual-simplex”. However, from an Interlan user’s perspective, each flow is a single entity. So, in the Interlan language, all flows are full duplex. The language says nothing about the fact that backbone flows are routed along paths and backbone paths in particular, are dual-simplex. This contradiction is resolved by ensuring that the pair of simplex end-points which comprise one end of a path are never allowed to separate.

4.12 Mobility and portability

“Mobility” is a one-word reference to a host that moves while maintaining a connection. The word is most often used in reference to hosts that move at some speed while continuing to carry on a conversation. The challenge is not only that the host changes its location but also any conversation that is in progress needs to continue without interruption. A different challenge arises when a host is ported from one location to another (it may be asleep while being moved). In that case its new point of attachment to the network is not so easily predicted, and in the process the host may switch from using one network technology to using another.

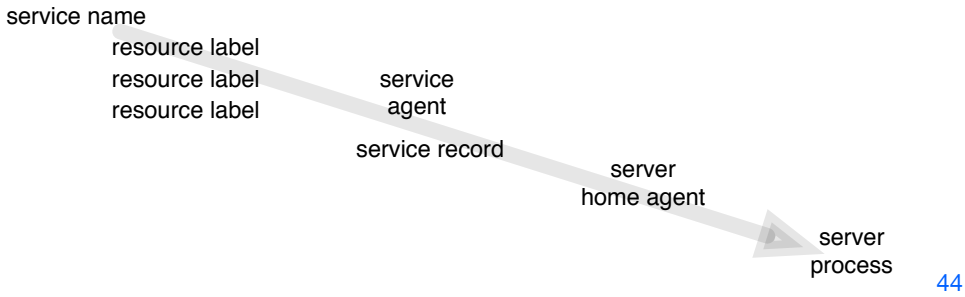
A goal for Interlan architecture is transparent mobility and portability for all hosts. Transparency means conversation continuity. For a portable host it may mean re-establishing the flow after a move. In both cases the host and network must confirm each other’s identity before communication resumes. In the case of a mobile host that can communicate with the new base station before breaking connection to the old base station, that handshake can be completed without interrupting the user’s conversation. Mutual authentication may use an abbreviated protocol which employs a shared temporary secret.

An Interlan’s use of agents parallels that which was pioneered for cellular telephony. At any moment, all communications to and from a given host are by way of an agent that is in the same region as a wireless base station within range of the host. That agent is, for the time being, the host’s “local agent”. When the host moves from one region to another it may acquire a new local agent. In order that the mobile host can be reached by others, one agent is permanently assigned the task of keeping track of the host’s present position. That agent is the host’s “home agent“. Usually the home agent is the host’s local agent in the host’s home location.

additional topics - mobile networks, paths as an aid to mobility, extending the flow via neighbor trunks, volume control and mobilty (picture).

4.13 Connecting to a mobile service

A service is an abstract entity, a type of resource defined within the Interlan language. The service resource serves as a rendezvous for clients and servers. The service name translates to one or more resource labels each of which contains the address of an agent.



A service agent keeps note of the home agent for each server that has volunteered to provide the service. When a client arrives, the service agent steers the client to a suitable server. Therefore there is really no such thing as a “mobile” service; the service is represented by a database record administered by an agent, neither of which is mobile. The term “mobile service” really is a reference to a server process in a host which is mobile. It is also possible for a server process to be encapsulated by a virtual machine that moves from one host to another.

From a service perspective there is little difference between static and mobile servers. All are processes that have made a listen( ) syscall that “connects” them to the service resource. From an Interlan language perspective, the connection between server and service has the properties of a flow between processes, but the service is not a process and the agent which administers it is located within the control network where payload connections are not allowed. So the connection between server and service is virtual and carries syscalls only. Syscalls stimulate a host operating system to generate syscall messages that are routed by way of a host’s control flow for delivery to an Interlan agent. The connection persists until it is closed.[10]

Figure xx illustrates the rendezvous procedure when a client connects to a mobile server.

[10] There is an argument in favor of giving each server a separate control flow that is established using the credentials of the server. That is not inconsistent with the application interface to the present design, but it does expand the security perimeter of the network operating system.

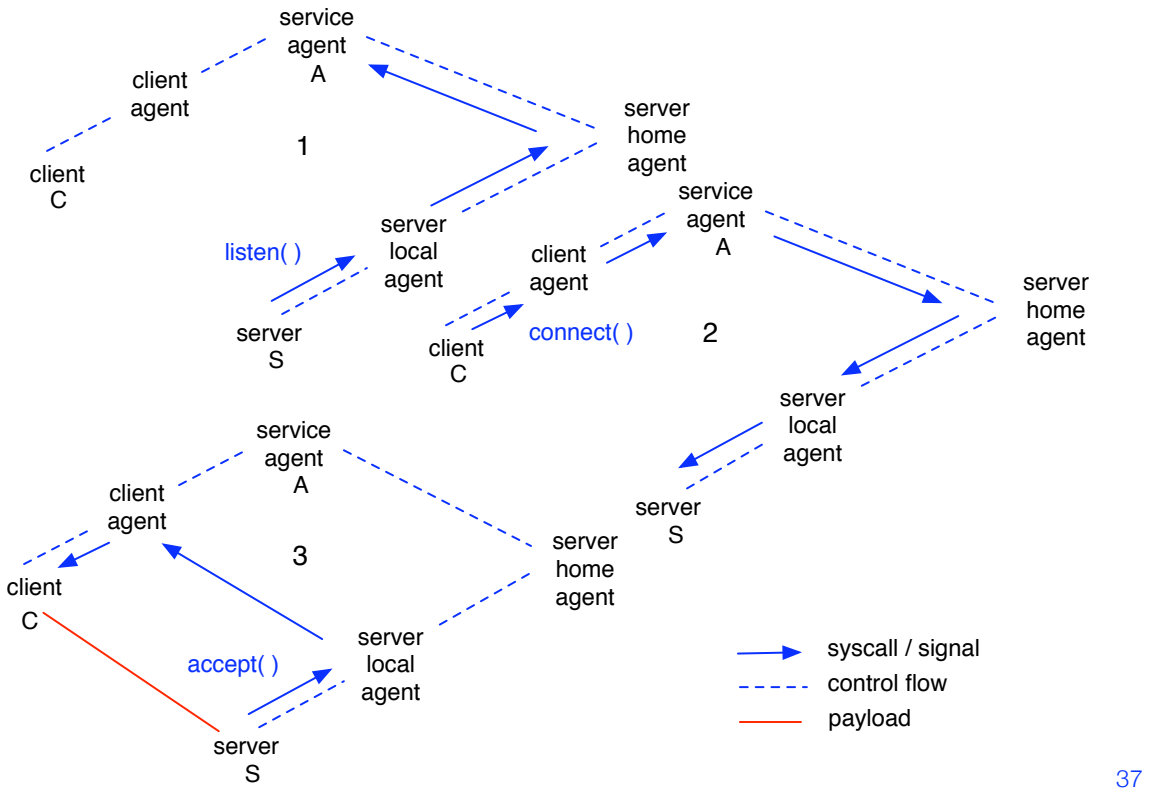


Figure xx. Syscall and signaling message flow when client C connects to a service S.

The rendezvous takes place in three steps: First, the server S offers to provide the service by sending listen(<service name>) to the service agent A. Second, client C wishing to receive that service sends connect(<service name>). That connect( ) message also goes to agent A. The fact that S has sent listen( ) and C has sent connect( ) is noted in that resource record. Agent A sees that C can use S, so A sends C's connect( ) message to S. Finally, the server S accepts the request by sending an accept( ) message to C. As the accept( ) makes its way through the network it finds the shortest route between S and C. (It is not necessary for the accept( ) message to take the longer trip that goes via service agent A.) The accept( ) moves from agent to agent as the route of the connection is developed. Concurrently, the agents configure a "provisional" flow between S and C. That flow is recorded in the forwarding engines along the route. A provisional flow has the property that C can use it to send packets to S, but S cannot transmit packets on that flow until S has received a first packet from C. This gives C the opportunity to check S's credentials before S can send payload to C. If C has a problem with those credentials it can disconnect the provisional connection.

A client requests connection to a service by issuing a connect( ) syscall. The client's agent first consults the name server to obtain the location of the service agent. The connect( ) is then routed by way of the service agent and the server's home agent to the server's local agent and hence to the server. Further interactions between client and server involve only their respective local agents.

For as long as a host stays in the same region as its agent, that agent and its forwarding engine will handle packets for the host. When the host moves to another region, an agent in that other region is appointed to handle the host's packets. So, in general, there is a local agent for the host that is in the same region that the host now occupies, and there is a home agent which is located in the region which the host considers to be its home. The home agent is also the local agent when the mobile server is at home. It is the job of the home agent to stay connected to the server's local agent when the server moves away from home. For that purpose each time that the server acquires a new local agent, that local agent and the home agent handshake with each other. Messages sent to the retired agent are routed via the home agent. It is this contact between agents that allows the local agent to gain access to the credentials which allows mutual authentication between client and server.

When a host leaves the network its local agent, if it is not the home agent, drops out of the picture. Any listen( ) flows for the host are disconnected. That does not mean that the service being listened on is not available. Other hosts may be listening on the same service. There is also the possibility that the name service has information about more than one resource record for the one service. Various strategies are available for use when a client requests to connect to a service that has multiple resource records, or when a single resource record has multiple servers that are listening. For example, the servers may be chosen in rotation, or a server close to the client might be selected. Other criteria in common use by other networks include the time of day and the work load which a server is currently handling.

When a mobile host comes online it must register with the network, which means that it makes contact with an agent in the local region. At first, a temporary agent is assigned, to find out which host is registering and to authenticate its identity. The host must oblige by providing an identifier, probably an account number, from which the temporary agent obtains the location of the home agent. Then, using credentials provided by the host, the temporary agent and home agent authenticate the host's identity, and the temporary agent acquires information about the host's service state just prior to disconnecting from the network. If the home agent is distant, a local agent is assigned, otherwise the home agent takes over. In order that there should be fast response when a mobile device temporarily loses contact with the network, or a laptop takes a short trip from one network connection point to another, the local agent will usually stay attentive after the disconnect, retaining the host's status for a period of time before dropping out of the picture. For the same reason there may be a fast re-registration procedure based upon a temporary secret that was provided by the local agent. See (ref) for a more complete discussion of authentication.

4.14 Service quality

It was twenty years ago that the Internet exploded onto the world stage. Much was then and has since been learned about the required service characteristics of a global network and how to provide them. But large scale deployment made evolution of the Internet's underlying architecture difficult. While innovation and research continue, the network has not kept pace. Now, the exponential improvements in device performance are coming to an end, and concerns about security are pressing strongly for change. Perhaps this is the time to benefit from what has been learned about service quality and take the necessary steps to catch up with the research program.

Telephony and video have unique requirements. Telephone use is interactive and users become dissatisfied when round-trip delay approaches 100 msec. Video is by far the highest bandwidth application that is ubiquitously deployed. While variable rate coding is available the advantage of statistically multiplexed video is small, so video is in practice seen as a constant rate flow. There is an interactive component to other networked applications but there is no single characteristic for their behaviors. Perhaps the most useful observation to make is that the resulting traffic has two components: Approximately 9/10 of the messages are short - on the order of 10 packets or less. The remaining traffic, accounts for more than half of the total bandwidth and is contained in messages that vary greatly in length. As with telephony, interactive data traffic requires low delay, while large message traffic requires high bandwidth in order that expected transfer completion time can be reasonably short. Sadly, a further dimension was added to the Internet's traffic profile when, rather late, it was realized that TCP needed some means of controlling congestion. The day was saved by Van Jacobson who proposed the now widely deployed AIMD protocol(s). The result is that TCP traffic now has its own characteristic behavior that constantly oscillates between high and low speeds. This "saw-tooth" traffic pattern makes it difficult to obtain high bandwidth utilization, interferes with video and other traffic flows that prefer a steady rate of service, and makes routing more complex than it otherwise would be.

Early attempts to control data transfer were concerned with preventing a sender from transmitting faster than the receiver can receive. Window flow control solves that problem but causes traffic jams in a wide area network if additional measures are not taken. Circular dependencies between controlled flows that share queue space lead to traffic deadlock. A synchronous underpinning, enabled by carrying traffic in small fixed size cells looked like being the basis for a high quality service. Unfortunately, host architecture does not fit well with paced delivery of small packets, and best use of host interface bandwidth is obtained when large messages are sent in large packets. However, it was in the study of traffic control for cell networks that rate control was shown to yield beneficial results. Rate control, as originally conceived, requires a traffic

source to pace its transmissions so that there is a controlled gap between one cell and the next. More recently it has been shown that rate control can work in packet switched networks even when message size varies greatly.

Many of the problems mentioned above go away when host interfaces and network backbones are fast enough. So there has been a strong focus on increased transmission speed. That complements the very successful escalation in host performance for the past 4 decades. But there is one more bottleneck - relatively slow upstream flow in the recently upgraded aggregation networks. Upstream flow is technically challenged because there is a need to quickly resolve contention between the homes that simultaneously want to transmit. (Switched optical networks do not have that problem but network operators baulk at the cost of providing power and maintenance for switches in the outside plant.) It will probably be a while before that problem goes away. Meanwhile, it would be wise to plan for that event.

Four decades have elapsed since the objective of a single multi-purpose network was conceived. One may conclude from that experience that the technical challenge of providing good and appropriate service quality for all applications cannot be under estimated. An important contributor to the difficulty has been the challenge of upgrading millions of hosts while the network continues to operate. So perhaps it is time to accept the need for fundamental change. Bite the bullet and see whether the research insights of recent years can provide the service quality that society needs now and certainly will require in the years to come. While there are as yet no large scale multi-service networks that provide high performance low delay service in a non-congesting wide area infrastructure, there are research results that point the way. The following paragraphs outline the best plan that we can derive from current work. The plan relies heavily upon the work of Katabi[11] [12] and Dukkipati[13] [14], and their respective colleagues.

4.15 Congestion control

In a city, traffic lights provide the first defense against rampant congestion. Traditionally the lights are placed right at the road junctions where congestion occurs. Today, by various means, motorists are warned about road conditions before they enter the troubled area, so that they have time to react and stay out of trouble.

Traffic control in packet switched networks is now evolving in a similar way. At first, the focus was on queues in the switches and routers located where traffic flows intersect. Here the capacity of an outgoing transmission line is shared among the incoming traffic flows. A queue builds up and traffic may be discarded when the transmission line is saturated. But there is not much room for alternative strategies in such a confined space, and every discarded packet represents profitless work done by the network. So a family of control systems has evolved that send feedback from bottleneck nodes, through the network to the traffic sources. In that way a source is persuaded to hold back traffic that would otherwise cause congestion. Packet loss can thereby be avoided. However, the distance between the point of congestion and the point of traffic control puts significant delay into the control loop. The potential exists for instability leading to oscillations that impede the packet flow. It is in this context that rate control has found its home, but the parameters of the feedback signal must be carefully chosen.

The following paragraphs describe two forms of traffic control with feedback. “Rate control” applies to long distance communication, and “volume control” is used in the upstream path between user premises and the nearest regional center. The combined system has several benefits:

It evens out the backbone traffic flow,

---

[11] D. Katabi, M. Handley, and C. Rohrs. Internet Congestion Control for High Bandwidth-Delay Product Networks. In SIGCOMM, 2002.

[12] Srikanth Kandula, Dina Katabi, et al., “Walking the tightrope: Responsive yet stable traffic engineering”, SIGCOMM 2005, Philadelphia.

[13] Nandita Dukkipati, "Rate Control Protocol (RCP): Congestion control to make flows complete quickly", Ph.D. Thesis, Department of Electrical Engineering, Stanford University.

[14] Nandita Dukkipati and Nick McKeown , “Why Flow-Completion Time is the Right Metric for Congestion Control", Computer Communications Review, Volume 36, Issue 1, January 2006

It regulates that flow so that queue overflow will be rare,

It avoids the need for significant change in host-network interface design,

It protects the network from attacks launched by way of the congestion control system, and

It creates a framework for multiple classes of communication service.

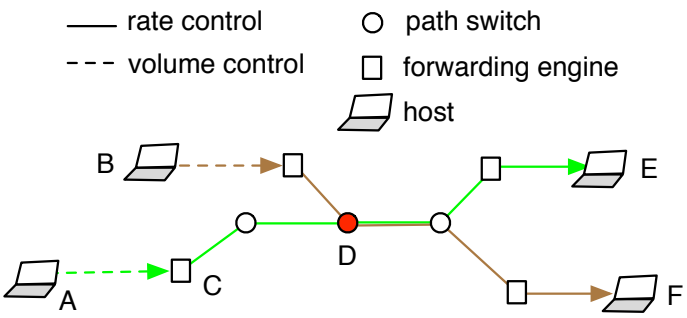


Figure xx. Combined volume and rate control

Figure xx illustrates two flows, one from B to F and the other from A to E. The traffic passes through three path switches that represent the network backbone. Traffic from A and B converges on path switch D, so it is the output port of D that is most likely to be congested. Dashed lines in this figure indicate that the upstream flows in the two aggregation networks are regulated by a volume control protocol, and the solid lines indicate that backbone and downstream flows are subject to rate control. Forwarding engine C contains a queue that sits at the junction between the volume and rate controlled sections of the flow between A and E. Rate control regulates traffic in the bulk of this network, and it is the responsibility of engine C to regulate the flow from A so that the flow moves as fast as it can while avoiding queue overflow.

4.16 Controlling backbone flow

It may seem strange to say this, but a network payload seems to behave like an incompressible fluid that has momentum once it starts moving. Any attempt to stop, or even retard the flow, leads to a growing heap of data that is typically a burden to handle. It is far better to treat that traffic flow as a fluid that must keep moving, and modulate the rate of flow so that the transmission system is not overloaded. However, if the traffic is moving in a constrained environment where the constancy of a flow is in doubt, then it is better to count the packets and be prepared to stop at any moment. Rate control protocols do the former and volume control does the latter. Both are employed in the Interlan architecture for traffic control.

Paths define the routes which are available for Interlan traffic flows to follow. A backbone path spans the distance between one regional switching center and another. Each backbone network has its own network of paths wherein there is at least one such path between every pair of regional switching centers. In most cases there are many alternate paths which might be followed to reach a given region. The experience of Internet service providers coincides with network operating practice used in the AT&T telephone network[15]. The path structure is chosen by an offline process which makes use of information about network topology and persistent traffic patterns. Once a set of paths is defined, that set becomes the framework within which traffic is routed.

Several paths may share a transmission channel which today is most probably a color of light in a fiber or a radio frequency in a wireless system. Channels are interconnected with path switches so that the path network can be reconfigured when necessary. The price of using shared channels to implement the path network is potential congestion in the switches. So the rate at which traffic is flowing through the switch egress ports is constantly monitored. Measurements thereby obtained give rise to advice that is constantly being fed back to the points at which that traffic enters the network.

---

[15] Srikanth Kandula, Dina Katabi, et al., “Walking the tightrope: Responsive yet stable traffic engineering”, SIGCOMM 2005, Philadelphia.

In the design of rate controlled packet switch networks it has been customary to send that feedback directly to network hosts by way of extra bytes in payload packets. It is as though the traffic police hand messages to drivers for delivery to toll-booth attendants whose job it is to regulate the incoming traffic flow. Propagation delay in this control loop increases as congestion builds and one must hope that upstream and downstream flows do not both become congested at the same time. A more pernicious concern is that people with contrary interests can cause chaos by interfering with the message flow.

Present-day switches and routers use electronics for the transmission path as well as for control. However, there is an emerging fleet of optical switches that will surely take over on major transmission routes. In that case it is not so easy to put feedback into the payload flow. Electronic circuits have a hard time keeping pace with the optical signal, and there is no practical form of optical memory. A separate, possibly narrow-band channel for the express purpose of carrying feedback is recommended.

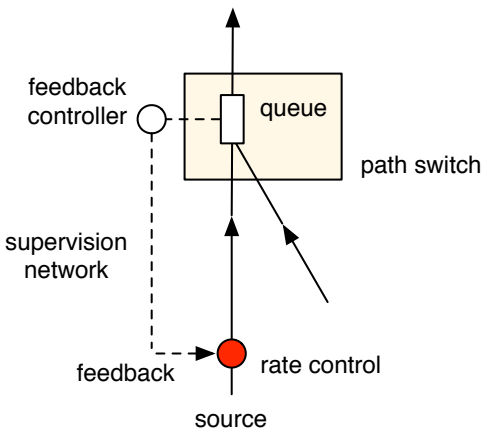
The Interlan supervision system is a narrow-band network designed to carry measurement and status reporting messages. The system touches all critical points of an Interlan infrastructure. The purpose is to provide a low delay monitoring system which provides aspects of the network operating system with time-critical information about how well the network is working. The supervision network also carries critical control messages whose prompt delivery is essential if rapid service restoration is to be guaranteed when there is a network problem. Sonet and other transmission systems have channels of this type built into them. It is unclear why that practice is not a prominent part of Internet design. For as long as the Internet rides on a patchwork collection of transmission facilities that were once designed for telephony, there may be sufficient framework for high quality service. But eventually it will become clear that an architected network-wide supervision system is essential for first class service to the customer. Flash crowds, denial of service attacks and the threat of cyber-war make that design philosophy even more important.

There is one branch of the supervision network in every path. Feedback messages generated in path switches are propagated through the supervision network to forwarding engines at the path ends. The feedback is filtered because the engines need only know about feedback from bottleneck nodes. Then the feedback is replicated so that it reaches all the relevant path forwarding table entries in a regional switching center.

The supervision system is more than a network; it includes computation and supportive databases. Its ongoing task includes monitoring the performance of network nodes and keeping track of configuration changes. From these data the routing system derives its understanding of network topology and develops a profile of traffic levels within the network. Thus routing and congestion control are inter-related and can be coordinated.

4.17 Rate control

XCP and RCP are rate control protocols designed for use with the Inernet. In an Interlan network there would be a feedback controller in every path switch. That controller measures the queue size and traffic volume as payload packets pass through each egress port on a switch.



47

Figure xx. Path switch with feedback control and rate controller

In an Internet application the feedback controller collects information that is carried in each payload packet. In particular, XCP packets carry the sender's current congestion window, the round-trip time for the flow and a feedback signal. Feedback is for rate controllers in hosts and other traffic sources. The feedback tells them when they can speed up and when they must slow down.

Feedback for a rate-controlled network is generated by a periodic process with a time interval that is approximately equal to, and not less than, the average round-trip time for all flows which pass through a network node. For an Interlan network, the nodes are path switches which populate the backbone networks. The following is a description of the computation which takes place during each time period.

Measurements made during the "current" time period contribute to the feedback quantity Φ that represents the anticipated spare bandwidth (bits/sec) at the end of the "next" time period. If the network is overloaded, Φ is negative.

$$\phi = \alpha \cdot d \cdot S - \beta \cdot Q$$

In this expression d is the length of the measurement time period, S is the spare bandwidth defined as the transmission line bandwidth minus the input payload rate measured during the current time period, and Q is the volume of data held in queue at the end of the current time period. The two constants α and β are chosen so that the feedback system is stable.

Katabi and Dukkipati have studied two different ways in which the estimated spare capacity might be shared among all flows which are routed through a bottleneck node.

(a) "AIMD fairness" - If $\Phi \geq 0$ feedback authorizes an additive increase in a source's current transmission rate, and if $\Phi < 0$ the source must multiplicatively reduce its transmission rate. The amount of increase and the size of the multiplicative decrease must be scaled so that the sum of the changes in flow rate for all flows does not exceed Φ.

(b) "processor sharing" - Let R' be the feedback sent to every source during the previous time interval, and let N be an estimate of the number of flows whose sources are currently operating below their maximum capability. If extra bandwidth is available these N flows most likely will use it. So feedback equal to R is computed as follows.

$$R = R' + \Phi/N$$

This value is sent to every flow, and every flow must limit its future transmission rate to that feedback value. The estimated value of N for the current time period is equal to the ratio of the transmission line bandwidth C to the feedback R' that was transmitted to each source during the previous interval.

$$N = C / R'$$

These two methods of traffic control yield different feedback values, yet both yield a result that is approximately max-min fair[16]. The processor sharing method is preferred for Interlan networks because it is known to give close to the minimum flow completion time, and the computation of rate adjustments is by far the easiest to implement in a distributed system. AIMD is difficult to implement because computation of the required scaling adds significant complexity to the feedback process.

The processor sharing method has one other significant advantage: fast startup. During the initial connect( ) / accept( ) handshake which launches a new flow, the Interlan operating system chooses the route and negotiates the first transmission speeds that the rate control system will support on that flow. Arguments of these two syscalls include for each of the client and server the rate which they would each like to use for transmission, and the maximum rate of input that each can handle.

<flow> = connect(<name>, <desired transmit rate>, <max receive rate>);
accept(<flow>, <desired transmit rate>, <max receive rate>);

[16] A capacity allocation is max-min fair if an attempt to increase the allocation of any flow necessarily results in a decrease in the allocation of some other flow with an equal or smaller allocation.

To compute the authorized rate of transmission in a given direction, take the minimum of the three values which are obtained from connect( ), accept( ) and rate control feedback for the route of the flow.

max flow rate = minimum(desired transmit rate, max receive rate, feedback rate)

Once the client has received accept( ) it can begin transmitting at its maximum flow rate. There is no slow start for an Interlan flow.

4.18 Controlling upstream flow

The need for volume control on the upstream flow from a host into the network stems not only from the limitations of the host-network interface but also from the risk that the network can be attacked from a host if the host has a trusted role in network operation. The host must not be relied upon to do what is necessary to make backbone congestion control work properly. So it is proposed that an Interlan backbone network shall have a self-contained congestion control system (as described above) and that separate arrangements will be made for controlling upstream traffic in aggregation networks. This architectural choice has the further benefit that the backbone congestion control system can be contained within the protective framework of the network operating system. The method allows rate control where it is most important - in the backbone networks that everyone shares. And it does this without forcing an upgrade in the host-network interface for devices that would use the network.

A rate control system might be a satisfactory means of upstream traffic control if hosts were equipped with suitable host-network interfaces. There is an additional complication introduced by the present-day broadband aggregation networks (EPON and DOCSIS particularly). Those systems use a request-grant protocol to resolve contention among competing homes. This protocol destroys the carefully timed transmissions that characterize rate control. So Interlan networks use a simple window protocol which operates on a per-flow basis between a host and its forwarding engine. We use the term “volume control” when referring to this mechanism because, in contrast to rate control, this window-style protocol allows the host to transmit as fast as it can while limiting the volume of payload that at any moment is held in the forwarding engine’s input queue. This leaves the forwarding engine free to control the rate of flow in the backbone. At this point in time it is less expensive to implement rate control in a forwarding engine than look for changes in host-network interface design and aggregation network technologies.

Fortunately, the forwarding engine is the natural place from which to control backbone flow rate. The engine is a hardware device designed for high performance packet queuing and header processing on a per-flow basis. The rate control logic needs to have those same qualities. The engine is on the boundary between the well-protected backbone and the more vulnerable outside plant which leads to user premises. Furthermore, there sits behind the forwarding engine the massive protected computational capability that is home to network agents and other aspects of the network operating system. The engine is therefore securely connected to all sources of control information.

The forwarding engine is the natural home for the logic which supports volume control. The same high performance queue space which holds payload waiting for output into the backbone, can also provide the queue space for volume controlled upstream traffic. As with connection management, the forwarding engine does the high speed part of the job and the rest is left to the agents.

It is evident that whatever support a host provides for the volume control system, that support is going to be in the form of software, either in an Ethernet driver or some other low-level part of the host operating system. The hardware support for data transmission will remain as it is today - in essence, an Ethernet chip.

(As an aside, we note that Ethernet lacks a suitable means of controlling high speed flow between hosts on a LAN. Conceivably, the same mechanism which underpins Interlan volume control can also serve a useful purpose between one host and another. Therefore, we present a symmetric protocol, even though the wide area network requires volume control for upstream traffic only.)

The protocol is reminiscent of the Universal Receiver Protocol in that traffic control strategy is in the hands of the traffic source, and each receiver provides a few generic functions that can support a range of host-network interactions. It is an

asymmetric window management protocol where the transmitter does most of the work. It is light weight in that it is concerned only with flow regulation so that the receive queue does not overflow. All negotiations (if any) concerning the start and cessation of flow, the assignment of queue space, the maximum packet size and where the traffic goes after it leaves the queue, are implemented in higher-level software. In an Interlan network, these matters are addressed by the use of syscalls.

A forwarding engine is built around an array of high bandwidth queues. As each packet arrives at the engine, the header is examined, the forwarding process is completed, and a pointer to the stored packet is moved into the appropriate output queue. During this time of header processing, the engine examines the packet's volume control field and implements the volume control protocol functions.

The volume control protocol operates on a per-flow basis. There is no support for packets arriving out of sequence or getting lost. If those services are required, the application is advised to use an end-to-end transport protocol at one level higher than this link layer protocol.

4.19 Volume control protocol

The source and destination flow labels in an Interlan header have space for a 24-bit flow number and an 8-bit control field.

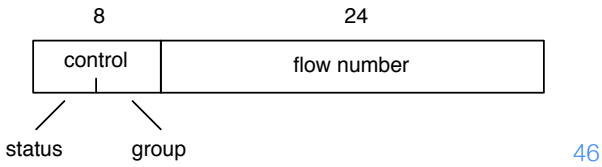


Figure xx. Flow number with flow control field

The volume control protocol uses this control field to prevent queue overflow when transmitting payload in either one or both directions on a full-duplex flow. For the Interlan application, this protocol is used only for upstream flow out of a host and into a forwarding engine. Commands and responses are carried in the control field. "Commands" travel in the destination flow labels of upstream packets, and "responses" travel in the "source" flow labels of downstream packets.

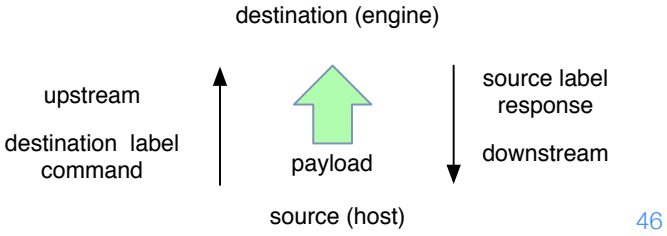


Figure xx. Control of an upstream flow from host to forwarding engine

At the destination end of each simplex flow there is a queue and a status register, VSR, that contains queue status information. The content of this register is transmitted in the responses that a source host receives.

Figure xx illustrates the upstream flow of payload from host to engine and the corresponding flows of control information.

Payload from a host travels upstream into an engine's queue memory where a certain amount of space has been assigned for use by this upstream flow. When the engine has done its work and when backbone rate control allows, the packet is removed from the engine's queue memory and sent on its way.

Every Interlan flow is full-duplex. It is the upstream flow into the network that is subject to volume control. (Downstream flow is rate controlled.) So, queues in the forwarding engine come in pairs - one for the upstream flow and the other for the downstream. These two queues are coupled in the following respect: Each queue has a status register VSR. In order that the host which is sending traffic upstream can learn about that queue's state, packets traveling in the downstream flow obtain copies of the upstream VSR as they head towards the host.

Upstream flow is divided into packet groups; group size being determined by the host. A group sequence number N is the basis of upstream flow control. Each upstream packet's command field contains both a command code and the number of the group to which the packet belongs. These two values are carried in an upstream packet's destination flow label. The packet sits in the forwarding engine until it is time for it to be transmitted into the backbone network. At that moment the upstream VSR is updated (as will be described). When a packet traveling downstream in the same flow leaves its queue and heads towards the host, a copy of VSR from the upstream queue is written into the downstream source flow label. Thus, VSR is delivered to the host for use by the upstream transmitter.

The command field contains one of the following code values:

0	payload not subject to volume control
xmt	payload subject to volume control
query	query the status of the receiver
reset	reset the receiver's queue

Query and reset prompt an immediate response containing the current VSR content. The response to an upstream payload packet depends upon the state of the engine's upstream queue. If the queue is full the payload is discarded, the queue state indicates that overflow has occurred and an immediate response is returned. Otherwise the packet is placed in the queue and response is delayed until there is a naturally occurring downstream packet in which the response can be carried.

There are two fields in the VSR:

VSR.status	Status information relating to the engine's upstream queue.
VSR.group	The group number from the packet most recently transmitted out of the engine's upstream queue.

The value contained in VSR.status is one of the following.

0	null response
empty	queue is empty
ready	queue can accept more payload
full	queue is full
ovr	queue has overflowed
clr	queue has been reset

Parameters which govern upstream traffic flow may include the maximum queue size, maximum packet size, maximum group size, etc. These properties of the flow may be negotiated but that negotiation is not part of the volume control protocol. In the case of upstream flow from an Interlan host, parameter negotiation is handled by syscall.

The strategy for using this mechanism is quite simple. The transmitter chooses a window size and maximum packet size that are acceptable to the network (receiver). The receiver assigns queue space equal to the window size. These sizes are not restricted by the volume control protocol, but the protocol allows only 15 packet groups in a window. Group size is a number of packets which is equal to or greater than 1/15 of the queue size. The transmitter sends packets for as long as its calculation indicates that there is queue space available in the receiver.

All packets contain the current group number. When a group has been sent the group number is incremented. The transmitter can determine from the values of VSR the approximate volume that remains in the queue. It is the transmitter's task to pause before the queue overflows. The receiver will discard a packet if its reception would overflow the available queue space.

Along with the pair of flow numbers and Ethernet addresses, the Interlan packet header contains four control fields - one associated with each source flow label and the other with each destination flow label. These two fields carry commands and responses between host and forwarding engine.

Therefore there is sufficient information in an Interlan flow header to support one volume control protocol per flow, or a single volume control that administers queue memory for multiple flows. There can be a separate queue for each flow, or they all can share one queue. The protocol is agnostic with regard to transmission rates and window sizes. For most Interlan users,

the distance to a regional switching center will be no more than about 100 miles. Round-trip time in the aggregation network (with no allowance for aggregation network phenomena) will be less than 1.2 msec. So a queue memory size of about 300 KBytes will support 2 Gb/sec continuous transmission. The host can transmit in large bursts and the engine will empty the queue at the controlled rate of the backbone network.

The combined presence of volume and rate controls means that host software is isolated from much of the complexity that is required to achieve a balance between high performance and a stable traffic flow. The upstream rate of flow is constantly adjusted so that best and fair use is made of the network's bandwidth. Volume control makes sure that the source of a flow does not go faster than rate control allows. So the application process has no direct involvement in rate control. When circumstances change the feedback controller and the agent make necessary adjustments which lead automatically to a change in the rate that the forwarding engine enforces. That in turn leads to an adjustment in the behavior of volume control and a consequent change in network performance as seen by the application process.

4.20 Negotiated service quality

There are situations which will cause a network user seek a change the parameters of the service that he is receiving. That will usually happen at the start of transferring a new document, or at the launch of a new application. In that case the new service parameters can be expressed when using a connect( ) syscall. The request for a specific flow transmission rate has already been mentioned. In that case the desire for a certain transmission rate and specification of the maximum input rate are the beginning of a framework where client, server and network can jointly negotiate the parameters of a new connection. In the past, there has been no dynamic framework within which to express the needs of an application or the preferences of a user. Interlan syscalls are seen as having the flexibility to provide such a framework.

There are some applications for which best effort service is not good enough. With rate control and the organization of backbone traffic in flows and paths, it becomes possible to respond to special needs. For example, high quality real-time video demands a high bandwidth that is continuously available. The quality of video reproduction may be effected by other traffic (such as TCP) which has a very dynamic behavior. The problem can be solved by choosing separate routes for these two traffic types, or at least by giving the video higher priority than TCP.

In general, specification of service parameters for a flow should be made before a route is assigned to the flow. That implies that connect( ) and accept( ) are appropriate syscalls in which to express particular service needs. However, that is not always the most convenient time to make the request. For example, a connection is established to a service for which there are access controls. Then it is required to send a video to the server. The connection that is in place was not established with the appropriate service parameters, and using connect( ) to establish a new connection will force the user into repeating the service's access controls. That may be at least inconvenient if not impractical. If there are many servers for the one service, making a new connection will not necessarily lead (at first) to the required server. While that problem can be worked around it is wasteful in user time and server resources.

In this situation the client needs to be able to leverage the fact that his credentials are already established as evidenced by the connection that he already has in place. Hence the need for clone( ), a syscall that leverages the existence of one flow to create another. The syscall

fn = clone(fc, <service options>);

creates a new flow between the same two processes that are already interconnected by the flow fc. While the endpoints of the new flow are the same as the existing flow, the route and service features may be different. Once accept( ) has been received, the flows fn and fc become independent of each other. Either flow may be terminated by means of close( ) and the other will not be effected.

By developing an appropriate catalogue of service options a network operator may be able to improve his service even to those users that do not use service options. For example, the time may come when the TCP saw-tooth characteristic is considered to be an unsatisfactory traveling companion for a wide range of normal services. So TCP as a service option would allow it to be routed in a protected environment, thereby relieving other traffic from its impact.

Similarly, if a user has a very large file to download and his aggregation network has limited bandwidth, it might be helpful to schedule the transfer for a time when the aggregation network is lightly loaded. An option which specifies the size of the file

and the willingness to have the transfer delayed, would avoid dragging down the quality of service to other users. The same feature might be used in another way, to reduce network operating costs. A backbone subnet constructed with dedicated paths might be devoted to large document transfers. It may usually be practical for users of this service to specify the (approximate) size of the document (it is probably already in the file system), then the network can tightly schedule the subnet bandwidth without risking complaints from interactive network users.

Bandwidth and routes are two of a network operator's resources that can benefit from the flexibility that comes by way of the syscall interface to network users. Rate control is a realistic option for the network when accompanied with volume controlled access. In combination these resources can be used to raise the quality of service for network users and to also make better use of the network's facilities.

4.21 Flow implementation

In outline, the plan is to create a global Ethernet backbone that can concurrently carry two generations of traffic: Internet packets and Interlan flows. Then, create a framework in which evolution can take place based upon the dependencies that one person has with another rather than attempting large scale coordinated transitions involving large groups of people.

The Interlan architecture has sought to raise the intellectual level of network service, just as compilers lifted programming out of the detail of assembly language programming. The concept in outline was described in Chapter 3. The focus now is to understand how it can be implemented. The priority items in the implementation of this vision are:

Globally unique address of sufficient size and a plan for long-term growth.
Agents with resource records linked to the name space.
Syscalls as a manifestation of a higher level view of networking.
Registration of users and hosts, and access controls to improve network security.
A forwarding engine that instantiates a higher quality of service.
Partitioning of architectural components so that there can be incremental evolution.

The originally priority for this program of work was to show that these things could be achieved without requiring any change in host software. That is an achievable result and it is described in this report. But we judge that it cannot yield a satisfactory outcome in the long run. Too little of what is worth achieving would have been attained. It will take some years to develop the technology that will solve today's problems, and in that time at least some progress can be hoped for in the matter of host software, even if it does leave some equipment locked into older protocols.

Switching center

A regional switching center is at least as much a computer system as it is a switching system. The two systems may be roughly equal in size. A clustered computer system spanning multiple racks is supported by a large storage array. That is adjacent to clustered racks of forwarding engines, a high performance Ethernet switch, and racks of line cards. Trunk and aggregation network cables feed into the line cards. The general idea is to keep the line cards as simple as possible because that part of the physical system is more prone to failure than the forwarding engines and processors. By placing the high capacity computer complex adjacent to the forwarding engines and away from the clutter of fiber terminations, it is intended to create a close coupling between engines and computers. That configuration allows engine architecture to focus on that which custom hardware does well. It leaves virtually all of the computing load to generic computer systems. Figure xx illustrates the concept.

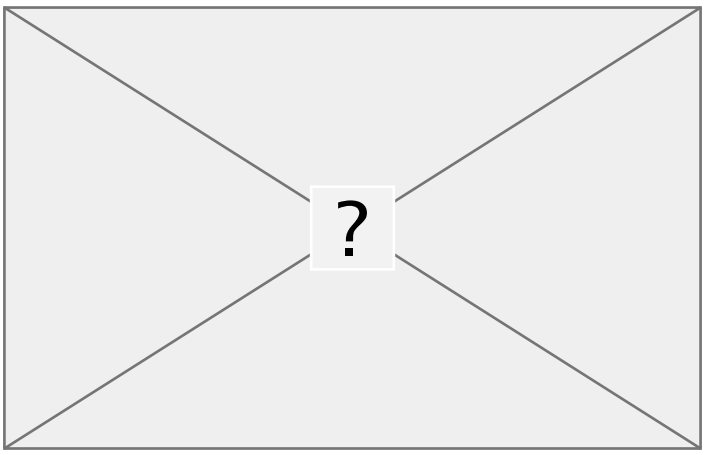


Figure xx Switching center concept

Packets enter and leave through the root and trunk line cards. Each packet's destination address is the Ethernet address of an engine. Address translation has a built-in function that makes sure that all incoming traffic goes to a forwarding engine, rather than another line card. This location of the switching network allows traffic to be diverted from one engine to another in order to permit some specialization among engines and to support load balancing and fast service restoration. That part of the figure which is drawn in red represents the physical assets of the network operating system. It is a computing complex with tight coupling to the forwarding engines. In practice, agents are processes (not processors) that can be relocated as necessary in the computer array. The control network is a message switched system.

Each host has a control flow that connects to an agent which is its interface to network operating system functions. Nothing can be done on the network until a control flow has been obtained.

Obtaining a control flow

When a host comes online it should immediately acquire a control flow. That can be done by using DHCP to obtain the Ethernet address of the "registration agent". That agent interprets a packet addressed to its port zero as a request for a control flow. The source flow number in the agent's reply is the agent flow number to be used in all communications with that agent.

The format of the packet which requests a control flow indicates to the agent what protocol the host will be using. If the packet has an Interlan header, then the host is expected to use Interlan protocols. If the request packet is a plain Ethernet packet, then the host is expected to use IP protocols. The first task for the host is to register, which means that it must claim and authenticate a host identity. Then the user must login. (Perhaps both can be achieved in one action.)

Registration is handled in the host by an Interlan deamon. That daemon remains active so long as the host is active on the network. (If the daemon dies and is not restored the host drops off the network.) The daemon's task is to represent the host operating system when communicating with its agent. In particular the deamon is the process at the end of the control flow.

The daemon is a user-space extension of the host operating system. It relies upon certain privileges granted by the host operating system. Among them is the ability to use certain system calls which relate to the administration of flows. To illustrate how that works we describe the implementation of connect( ).

Connection establishment

The "socket system" is the most widely used communications interface for host operating systems. Variants of that system appear in all popular operating systems. The socket system coordinates communication between processes across a network. Each process acquires a "file handle" which, when attached to a socket, allows the process to launch or receive a network connection. The socket system assigns a "port number" to the local end of a connection. The client and server port numbers appear in every Internet packet. Interlan syscalls connect( ) and listen( ) correspond approximately to socket system functions with the same name.

The socket system ordinarily operates in terms of IP addresses. For user convenience the system provides a means of using the network's name service to translate a host name into the host's address. However, most of the functionality is under the surface, unseen by most users. It includes such things as preparing and checking protocol headers, assigning buffers in which to temporarily put incoming and outgoing packets, and operating the interface to the hardware device which sends and receives packets on the Ethernet.

The key here is that the agent's flow number is decided up front before any payload packets are sent
so there is no "trick" such as decoding packet mismatch errors, and all is done by table lookup

Within a host, Interlan flows are made to resemble Internet connections which means that Interlan users can benefit from the socket system functionality. But, of course there are limits to how much commonality can be achieved that way. The Interlan name system has much more functionality than the Internet name service, and the socket system does not know about Interlan access controls. Nor does it know how Interlan flows operate, and how congestion is controlled. These are important differences which reflect the value that Interlan architecture brings to network users. So a method of working around these differences has been developed. The Interlan software strategy is to add Interlan capability to a host without disturbing its implementation of the socket system.

The Interlan daemon plays a key role. When a host application process does connect( ) on an Interlan network the daemon generates a connect( ) syscall that includes critical information extracted from the operating system. That information includes, the user ID of the client, the source port number, the local Ethernet address for the new connection, the name of the resource that is being connected to, and the transport protocol that will be used. This information is sent to the Interlan agent by way of the control flow. As the agent processes this information it translates the resource name using the Interlan name service. An agent for the server evaluates any access controls. That may involve interaction with the client in order to acquire credentials. If that goes according to plan, an Interlan flow is established. The client host receives confirmation of the connection along with various useful information including destination port number, Ethernet address, IP address (if any) and resource name. The daemon uses one of its special system calls to transfer that information into the socket system, so that the socket system can complete its part of the connection setup.

If the connection uses IP protocols it can now proceed as usual for communication with an IP host. The packets will travel within the new Interlan flow, which means that the conversation will be more secure than it might have been. If the client uses Interlan protocols, the socket system will implement volume control on the new flow. These aspects of an Interlan flow are transparent so far as the application is concerned. Volume control means that transmission will be regulated to stay within the available capacity of the network and will be in accordance with rate control in the backbone.

Interlan evolution

The Interlan system is design to evolve through multiple levels of technology. Ethernet is the core standard around which the design revolves. Hosts connect to Ethernet LANs. Initially that connection will use 64-bit flow labels. An upgrade in the host operating system will allow 72-bit flow labels and will enable volume control in the aggregation network. Forwarding engines support both 64-bit and 72-bit labels. Rate control does not depend upon host software. For those hosts that cannot be upgraded it is possible to avoid the need for any operating system change, but there is need to install a software daemon.

need to explain this in more detail, also explain the three forwarding engine technologies (Interlan, IP and Ethane) See 4.27

Non-upgraded hosts in an Interlan network

IP over Ethernet can be switched in a forwarding engine that swaps port numbers. Is this a way in which non-upgradeable hosts can join in an Interlan network. If the host can install a daemon it can get an account and register. Otherwise it would be a breach of security to allow the host on the network.

One can conceive of a non-upgradeable operating system that installs a daemon and so obtains an Interlan account. The daemons can send signals that create a tunnel without telling the host OS anything about it. That now is a virtual private flow. Then they direct the socket system to set up a connection that uses the appropriate port numbers. During the first

step the daemons are talking through their control flows. During the second step the action is entirely with in the two operating systems. For the user, this is just another form of connection establishment.

Therefore it is possible to have IP over Ethernet with port swapping in a network of Interlan users. The hosts have Interlan daemons but no host operating system upgrade. All that is required is that there be syscalls for setting up a tunnel without using the tunnel in the setup protocol.

4.22 Garbage collection

A flow is open when the exchange of syscalls, connect( ) and accept( ) has been completed and the client has chosen not to reject the server's credentials as provided in the accept( ) message. The client's first payload packet enables the flow for which a route has been established. Receipt of this packet should be taken by the server as confirmation that the flow is open. However, it is not always so simple to ensure a reliable start to a conversation, particularly in networks that do not carry connection identification in the packet header. In general, there are three sources of difficulty.

a) Packets left in the network after one conversation finishes can be mistaken for packets that might be used at the start of the next conversation.

b) Transmission errors leading to packet loss can add confusion at the start of a conversation. The two applications may not properly understand the state of the protocol at the other end of the connection.

c) An application may terminate prematurely or a host may crash leaving its partner confused when packets start to arrive from another host looking to start a new conversation.

It is not just a question of the probability that one of these events may take place, but there is also the risk that an attack might be based upon a host's vulnerability in these cases. Therefore it is important that each conversation shall start with a clean slate and the two participants should be robustly synchronized. It is also important that there should be no residual left in the network from a conversation that has been completed, even when the conversation ends prematurely as a result of error or a host becoming disconnected from the network. Therefore, each forwarding engine resets the queues that are to be used for a new conversation, so packets cannot be inadvertently carried forward from one flow that has terminated into another flow that is just starting up. Then each connection begins with the 3-way handshake conducted by the agents That protocol is implemented in the control network.

When a conversation ends, the hosts are expected to use a disconnect( ) syscall. However there are many reasons why that may not happen, so the Interlan daemons take part in a periodic garbage collection process that gathers up abandoned and broken network connections. This process that is initiated by the local agents. Daemons keep records of the flows that are created and deleted. These records are checked against the state information held in each host operating system. Agents compare the status of the flow labels that are presently assigned for use in a given path with the records that are held by the daemons. Discrepancies lead to some flow segments being closed.

These tasks are assisted by three facts: (a) flow labels are globally unique by virtue of the Ethernet address which each one contains[17], (b) each label is assigned for use on a specific flow segment, and (c) each flow segment is assigned to a linear (not circular) path.

the remnants of conversations that have ended but were not terminated properly. This process is conducted by the Interlan daemon.

[17] The uniqueness of Ethernet addresses today is not guaranteed. In particular, virtual machines are known to use Ethernet addresses that have not been issued through the IEEE standard process. Interlan will need to ensure that all hosts that register do so with Ethernet addresses that are unique among all Interlan registered hosts and forwarding engines.

4.23 Transition to a new network

It is important that there should be a smooth transition for Internet users as they move from IP to Interlan protocols. Existing software should continue to run correctly, existing equipment should be able to work with the new network, and users should not lose contact with the colleagues and services that they presently depend upon. Fortunately, the large majority of Internet users spend most of their time engaged in TCP sessions and other applications that employ unicast communications protocols. For these applications each unicast connection is implemented as an Interlan flow. That flow acts as a tunnel in which the IP packets are carried. Thus the Interlan network transparently provides security for the IP payload transport. Applications immediately have the benefit of a network with access controls and superior user authentication. Hosts that do not intend to be servers for the general public are not exposed to attack. Those that do offer service are only exposed through the port through which the service is offered. Exploratory attacks have to work through the name space which is much more difficult. All users are protected from attacks that snoop[18] on client-server communications and they are protected against packet insertion attacks.

Every service that is to be offered on the new network must be registered as an Interlan resource. The Interlan name of the service can be the same as the name which is used by IP applications. However, when making that registration the service owner will want to consider what access controls to use for the service in the new network. Also, the Interlan resource structure allows considerable flexibility in the way that multiple servers share the load of supporting a single service.

The security of personal computers on home and business premises has been a particularly vexing problem for Internet security experts. Even well-intentioned people have difficulty maintaining a secure environment within their homes and businesses. Therefore Interlan networks draw clear distinction between services that are registered as resources in the Interlan backbone network and the resources (including named hosts) that are attached to a local area network. The Interlan architecture requires that all assets which are on a local area network employ a name space that is dedicated to the home or business premises in which the LAN is installed. The database for that local name space must be inaccessible from the network outside that home or business. By adopting that policy it is accepted that accidents and oversights happen on all premises and that criminals operating from around the world have become experts at exploiting those situations. The policy also addresses the concern that existing applications and practices associated with IP networks include many that are intrinsically vulnerable to attack and it will be impractical to ask that these security risks be removed as hosts transition to an Interlan network. Hence the concept that a LAN is like a residential or business premises; it must be constructed with a security perimeter that is easily understood. Giving the LAN its own name service that cannot be accessed from outside the LAN does that.

That perimeter must be established before making the transition to the new network.

Focus on the naming system follows from an Interlan network objective to make it impossible to use the knowledge of a host address as a basis for attacking that host. It is for that reason that the Interlan syscall language (with one exception discussed below) does not provide any means of connecting to a host by presenting the host's Ethernet address or by presenting an Interlan label which contains that address. The Interlan language is based on names (and is the richer for it.) However, there are legitimate (non-criminal) applications which have been written to exploit the use of addresses. These applications require work before they can be moved over to an Interlan network. That we see as the price for removing the scourge that is presently causing so many people great pain, grief and loss.

The exception is presently a big one and we continue to look for a way to diminish or remove it. That exception is to be found in the implementation of private networks. A host that does not participate in a private network is not vulnerable, and many hosts that participate in private networks will not be vulnerable because the members of that network have no intention of doing harm to others. By confining the awareness and manipulation of addresses to private networks, the Interlan architecture provides users with a simple way of staying safe - don't join a private network. But, of course, that cannot satisfy everyone and there is a risk that a member of a private network will attract a virus and then, when operating outside the private network, will spread that virus to others. That is the risk that we would like to eradicate.

---

[18] At the time of writing (2010) it is not known how to protect a wireless channel from an expert cryptographic attack.

Multicast networks are a special case. They are private networks because in practice these networks employ techniques that require the manipulation of addresses. In principle, the Interlan language should be able to include a multicast functionality that is safe to use. It is not yet clear whether that will be useful. For example, Ethernet multicast gives the multicast source no way of controlling which network users can or cannot receive the multicast. On the other hand, a multicast implemented in the forwarding engines does have that functionality but is less efficient and is not backward compatible with existing applications.

As Internet users migrate to Interlan networks they should evaluate the applications that they use. Be very cautions before using an application which requires membership in a private network. There may be user preferences which prevent applications launched on the user's account from engaging in certain risky or unusual practices.

And there are no IP addresses which allow the naming system to be bypassed. The names of publicly available services are registered in the global name space, and the fact that a host may offer a public service does not mean that its other locally useful services are exposed to the public. So, while the transition from an IP to an Interlan operating environment requires that new directory entries must be made, most people will find that with insignificant effort they can immediately have much better control over whom they talk to and who can use their resources.

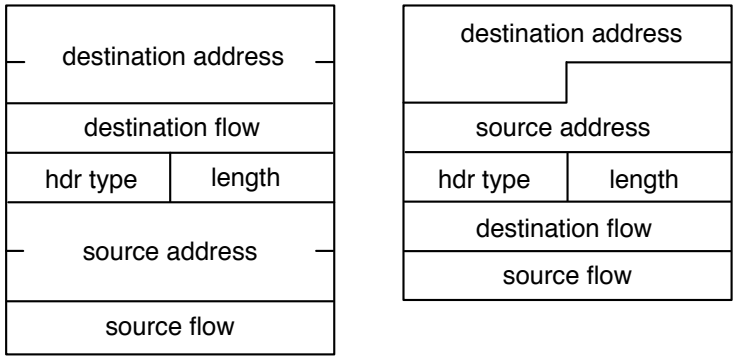
Address space issues

Migration from Internet to Interlan does not require the immediate purchase of new hardware. However, users may wish to disable or get rid of NAT boxes. Migration from IP addressing to an Ethernet based addressing system will be underway once an Interlan backbone is in place. That backbone relies entirely on Ethernet addressing and should be good for many years to come. If in the long run 48-bit addressing is not considered sufficient, there is an easy transition to an Interlan network with 64-bit Ethernet addressing.

Interlan resources are identified by name. The connect syscall does not accept an address as a valid form of resource identification. Addresses are used for efficiency but the interface between one user and another does not require shared knowledge of any address. This result follows from the design of the forwarding engines and the roles played by agents. Forwarding engines swap out the addresses used in the local area and replace those addresses with other addresses that are valid only in the backbone. An engine in the destination region swaps out the backbone addresses and installs a header that uses the addressing method of that destination region. If it were not for mobile devices and the emergency arrangements which cause one network to share equipment with another, each region could have an autonomous addressing system.

The backbone does have a global addressing system but the design does not require as large an address space as one might suppose. That comes about because of the system of resource labels. Each label is globally unique, but it contains two fields - a backbone Ethernet address (48 bits) and a resource number (24 bits). In effect the address space for the backbone is contained in 72 bits.

The backbone address has structure, so it is not as efficiently used as it might otherwise be, but the loss in coding efficiency is totally obscured by the overall size of the resource label. Also, backbone addresses are out of reach for network users, so there will not be the build-up in vested interests that made IPv4 addresses so difficult to manage. If in the centuries to come there is pressure on the 48-bit addressing system it will be because there is an extremely large number of mobile devices. In that case 64-bit addressing can be phased into service as a superset that contains the 48-bit address. The Interlan header structure can handle that in a backward compatible way so that local area multicast continues to work. Figure xx shows the header structures which support 64-bit and 48-bit addressing respectively. The 64-bit header is distinguished from a 48-bit header by the header type code that is contained in the fourth 32-bit word.



39

Figure xx. Interlan headers with 64-bit and 48-bit addresses

IP addressing

The next part of this story describes the work of forwarding engines and agents. A forwarding engine is managed by an agent which responds to syscalls. These are statements in a names based language. The agent obtains name translation which usually means that it obtains a label that refers to a resource, or more accurately - it refers to record which describes the resource. From that record the agent can obtain a flow labels and an IP addresses for the two ends of a flow. Backbone packets use the flow labels to find their way across the backbone network. The IP address is used when the journey ends on an aggregation network or LAN that relies upon IP addressing. Otherwise, the label provides the Ethernet address of the destination. This mechanism operates independently for each end of a connection. Therefore an Interlan flow can have IP at one end and Interlan at the other. So long as the Interlan host provides an IP header in addition to the Interlan header there can be effective communication end-to-end.

Notice that IP addressing serves no real purpose for local area communication once host software at both ends of a connection have been upgraded. Thus the matter of software upgrade has been reduced to independent bilateral agreements between clients and servers. The use of IP can continue for a long time without being a burden on other folk. Really the cost is not much more than the few bytes of packet space that the IP header consumes.

This, then is the last stages in a 30-year transformation during which time names of abstract objects have replaced the addresses of computers that implement them. In this way the Interlan architecture brings the network into line with operating systems and programming languages where names denote services and other assets of value to users. Compilers and operating systems translate from the abstract concept to its implementation in hardware.

Chapter 5

Multicast forwarding - see Aug 2 log

Private networks (with gateways)

Configurable engine

Datagram forwarding (IP and Ethernet)

This may be the point to begin discussing roles of agents, engines, drivers within a switching center.

Effect of nested Ethernets on header processing (simple and multicast flows)

Perhaps some discussion of restoration in a network of fat trees.

Network system architecture

Chapter 3 explained the separation of control from payload transport. These two aspects of network architecture rest upon a shared physical infrastructure but, for security, the two networks have very restricted information flow between them. The control and payload networks have the layered structure illustrated in figure xx. Does not include network operations systems which are outside the trusted systems boundary - perhaps use “admin” instead of “operations systems”.

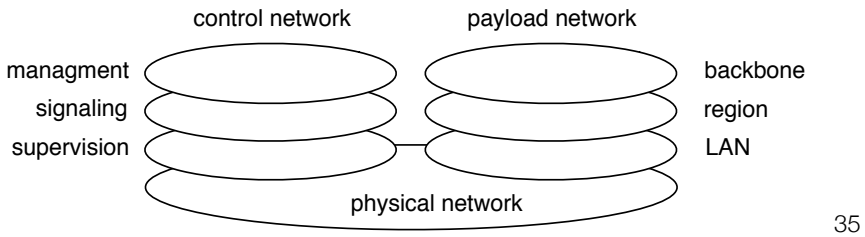


Figure xx. Communications infrastructure for payload transport and control

The control network carries messages within the network operating system. Each instance of the operating system consists of many processes, each one working for a network user or on some aspect of network operation. The distributed nature of communication service means that these processes do not work alone but they make progress by interacting with each other. Thus the life-blood of the operating system is a constant flow of control messages between processes distributed among switching centers around the world.

There are three levels of control: supervision, signaling and management. Supervision is the central nervous system of the network operating system. It is a network-wide function which constantly scans every item of equipment that makes up an Interlan network. The scan is designed to discover the equipment’s identity, track its movements and report on its current state. That is typically not a lot of information, but the scan must be repeated every few seconds. In addition, the supervision system transports alarms. These are short messages generated by the network equipment, but in this case the equipment takes the initiative and its alarm must be transmitted reliably and quickly to a relevant destination in the operating system. The supervision system does not require a high performance control network but that network must work reliably without interruption, and it must have tentacles that reach into every piece of equipment. Furthermore, the supervision system should be difficult for network users to attack.

The second level control system transports signaling messages. Signals are the internal messages which agents send to one another when they establish connections and engage in other collaborative tasks in support of user services. Signals are the network operating system's internal equivalent of the syscall messages that an agent exchanges with a host operating system. (ref) These messages must be transmitted with error detection and if the packets of a message get out of sequence they must be reassembled before delivery to the recipient control process. Signaling messages are not always as short as one might expect because of the need to transport public keys. Corrupted messages are not retransmitted by the control network; that task is left to the agents.

The third level system has more of the character of a data processing system. It is the management system which oversees operation of the entire network. This system has a number of sub-systems that are critical to the continued reliable operation and security of the operating system. One class of sub-system processes traffic measurements collected from the supervision system. From that data the operating system generates routing strategies that make best use of the available network capacity. It is this system which allows the network to respond effectively to emergencies, such as natural disasters, flash crowds and denial of service attacks. Another management subsystem conducts a continuous audit which is designed to combat the inevitable faults that occur in any large hardware-software complex. The audit system is counterpart to the supervision system. While supervision monitors equipment state, the audit systematically monitors the state of the operating system software, its internal consistency and its consistency with the reported state of the equipment. Inconsistencies, even apparently harmless inconsistencies, can escalate to become serious problems. Therefore the audit system forcibly removes any inconsistencies as soon as it finds them[19].

An overview of actions taken by the network operating system is contained in Chapter 3. Aspects of those actions are described throughout this and the remaining chapters of this report. They will not be elaborated here. However, it should be understood that the network operating system, the computing systems upon which it runs, and the control network which makes it a global facility, draw inspiration from a broad range of existing and historical systems including computer systems, host operating systems, database systems, and the control software developed for telephone and Internet services.

Payload network

The payload network is devoted to transporting user data. Its architecture has three levels: local area networks (LANs), regional networks, and backbone networks. The lowest level includes the local area networks that interconnect computers and other application devices on user premises. Wired and wireless LANs are both at this level as are the wireless segments of cellular and WiFi networks. Regional networks make up the second level. These networks connect local area and wireless networks to regional switching centers. A variety of aggregation network technologies are used for this purpose including DOCSIS, DSL and EPON. Then, at the third level are the backbone networks which transport packets across the country and around the world. The ownership and structure of these networks depend upon the size and policies of each country.

The technologies for local area networks, regional networks and backbone networks are quite different. Local area networks operate in premises that are generally protected from the weather and are isolated from public view. Cost is a major constraint on the quality of this equipment and security is limited by the capabilities of network users. This report assumes that Ethernet dominates here. The distance covered by a LAN is usually small, typically 10s or 100s of feet for a residence.

Regional networks must efficiently and economically reach into every home and business. These networks connect each premises to the local switching center. The network installation is technically challenged by its exposure to the weather and other hazards which arise in public places. At its edge a regional network has a separate connection into each home, which means that the usual economy of a shared transmission line does not apply and makes this an expensive part of the network plant.

Backbone networks on the other hand are high performance long-distance transmission systems that are shared by many people. While they are individually very expensive the cost per user is relatively small. A single high performance

[19] Audit is a practice carried over from telephone switching systems. It contributes to their high degree of reliable operation.

transmission line costs much less than a bundle of low performance cables with the same aggregate capacity. Interlan can use the existing infrastructure.

From a network operator's perspective the transmission capacity of regional and backbone payload networks must be organized in a uniform fashion so that differences in transmission technology are mostly hidden from the upper layers of the architecture. In this way new technology can be installed and legacy equipment can be phased out with minimum impact on other parts of the system. Interlan architecture achieves that result on two levels

(a) At the application level packet flows use a uniform transmission format derived from Ethernet.

(b) At the interface with the underlying physical plant the network is set of paths.

Services defined in terms of paths are largely independent of the underlying technology. Service features designed for the benefit of certain applications can employ paths in novel ways without necessarily forcing a change in the underlying equipment. Similarly, applications which use flows for their communications are largely independent of network implementation issues.

Each flow is a series of packets which follow a defined route. The route for a flow is a series of one or more paths. Paths are created by an autonomous process. They hide irrelevant details of the underlying hardware and serve a purpose comparable with Ethernet's spanning tree. Spanning trees are the basis for routing flows in aggregation and backbone networks. Flow routing in a backbone network employs a mesh of paths.

Redo this paragraph The "flow layer" responds to syscalls issued by application processes, and the "path layer" reflects the underlying network topology as discovered by the network supervision system. The flow layer consults the path layer when a route is being sought for a new flow. Otherwise the flow layer responds to the needs of network users while the path layer adapts to the properties and problems of different physical layer technologies. Congestion measurements made by the path system control the rate at which the packets of a flow are transmitted.

Flow segments are assigned to paths according to preferences expressed by users and according to traffic conditions observed by the supervision system. Path assignment for a flow also varies as the participating processes move from one place to another. Flow endpoints are permanently attached to processes, not hosts. Paths may be rerouted in response to changing conditions in the network. For efficient backbone utilization each path consumes only a small fraction of a transmission line's bandwidth so that rerouting one path makes no more than a small albeit significant change in traffic density.

Mobility is an aspect of communication service which depends upon the network's knowledge of individual flows. When a mobile device moves from one network location to another, the network uses its knowledge of which flows the device is participating in to make the necessary route changes. This capability is built into all aspects of the network's design so that it is available to every device with a network connection. Universal support for mobility is possible because the support for flows is universal.

Cellular telephony became a popular service in the late 1980s, and it was during the 1990s that wireless networking spread to laptop computers. Today the distinction between computing and telephony is becoming blurred. So too is the role of wireless communication and the definition of what constitutes mobility. At one extreme, mobile communication is when a conversation persists even while the end devices are moving about. But then there is the case of a laptop which can easily be transported between one network hot-point and another, with resumption of the application process when the laptop regains its network connection. Many people use wireless communication today for the convenience of not trailing a wired connection, even when their use of a computer or telephone is static. For the purposes of the present network design, the word "mobility" is broadly interpreted to include any situation where the physical connection between a device and the network is not continuous but the pattern of network use requires that flow connectivity persists. If there is a break in the physical path, security requires a repeat of device registration before all flows previously available are once again active. This applies for a cellular telephone that temporarily loses contact with one base station and moves to another, and for a laptop

with a wired network connection which is moved from one wall socket to another. The registration process in these circumstances can be expedited and does not necessarily involve the network user.

Privacy is another universal property of communication in an Interlan network. Each flow is identified with the application processes that it was established to interconnect; all other processes are excluded. The design intention is that no other process can listen into the conversation, nor can any other process inject packets into the flow. These characteristics of the way that flows are implemented are fundamental not only to the privacy of individual conversations but also to security of the entire network. For this reason flows in the payload network are designed for hardware implementation. In contrast with Internet routers which contain large software systems, the forwarding engines that implement flows contain very little software and are much less vulnerable to attack. In this way communications which take place in the payload network are protected from one another. Packets can only reach those places which are authorized by the network operating system.

Service quality is the third aspect of communication service which relies upon and is integral to the flow-based network design. Each application that uses the network can express preferences with regard to the performance, correctness and cost of payload transport. There is no single set of service parameters which will match every user's requirements. Live video, for example, has a certain demand for bandwidth and its quality degrades when that bandwidth is not continuously available. Telephony and other forms of interactive communication demand low delay. Web browsing stimulates many short messages and some that are very large. For economy, all of these traffic types need to share the same network. The challenge is to manage that sharing while implementing application preferences. User demand for network service varies with time, and sometimes this demand changes rapidly. But network capacity is fixed and is determined by the investments that network operators have made. Thus there needs to be dynamic control which balances user requirements and preferences against available capacity. That clearly cannot be done without the network being aware of individual needs, which is why a flow-aware network design with a syscall interface between application and network operating system is key to customized service quality. A user can express his preferences when creating a flow, and the network can associate those preferences with a flow as it is being routed.

Scale is also important. Variations in demand can be compatible with shared use of fixed capacity transmission systems when there is some flexibility to chose the routes over which the traffic flows, but routing must be aware of application preferences. It is also helpful to know what the user can be expected to do. Most transport systems, from the post office to the shipping lines, expect to be able to classify their cargo according to size and urgency. The network's need is the same. So, traffic type can be included in a statement of user preferences. In this way a flow-aware network operating system can do much to resolve the conflict between a dynamic traffic load, conflicting requirements for service quality, and a finite capacity network.

Regional switching center

<Consider starting out with a section on the plausible performance of a forwarding engine.>

queueing, path and flow header processing, rate control and flow control.

Much of this report relates to functions which, at least in part, are found within the walls of a switching center. The purpose of the following paragraphs is to provide perspective view of what that center might contain and how large it will need to be.

We start with the proposition that in all but perhaps the largest regions there will be a single switching center that handles the region's traffic. At the time of writing (2008) the industry structure seems unstable. In the United States there are three large network operators (AT&T, Comcast, Verizon) which continue out of momentum from the years prior to 1984 (the birth of the Internet and the break-up of AT&T.) Momentum for these companies is measured in the miles of cable that hang on telephone poles or lie buried in the ground. A strong stabilizing factor is the cost of building a new aggregation network. It is far more expensive to build from a clean start than to hang an extra cable where you already have one installed. To make matters even more difficult, many telephone poles have no spare space that can be leased. Large cellular network operators are slightly more numerous, but they rely on an extensive wired infrastructure which interconnects their base stations. As wireless transmission speed increases so antennas will be more closely spaced and the wired infrastructure will grow more complex. There are in addition many small network operators, and Internet Service Providers that use other

people's wires. We estimate that this picture will not change as much as some people may hope, so from a technology planning perspective we need to make sure that our plans are practical for large scale wired network operators, and in particular to understand the implications and practicalities of just one or two switching centers per region.

It is unclear what the maximum size for a switching center will be. Of the 109 regions used in our initial regional design for the USA, 39 regions contained more than one million households, and only 2 exceeded 5 million. Looking elsewhere in the world to foresee the future of population growth, we note that China, which has a population that is about four times that of the United States, can be divided into about the same number of regions with 100 mile radius. For that country it is estimated that 40 regions will have more than 5 million households in each.

Concerns about region size include the vulnerability of a large switching center to natural and man-made disasters, including war and power outages. There is an economy of scale when backbone traffic is concentrated onto a few high speed long-distance transmission lines, and under these circumstances there is a reduction in delay which makes the network's performance fast and predictable. From a security standpoint it is probably safer and less expensive to secure a small number of large centers.

For these reasons we have focused attention on the design of switching centers that serve one to ten million households. Given that the average payload transmission rate in the busiest time of the day may reach 20 Mb/sec per household (ref) we plan on a switching system design that can carry 200 Tb/sec of offered traffic.

To understand the consequences of this objective, consider that the traffic comes from and goes to households that are connected to aggregation networks which probably operate at a peak rate of 10 Gb/sec. So a switching center for one million households probably terminates between 1,000 and 2,000 aggregation networks. In the past, major switching centers have been dominated by racks of transmission line interface cards. It seems likely that that will continue to be the case, although modern electronics will reduce the total size. For planning purposes we shall assume that traffic from aggregation networks enters a switching center at 10 Gb/sec, and that each of these composite flows requires its own electronics.

Among this traffic will be an uncertain volume that comes from mobile devices. The number and transmission speed of these devices is difficult to judge, but two present-day facts are influential. (a) There is a convergence of traffic types. Voice, video and data applications run on computers whether they be static or mobile. (b) Mobile devices already outnumber static devices and the disparity is growing. Furthermore, many supposedly static computers are portable. (c) As data rate for mobile devices increases cell size (and the size of a "hot spot") will become smaller and the number of base stations will grow much larger. The economic means of back-haul for this traffic will be on the same aggregation network facilities which provide wired service to homes and businesses.

Automatic service restoration has a close relationship to mobile service. Both are concerned with quickly establishing a new route for a connection that has failed. In the case of restoration, failure may have natural causes outside of the network operator's control. For mobility, failure is predictable as a mobile device moves out of range for one base station and, hopefully, into range for another. Thus, the plan to include universal support for mobility, is consistent with providing a highly reliable service through automatic service restoration.

For all of these reasons we are focused a switching center that offers automatic service restoration, mobility and portability on every network connection.

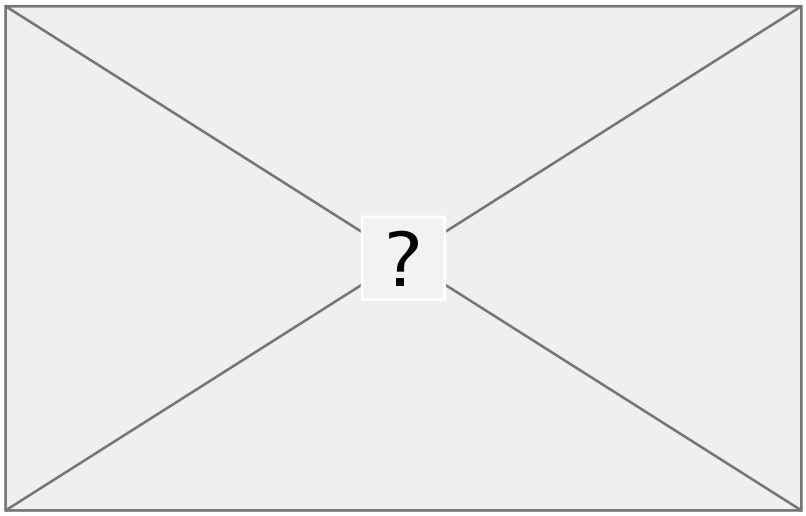
The primary service functions have been mentioned in earlier sections of this chapter. Agents, on behalf of the network operating system, and forwarding engines that implement switched services are the key elements. Forwarding engines in particular must be engineered for both high speed and reasonably fast restoration Forward engine functions include the following.

a) Flow forwarding for point-to-point flows
b) Path termination for backbone and aggregation networks
c) Rate control for all traffic that is ingress to the backbone
d) Volume control for all traffic sources that ingress via an aggregation network

e) Restoration of mobile and failed flows
f) Traffic measurements for management and billing purposes
g) Gateway functions for agents that receive control flows

The following figure is a concept diagram for a large switching system which meets the above mentioned objectives.

Show computing cluster, regional database



The principal components are these:

Agent

Computing capacity which is the primary source of 'intelligence' for user-facing functions in the switching center. The agents along with other processes that make up the network operating system will execute in rack-mounted processor clusters.

Resource database

A disk farm or other large persistent memory with high redundancy in which is recorded the network state, including a names service database, a database of resource records, billing records and other operations service records.

Control network

A secured high performance message-switched communications infrastructure which serves agents and other elements of the network operating system.

Forwarding engine

A special-purpose device which handles header processing and implements flow protection and rate control for individual connections. The forwarding engine is a table-driven machine that is a peripheral device on a processor cluster.

Root module

The interface to an aggregation network. This device sits at the root of what is usually a tree of multiplexers and switches connected to home and business networks.

Trunk module

The interface to a backbone transmission line. This module in common with a root module distributes ingress traffic to forwarding engines, and collects traffic for transmission out of the switching center.

Switch

A packet switching network that interconnects all of the switching center elements that make up the payload network.

CCCCCCCCCCC needs lots of work, architecture of engine and switch

Think about Ethernet address translation for a mobile device's engine engine table

Provide confidence that forwarding engines and switch can be constructed.

Discuss value of cluster computing with engines as peripheral, rather than processing on line cards.

Among the challenges facing this architecture is high speed translation of Ethernet addresses into slot numbers. Slots are the points where the switch connects to forwarding engines, trunk modules and root modules.

Aggregation network is the lowest speed and most numerous interface. It is obliged to translate the destination Ethernet address of an incoming packet in order to know which forwarding engine (slot) the packet should go to. The trunk also needs to translate the destination Ethernet address. But the speed is likely to be 100 times the speed of an aggregation network. Fortunately the Ethernet addresses used in packets arriving from the backbone network belong to the set of forwarding engines, a much smaller set than is connected to the aggregation networks. It is proposed that the backbone Ethernet addresses be drawn from a specific block, so that they can be translated by table lookup and yet they are as valid as any other Ethernet address which a forwarding engine will need to handle. (Trunks and aggregation networks, for reasons of security, are prevented from talking to one another.) Finally, the forwarding engine needs to match the speed of the trunk, but the engine has a privileged position. All of its table entries are provided by the agents, so the translation of Ethernet addresses can be stored in the forwarding table, thereby circumventing the needs for high speed Ethernet address translation there. By these means one of the serious bottlenecks in the construction of a large high performance Ethernet switch is circumvented (There is probably also a significant cost saving in complexity and cooling).

A typical packet flow enters a regional switching system through a root module, travels across the switch to a forwarding engine which transforms the packet header so that it refers to a forwarding engine in the destination region. Then the packet passes one more time through the switch to a trunk module which transfers the packet into a backbone network.

In smaller switching systems, and in many datagram switches the functions performed by agents and forwarding engines are implemented in the root and trunk modules. That design avoids sending each packet twice through the switch but it adds complexity in other ways. Putting an agent and forwarding engine on each root and trunk module adds considerably to its complexity in many ways, resulting in a power hungry and expensive pieces of equipment. The same amount of computing is more efficiently obtained by using a computer cluster where standard computer equipment can be used. Furthermore the computer cluster puts all agents in one place where they can interact more effectively, and the cluster can easily be expanded to include extra functionality. This design results in a small forwarding engine which can be optimized for performance, and it allows the forwarding engines to be clustered with the result that communication between agents and forwarding engines can be routed efficiently. That is particularly important for the very large switching systems which are required by the proposed regional design.

When a mobile host moves from one wireless base station to another the host may also be forced to move from one root module to using another. If connection state was held in the root module then that state would have to be copied from one module to another. By keeping flow-related state information in a free-standing forwarding engine we avoid most of that overhead and related complexity, thereby enabling faster response to device movement and providing good service to large vehicles with many passengers such as trains.

There are other benefits from this design. Automatic service restoration in the outside plant can be achieved by rapidly rerouting the traffic to avoid a broken piece of equipment. That reroute has the same impact on traffic flow in a switching

center as does large scale mobility. Therefore, a free-standing forwarding engine also simplifies the task of providing rapid service restoration.

The price to be paid for these benefits is a switch with twice the throughput. However, as we shall show in (ref) high capacity switching is made possible by the continued advances in device technology.

7 Discussion

A vision of the future which merges the host and network control systems belongs in Chapter 7.

/*

At some point I want to put together a discussion chapter, comparing our work with that of others and explaining why things are the way that we describe them.

*/

7.1 Ethernet address administration

Administration of Unique Ethernet addresses

User authentication as part of device authentication means that a device cannot register without the support of a user who can authentic himself. The device may also be required to contain a secret which allows it to prove its identity.

Interlan will attempt to prove that every address is being employed by a device and a user that are entitled to employ it, and that every address is associated with only one Interlan network account. (For this purpose, we must talk about the network of Interlan networks.)

It is proposed that Interlan networks participate in a global registration system for Ethernet addresses. That system might be supported by IEEE. (That is a policy matter that we will not address.) We postulate an Interlan specific Ethernet address registration system which is designed to ensure that no two devices registered with an Interlan network have the same address. The registration system requires a global directory of device addresses. It is already required that each user have an account record and that each device authenticate its identity each time that it becomes active on the network. That directory is accessible only from an Interlan operating system. The first time that a device (anywhere in the world) registers an Ethernet address the global directory is searched with a view of either identifying an existing address record, or creating a new one. If there is an existing record, the applicant must authenticate his claim against the credentials contained in the address record. Failure to authenticate in this way causes the registration request to fail. Otherwise the applicant must provide credentials which are stored in the new address record. (There may be other procedural matters designed to prevent abuse of this system.)

Once the first registration has been completed, any subsequent attempt to register with that Ethernet address must be accompanied by suitable credentials. In order to expedite these subsequent registrations, the Interlan home agent for the device may associate the authorized Ethernet address (or addresses) with the device's account. In this way, the global database is used only on the first occasion that a device attempts to register with a given Ethernet address.

If, in effect, hostilities disrupt global network administration, then there may temporarily need to be a separate instance of the address database, and Interlan networks will need to rely upon the properties of Interlan architecture which allow regional autonomy.

An incidental benefit of this method is that, at least as far as the network of Interlan networks is concerned, there is in this proposal the beginning of a recycling program for Ethernet addresses. Perhaps, title to an Ethernet address must be renewed every year. That can be an automatic byproduct of the usual registration process.

Authenticated use of Ethernet address

The legitimacy of a device address can be tested when the device registers, which it must do each time that it comes online. Interlan, for other reasons (ref), requires that device and user identify themselves and authenticate that identity. It is a feature of the registration system described above that each device address is authenticated and there is confirmation of its global uniqueness among all addresses used by the network of Interlan networks. Therefore, a fraudulent device that changes its address cannot be successful unless it emulates a registered device which uses that address. Given that the forwarding engines check that the (aggregation network) path on which packets arrive, the fraudulent device must either be co-located with the device that it is targeting or the physical hardware of the aggregation network must be tampered with. In (ref) we discuss Interlan defenses against attacks on aggregation networks. So it remains to consider the problem of a fraudulent device that emulates another device on the same LAN. That problem is discussed in (ref).

APPENDIX how to cope with non-unique host Ethernet addresses,

non-unique host Ethernet addresses, registration of host by the OS,
Ethernet addresses may not be large enough, registration of addresses by the state = header format

However, in fact neither of these requirements is fully met. IEEE's responsibility ends when an Ethernet address has been handed to a device manufacturer, but that is not enough to ensure proper use of that address. So, in practice Ethernet addresses are not necessarily unique. Local area networks are small islands of connectivity. It has not yet been demonstrated that Ethernet addresses can be the basis for reliable packet delivery on a global scale. The problem is that an Ethernet address does not contain location information and address translation at high speed on a large scale is difficult. There is also a concern that a 48-bit address may not be large enough to meet all future needs. These matters are discussed in the following paragraphs.

Interlan packets are routed to hosts and forwarding engines based upon their Ethernet addresses. If two hosts use the same Ethernet address either one may receive packets addressed to the other. While it is IEEE policy that each host/network interface should be provided with a unique Ethernet address, present usage patterns do not always conform.

a) IEEE issues address blocks to manufacturers who (today) are expected to assign each address to just one interface. However, that does not always happen as expected. For example, a manufacturer who has used up all addresses issued to him, may be reluctant to halt a production line in order to wait for a new block of addresses to be issued by IEEE.

b) Virtual machines are created by software and, for consistency with traditional machines, each virtual machine has a separate network interface. Today, these "virtual network interfaces" use Ethernet addresses that are not always issued through the standard IEEE process.

c) A hacker may contrive an attack, knowing that Interlan networks rely upon the uniqueness of Ethernet address assignments. He could use the fact that, in most (perhaps all) hosts, suitably privileged software can change the host's Ethernet address. A host that emulates another host is said to be "spoofing".

d) Ethernet switches in use today do not necessarily authenticate and monitor the address used by a host. Furthermore, the wide area network does not receive sufficient information to authenticate the addresses used on a given premises.

e) A growing number of people believe that the world is heading towards cyber war. Assumptions and agreements made in time of peace probably cannot be relied upon in times of war. In particular, it may be unsound to assume that IEEE can administer a global address space under these circumstances.

how to cope with mobility,
when Ethernet addresses are not globally unique

how to operate at high speed and on a large scale,
how to deal with the possibility the existing Ethernet addresses may not be large enough.

What happens if 48 bits are not enough

64-bit header format, or 48-bit format with the m.s. byte of the flow number field to contain the extra 8 bits.

21

An extension to the 48-bit Ethernet address might become essential if centralized control of the Ethernet address space by IEEE breaks down. If that happens address administration will probably devolve to individual nations and a global Ethernet address will need space for a country ID. The most likely causes are a world-war or a need to recycle Ethernet addresses.

Ethane

/*

See log for 27 October 2009 for a discussion of the two forwarding methods, and implications for one network using the other's forwarding method.

*/

Glossary

Flow A flow of packets traveling on a prescribed route, usually between two endpoints. The endpoints of a flow are usually host processes.

Control flow A flow between a host and an agent, used for control of the communication service which the host receives. This flow is usually encrypted.

Point-to-point flow A flow between two endpoints.

Multipoint flow A flow with multiple endpoints where each packet travels between just two of the endpoints of the flow.

Multicast flow A flow from a single source to multiple destinations.

Flow setup Establish the data structures which define the route of a flow.

Flow disconnect Removing the data structures which had defined the route of a flow

Path Manifestation of a route which a certain set of flows will follow.

Backbone path A path through a backbone network. Endpoints are usually forwarding engines.

Aggregation path A path through an aggregation network. The endpoints are usually a forwarding engine and a premises NIU or a wireless base station.

Switch A device with multiple ports. A packet entering at one port carries a destination address which implies the identity of the port through which it will leave the switch.

Path switch A switch wherein the packet destination address is a path number.

Ethernet switch A switch wherein the packet destination address is an Ethernet address.

Forwarding Engine A device which processes packets according to the content of a packet forwarding table, that table having been configured by an agent.

Network Interface Unit (NIU) A network interface unit, sits between a premises network and an aggregation network.

Premises gateway A multifunction unit that sits between a premises network and an aggregation network.

Aggregation network Connects a user premises to a switching center

Backbone network Interconnects switching centers.

National backbone Interconnects switching centers within one country or group of countries.

International gateway Interconnects a national backbone with one or more international backbones.

International backbone Interconnects international gateways.

Mobile	A device is said to be mobile if it can communicate while it is moving from one place to another.
Portable	A device is portable if it can be moved and it can communicate but cannot do both at the same time.
Agent	A process that implements the interface between a host and a communication network. Agents are located in a switching centers.
Name service	The service provided by a name server
Name server	Translates names upon request from agents. Names are looked up in a directory of names.
Directory	A list of names, each name has an associated set of attributes. Names refer to network resources.
Network resource	A service, a network, a host or a directory.
Name translation	The act of acquiring some or all of the attributes associated with a name.
Root directory	A directory which at the top level in the hierarchy of directories.
Trunk	A transmission line that is part of a backbone network
Line	A transmission line that is part of an aggregation network
Transmission line	The physical means of transporting a packet between one place and another. A transmission line is made of metallic wire or glass fiber
Radio channel	The physical means of transporting a packet through the ether without benefit of a transmission line.
Base station	A fixed endpoint of a radio channel, includes an antenna for radio transmission.
Host	A computer with a network connection not contained within the network core.
Server	A host that offers a service
Client	A host that makes use of a service
Service	A computational capability offered by a host for the benefit of others which use the network to connect to the service. Several clients may concurrently use one service, in which case each client may be served by a different server.
Listener	A server that is available to serve a new client.
Network operating system	The service which the agents collectively provide to networked hosts.
Payload network	That part of the public network which carries packets generated and consumed by hosts.
Network core	The trusted part of the public network. The payload network and network core are exclusive parts of the public network.
Public network	That part of the worldwide network which is not privately owned.
Flow control	The means by which two hosts regulate their transmissions so as not to flood the other with transmitted data.
Congestion control	The manes by which the network infrastructure is protected against being flooded with packets.

Rate control A means by which a host or forwarding engine regulates the frequency of its packet transmissions so as not to flood a bottleneck node in the network.

Supervision A continuous process that monitors the state and configuration of the network infrastructure and, if necessary, the immediate actions taken to combat a problem within the network.

Control network The network which interconnects agents, name servers and other computers located within the network core.

Supervision network The network which transmits supervision information and commands throughout the public network.

Supervision system The set of computers that supervise the network

Ethernet A protocol for transporting packets

Ethernet address The address assigned to a device so that it can communicate. Ethernet addresses are globally unique 48-bit or 64-bit addresses.

Flow segment That part of a flow which extends between one device (host or forwarding engine) and the next along the route which a flow follows.

Flow number A number assigned to the end of a flow segment by a device (host, forwarding engine) through which the flow passes.

Flow label The combination of an Ethernet address and flow number which uniquely identifies the end of a flow segment.

Resource number A number assigned to a resource by an agent which administers that resource. The resource number is unique within that agent.

Resource label The combination of an Ethernet address and resource number which uniquely identifies the resource within the public network.

Region The area served by a regional switching center and its associated aggregation networks.

Regional center A building which houses a regional switch and related communications equipment.

Regional switch A switching system upon which aggregation networks converge.

Root interface module which connects an aggregation network to a switching center.

Syscall A control message transmitted between a host operating system and an agent. Each syscall is a single packet transmitted in a control flow.

Signal A control message transmitted in the control network between one control computer and another.

Control network The network which interconnects computers in the network core.

Control computer A computer that is part of the network core. The computers which house agents and name servers are part of the network core.

Path network A network of paths and path switches. Aggregation networks and backbone networks are instances of path networks. Switches within a path network are used for load balancing and service restoration

Restoration The action taken to restore path connectivity after there has been an equipment failure or a mobile device has moved to a different base station.

Service quality The timeliness, reliability and rate of packet delivery provided for a flow.

Carrier grade service A standard of service quality to which network service providers aspire.

Datagram A message that occupies a single packet.

Trapezoidal routing A packet routing method whereby packets transmitted between a source and destination pass through at most two switching centers.

Flow protection Protection provided to a flow which ensures that the flow is not visible to and is not polluted with packets generated by non-participants in the flow.

Network security Protection provided to a network which ensures that the network operation is not compromised by the actions of any single host or group of hosts acting in unison.

Connection A flow between two hosts.

Local area network (LAN) A private network operating on user premises.

Bridge A switch that interconnects local area networks. Datagrams are switched within the bridge according to the Ethernet addresses which they carry.

Bridge connection A flow that connects a bridge to a local area network, or a flow that connects one local area network to another.

Bridge network A virtual network which consists of a bridge with connections to multiple local area networks.

Ethernet bridge A bridge that switches Ethernet datagrams.

UDP bridge A bridge that switches UDP datagrams.

Virtual network A bridge network or a flow network which has a defined membership.

Network membership A virtual network is represented in the network name space. Members connect to the network name and must satisfy access controls.

