

Internetworking Technologies

An Engineering Perspective

Rahul Banerjee

Computer Science & Information Systems Group
Birla Institute of Technology & Science
Pilani, India

Prentice-Hall of India

This small initiative is dedicated to my loving parents

*Mrs. Purnima Banerjee
&
Mr. Ramanand Banerjee*

*Who have been the guiding lights of my life and to whom I owe whatever
little I have been able to achieve.*

-Rahul Banerjee

Preface

Imagine a child sitting in the lap of her mother and watching endless stars in the sky. Those inquisitive eyes, small and innocent queries about everything she notices and finds either interesting or frightening, make the mother sometimes cuddle the child with all her affection and at times feel a bit irritated due to the same question being asked time and again. The same is the story of an inquisitive student population and a teacher who loves to impart whatever little he knows in a way that could inspire his students to learn more – often beyond the limits set by the basis course-structure! The situation becomes more involved when there is no single place wherein students may find every basic information they may need. And, that's when a small enterprise takes its root in some corner of the teacher's mind so that the hardship of his own students could be somewhat reduced, if not completely eliminated. This is exactly what had prompted me to begin a modest effort towards developing a Web-based book in the early 1999. The book, that originated from my lecture-notes, was made available at my website along with a lot of other supporting aids including customizable slides, FAQs and On-line Discussion Forum etc. The EAC 451 students doing this course on the campus, therefore, had to test the worth of this small initiative. What is in your hands right now is the *print version* of part of this work. The Web-based version is updated on a regular basis and is available at the URL: <http://www.bits-pilani.ac.in/~rahul/>. Part of this work contains case studies of select research projects carried out at the Centre for software Development, BITS Pilani (India). The presented material has been extensively classroom tested and used by on as well as off-campus students of the university.

The presented material should be adequate for a one-semester course at the senior undergraduate / graduate level. The organization is largely modular and therefore would permit an instructor to choose his own set of chapters in almost any sequence he considers suitable. The book assumes a basic knowledge of Data Structures, Graph Theory, Queuing Theory, Operating Systems and some exposure to Compute Networks on part of the readers, though it attempts to provide some basic concepts in a nutshell in the introductory chapters.

The book has been written as a text on internetworking technologies that should also cater to the needs of the working engineers who wish to update themselves about various associated technologies or those who wish to have a brief survey of the state-of-the art so as to decide the exact direction they may wish to take for their research and development initiatives. However, this small volume can very well serve as the secondary reading material for an advanced course in Internetworking. It takes a simple approach to illustrate intricate concepts as well as encourages the reader to take his first critical step forward through end-of-the-chapter exercises.

The book begins with a set of introductory chapters on internetworking concepts and gradually builds up the state-of-the-art technology and design concepts in the areas of Next Generation Networking (with specific emphasis on IPv6-based internetworking, mobile networking and interworking), Routing Architectures, and Desktop Video-on-Demand over NGNs and Internet Security Systems.

The book has been organized into twelve chapters and four appendices divided into three parts. First part introduces the uninitiated about certain basic technology terms

and related important concepts. The second part of the book takes up the system-level architectures. Third part of the book primarily comprises of application-level architectures and a small Internet programming primer. Finally the Appendices present a set of research / development draft papers that have emanated from the projects discussed in the Part-three. Appendices also include a literature guide and a bibliography to help readers in quickly identifying the initial foundation documents and related status reports wherever applicable.

Like any work of this nature, this work may have a few errors that may have escaped unnoticed. Students and peers are the best judges of any such endeavour and their constructive criticisms as well as suggestions are most welcome.

I would fail in his duty if I do not gratefully acknowledge the support, encouragement and inspiration that I received from my friends and colleagues. I am thankful to Dr. S. Venkateswaran (Director: BITS), Dr. B. R. Natarajan (Dean of DLP at BITS), Dr. K. R. V. Subramanian, CEO: Answerpal.com Bangalore, Dr. Rajeev Kumar of IIT Kharagpur, Dr. Sathya Rao of Telscom SA (Switzerland), Dr. Pascal Lorenz of UoHA (France), Dr. Bernardo Martinez of Versaware Inc. (Spain), Dr. Torsten Braun of UoB (Switzerland), Dr. Robert Fink of UCB (USA), Mr. Ishwar Bhat (Librarian: BITS) and Dr. Latif Ladid of Ericsson (Luxembourg) for their support and encouragement in many forms. In particular, I wish to express my gratitude towards my parents: Mr. Ramanand Banerjee and Mrs. Purnima Banerjee; my life-companion: Reena and little Ananya for all their love and support. Prof. Mahesh M. Bundle, Ms. Krishnapriya D. Bhardwaj, Mr. Ashaf Badar and Mr. Anand Gangele deserve special thanks for being there all the time whenever I needed them. Mr. Narendra Saini and Mr. Ashok Jitawat took expert care of typesetting in the camera-ready form and my heartfelt thanks go to them. The Prentice-Hall team of Mr. Ashok Ghosh, Mr. Vasudevan, Mr. Malay Ranjan Parida and Mohd. Shamim were instrumental in timely execution of the project.

Finally, I am also thankful to all my students – present and past — for providing me the reasons to take up this project.

BITS, Pilani
November 21, 2002

Rahul Banerjee

Contents

Preface

Part-I Internetworking, Multimedia, Compression and Intelligent Agent Technology Basics

1. Introductory Concepts in Internetworking

1.1	Introduction	1
1.2	Constituents of an Internetwork	2
1.3	Hierarchy in Internetworks	2
1.4	Classification of Internetworks	2
1.5	Local Area / Campus Internetwork Design: Practice and Trends	2
1.6	Competing LAN Technologies	3
1.7	Wide Area Internetwork Design: Practice and Trends	4
1.8	Competing WAN Technologies	4
1.8.1	Wide Area Technology: Other Classification Schemes	5
1.9	Steps Involved in Internetwork Design	5
1.10	Primary Design Goals of Internetwork Design	6
1.11	The Hierarchical Internetworking Design Models	7
1.11.1	The Hierarchical Internetworking Design Models: The Architectural View	7
1.12	Summary	7
1.13	Recommended Readings	8
1.14	Exercises	9

2. The Multimedia Internetworking Technology Basics

2.1	Introduction	10
2.2	Elements of Multimedia Communication	10
2.3	Defining Multimedia Internetwork	11
2.3.1	Examples of the Multimedia Internetwork in Action	11
2.4	Multimedia Internetworks: When to go for them?	11
2.5	Principles of Redesign and Upgrading of Data-Intranets to Multimedia Intranets	11
2.6	Multimedia Internetwork Requirements	12
2.7	Multimedia Internetwork Integration	12
2.8	A Generic Classification of Multimedia Internetworks	13
2.9	Link based Classification of Multimedia Internetworks	13
2.9.1	Point-to-Point Unidirectional Multimedia Internetwork applications	13
2.9.2	Point-to-Point Bi-directional Multimedia Internetwork applications	13
2.9.3	Point-to-Multi-point Unidirectional Multimedia Internetwork applications	13
2.9.4	Point-to-Multi-point Bi-directional Multimedia Internetwork applications	14
2.10	Interactive Multimedia Internetworks: Major Design Factors	14
2.11	Estimating Bandwidth Requirements for Multimedia Internetworks: Factors and Issues	14
2.12	The Bandwidth Factor	15
2.13	Networked Interactive Multimedia Video	15
2.14	Videoservers	16
2.15	Multimedia Broadcast Standards	16

2.16	Summary	17
2.17	Recommended Readings	18
2.18	Exercises	19

3. The Data Compression Technology Basics

3.1	Introduction	21
3.2	Space / Storage Compression	22
3.3	Lossy versus Lossless Data Compression	22
	3.3.1 Lossless Compression	22
	3.3.2 Lossy Compression	22
3.4	Graphics Metafiles	23
3.5	Language-based Redundancy Probabilities	23
3.6	Primary Classes of Data Encoding Techniques	23
	3.6.1 Entropy Encoding	23
	3.6.2 Source Encoding	23
	3.6.3 Statistical Encoding / Arithmetic Compression Technique	23
	3.6.4 Repetitive Sequence Suppression based Encoding Technique	23
	3.6.5 Differential Source Encoding Techniques	24
	3.6.6 The Transform based Source Encoding Techniques	24
	3.6.7 Huffman Encoding Techniques	24
	3.6.8 Adaptive Huffman Encoding	24
	3.6.9 The Lampel-Ziv Encoding Techniques	24
	3.6.10 The Lampel-Ziv Welsh (LZW-78) Encoding Technique	25
	3.6.11 The V.42 bis / British Telecom Lampel-Ziv (BTLZ) Compression	26
	3.6.11.1 Dictionary Pruning	26
	3.6.12 Discrete Cosine Transform based Compression Scheme	27
	3.6.13 Wavelets based Compression Scheme	27
	3.6.14 Fractal Compression Scheme	27
	3.6.15 Digital Video Interactive (DVI) Compression Scheme	28
	3.6.16 Other Compression Tools	28
3.7	The GIF Compression	28
3.8	The PNG Compression	29
3.9	The JPEG Compression	29
3.10	The MPEG Compression	30
3.11	Summary	31
3.12	Recommended Readings	31
3.13	Exercises	32

4. The Intelligent Agent Technology in Internetworking

4.1	Introduction	34
4.2	Intelligent Software Systems	34
4.3	Intelligent Agents	35
4.4	Attributes of Intelligent Agents	36
4.5	Intelligent Architectures	36
4.6	Internetworking Applications of Intelligent Agents	37
4.7	Role of Agents	37
4.8	Components of IA based Distributed Systems	37
4.9	Other Aspects of Intelligent Agents	38
4.10	IBM Aglet Technology Architecture	39
4.11	The Stanford's JAT Technology Architecture	40

4.12	The JAFMAS Technology Architecture	41
4.13	Summary	41
4.14	Recommended Readings	42
4.15	Exercises	43

Part-II Internetworking System Architectures

5. The TCP/IPv6 Internetworking Architecture

5.1	Introduction	44
5.2	The TCP/IPv6 Architecture: An Introduction	45
5.2.1	The Application Layer	45
5.2.2	The TCP/UDP Layer	45
5.2.3	Internet Layer	47
5.2.4	Host to Network Interface	48
5.3	The Internet Protocol	48
5.3.1	IPv4 Options	50
5.3.2	IPv4 and the World of Classes	50
5.3.3	Concept of Subnetting and Supernetting	51
5.3.4	On the Internet Control Message Protocol (ICMP)	53
5.3.5	On the Internet Group Management Protocol (IGMP)	53
5.3.6	The Address Resolution Protocol (ARP)	54
5.3.7	The Reverse Address Resolution Protocol (RARP)	54
5.3.8	Mobile IP	55
5.3.9	The Internet Protocol Version 6 (IPv6)	56
5.3.9.1	Major Goals of IPv6 Design	56
5.3.9.2	On the EUI-64 Addresses and the Link Local Addresses	57
5.3.9.3	How to convert a 48-bit Ethernet Address into the IEEE EUI-64 Address?	57
5.3.9.4	What about the networks for which no IEEE 802 address is available?	57
5.3.9.5	The IPv6 Base Header Design	58
5.3.9.6	The IPv6 Extension Header Structure	59
5.3.10	IPv6 Versus IPv4: A Brief Comparison	62
5.3.11	The IPv6 Address Notations	63
5.3.12	Address Issues in IPv6	63
5.3.12.1	Valid Address-Lifetime	64
5.3.12.2	Preferred Address-Lifetime	64
5.3.13	Address Autoconfiguration / Plug-and-Play Support in IPv6	64
5.3.13.1	Associated Factors of Autoconfiguration	64
5.3.13.2	Stateless Autoconfiguration	65
5.3.13.3	The Stateful Autoconfiguration	65
5.3.14	Time-sensitive IPv6 MM Traffic Over the Ethernet	67
5.3.15	A Quick Note on Mobile IPv6	69
5.3.16	On the Current State of IPv6 Research, Development and Deployment Around the World	69
5.4	On the Congestion Control in Internetworks	71
5.4.1	Congestion Control Strategies	71
5.4.1.1	The Anticipatory Buffer Allocation Scheme	71
5.4.1.2	'Arbitrary Packet Rejection-based' / 'Reject-on-Getting-Full' Congestion Control Scheme	72
5.4.1.3	Selective Packet Rejection based Congestion Control Scheme	72
5.4.1.4	Permit-based / Token-based / Isarithmic Congestion Control	72

	Scheme	
	5.4.1.5 The Choke Packet Scheme of Congestion Control	73
	5.4.2 Deadlock due to congestion	73
5.5	More on the Generic Transport Layer Concepts	74
	5.5.1 Transport Layer Responsibilities	74
	5.5.2 Generic Transport Service Primitives	74
	5.5.3 Generic Transport Service Primitives	74
	5.5.4 Transport Service Primitives: The Berkeley Sockets Set for the TCP	75
	5.5.5 The Transport Service Access Point (TSAP) and the Network Service Access Point (NSAP)	75
	5.5.6 QoS Considerations in the TL As Used During the Option Negotiation Process	75
	5.5.7 Inside the TCP	75
	5.5.7.1 About the TCP Ports	76
	5.5.7.2 The 3-Way Handshake in TCP	76
	5.5.7.3 Of the Crashes and Crash Recovery Mechanisms and Strategies applicable to the TCP/IP Architecture	77
	5.5.7.4 Client Crash Recovery Strategies	77
	5.5.7.5 Server Crash Recovery Strategies	78
5.6	About Application Client and Application Server Processes	79
5.7	Summary	79
5.8	Recommended Readings	80
5.9	Exercises	81

6. The Internetwork Routing Architectures

6.1	Introduction	84
6.2	About Routing Terminology	85
6.3	Classification of Routing Architectures	86
6.4	Shortest Path Routing	87
	6.4.1 Dijkstra's Algorithm	87
6.5	Flooding Based Routing	88
	6.5.1 Pure Flooding Algorithm	88
	6.5.2 Hop Count based Flooding Algorithm	88
	6.5.3 Selective / Direction-Constrained Flooding Algorithm	89
6.6	Flow-based Routing Algorithm	89
6.7	Distance Vector Routing Algorithm	89
6.8	Link-State Routing Algorithm	91
6.9	Hierarchical Routing Architectures	92
	6.9.1 The Interior Gateway Protocol (IGP)	93
	6.9.2 The Interior Gateway Routing Protocol (IGRP)	93
	6.9.3 The Exterior Gateway Protocol (EGP)	93
	6.9.4 The Border Gateway Protocol (BGP)	94
6.10	Issues in Hierarchical Routing Architectures	94
6.11	Summary	94
6.12	Recommended Readings	95
6.13	Exercises	96

7. Internetwork Management Architectures

7.1	Introduction	98
7.2	The Simple Network Management Protocol	
7.3	The Remote Monitoring (RMON) Scheme	
7.4	Role of Intelligent Agents in Internetwork Management	
7.5	Summary	
7.6	Recommended Readings	
7.7	Exercises	

8. Internet Security Architectures

8.1	Introduction	113
8.2	Security Issues in Intranets and the Internet	
8.3	Encryption-based Solutions	
8.4	Authentication-based Solutions	
8.5	Summary	
8.6	Recommended Readings	
8.7	Exercises	

Part-III Internetworking Application Architectures

9. Internetwork-based Video-on-Demand Architectures

9.1	Introduction	127
9.2	Types of Video-on-Demand Technologies	127
9.3	The Video-on-Demand System	127
9.4	The VoD Architecture	128
9.5	Basic Issues in VoD Design	128
9.6	Constituents of a VoD System	129
9.7	Internetworking Aspects of Video-on-Demand Technology	130
9.8	Case Study of the Cisco's IP/TV Solution	130
9.9	Case Study of the Ichcha-Drishti: Case Study of the World's First Native IPv6-capable VoD System (VoDv6)	132
9.10	Summary	133
9.11	Recommended Readings	133
9.12	Exercises	134

10. Internetwork-based Digital Library Architectures

10.1	Introduction	136
10.2	Classification of Digital Library Architectures	137
10.3	Major Digital Library Architectures	137
10.4	Basic Issues in Digital Library Design: Internetworking Viewpoint	138
10.5	Constitution of a Digital Library	138
10.6	Internetworking Aspects of Digital Libraries: Multimedia Object Handling	139
10.6	Case Study of the Stanford Digital Library Architecture	139

10.8	Case Study of the CMU Digital Library Architecture	140
10.9	Case Study of the JournalServer SM Virtual Digital Library Architecture	141
10.10	Summary	142
10.11	Recommended Readings	142
10.12	Exercises	143

11. Internet Commerce Architectures

11.1	Introduction	144
11.2	Principal Objectives of Internet Commerce	145
11.3	Fundamental Components of Internet Commerce Frameworks	145
11.4	Electronic Data Interchange (EDI)	145
11.5	The EDI Architecture	146
11.6	Electronic Funds Transfer (EFT)	146
11.7	Secure Electronic Transactions (SET)	147
11.8	The SET Architecture	147
11.9	The X.400 Standard-based Solution	148
11.10	The MIME-based Solution	148
11.11	Smart Cards and other Solutions	149
11.12	On the Digital Signature and Digital Certificates	149
11.13	The I-Commerce Gateways	152
11.14	Summary	152
11.15	Recommended Readings	153
11.16	Exercises 153	153

12. Internet Programming

12.1	Introduction	154
	12.1.1 Linux Network Programming Basics Revisited	154
	12.1.2 A Subset of Address Families Used in Linux Environment	155
	12.1.3 A Subset of Protocol Families Used in Linux Environment	155
	12.1.4 Socket Errors (ERRNO VALUES)	156
12.2	The World Wide Web and the Hypertext Transfer Protocol	156
12.3	The World Wide Web and Uniform Resource Locators (WWW & URLs)	157
12.4	The World Wide Web and File Transfer Protocol (WWW & FTP)	157
12.5	The Common Gateway Interface (CGI)	157
	12.5.1 The Common Gateway Interface (CGI) and PERL	158
	12.5.2 Invoking the PERL	158
	12.5.3 Select command-line switches and options	158
	12.5.4 Data Types in PERL	159
	12.5.5 File Handles in PERL	159
	12.5.6 File Access Symbols	159
	12.5.7 Relational Operators	159
	12.5.8 Logical Operators	159
	12.5.9 Conditional Operators	159
12.6	The Server Side Includes: An Example	159
12.7	Java Technologies	160
	12.7.1 The Concept of the Java Threads	160
	12.7.1.1 Creating threads	160
	12.7.2 The Java Script: A Scripting Language	160
	12.7.2.1 Java Script, HTML and Frames	161
	12.7.2.2 Java Script: A Partial Event List	161
	12.7.2.3 The Visual Basic Script and its Position vis-à-vis Java Script	161

12.8	The ActiveX Scripting Services	162
	12.8.1 Classes of ActiveX Scripting Components	162
	12.8.2 The VB Script and the Visual Basic	162
12.9	XML: A Quick Look	162
	12.9.1 XML and Java: A Quick Look	163
12.10	Summary	163
12.11	Recommended Readings	164
12.12	Exercises	165

Appendices

A-1	<i>A Revised Version of the IETF Internet Draft on the IPv6 Quality-of-Service through the Modified Flow-label Specification</i>
A-2	<i>A Revised Version of the IETF Internet Draft on the IPv6 Quality-of-Service through the Modified Hop-by-Hop Extension Header Specification</i>
A-3	<i>A Quick-view Chart of Major Internetworking Research and Development Initiatives Around the World</i>
A-4	<i>Bibliography</i>

Index

Chapter –1

Introductory Concepts in Internetworking

Interaction Goals

Objectives of this chapter are to define internetworks, discuss their basic constituents, learn about the advantages they offer, realize the design problems they pose, learn various design-specific concepts and appreciate the wide spectrum of applications they may be closely associated with. Additionally, the chapter also attempts to motivate further exploration by providing certain useful pointers, Self Assessment Questions and Exercises -- together; these aids aim to extend the coverage of the topic beyond the classroom interaction.

At the end of this chapter, you should be able to:

- Identify an internetwork as the Internet, Intranet or Extranet;
- Identify the design issues in each of these cases,
- Identify the right way to hook-up two internetworks,
- Analyze the correctness of the internetwork design approach,
- Tell about how to extend an existing design without throwing away existing setup.

The treatment presupposes the working knowledge of Computer Networks and some exposure to Operating Systems and Data Communication areas

1.1 Introduction

With each passing day, the people living in all parts of the world are getting closer to one-another, thanks to the years of human quest for making this world a better place to live! Several thousands of man-hours have made this journey towards this level of technological advancements possible. One of the basic tools that made us witness this global shrinking possible is the *computer communication* ('*compunication*' to the gifted coiners of the words!). An outstanding contribution that has accelerated this growth of information technology and thereby helped people to come closer than ever, in terms of collaborative activities at the least, is the *Internet*.

As the computers got smaller, cheaper and yet more powerful, more and more organizations, companies and people began having their own *private networks*--- even *internetworks*, in case of large organizations. Most of them wanted to join the rest of the information world by further connecting to the Internet. In fact, some of the organizations went a little ahead! They used the *Internet* as a vehicle of communication between their remotely located private *networks* / *internetworks*. Clearly, all of these developments saw the *internetworking technology* to evolve as an important technology in its own right! Times changed. And, as usual, this technology saw itself growing into several divergent but interrelated segments -- from Telerepair to Telemedicine to *Interactive Video-on-Demand* -- not to mention the *Internet Commerce* that glued it all. This work attempts to introduce you to this wonder world of technology in a step-wise and guided manner!

1.2 Constituents of an Internetwork

An *Internetwork* may be defined as a network of computer communication networks every authorized member of which could communicate with every other authorized member (node) directly or indirectly.

It may consist of several Local, Metropolitan or Wide Area Networks interconnected via a *LAN*, *MAN* or a *WAN* oriented communication technology, depending upon the specific context of use.

1.3 Hierarchy in Internetworks

Theoretically speaking, a single level *hierarchy*, i.e. a flat hierarchy is possible to attain in case of any network. Similarly, a *flat internetwork* is possible. Unlike the *local area networks*, where *hierarchical architecture* is seldom used, it is common to find both local as well as *wide area internetworks* having a two or greater levels of hierarchy. Reason can be many -- the greater degree of administrative control, the reduced routing table space requirements, drastically lesser search time or support for incremental growth. An internetwork may have a flat or *multilevel (Tree-like) hierarchy*. The number of levels depends upon several factors:

- Costs, Capacity and Number of *Routers* in the Internetwork
- Total Number of Networks in an Internetwork
- Degree of Administrative and Security Control Desired

F. Kamoun & L. Kleinrock suggested a simple rule of thumb for determining an acceptable number of *levels of hierarchy*:

*If number of routers is 'N', then
Number of levels of hierarchy = $\ln(N)$*

1.4 Classification of Internetworks

There exist three classes of Internetworks for most of the practical and analytical purposes:

- The Global Public Internetwork: The **Internet**
- The Wholly Owned / Private Internetworks: **Intranets**
- The Hybrid Internetwork-- private networks / internetworks connected through the Internet: **Extranets**

1.5 Local Area / Campus Internetwork Design: Practice and Trends

Traditionally, a *Campus Internetwork* is a campus-wide internetwork of individual LANs, which may be geographically spread over the part or whole of a single campus. In common practice, a single organization or institution wholly owns the entire campus internetwork including its communication subnet.

Usually, the campus internetworks use LAN technology; however, it is possible to use WAN technology, when so desirable. The latter may be desirable in some cases when

the campus is very large and comprises of a vast set of buildings spread over it. Protocols used in both of these cases are, generally, different.

Examples of the LAN technologies include the popular *Ethernet*, *Fast Ethernet*, *Gigabit Ethernet*, *Token Bus*, *Token Ring*, *FDDI* and *ATM LAN*, whereas examples of WAN technologies include *VSAT*, *Radio*, *Global System for Mobile communication (GSM)*, *Cellular Digital Packet Data (CDPD)*, *CDM*, *ATM WAN* etc.

Generally, WAN technologies are notorious for their severe cost constraint (initial as well as recurring) for high bandwidths.

This, however, is a non-issue for a *campus-wide internetwork* (except for the relatively high one-time upgrading / installation cost). This is because relatively smaller distances are involved than in the WANs and also because no post-installation recurring charges are payable to any external infrastructure / service provider.

Many designers prefer using a combination that could be a subset of *Shared Hubs* (conventional / intelligent type), *ATM Switches*, *CDDI / FDDI Concentrators*, *DLL Switches*, *Multi-layer Switches*, *Transparent / Source Routing Bridges*, *Routers (single / multi-protocol type)* and other existing devices / media in such a manner that the design could provide an extensible, cost-effective and acceptably efficient internetwork setup.

Choice of an exact combination of technologies is primarily dependent on the available budget, applications' requirements including the expected *Quality of Service (QoS)*, estimated *technology-lifetime*, available time (for upgrading / installation) and future projections.

1.6 Competing LAN Technologies

Major Competitors in this category include the Switched Routing of Packet and Cell Switching types. These may be further categorized as:

- *LAN Switching* (Layer-2 / Layer-3)
- *ATM LAN Switching*
- *Traditional Routing* (IPv4 and IPv6 routing included)

Major Features of *Layer-2 LAN Switches* include the following:

- Layer-2 LAN Switches (Ethernet / Token Ring) operate at the Data Link Layer.
- They permit *Source Routing / Transparent Bridging* options.
- They offer greater *bandwidth per node-pair* and improved performance cost-effectively.

Major Features of *Layer-3 Switches* include:

- *Layer-3 LAN Switches* (often a functional element of a multi-layer LAN switch) operate at the *Network Layer*.
- They provide switched routing functions with great degree of configurability in terms of *QoS, Traffic Control, Subnet Security* etc. apart from *Scalability* and *Stability*.
- They are, however, relatively poorly suited to *real-time traffic*.
- Choice of a conventional router or a Layer-3 Switch depends on several factors including *connection issues, cost* constraints and level of required *security* etc.

Major Features of *ATM LAN Switches* are as follows:

- ATM LAN Switches offer *high-speed LAN switching* and allow a high *bandwidth*.
- They provide *switched routing functions* in a way somewhat similar to the non-ATM LAN switches.
- They also offer a *guaranteed QoS, guaranteed orderly arrival of data units, easy Traffic Control, Subnet Security* etc.
- They inherently suit *real-time traffic requirements*. The *ATM LANE* technology allows *MAC-sub layer compatibility* with other common *LAN protocols* and therefore existing LAN applications may continue to run atop an ATM LAN as if they are running in their native LAN environments.
- Additionally, this permits the *VLAN (Virtual LAN)* technology to be employed, when so desired.

1.7 Wide Area Internetwork Design: Practice and Trends

The term 'wide area' in the world of networking refers to geographically separate areas and is different from the term 'metropolitan area'. Basically, what is a *LAN* or a *LAI* to a 'local area' the same is *WAN* or a *WAI* to a 'wide area'.

Design considerations for a *WAN / WAI* are, however, radically different than those of the *LAN / LAI*. Technology classes for local and wide area networks and internetworks overlap each other.

1.8 Competing WAN Technologies

Circuit Switching Technologies:

- Users can use the whole *channel bandwidth* assigned to them without any fear of blockade, infringement or delay.
- Well suited to *real-time applications* and the applications where delays can create serious problems.

- Once allotted, the channel and its entire *bandwidth* is reserved for the user until the circuit is explicitly released / terminated even when the channel is idle or only a fraction of the bandwidth is in use. This leads to inefficiency, poor channel utilization and longer waiting periods for others willing to use the channel.

Packet Switching Technologies:

- Users can share the available *channel bandwidth* amongst them without being aware of this fact.
- As the channel and its entire bandwidth is not reserved / monopolized, whenever the channel is idle or in partial use, anyone else is allowed to make use of it; and hence it offers greater average efficiency, better channel utilization and smaller mean waiting period for others willing to use the channel.

Virtual Circuit Switching Technologies:

- These technologies attempt to provide the best of packet switching as well as circuit switching worlds and display some of the features of each of these.
- These technologies offer low latency period and promise high throughput.
- As the bandwidth requirement soars, in many situations, these technologies actually offer cheaper routing elements compared to those of the packet switching schemes.
- Generally, these technologies demonstrate greater suitability to real-time traffic than their packet switching counterparts.

1.8.1 Wide Area Technology: Other Classification Schemes

In yet another classification, we may further regroup these technologies into classes like *ATM (WAN / WAI) / Frame Relay / X.25 / ISDN / Leased Line / VSAT / Cellular Radio / Terrestrial Microwave / Switched Multimegabit Data Service*.

In a nutshell, it may be said that there may be several overlapping classification schemes that may be applied to any set of such technologies. Some of the schemes may consider the PL features as the basis whereas some other schemes may consider DLL (MAC sub layer in particular) or NL features as their basis of classification.

What is common to all of the *WAN classification schemes* is the fact that none of them is usually classified with respect to any layer higher than the Layer-3 (i.e. the Network Layer in the *OSI model*).

1.9 Steps Involved in Internetwork Design

Requirement analysis: Statistical analysis of the specific and general requirements of an *internetwork* and its various segments in terms of hourly, six-hourly, twelve-

hourly, daily, weekly, monthly and yearly traffic is one of the key steps in the internetwork design. This analysis also helps in situation-specific or time-specific traffic estimation, availability analysis, maintainability analysis etc.

Projections: Projection of near and to some extent distant future requirements of an in-design / under-expansion internetwork is a necessary step that helps a designer to foresee the likely growth and usage pattern of an internetwork and make suitable provisions right at the architectural design stage.

Extensibility Analysis: It is an exercise that complements the previous step and helps a designer to discover whether his / her design shall pose any problems with respected extensibility in future. This step also guarantees investment protection to an appreciable extent.

Lifetime analysis: Every technology does have an *estimated lifetime*, beyond which it may have to be replaced with either an enhanced version or a radically new technology. It is a designer's responsibility to ensure that he / she does not use a technology, which is likely to necessitate sizeable re-investment in near future. Consideration for upward compatibility is, therefore, a thing that no designer can afford to overlook completely.

Technology and performance analysis: Analysis of the economics of the chosen technology vis-à-vis the expected *performance* is another step that may prevent certain seemingly attractive but inherently uneconomical design choices to be identified even before the pilot-implementation / prototype-building stage.

Sensitivity analysis: Most of the implementations tend to exhibit on or other type of sensitivity to their environment of operation. This, to a certain extent, may be desirable too -- particularly, for the sake of adaptability and auto-configuration type of requirements. However, there may be instances wherein a hypersensitive implementation actually may cause instability in part or whole of the internetwork. It is, therefore, designer's job to ensure that the network -- imbibes just the right degree of sensitivity by design, not by chance.

Design Validation / Simulation / Pilot Testing: These are the three ways to have a feel of the overall grand internetwork behaviour before actually building it in its entirety.

1.10 Primary Design Goals of Internetwork Design

Central design goals of an *Internetwork* include *Interoperability, Compatibility, Load Balancing, Consistency, Bandwidth Optimization, minimization of Information Storage and Retrieval Delay* while keeping the cost low, ensuring *FTRT processing* at intermediate nodes, provision for two or more levels of *Access Control and Authorization Checks*, provision for a verifiable mechanism for *Authentication, Application Transparency, High Availability, effective Congestion Avoidance / Control, Multi-Protocol Support*.

1.11 The Hierarchical Internetworking Design Models

Hierarchical Internetworking design models permit layered modular design of *internetworks*. They make it easy to accommodate design changes. Moreover, their modular design permits easy expandability of an internetwork as per the growing needs of the environment of operation.

Hierarchical Internetworking models compared to the huge monolithic *network design models / architectures*, obviate the need to make large-scale, and often expensive, changes influencing several component sub-systems. Another plus offered by these models is the ease and effectiveness of the *fault isolation*.

1.11.1 The Hierarchical Internetworking Design Models: The Architectural View

Hierarchical Internetworking models are basically *three-layer* models:

Layer-1 comprises of the functional building blocks, which ensure optimal Transport operations between the involved network locations. This layer handles high-speed switching and related issues and is often called the *Core* or *Backbone Layer*.

Layer-2 often called as the *Distribution Layer* is primarily responsible for providing connections between the requested sites as per a structured / default policy.

Layer-3 is the layer that is primarily responsible for controlling (and optionally monitoring) the user access to one or more segments of a designated internetwork / network. This layer is often called as the *Local Access Layer* for this reason.

1.12 Summary

Internetworks have come of age. Unlike the early days of internetworking, when only the computer science departments of a few privileged universities and select defense and telecom organizations were the major users as well as developers of this technology, now even laymen, housewives and children not only use these internetworks but many a times, actually contribute themselves to this core area. The best-known internetwork is the public Internet -- which saw unparalleled growth (or was that an explosion?) soon after emergence of the World Wide Web technology.

Although, it is the best-known type, the Internet is not the only known type of internetwork. Due to the reasons of varied degrees of privacy, security, administrative policy, distances, data transmission needs and associated economics of scale, a few other derivative technologies have begun evolving into their own -- most promising of these categories are the Intranet Technologies and the Extranet Technologies. Though they have a lot in common, because of the situations / circumstances of their use, they can be easily identified as different, though related, entities.

There exist several areas of overlap -- right from the switching technologies to the routing protocols and congestion control strategies! Each type of internetwork needs to address issues like stability, worst-case response time, availability, synchronization, concurrency control and resource sharing without policy violation as well.

Hierarchical or tree-structured internetworks are commonly used for the reasons of saving in terms routing table space and search time amongst several reasons like greater degree of administrative control such arrangements offer. However, not every such arrangement is always by choice -- at times, it just happens (for instance, as a result of incremental unplanned growth of networks within an environment).

Although, there do exist monolithic internetwork designs, mostly, these designs create serious problems in terms of technology upgrade and maintenance. The only advantage some of these designs do offer is their relatively low development time. Naturally, functionally layered architectural designs are becoming increasingly popular for medium to large internetworks. Often, these hierarchical design models are three layer architectures, comprising of the Core Layer / Backbone Layer, Distribution Layer and Local Access Layer. It is possible to have a design that may not really conform to this layering pattern necessarily. What cannot be ignored is the functionality that a layer is supposed to offer! Whatever be your design choice and strategy, you have to provide the minimal set of functionalities these layers put together provide.

1.13 Recommended Readings

1. B. O. Szuprowicz: **Multimedia Networking**, McGraw-Hill, New York, 1995.
2. C. Huitema: **IPv6**, Second Edition, Prentice-Hall PTR, Englewood Cliffs, NJ, 1998.
3. Cisco staff: **Internetwork Design Guide**, Cisco Press / Techmedia, New Delhi, 1999.
4. Cisco staff: **Internetworking Case Studies**, Cisco Press/Techmedia, New Delhi, 1996.
5. Cormac Long: **IP Network Design**, Tata McGraw-Hill, New Delhi, 2001.
6. D. Comer & D. L. Stevens: **Internetworking with TCP /IP**, Vols. 2-3, Prentice-Hall of India, New Delhi, 2000.
7. D. Comer: **Internetworking with TCP /IP**, Vol. -1, Third Edition, Prentice-Hall, Englewood Cliffs, 2002.
8. Dave Koiur: **IP Multicasting: The Complete Guide to Interactive Corporate Networks**, John Wiley & Sons, New York, 1998.
9. Garry R. McClain (Ed.): **Handbook of Networking and Connectivity**, AP Professional, 1994.
10. J. F. Koegel (Ed.): **Multimedia Systems**, ACM Press, Addison-Wesley, New York, 1994.
11. Marilee Ford et al: **Internetworking Technologies Handbook**, Third Edition, Cisco Press / Techmedia, New Delhi, 2002.
12. Nalin K. Sharada: **Multimedia Networking**, Prentice-Hall of India, New Delhi, 2002.
13. R. K. Arora et al (Ed.): **Multimedia 98 --- Shaping the Future**, Tata McGraw-Hill, New Delhi, 1998.
14. Rahul Banerjee: **Lecture Notes on Computer Networks**, Oct. 2002, BITS, Pilani, available on-line at: <http://www.bits-pilani.ac.in/~rahul/csc461/index.html/>
15. Rahul Banerjee: **Lecture Notes on Internetworking Technologies**, Oct. 2002, BITS, Pilani, available on-line at: <http://www.bits-pilani.ac.in/~rahul/eac451/index.html/>

1.14 Exercises

1. What are the situations in which, you, as an intranet designer, would opt for a Cell Switching Intranet technology?
2. Why is it more common to see Packet-Switched Campus-wide Intranets than the Virtual Circuit-Switched Intranets of the same set of capabilities? (An example is the popular preference to the Switched Gigabit Ethernet backbones over ATM backbones in campuses.)
3. Consider a situation in which your client, a large university, using IEEE 802.3 LANs, IEEE 802.5 LANs and a small ATM LAN wishes to replace / upgrade its existing IEEE 802.x LANs with / to a high speed setup capable of providing guaranteed quality of service for running heavy multimedia networking applications. The client also wants the VLAN technology to be available on demand. If the client demands that the proposed solution (to be offered by you) should not force it to throw away its older LAN-oriented application software, at least immediately, which internetworking technology out of those discussed in this chapter would you propose and why?
4. Look up the Web for Packet Service Internetworks and comment on the suitability of their application to the remotely located Indian rural areas for supporting the Tele-Medicine applications.
5. Study the IEEE 802.3x standard and the IEEE 802.11x standard. In case you have to integrate LANs based on these fixed and mobile networking-based standards, how would you plan interconnection such that seamless operation becomes possible at the user level?
6. Study the relevant IETF RFCs pertaining to the MPLS solution proposed originally by Cisco. What are the strengths and weaknesses of this solution in a multi-protocol environment? Why, in terms of classification, is it difficult to place this solution either at Layer-2 or at Layer-3?
7. Take a careful look at the Intranet of your organization and discuss its strengths and weaknesses from a designer and implementer's viewpoint.

Chapter-2

The Multimedia Internetworking Technology Basics

Interaction Goals

Interaction Goals of this chapter include defining the Multimedia Internetworks, identifying the fundamental components of Multimedia Communication, understanding of Design Issues, Bandwidth Requirement Analysis of the Shared Multimedia Applications, identification of the factors influencing Effective Link Bandwidth, developing a conceptual understanding of working and applications of Videoservers and a glimpse of current practices and future trends.

At the end of this chapter, you should be able to:

- ? do an effective analysis of the requirements of any prospective Multimedia Internetwork
- ? plan the location, number and functionality of basic internetwork building blocks
- ? take another look at your proposed design for a given situation in order to ensure cost-effective and reliable working of the 'in-design' internetwork
- ? suggest ways and means for improving / upgrading any existing internetwork to support desired level of collaborative multimedia environment.

Here, prerequisite is some exposure to Data Communication basics.

2.1 Introduction

In the previous chapter, we have explored the world of *internetworks* and attempted to pick up a few preliminary but fundamental concepts of associated technologies. We have just scratched the surface so far! It is about time we begin to focus on issues that plague the existing internetworks required to be upgraded to support an acceptable quality and volume of *multimedia traffic*. We shall also take a good look at the design of multimedia internetworks, related methodologies, architectures and technologies. This chapter shall form the basis for many other chapters like those addressing desktop videoconferencing, Video-on-Demand and education over the Net.

2.2 Elements of Multimedia Communication

There exist five major components of effective multimedia communication involving human being. These have been identified in the literature as:

- ❑ Capability of *media-based expression* of information
- ❑ Capability of effective use of various tools / means of articulation of a concept / idea
- ❑ Capability of reacting / responding in the *real-time*
- ❑ Capability of *collaborative communication*

- ❑ Capability of *unicasting, multicasting (or anycasting) and broadcasting*

Since most of the multimedia applications invariably focus on the *human behaviour, tolerance levels, adaptability, perception-patterns* and *intelligibility-characteristics*, all good multimedia internetwork designs need to model themselves on the abstractions suggested above.

2.3 Defining Multimedia Internetwork

An Internetwork of autonomous computers consisting of LANs and / or WANs, in which (depending upon the specific context of use) it could be possible for two or more participating entities to get an assured minimum *quality of network service(s)* during their exchange of one or more components of multimedia data is called a *Multimedia Internetwork (MMI)*.

2.3.1 Examples of the Multimedia Internetwork in Action:

There exist innumerable applications covered under the category of *multimedia networks* or *internetworks*. These include:

- ❑ *Desktop Videoconferencing* over the Internet
- ❑ *Scheduled Video over Internetworks*
- ❑ *Voice over Internetworks*
- ❑ *Video-on-Demand over Internetworks*
- ❑ *Multimedia-based Distance Learning* via the Internet (Virtual University models included)
- ❑ *Continuous-Media-based Digital Libraries*
- ❑ *Collaborative Workshops over the Net*
- ❑ *Telemedicine Consultancy* via the Internet

We shall learn about most of these applications and their typical *design requirements* from the *MMI* point of view in the subsequent chapters in adequate detail. In fact, a few case studies shall be taken up as the learning progresses.

2.4 Multimedia Internetworks: When to go for them?

There is no single best rule that could possibly advise on the exact point when to employ such internetworks. There exist, however, several factors which, when monitored, give an indication that the organization needs a *multimedia-capable internetwork*. These include *frequency of multimedia exchanges, exact nature and volume of such exchanges, duration of such simultaneous exchanges & number of users / entities involved per unit time*.

2.5 Principles of Redesign and Upgrading of Data-Intranets to Multimedia Intranets

There may exist several situations wherein it may be required to study an existing *cluster of generic networks* or an internetwork and selectively tune or upgrade it to a low-end or high-end *multimedia internetwork*. In some of the situations, particularly those wherein

the problem lies with the poor usage or configuration rather the hardware resources, it may be possible to get an acceptable performance just by putting your head down and tuning up the existing configuration or simply reallocation of resources. In a nutshell, not in all the cases of upgrade requests by your clients, an upgrade may really be necessary -- particularly where finance may be a major issue. However, in majority of the cases, selective upgrade may be a preferable approach. Only in very rare cases, actually the whole setup may be required to be coolly slipped into a museum of obsolete technologies. Steps that are normally helpful in *systematic upgrade* of existing internetworks to partial or full-fledged *MMIs* include:

- *Analysis of Bandwidth requirements*
- *Careful reallocation (preferably, dynamically) of network resources with the help of a priority policy*
- *Reconfiguration of the existing resources, if necessary*
- *Statistical analysis of user history profiles and authorization for selective priority based access control*
- *Structured grouping / regrouping of users*
- Exploring the *possibility of use of Intelligent Agents* and / or *Softbots* (Software Robots) for critical but frequent / repetitive tasks.
- *Upgrading the existing LAN(s), Inter-LAN Links* and, where necessary and viable, *WAN subnet components* for ensuring that the required number of simultaneous *Multimedia Data Streams* (usually, not more than five to ten) are possible to be provided by the internetwork without hampering other normal transactions / exchanges.

2.6 Multimedia Internetwork Requirements

Almost all *multimedia applications* are inherently *time-sensitive*. The *Time-Sensitivity Analysis* is, therefore, often a good way of moving towards a good *MM Internetwork* design. This requirement suggests that *Faster Than Real-Time (FTRT) processing* at various internetwork components (like *Hubs, Routers, Bridges, Gateways* etc.) is often necessary. Consequently, *Real-Time* or *near-Real-Time* traffic requirements suggest that *low-latency periods* are highly desirable. Put together, all of these factors point towards the need for some type of *QoS assurance* for such shared services.

2.7 Multimedia Internetwork Integration

As stressed throughout this and preceding discussion, primary needs and preferred features in an *MMI integration* include the capability to interoperate, exhibit stability, offer transparency, inherit controllability, demonstrate reliability and provide a high degree of availability -- and all this, without lowering of throughput and degree of service utilization. Also, in order to make the network / internetwork cost-effective and maintainable clean and patch-free *design* plays a crucial role. *Security issues* vary from situation to situation and it should be remembered that often networks with *adequate security by design* might prove insecure because of poor *configuration* or *access-control policy*. Too many *security levels* may actually serve to lower the *MMI performance* and should therefore be advised with caution.

2.8 A Generic Classification of Multimedia Internetworks

There do exist a variety of ways to place the MMIs in a specific category or the other. One of these is to consider the type of service-solicitation as the criteria for deciding a class. Based on this, a partial list of MMI classes might look like:

- ❑ *On-Demand Multimedia Internetworks*
- ❑ *Interactive Multi-location Telecollaboration-based Multimedia Internetworks*
- ❑ *Intelligent Multimedia Internetworks*
- ❑ *Desktop Teleconferencing-oriented Multimedia Internetworks*

2.9 Link based Classification of Multimedia Internetworks

MMIs can also be categorized on the basis of link classes. Going by this basis / yardstick, the major *MMI applications* can be grouped into four broad *classes*. These include:

- ❑ *Point-to-Point Unidirectional Multimedia Internetwork applications*
- ❑ *Point-to-Point Bi-directional Multimedia Internetwork applications*
- ❑ *Point-to-Multi-point Unidirectional Multimedia Internetwork applications*
- ❑ *Point-to-Multi-point Bi-directional Multimedia Internetwork applications*

Subsequent sections take a brief look at each of these classes and attempt to identify select applications in each of the categories. A later chapter shall discuss each major application in adequate detail.

2.9.1 Point-to-Point Unidirectional Multimedia Internetwork applications

Examples of Point-to-Point Unidirectional Multimedia Internetwork applications include:

- *One-way Teleconferencing with audio-callback*
- *One-way Video-Multicast using a stored video stream*
- *One-way Videoconferencing using a real-time stream*

2.9.2 Point-to-Point Bi-directional Multimedia Internetwork applications

Examples of Point-to-Point Bi-directional Multimedia Internetwork applications include:

- *Two-way Audioconferencing*
- *Two-way Videoconferencing (using real-time stream)*
- *Online Multimedia-based Training (real-time)*
- *Shared Whiteboard based Multimedia Collaboration*

2.9.3 Point-to-Multi-point Unidirectional Multimedia Internetwork applications

Examples of Point-to-Multi-point Unidirectional Multimedia Internetwork applications include:

- *Web TV*
- *Non-Interactive Real-Time Video Stream based Multicasting*
- *Non-Interactive Stored Video Stream based Multicasting*

2.9.4 Point-to-Multi-point Bi-directional Multimedia Internetwork applications

Examples of Point-to-Multi-point bi-directional Multimedia Internetwork applications include:

- *Interactive Video Distribution*
- *Multiparty Videoconferencing*
- *Video-on-Demand*
- *Voice-on-Demand*

2.10 Interactive Multimedia Internetworks: Major Design Factors

Not all multimedia internetworks are essentially interactive by nature of their operation. Interaction over *MMIs* are influenced by many factors including but not limited to the following:

- *Levels of multimedia information flow*
- *Type and Volume of multimedia content*
- *Number, Location and Frequency of entities involved in simultaneous multimedia information exchange*
- *Extent of Hardware and / or/ Software support required / available*

2.11 Estimating Bandwidth Requirements for Multimedia Internetworks: Factors and Issues

Each of the *basic multimedia objects* like *text, audio, video* and *graphics* has its own bandwidth requirement that widely varies from that of the other *objects*. Furthermore, factors like the proportion / degrees of use of two or more of such objects in a two-way or multi-party *multimedia exchange* influence the *bandwidth requirements*. Desired *transmission and reproduction quality* is yet another factor that influences such requirements. Number of parties involved and their geographic locations affect bandwidth requirements as well.

Amongst the other factors affecting the *bandwidth estimation* are dependent on the *physical characteristics of the medium / link* and *intermediate processing / switching / storage devices*, since each of these has potential to influence the *actual deliverable bandwidth specification*. Physical and logical *organization* of various *multimedia servers* (like *audio servers, video servers* etc.) and *multimedia databases* has a major bearing on the required bandwidth. *Router / Switch hierarchies* and the *network / internetwork topology* also play important roles in this matter.

Choice of Leased or on-demand bandwidth allocation depends upon the *economics of scale* and / or the critical nature of the intended applications. The choice of *Data Compression and Decompression / Recovery Scheme* plays an important role in all such matters.

2.12 The Bandwidth Factor

The maximum *Rate of Data Transfer* that a given *transmission link* may support, is called its *Maximum Bandwidth*. However, in an internetwork, often, it is the slowest intermediate link between two networks that influences the *maximum data transfer rate* actually achievable.

The *Effective Link-bandwidth* actually depends on several physical factors like:

- The *transmission quality* supported by a *guided or unguided medium*
- The *effect of proximity of adjacent signal frequencies*
- The *type of physical terminators and /or connectors* intended to use along with the *link*
- *Effect of noise(s) and external interference(s)*

Multimedia Traffic over an Internetwork may include one or more instances of:

- ❑ *Image* (10 - 500+ Kbps)
- ❑ *Voice* (4 - 100 Kbps)
- ❑ *Text / Data* (<5 Kbps)
- ❑ *Stereo Quality Audio* (125 Kbps - 1 Mbps)
- ❑ *HDTV Signals* (200 Mbps - 1 Gbps)
- ❑ *VCR Quality Video* (4 - 10Mbps)
- ❑ *3-D Scientific Visualization* (around 1 Gbps)
- ❑ *Animation*
< All of these values are approximate. Under test conditions, in certain cases, a higher estimate may be more valid. >

Average *degree of compression* in each case is different and has a bearing on the required bandwidth.

Types of *Compressed Video* include:

- ❑ *High Quality Compressed Video* (6-24 Mbps)
- ❑ *Medium Quality Compressed Video* (1.54 Mbps)
- ❑ *Low Quality Compressed Video* (100 Kbps)

In *multimedia internetworks*, the *medium quality compressed video* is often preferred because it offers a good *compromise* between *cost* and *quality*. At an average, it provides a performance comparable to 30 Frames / Second.

2.13 Networked Interactive Multimedia Video

The common types of Videoconferencing include:

- ❑ *One-way Videoconferencing with audio or textual callback*
- ❑ *Two-way Desktop or Integrated Videoconferencing*
- ❑ *Multi-point / Multi-way Integrated Videoconferencing*

Traditionally, the most common of these over the *multimedia internetwork* has been the *Multi-point / Multi-way Integrated Videoconferencing*. This provides a costly (medium to high cost) but highly collaborative option and permits the participants to exchange, modify, visualize and simultaneously view multimedia data that may be in the forms like Graphs, Charts, Images, and Text etc.

Desktop Videoconferencing is a low-cost option compared to the original *Integrated Videoconferencing* but is almost as effective as the latter except for its higher latency and poorer video quality. However, with the advancement of technology and the anticipated economics of scale, these drawbacks are no longer the major barriers.

Web TV and *LAN TV* technologies are other variations of *networked multimedia interactive video*. *Video-on-Demand* technology is a related technology but may exist with or without computer networks, though the former is more common.

2.14 Videoservers

A *Videoserver* is a server that is specifically designed and configured for:

- Handling efficiently and reliably *video traffic* over an existing network / internetwork.
- Converting *VHS (Video Home System)* signals into *digital video signals*.
- Converting *Analog Television signals* (where so applicable) into *digital video signals*.
- Compressing *compressible digital signals* before storage, forwarding or retransmission.
- Providing linkage between various *interacting components* using its services in a manner that is transparent to the participating clients.

Conventionally, the Videoserver software sits atop the *Network or Distributed Operating System (NOS or DOS)*. The exact amount of required bandwidth also depends on the capacity and speed of various components like Video Camera(s), Video Capture / Playback / Frame-grabber Adapter(s) and certain other factors including those mentioned earlier. In most of the real-life conditions, the major challenge of the *Multimedia Internetwork* is to *as closely match the internetwork capabilities and traffic demands as possible*. Like the factor of *acceptable Audio Latency*, *Video Latency* also proves a major factor in *bandwidth estimation as well as QoS-based Routing decisions* and that is why it is important to reduce *latency* and further take appropriate measures to nullify the effect (*jitter*) generated by *variable latency*. Physical distance involved as well as number of *hops involved*, play a very important role in case of the *WAs / WANs*.

2.15 Multimedia Broadcast Standards

There exist three major *standards for analog transmission of multimedia broadcast: NTSC, PAL and SECAM*. In a commercial scenario involving hybrid media a typical videoserver should be able to handle all the three formats on demand.

The **NTSC** (National Television Standards Committee) standard:

- Followed in the Central American countries, USA, Canada, Japan etc.

- ❑ Features 525 lines per frame and recommends 30 FPS (frames per second) refreshing rate. (Lines refer to Vertical Scan Lines here.)

The **PAL** (Phase Alternation Line) Standard:

- ❑ Followed in India, several European countries, gulf countries and many other countries.
- ❑ Features 625 lines and 25 FPS refreshing rate.

The **SECAM** (System Electronique pour Couleur Avec Mémoire) Standard:

- ❑ Primarily used as the analog multimedia broadcast standard in France, Russia and a few other countries
- ❑ Features 625 lines and 25 FPS refreshing rate.
- ❑ In a way, SECAM is a variant of the PAL standard.

2.16 Summary

Multimedia Internetworking can take several forms and depending upon the situation-specific / application-specific requirements, may need a different structured design approach in each of the cases. As usual in the real-life situations, no single design or design strategy, howsoever brilliant it may be, works well in all situations. Yet, a pool of a few time-tested strategies allows a decent workable solution if a careful analysis of the situation is carried out by the designer. One simple rule to keep in mind is that like in any other engineering design, economics may, at times, dominate your final decision so much so that a technically superior design may have to give way to a relatively inferior solution. Such a situation may arise specially when the designer, out of his / her enthusiasm for the best technical quality design commits a blunder of ignoring the feasibility factor and the budget of the client. Often a two-pronged approach of step-wise incrementing the quality and almost simultaneously assessing the associated cost helps to avoid such situations. Hierarchical design architectures in their industry-standard three-layer avatar help in arriving at a good design solution only when all these factors are constantly kept in mind.

Technology helps. But so does the common sense! For instance, an intelligent rearrangement of an existing setup or regrouping of users / applications or just sensible reallocation of available resources may make an existing internetwork to qualify as an acceptably good quality MMI. And, all this without any additional investment!

Cases of upgrade that really demand / warrant major changes are actually redesign problems. All redesign problems are inherently tricky and need more caution in handling than their 'fresh design' brethren. Steps suggested in the chapter, therefore, can play a very helpful role in these matters.

Most of the MMIs have to address soft real-time segments. This does not, however, lessen the magnitude of the problem except for the fact that you, as a designer, may offer a solution that may afford to reduce the price tag by slightly compromising on the FTRT requirement of processing. (Not all MMIs would perform acceptably in this way though!)

Just as the right hardware choice is critical for such designs / redesigns, the software choices, content-management strategies and transfer-of-control strategies play important roles in the actual performance of an envisioned design.

2.17 Recommended Readings

1. B. O. Szuprowicz: **Multimedia Networking**, McGraw-Hill, New York, 1995.
2. C. Huitema: **IPv6**, Second Edition, Prentice-Hall PTR, Englewood Cliffs, NJ, 1998.
3. Cisco staff: **Internetwork Design Guide**, Cisco Press / Techmedia, New Delhi, 1999.
4. Cisco staff: **Internetworking Case Studies**, Cisco Press / Techmedia, New Delhi, 1996.
5. Cormac Long: **IP Network Design**, Tata McGraw-Hill, New Delhi, 2001.
6. D. Comer & D. L. Stevens: **Internetworking with TCP /IP**, Vols. 2-3, Prentice-Hall of India, New Delhi, 2000.
7. D. Comer: **Internetworking with TCP / IP**, Vol. -1, Third Edition, Prentice-Hall, Englewood Cliffs, 2002.
8. Dave Koir: **IP Multicasting: The Complete Guide to Interactive Corporate Networks**, John Wiley & Sons, New York, 1998.
9. Garry R. McClain (Ed.): **Handbook of Networking and Connectivity**, AP Professional, New York, 1994.
10. H. Ghosh and S. Chaudhury: **An Abductive Framework for Retrieval of Multimedia Documents in a Distributed Environment**, Proceedings of the KBCS '98 International Conference, NCST, Bombay, Dec. 1998, pp. 153-165.
11. J. F. Koegel (Ed.): **Multimedia Systems**, ACM Press, Addison-Wesley, New York, 1994.
12. Marilee Ford et al: **Internetworking Technologies Handbook**, Third Edition, Cisco Press / Techmedia, New Delhi, 2002.
13. Nalin K. Sharada: **Multimedia Networking**, Prentice-Hall of India, New Delhi, 2002.
14. R. K. Arora et al (Ed.): **Multimedia 98 --- Shaping the Future**, Tata McGraw-Hill, 1998.
15. Rahul Banerjee: **Lecture Notes on Computer Networks**, Oct. 2002, BITS, Pilani, available on-line at: <http://www.bits-pilani.ac.in/~rahul/csc461/index.html/>
16. Rahul Banerjee: **Lecture Notes on Internetworking Technologies**, Oct. 2002, BITS, Pilani, available on-line at: <http://www.bits-pilani.ac.in/~rahul/eac451/index.html/>
17. **RFC 1009** (Requirements for Internet Gateways)
18. **RFC 1124** (Policy Issues in Interconnecting Networks)
19. **RFC 1175** (FYI: A very useful reference-list on Internetworking related information)
20. **RFC 1360** (Official Protocol Standards of the Internet Architecture Board)

2.18 Exercises

1. What are the situations in which, Jitter as well as Latency need to be minimized appreciably? How shall you overcome the Jitter in an Intranet, which is frequently used for heavy multi-party multimedia applications?
2. Consider a situation in which your client, a large university wishes to upgrade its entire existing intranet to a high-speed setup capable of multimedia networking. The client also wants the Video Streaming

technology to be available for Video-over-the Intranet on demand. What shall be your primary design choices in this case and why?

3. Look up the Web for Cisco's Internetwork Operating System (IOS) details and comment on the suitability of its application to the Intranet environments requiring multimedia traffic management.
4. Take a careful look at the Intranet of your organization and suggest what needs to be done to shape it as a setup for supporting multi-party desktop videoconferencing.
5. An area of key concern in the MMIs is the way multimedia files are stored and retrieved. More often than not, generic file-systems exhibit their inherent inefficiency in this regard and make the latency problem more severe. A few research groups around the world are working on Multimedia File-systems. Identify such groups over the Web and study their findings. Based on your analysis, suggest a possible file-system architecture that, in your opinion, would aid MMIs in performing better.
6. Why is it necessary to consider individual time-presentation styles for various multimedia objects like video, audio etc. and why these styles need to be preserved even when these objects are used together in a networked environment?
7. What are the issues involved in the synchronization of various multimedia components?
8. How can we apply the Ethernet technology, if at all, in IPv6-based time-sensitive Intranets?
9. What is the basic limiting factor for the multimedia information transfer over POTS and why?
10. Video Telephony typically uses a low frame rate (say 10-15 FPS) and a small picture – size. Furthermore, it uses an appropriate image compression scheme. Why?
11. If a small company requires occasional low-quality Video-Conferencing over analog line sometimes designers recommend the 'Switched 56' service deployment. Why?
12. T-1, T-2, T-3 and T-4 represent 1.544 Mbps, 6.312 mbps, 44.70 Mbps and 274.00 Mbps digital leased lines. All of these use TDM principle. A lower-speed leased line of 384 Kbps known as FT-1 can be used for Video-Conferencing as well. Which data-compression standards can be used for this purpose and why?
13. Compute the optimal bandwidth requirements of a network that has the following needs and constraints:
 - Provision for four concurrent two-party desktop video-conferencing systems using full-screen 1024*768 pixel windows with high-colour full-duplex audio and video exchange,
 - Provision for normal Web-browsing, E-mail transfer, File transfer and Remote Login for 100 concurrent users,
 - Support for 10 concurrent VoIP services using PC-to-PC, PC-to-IP Phone, IP Phone to PC or IP Phone to IP Phone devices.

Please note that in all cases, you may first compute the raw bandwidth requirements and subsequently reduce this raw figure to a realistic value by suggesting employment of one or more appropriate Data Compression scheme.

Chapter-3

The Data Compression Technology Basics

Interaction Goals

Learning objectives of this chapter include an appreciation of the basic techniques and strategies used in achieving data compression with specific reference to the MMIs. Naturally, a good understanding of fundamentals, a brief study of impact of compression and decompression on internetworking applications including the continuous media-based ones and a quick look at the current practices and evolving trends form the basic content that is expected to be assimilated.

At the end of this chapter, you should be able to:

- Select the right compression strategy for any MMI-based application,
- Write efficient software Codec(s) for the algorithm(s) you chose above,
- Work out the price-performance statistics of this program designed by you.

The treatment assumes a sound knowledge of data structures, time and space complexity analysis and mathematical theory of transforms.

3.1 Introduction

In comparison to the normal *time-insensitive traffic*, the *multimedia traffic* over *internetworks* has its own set of requirements and these specific needs range from *MM-specific Data Representation, Manipulation, Transmission, Storage, Management* to *MM-specific Retrieval*. This is more so because of the reasons like:

- *Time-sensitive nature* of the most of the *MM-traffic*,
- *Bandwidth constraints* of the *data pipes*,
- Large and varied sizes of *unstructured data (BLOBs)*,
- *Need for operational transparency* and
- *Associated economics*.

A natural requirement of such traffic, as discussed earlier, is to have a *continuous as well as steady flow of stream of multimedia data*. In other words, *Stream-based traffic mechanism / Isochronous traffic mechanism* is a requirement for the *MM-traffic*, particularly over the *networks* and *internetworks*.

All these requirements put together have necessitated specialized technology solutions for such traffic right from *reception* to *storage, retrieval* and *transmission*. *Data Compression Technologies*, therefore, address an important aspect of this problem.

3.2 Space / Storage Compression

Reduction of storage requirement of any entity is called its storage / space compression. (Theoretically, both space and time compressions are attainable. The focus here, is however on the space / storage requirements.) In an Internetwork, careful compression is essential for acceptable throughput. Bandwidth Compression is often a direct consequence of storage / space compression.

Majority of the *video-data compression schemes* employ mathematical *algorithms* for *smoothing out the minor / finer details within the original video-data those are not recognizable by the naked human eye*. The most common way to do it involves digitization of the original data followed by application of these *smoothing algorithms* (sometimes called *'filters'*) to it.

There exist two primary classes of *Data Compression: Lossy Compression and Lossless Compression*. Another way to classify *compression* may be *Symmetric Compression and Asymmetric Compression*. Yet another way to categorize *compression* may be based on the manner of compression within or between the successive *video-frames*; and may lead to two basic classes: *Intraframe Compression and Interframe Compression*.

3.3 Lossy versus Lossless Data Compression

As suggested in the foregoing discussion, there are two broad classes of any form of compression:

- *Lossy Compression* (e.g. *JPEG compression*)
- *Lossless Compression* (e.g. *RLE compression*)

Certain situations that may tolerate some degree of information loss are best suited to the *Lossy Compression* schemes.

3.3.1 Lossless Compression

In case, the *compression scheme / algorithm* ensures that while storing the information in the chosen compressed format, it does not leave any piece of it out; and, the *decompression scheme / algorithm* guarantees that uncompressed form and original form are exactly the same, such techniques are called *Lossless Compression Techniques*.

3.3.2 Lossy Compression

In case, the *scheme / algorithm* works such that while storing the information in the chosen compressed format, it does leave out certain pieces of it; and, even the *decompression scheme / algorithm* cannot retrieve the information-content of the entity-in-question to its original form, the compression technique is called *Lossy Compression Technique*.

3.4 Graphics Metafiles

A *graphics metafile* is often a file that provides *storage / space compression* by *describing the graphical details by using Meta tags / descriptive notations*.

An example of a *Graphics Metafile*: The description may involve naming a regular shape; starting coordinate and other associated attribute(s). For instance, the description: *Square 20,2,38* may refer to a square which is anchored at the screen coordinate (20,2) and which has width as well as length 38 pixels. Similarly, *Circle 20,2,40* may refer to a circle whose centre coordinate is (20,2) and whose radius is 40 pixels-long. Not all graphics may be expressible in so simple a manner though!

3.5 Language-based Redundancy Probabilities

Language-based redundancy probabilities may be of many types including *Letter Repetition / Redundancy Probabilities*, *Word Repetition / Redundancy Probabilities*, *Special Character / White space Repetition / Redundancy Probabilities* and *Notation / Symbol Repetition / Redundancy Probabilities*. Any / all of these redundancies can be exploited to obtain varying degrees of compression of textual documents.

3.6 Primary Classes of Data Encoding Techniques

3.6.1 Entropy Encoding: This is a *lossless encoding technique* that does not make any distinction between data-bits on the basis of its characteristics. It has two sub-classes:

- *Statistical / Arithmetic encoding technique*
- *Suppressive Repetitive Sequences-based encoding technique*

3.6.2 Source Encoding: This technique takes characteristics of the components of *compression object* into account.

3.6.3 Statistical Encoding / Arithmetic Compression Technique

In this case, the given textual data / file is analyzed and a *Concurrence Table* (i.e. a table of repetitive usage) is generated for select *patterns* (of sequence of characters) and thereafter using a specifically designed *compressed representation format* every such occurrence is encoded (normally, with lesser number of bits).

Two such techniques are:

- *Morse Code Encoding Technique*
- *Huffman Encoding Technique*

3.6.4 Repetitive Sequence Suppression based Encoding Technique

In this case, a given data is analyzed to:

- *Determine the presence and locations of long repetitive bit-sequences (in succession) in the data / file; and thereafter,*

- *Replace (i.e. suppress) each of such sequences by a shorter (specifically assigned) symbol / special bit-pattern.*

One such technique is the *Run Length Encoding (RLE)* technique, in which any such repetitive sequence / character is replaced with a *flag* followed by the number of repetitions which is further followed by the original bit-sequence / character that was found to be repeated in succession in the original data.

3.6.5 Differential Source Encoding Techniques

Such techniques are employed when *data blocks* (say each block represents a *Frame*) have only small degree of changes with respect to their immediate predecessors and successors. (A continuous audio signal or a motion video is a good example of such a case.)

Some of such encoding techniques include:

- *Pulse Code Modulation (PCM)*
- *Delta PCM*
- *Adaptive Delta PCM*
- *Differential PCM*

3.6.6 The Transform based Source Encoding Techniques

This technique makes use of any suitable *mathematical transform* for attaining the *reduced storage / bandwidth requirement* for a give data. *Important / strongest coefficients* are encoded precisely and *less important / weak coefficients* are often encoded with less precision in such *Transform Encoding* cases.

Examples of such transforms include:

- *Fourier Transform*
- *Discrete Cosine Transform*

3.6.7 Huffman Encoding Techniques

These are the encoding techniques, which fall in the category of 'general' data compression techniques.

Pure Huffman Encoding: This involves use of a *variable length code* for each of the elements within the information. Like the statistical paradigm, this technique determines occurrence probabilities and then encodes the most probable elements with lesser number of bits whereas encoding of the least probable elements is done using greater number of bits.

3.6.8 Adaptive Huffman Encoding

This variation was first suggested by Faller and Gallager and subsequently modified by Knuth. Therefore it is also known as *FGK Encoding Technique*. Unlike its pure version, the adaptive version provides *optimal encoding by adapting the encoding process as per*

analytical statistics of a piece of data. The *encoder* thus learns to react to the *locality-specific* needs. Also, only *one pass scan* is adequate in this case.

This scheme utilizes the *Sibling Property*, which suggests that if, in a *Binary Code Tree*, each (non-root) node has a *Sibling* and if these nodes form a *non-ascending weight based node-list*, the tree is said to have the *Sibling Property*.

3.6.9 The Lempel-Ziv Encoding Techniques

These techniques make use of *Adaptive Dictionary-based Data Compression Schemes*.

- *Pure (LZ)*
- *Welsh variation (LZW)*
 - *Fixed Length*
 - *Variable Length*

As the original (*LZ*) technique was developed in 1977, it is sometimes called '*LZ-77*' technique. In this scheme of compression the first step is to locate *type and frequency of repetition*. This repetition may be in many of the ways including:

- *Binary repetitions*
- *Textual repetitions* (includes letters / words etc.)

A special identifier called '*Flag*' is used for distinguishing *compressed data* from *uncompressed data*.

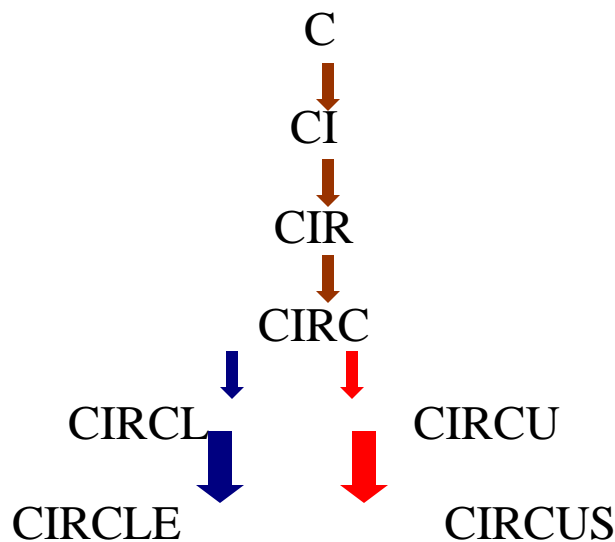


Fig. 3.1: A Logical Tree-based Learning Process

3.6.10 The Lempel-Ziv Welsh (LZW-78) Encoding Technique

This technique was originally suggested in 1978 as an improvement over the *LZ-77*. Its basic idea is to locate the *type and frequency of repetition* (this repetition may be in many of the binary repetitions or textual repetitions (includes letters / words etc.) type), build a *dictionary of the Most Frequently Used characters / bytes* and use a special identifier called '*Flag*' for distinguishing *compressed data* from *uncompressed data*.

Storage or transmission of this dictionary, as the case may be, before decoding the compressed data is necessary in this scheme. This technique proves acceptably good for textual compression although not so good for image compression. In most of the practical implementations of this scheme dictionary size is 4K or above. The storage structure in such cases uses addresses 0 to 255 for storing bytes / characters (single character) and the remaining addresses 256 and above are used for storage of strings containing 2 or more than 2 characters. The encoding scheme depends in a way on the dictionary size as well. For instance, in case of 4K-dictionary size, 12-bit encoding scheme is used.

3.6.11 The V.42 bis / British Telecom Lampel-Ziv (BTLZ) Compression

In 1988, British Telecom proposed this compression scheme to the ITU (earlier called CCITT). In 1990, ITU accepted it under the name *V.42 bis*. This *compression scheme* had the following characteristics that made it suitable for use in *dial-up Modems*:

- It can be easily implemented on *8/16-bit microprocessors*.
- It has low resource requirements specifically in terms of *Memory*.
- It has incorporated its *dictionary-size* including its *codeword notation* as well as *representation scheme*.
- It supports *dictionary pruning*.
- *Control codes* 0, 1 and 2 are reserved.
- Allows 256 *strings* of one character length.

Due to the provision '1', the initial strings are required to be indicated by the indices 3 through 258 (instead of the default 0-255). Due to the provision '3', index of the new strings / entries in the dictionary begin with 259. The *data structure* supported in this case is the *Trie data structure*. The basic *character-set* supported includes 256 characters (indexed 3-258 in the dictionary) and therefore any new entries, as mentioned earlier, begin only after these (i.e. index 259 onwards)! One possible situation describing the physical Trie structure for the *V.42 bis / BTLZ scheme* is depicted below.

The *Physical Trie Structure* looks like:

Node No.-> | **Character** | **Parent** | **First child** | **Dependent** |

3.6.11.1 Dictionary Pruning

This refers to the act of removing *dictionary entries*. The *V.42 bis* uses the *Least Recently Used (LRU) Algorithm* for selecting the strings to be removed from the dictionary.

3.6.12 Discrete Cosine Transform based Compression Scheme

This is conceptually similar but technically superior to the well-known *Fast Fourier Transform (FFT)* based *compression scheme* in terms of *speed of convergence* as well

as *compression ratio*. As with the other schemes in this class, the *DCT-based compression schemes* eliminate redundant visual data in the block of Pixels. The *JPEG* (Joint Photographic Experts Group standard), *MPEG* (Motion Pictures Experts Group standard), *H.261* (Video-conferencing standard) are the well-known compression standards based on this principle.

There are two basic forms of the *DCT*; namely, *Forward Discrete Cosine Transform (FDCT)* and *Inverse Discrete Cosine Transform (IDCT)*.

Although, theoretically, the *DCT-based* scheme may provide *data compression* to a maximum of 800:1, the practical upper limit has not been able to cross 230:1 as of this writing. As a result, for more MM-heavy *MMI applications*, improved techniques / schemes are being investigated / evolved.

3.6.13 Wavelets based Compression Scheme

Wavelet Compression scheme was developed at the AT & T Bell Laboratories with the objective of providing higher compression efficiency than the FFT and DCT-based solutions. It does not provide, however, any spectacular performance improvement over the latter! Unlike the DCT, this scheme uses *Pixel blocks of smaller size for fine detailing of the relevant video-data-area and Pixel blocks of larger size for the coarse detailing of the visually less relevant data-area*. In many other ways, otherwise, *Wavelet* and *DCT* scheme act similarly.

3.6.14 Fractal Compression Scheme

The term *Fractal* has its origin in the phrase "*Fractional Dimensional*", a phrase commonly used in the world of mathematics for referring to a *fractional element of a graphic object generated by repetitive application of a compression algorithm until the point of convergence is reached and the algorithm terminates*.

The primary strategy employed in *Fractal Compression* is the identification of one or more 'basic' shapes and / or patterns within a block of graphic image so that the original graphic could be represented merely by a set of mathematical functions. This results in smooth degradation of the graphic image. Unlike all other schemes, discussed so far, in this case, *algorithm* itself is transmitted over the data-pipe and not the image itself! (And, that's the secret of the high compression ratio!)

Although less relevant at the current state of technology, primarily due to large *compression time requirement*, this scheme of data compression is an active area of interest and research for *MMI researchers*. The root of this interest is the *potential compression ratio* that, at least theoretically, is one of the highest offered by any other scheme – a theoretical upper limit of 10000:1 has been computed – practically, *compression ratio* of the order of 2500:1 has been possible to achieve!

This scheme, like many others, is an *Asymmetric Data Compression scheme*, as it *requires greater time in data compression than the decompression*.

3.6.15 Digital Video Interactive (DVI) Compression Scheme

The *DVI Compression Scheme* is a form of *Vector Quantization-based data compression scheme*. It was originally developed at the Saruff Laboratory of the Radio Corporation of America (RCA) and subsequently improved by the IBM Corporation and Intel Corporation.

It permits real-time video editing of the compressed data and offers *Real-Time Video (RTV) data compression performance* (in terms of *compression ratio* as well as *quality*) that is comparable to the *Motion-JPEG* (a variation of the JPEG that had a short life until the *MPEG* arrived really). It offers *Production-Level Video (PLV) compression ratio* of the order of 120:1 that is a remarkable feature.

A *programmable compression scheme*, the *DVI* is basically an *Asymmetric Data Compression scheme* that requires specialized hardware as well as software support for being used. Downside of this technology is its very high computing needs due to which it could not be popular with most of the less demanding *MMI applications*.

3.6.16 Other Compression Tools

Intel's *Indeo*, Apple's *QuickTime*, IBM's *Ultimotion*, Progressive Network's *Real Video*, Microsoft's *Video for Windows*, Duck's *True Motion*, VDOnet's *VDOWave* and the *H.261* are some other well-known solutions offered for *video-data compression*. Some of these solutions are completely *Software Codec* based whereas some other solutions require specialized *Video Digitization Hardware* as well as the *Software Codec*. Pure Software Codecs often provide *smaller video-window sizes* for acceptable resolution; therefore, *Full-Screen True Colour Motion Video* often requires the hardware support of the said type.

Most of these solutions are primarily based upon one of the basic *compression schemes* discussed above. For instance, the *Real Video* offers *Fractal Compression* based *streaming* solution whereas many other *Software Codecs* use one or other form of *Vector Quantization based compression*.

3.7 The GIF Compression

The term *GIF* stands for the *Graphics Interchange Format*. It comes into multiple flavours primarily emanating from two versions: *GIF 87a* and *GIF 89a*. This format was popularized by the CompuServe in the eighties and is a commonly used scheme for encoding still images in *normal*, *interlaced* and *animated* forms. (Fig. 3.2)

The *GIF* algorithm is based on a variant of the *LZW* scheme described earlier. *It can be briefly described as below:*

- Initialize the string table;
- [Start-prefix] = Null;
- NextChar = next character in character-stream;
- Is [Start-prefix] NextChar present in string table?
 - If yes: [Start-prefix] = [Start-prefix] NextChar; go to Step-3;
 - If no: add [Start-prefix] NextChar to the string table;
 - Write the code for [Start-prefix] to the code-stream;

- [Start-prefix] = NextChar;
- Go to Step-3;

The definition of the *GIF Format* includes a *Data Stream* comprising of the *Header*, the *Logical Screen Descriptor*, a *Global Color Table* and the *GIF Trailer*. It cannot support more than 8-bit colour description (i.e. 256 colours).

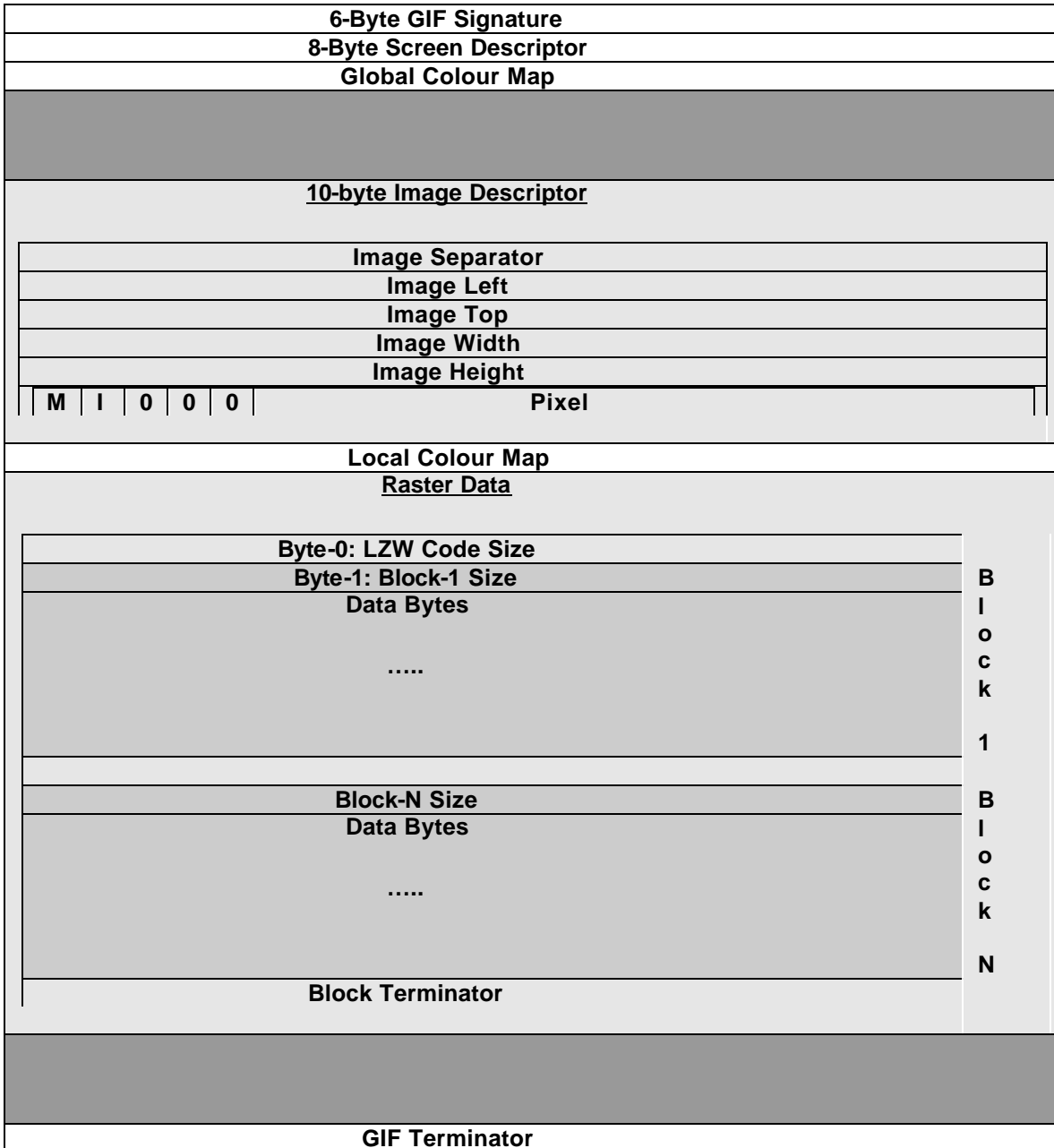


Fig. 3.2: The GIF Format

3.8 The PNG Compression

The term *PNG* stands for the *Portable Network Graphics*. It was based on a W3C recommendation document for still images. The basic idea was to improve upon the GIF capabilities and features as well as to provide a network-friendly format that could be free from patenting-specific issues and could be thus used freely by developers and users alike.

The PNG scheme is a combination of two schemes: *Predictive Encoding Scheme* and *Entropy Encoding Scheme*.

3.9 The JPEG Compression

The term *JPEG* stands for the *Joint Photographic Experts Group*. This standard was created for still images in the late 1980s and has an associated ISO / ITU-T document that describes it. The original *JPEG* was a *DCT-based scheme* and had following modes:

- *Lossless Mode*
- *Lossy Mode*
- *Baseline Mode*
- *Progressive Mode*
- *Hierarchical Mode*

JPEG uses a *set of algorithms* some of which are interchangeable in terms of functionalities of compression category and quality. The currently prevalent *JPEG* standard [ISO-JPEG-1] has forty-four modes, many of which are application specific and are not used commonly. There are several variants of *JPEG* based on these modes and a few small enhancements. These include:

- *L-JPEG*
- *LS-JPEG*
- *Motion-JPEG*

The last one is a variant of the original *JPEG*, meant for a temporary solution for movie image format based on a sequence of still image frames.

The latest addition to the *JPEG-family* is the *JPEG-2000*. This is a radically different format as compared to its immediate predecessor. This uses a *variant of the Wavelet Transform* called *Discrete Wavelet Transform (DWT)*. *JPEG-2000* uses *scalar quantization, context modeling, arithmetic encoding* and *post-compression (transfer) rate allocation*. It offers a low bit-rate compression (< 0.25 bits per pixel) for high resolution images, large image handling (> 64k x 64k pixels), better quality of transmission in noisy environments like mobile radio / telephony and capability to handle natural as well as synthetic images.

3.10 The MPEG Compression

The term *MPEG* stands for *Motion Picture Experts Group*. This is a *layered encoding scheme* that comes into a variety of flavours and versions including the following:

- *MPEG-1*
- *MPEG-2*

- MPEG-4
- MPEG-7
- MPEG-21

Standards like *MPEG-1*, *MPEG-2*, *MPEG-4*, *MPEG-7* and *MPEG-21* belong to the *Continuous Media* category of *media objects*. Put together, these are often referred to as MPEG-x standards.

The popular audio format *MP3* actually stands for *MPEG-1 (Audio) Layer-3*. Similarly, *MPEG-4 VTC* stands for *MPEG-4 Visual Texture Coding*.

Whenever two or more *multimedia objects* need to be *embedded*, they need to *synchronize* their *time-presentation styles*. This temporal relationship between various MM objects is often called '*Synchronization*'. For *network-oriented applications*, not only such *temporal relations* and their *embedding schemes* play important roles but also their *data-compression mechanisms / algorithms / formats* do matter in a big way.

Movie clips and Animation clips are examples of *Continuous Media* or *Time-Dependent Multimedia Objects*. Still Images, Textual data are the examples of *Time-Independent Media Objects*.

MPEG-1 is a five-part standard:

- ISO/IEC 11172-1:1993 Information technology -- Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbits/s -- Part 1: Systems
- ISO/IEC 11172-2:1993 Information technology -- Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbits/s -- Part 2: Video
- ISO/IEC 11172-3:1993 Information technology -- Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbits/s -- Part 3: Audio
- ISO/IEC 11172-4:1995 Information technology -- Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbits/s -- Part 4: Compliance testing
- ISO/IEC TR 11172-5:1998 Information technology -- Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbits/s -- Part 5: Software simulation

MPEG-2 is nine-part standard. One has been withdrawn later.

- ISO/IEC 13818-1:2000 Part 1:Systems
- ISO/IEC 13818-2:2000 Part 2:Video
- ISO/IEC 13818-3:1998 Part 3: Audio
- ISO/IEC 13818-4:1998 Part 4: Conformance testing
- ISO/IEC TR 13818-5:1997 Part 5: Software simulation
- ISO/IEC 13818-6:1998 Part 6: Extensions for DSM-CC
- ISO/IEC 13818-7:1997 Part 7: Advanced Audio Coding (AAC)
- ISO/IEC 13818-9:1996 Part 9: Extension for real time interface for systems

- decoders
- ISO/IEC 13818-10:1999 Part 10: Conformance extensions for Digital Storage Media Command and Control (DSM-CC)

The *MPEG-2 Video* was found to be better than certain earlier standards or specifications developed for high bit-rate or studio applications.

Like its predecessors, MPEG-4 standard is also a multi-part standard (ISO/IEC 14496-x). It permits synchronized delivery of streaming data from source to destination. It uses a two-layer multiplexing scheme so as to be able to exploit the Quality-of-Service (QoS) provided, if any, by the underlying internetwork. MPEG-4 has its evolution inspired by the advances in the areas of Digital Television (Normal and High Definition), Artificially Generated Interactive Graphics and Distributed Interactive Multimedia. It has integrated features covering a whole range of services including production, content-distribution and content-access. It has the capability to describe visualization of a complex scene comprising of hierarchy of a variety of media objects.

IETF's AVT Working Group and the ISO's MPEG-4 community collaborated to develop the exact specification to carry the MPEG-4 over the IP. These efforts have resulted in a framework standardized in the document ISO/IEC 14496-8 that is often seen as an broad specification for the transmission and use of MPEG-4 sessions over IP (Supported protocols, as usual include the RTP, RTSP, UDP, HTTP etc.). Similarly, relevant RTP payload format specifications have been formalized. MPEG-4 standard has the following eight parts:

- ISO/IEC 14496-1 (Systems)- It has tools including BiFS, Object Descriptors, FlexMux, MP4 File Format etc.
- ISO/IEC 14496-2 (Visual)- It has specification for natural and synthetic coding as well as Facial and Body Animation.
- ISO/IEC 14496-3 (Audio)- It has specification for Speech coding, General Audio Coding, Structured Audio, Text to Speech interface, Parameteric Audio etc.
- ISO/IEC 14496-4 (Conformance)
- ISO/IEC 14496-5 (Reference Software)
- ISO/IEC 14496-6 (Delivery Multimedia Integration Framework)
- ISO/IEC 14496-7 (Optimised software for MPEG-4 tools)
- ISO/IEC 14496-8 (MPEG-4 on IP framework)

The MPEG-7 standard deals with the description and specification of the multimedia content rather than the compression itself. MPEG-21 is currently under evolution and it aims at defining an open framework for multimedia delivery.

3.11 Summary

Specific needs of the MMI-specific applications range from MM-specific Data Representation, Manipulation, Transmission, Storage, and Management to MM-specific Retrieval. This is why MM traffic over internetworks deserves some special treatment so as to provide an optimal performance in a given situation.

Many solutions to this set of requirements have been suggested. Some of these solutions are completely Software Codec based whereas some other solutions need specialized Hardware as well as the Software Codec. Pure Software Codecs often provide smaller video-window sizes.

Intel's Indeo, Apple's QuickTime, IBM's Ultimotion, Progressive Network's Real Video, Microsoft's Video for Windows, Duck's True Motion, VDOnet's VDOWave and the H.261 are some of the popular video compression solutions.

The MPEG-1 and MPEG-2 standards have enabled the wide spread use of MP3, digital audio broadcasting (DAB), digital television and several experimental Video-on-Demand systems among others. MPEG-4 is a multimedia representation standard that models audiovisual data as a composition objects. MPEG-4 also supports the mobile multimedia. The MPEG-7 standard, called as the "Multimedia Content Description Interface", provides standardized tools for description of multimedia content.

3.12 Recommended Readings

1. Gilbert Held: **Data and Image Compression**, 4th Edition, John Wiley and Sons, Inc. 1996.
2. D. Santa Cruz, T. Ebrahimi, J. Askelof, M. Larsson and C. Christopoulos: **Coding of Still Pictures: JBIG and JPEG**, ISO/IEC JTC1/SC29/WG1 (ITU-T SG-8), N1816, July 2000. (Status: Informational document) This is based on two papers:
 - D. Santa Cruz, T. Ebrahimi, J. Askelof, M. Larsson and C. Christopoulos: **JPEG 2000 Still Image Coding versus Other Standards**, Proceedings of the SPIE. Vol. 4115.
 - D. Santa Cruz, T. Ebrahimi: **An Analytical Study of JPEG 2000 Functionalities**.
3. ISO/IEC: **JPEG 2000 Image Coding System: Core Coding System**, WG 1 N 1646, March 2000, available at the URL: <http://www.jpeg.org/FCD15444-1.htm>.
4. ISO/IEC: **Information Technology – Coding of Audio-Visual Objects, Part-2: Visual**, Dec. 1999.
5. W3C: **PNG (Portable Network Graphics) Specification**, Oct. 1996, available at the URL: <http://www.w3c.org/TR/REC-png>.
6. ISO/IEC: **Information Technology – Lossless and Near-Lossless Compression of Continuous-Tone Still Images: Baseline**, Dec. 1999.
7. ISO/IEC: **Information Technology – Coded Representation of Picture and Audio Information – Progressive Bi-Level Image Compression**, May 1993.
8. Bohdan O. Szuprowicz: **Multimedia Networking**, McGraw-Hill, Singapore, 1995.
9. Nalin K. Sharda: **Multimedia Information Networking**, Prentice-Hall New Jersey, 1999.

3.13 Exercises

1. Study the videoconferencing solutions offered by any three major solution providers (say PictureTel, Microsoft and Compression Laboratories) and

compare them in terms of compression efficiency, convergence rate, video-quality, underlying technique and limitations.

2. Many Codecs (coder-decoder combine) accept analog audio and video signals from the video cameras, sensors, microphones etc. and compress them after digitization. These compressed digital video signals may be transmitted over the MMI and at the receiving end, entire process is reversed and the resultant analog audio and video signals may then be sent to the playback devices. Survey the Codec product sites and identify five major Codecs that claim to offer the video quality comparable to the high quality of the broadcast TV services. Are they cost-effective over commonly available MMI links for WANs?
3. NI-GIF, I-GIF and Animated GIF have come out of a common GIF format. What is that format and exactly how these effects are derived with its help? Also comment on the preferential class of internetworking applications where these forms could be possibly employed in an efficient way.
4. Comment on the comparative features of JPEG-2000 and currently prevalent variants of JPEG from the viewpoint of compression-control, quality of compressed image, compression and decompression speeds and bandwidth requirements for WAN-oriented image transfers.
5. MPEG-1 standard is a five-part standard namely ISO/IEC 11172-1:1993 Systems, ISO/IEC 11172-2:1993 Video, ISO/IEC 11172-3:1993 Audio, ISO/IEC 11172-4:1995 Compliance Testing, ISO/IEC 11172-5:1998 Software Simulation. Some of these parts have modifications issued on later dates, latest one being introduced as recently as 1999.
6. Discuss the internal functional structure of the ISO/IEC 11172-x Encoder and Decoder with the help of necessary diagrams and briefly comment on the Temporal MPEG-1 image structure.
7. MPEG-2 standard is currently a nine-part standard (ISO/IEC 13818-1 through 13818-10: one part being withdrawn). Briefly discuss the structure of a sample MPEG-2 Audio Data Block.
8. Show a Reference Configuration for the Real-Time Interface to the ISO/IEC 13818-x Transport Stream Decoders with the help of a simple diagram.
9. Briefly compare JPEG and JPEG-2000 standards in terms of Internet-based still-image delivery considerations

Chapter-4

The Intelligent Agent Technology in Internetworking

Interaction Goals

Learning objectives of this chapter include a sound understanding of fundamentals of the Agent and Intelligent Agent Technologies, realization of the impact of Intelligent Agents and Mobile Agents on internetworking practices, taking a look at the currently prevalent practices and evolving research directions.

At the end of this chapter, you should be able to:

- Identify the situations wherein Softbots, Centralized Agents, Distributed Agents, Mobile Agents and their Intelligent counterparts may be cost-effectively used,
- Select the right Agent technology for the right occasion,
- Understand the performance and security trade-offs involved in certain combination of technologies.

The treatment presupposes knowledge of Java as a programming language and some initial grounding in Artificial Intelligence.

4.1 Introduction

There do exist entities that with or without our knowledge help us or our applications or even our computing systems to exhibit their best-expected *performance*. These may be often small in size and get automatically into an action whenever the associated events take place or even in a pro-active manner, at times! These are not *Operating Systems* or top-level applications but merely *intermediaries* called *Agents*. This chapter introduces you to this world of *Agents* and their more complex tribe called *Intelligent Agents* in the context of *Internetworking*.

4.2 Intelligent Software Systems

An *Intelligent Software System* may be defined as a software system that incorporates the *knowledge-based technology* along with one or more *adaptive technologies*; and, that may be seen as comprising of *intelligent agents* that continuously or periodically or in an event-triggered manner perform one or more of the following functions:

- **Perception** of dynamic state of the environment of application,
- Action to effect select conditions in the environment of operation,
- **Reasoning / interpretation** of its perceptions,
- Derivation of rational **inferences**,
- Automatic or pre-programmed **actions** based on drawn inferences and system / environment configuration control parameters.

An *Intelligent System* is often a **knowledge-based multi-agent system**.

4.3 Intelligent Agents

An *Intelligent Agent* is an entity that may be used to accomplish a set of pre-defined tasks in a manner that may be transparent to the applications and user and which may involve handling known, partly known or unknown situations.

A powerful strategy of enhancing *performance* of traditional standalone *knowledge-based computing systems* is to put / place the system in a *society of systems* where it can draw on the expertise and capabilities of itself and others in the society. The overall *distributed artificially intelligent system*, in such a case, becomes simply a *co-operative collection of individual modules that interact with one-another for solving a complex but ill-structured problem*. Each of these modules is then an *Intelligent Agent* in its own right.

As referred above, an *Agent* may be of non-intelligent type or of intelligent type; and, in general may be seen as belonging to one of the following basic classes:

- *Gopher Agents*
- *Predictive Agents*
- *Service Agents*
- *Leader / Pro-active Agents*
- *Adaptive Agents*

Simple *Gopher Agents* act as an intelligent or non-intelligent *interface agent* that may, for instance, act as appointment tracking and alerting system.

Service Agents are explicitly designed to perform specialized services. *Webbots*, *Taskbots*, *Userbots*, *Schedulerbots* etc. are a few examples of this type.

Data Mining Agents and *Intelligent Monitoring Agents* are the examples of the *Predictive and Proactive Agents*.

Chatterbots, *Searchbots*, *Spambots*, *Spiderbots*, *Jobots*, *Newsbots*, *Hotbots*, *Clonebots*, *Musicbots*, *Modbots*, *Docbots*, *Javabots*, *Annoybots*, *Knowbots*, *Mailbots*, *Shopbots*, *Combots* etc. are other possible *agents over the Internet*. *Chatterbots* are *Chat Agents*, *Searchbots* are used for generic web-based searches, *Spambots* are used by malicious attackers to spam a designated mailbox with unsolicited mails while *Spiderbots* are web-based variant of *Searchbots* with a specialized mission of constantly / periodically looking for new relevant pages / sites (and / or pages / sites wherein the contents have changed since last search / update). *Jobots* are Job-search Agents, *Newsbots* are meant to collect relevant news and *Hotbots* are used to locate 'predefined' category of so-called 'hot' content pages / sites (what is treated as hot may vary from situation to situation). *Clonebots* are agents employed for cloning the programmes, *Musicbots* locate designated music strips / pieces while *Modbots* are agents used to modify pages / sites in a preferred manner. *Docbots* are used to locate Medical Doctors on-line, *Javabots* are employed to locate Java Applets over the Net and *Annoybots* are used by the malicious attackers for creating annoyance / disruption. Similarly, *Knowbots* or *Knowledgebots* are agents looking for a specific category of knowledge over the Web, *Mailbots* are used to

manage / monitor / filter electronic mails, *Shopbots* are *Shopping Agents* and *Combots* are employed for network / internetwork communication. Anderson Consulting's 'BargainFinder' and Excite's 'Jango' are examples of *Shopping Agents / Shopbots*.

4.4 Attributes of Intelligent Agents

Intelligent Agents are expected to display certain degree of *autonomy* and must exhibit an *appropriate sub-set of following behavioural attributes* depending upon the *goal(s)* of the respective agent:

- *Ability to act at its own*
- *Capability to respond to a situation*
- *Capability of taking Initiative*
- *A subset of what is perceived as Human Intelligence*
- *Goal Directed Behaviour*
- *Network / Internetwork Mobility*
- *A subset of what is perceived as Rationality*
- *Selectivity and Interactivity*
- *Fault-tolerance*
- *Avoiding propagation of false data / information*
- *Capability of avoiding conflicting goals of other Agents where possible*

4.5 Intelligent Architectures

Principal *Intelligent Agent architectures* can be roughly classified as *Deliberative, Reactive* and *Hybrid Architectures*.

Deliberative Architectures

Agent architectures based on explicitly represented *symbolic models* and well-defined courses of action for each of the specified goals belong to this class. The deliberative architecture based agents may involve *behavioural traits and their representations*. Such agents often use *logical / symbolic reasoning* schemes.

Reactive Architectures

Agent architectures not based on any explicit (primary) symbolic model and symbolic reasoning but having the *capability of responding in an interactive environment* belong to this class.

Hybrid Architectures

These architectures *inherit properties from both* of the above-referred classes of architecture.

In hybrid architecture based agents, *for complex tasks the power of deliberative model* is used whereas *for handling routine trivial tasks reactive scheme's* simplicity and quick

response capability show good results.

4.6 Internetworking Applications of Intelligent Agents

Apart from the situations discussed in Sections 4.4 and 4.5, there may exist many possible avenues of *Agent deployment*. These include the following:

- *Network / Internetwork Monitoring*
- *Traffic Management*
- *Security Management*
- *Privacy Management*
- *QoS Management*
- *Failure Management*
- *Response Customization*
- *Pro-active Marketing*
- *Access History based disaster avoidance*

4.7 Role of Agents

Agent Technology can play a variety of *roles* including those classified in the preceding discussion. In most of the real-life situations in the world of *Multimedia Internetworking*, an *Agent* is used because of several reasons including the following:

- Many applications are *inherently distributed* and therefore, they are not well suited to a centralized scheme of things. *Internetwork Management / Network Management / Collaborative Multimedia Content Development / Virtual Employee Coordination* are some such examples.
- Since the work is distributed amongst the individual Agent's *architectural and structural modularity* becomes an inherent feature of such systems.
- As the work gets divided, naturally, *for complex or compute intensive tasks, overall processing / response time is appreciably reduced* in most of the cases. In other words, system often responds faster than before!
- *Higher efficiency* is another motivating factor for use of *Agents* in most of the cases.
- *Capability of reasoning in a heterogeneous environment* is a major reason behind adoption of *Intelligent Agents*.
- *Higher degree of reliability and robustness* provided by a set of well-designed *Agents* often reduces the cost of maintenance in unattended application segments.

4.8 Components of IA based Distributed Systems

Intelligent Agents are the integral parts of *Distributed Reasoning Architectures*. Such architectures have the following basic components:

- *Agent Co-ordination Strategy / Policy*
- *Inter-Agent Communication Mechanism*
- *Distributed Locality-and-Task-specific Reasoning Methods*
- *Conflict Resolution Mechanism*

Co-operation and co-ordination amongst Intelligent Agents is possible to be arrived in many ways including the *schemes that may be typically divided into following basic classes*:

- *Supervisory Agent based Agent co-ordination Scheme*
- *Master / Slave Co-ordination Scheme*
- *Centralized Assignment Decomposition scheme*
- *Shared Goal based Agent co-ordination schemes*
- *Competing Goals based Agent co-ordination schemes*

4.9 Other Aspects of Intelligent Agents

Centralized Assignment Decomposition scheme based implementations involve the following stages by the *Supervisory Agent*

- Problem Analysis,
- 'Agent Capability Analysis / Survey' and 'Load-balancing',
- Decomposition of the Problem / Assignment,

Assigning different parts of the assignment to individual Agents (I.e. suggesting sub-goals of the overall goal to various Agents),

- *Synchronization of distributed tasks,*
- *Collecting results of individual task-executions.*

The ***Contract Net Mechanism*** based ***Assignment Decomposition and Inter-Agent Negotiation*** (suggested in 1983 by Davis and Smith) involves Agent's acting as:

- Supervisor / Manager
- Contractor / Sub-contractor

Steps involved in this case may include:

- *Task Broadcast*
- *Distributed Task Evaluation*
- *Contract / Sub-contract Bidding*
- *Contract Assignment / Award*
- *Result Collection*

Transportable Agent Environments / Systems including the following are specifically appropriate for *specialized but constraint-driven applications*:

- MIT's Sodabot
- General Magic's Telescript
- Microsoft's ActiveX
- IBM's Aglet
- University of Kaiserslautern's ARA
- DEC's Oblique
- Objectspace's Voyager

In the following section, one of these *Agent Architectures* shall be discussed in some detail.

4.10 IBM Aglet Technology Architecture

Aglet is an *autonomous software agent technology* developed by the IBM that uses Java. The *Java Aglet* technology effectively extends the *model of mobile (over the network / internetwork) code* already witnessed in the *Java applet* technology. Like a *Java applet*, the class files for a *Java Aglet* can move within internetworks; but unlike the former, when a moving *Aglet* carries its *state*. *Since an Aglet carries its state, it can travel, as directed or required, to many destinations on a network and may even return to its point of start of journey.*

An *Aglet* is run as a thread inside the context of a host application. An *Aglet* needs a host Java application called as *Aglet Host*, executing on a computer / node for visiting that computer / node. When *Aglets* travel across a network, they migrate from one *Aglet host* to another. Each *Aglet* host installs a *security manager* to enforce *restrictions* on the activities of *untrusted Aglets*. Hosts upload *Aglets* through *class loaders* that know how to retrieve the *class files* and *state of an Aglet* from a *remote Aglet host*.

An *Aglet* may be:

- *Created*
- *Cloned*
- *Dispatched*
- *Retracted*
- *Activated*
- *Deactivated*
- *Disposed of*

Aglet hosts use *object serialization*, available in JDK / SDK with the *RMI (remote method invocation)* add-on, to *export the state* of an *Aglet object* to a *stream of bytes*. Through this process, the *Aglet object* and the *tree of serializable objects* reachable from it, are written to a stream. The *Aglet-state* can be reconstructed from the *stream of bytes*. The state of the execution stacks and program counters of the threads owned by the *Aglet* are not serialized. *Object serialization uses only data on the heap*. Therefore, when an *Aglet* is *dispatched, cloned, or deactivated, stack-based State information* of it, as well as *Program Counter* of any associated thread is lost.

Unlike expectations of the theoreticians interested in the behaviour of *Classical Mobile Agents*, *Aglets* are incapable of *retaining their state* after cloning etc. due to the reasons discussed above. This behaviour of *Aglets* is due to the *architecture of the JVM*, which doesn't allow a program to directly access and manipulate *execution stacks*. This is part of the *JVM's built-in security model*. Unless there is a change to the *JVM*, *Aglets* and any other *Java-based mobile agents* will be unable to carry the *state of their execution stacks* with them as they *migrate*. The inability of an *Aglet* to migrate with its execution stacks is not really an unreasonable limitation. *Aglet may be modeled as an FSM with the heap as the only carrier of the state.*

The *Aglet* development and run-time environments provide a library of Java classes for creating and executing the *Aglets*. To create an *Aglet*, we have to subclass class *Aglet* that includes methods, which can be overridden to customize the behavior of an *Aglet*.

The Aglet's counterpart to the `init()` method of applets is the `onCreation()` method. To initialize an Aglet, `onCreation()` is overridden. The `onCreation()` method is invoked only once in an Aglet's lifetime and should be used only for initialization. The Aglet also has a `run()` method representing the entry point to its main thread. Before any major event, a "callback" method is invoked to allow the Aglet to respond to the call of the event. An Aglet learns about its serialization just ahead of its use. The method `onDispatch()` is a "callback" method because the Aglet host invokes it some time after another method, `dispatch()`, is invoked. An Aglet can invoke `dispatch()` on itself or on another Aglet, with due authorization.

4.11 The Stanford's JAT Technology Architecture

The *Agent building* has come of age now although the technology is still evolving very rapidly. Apart from the *Aglet Development Toolkit* provided by the IBM, there do exist a few other solutions resembling the *RAD tool-like* features. The *JAT Lite* tool developed at the *Stanford University* is one of the earliest *generic agent creation tool-kits*.

The *JAT Lite* is, in fact, a set of Java Programs intelligently packaged together. It aims at providing an infrastructure that may be interpreted as:

- ❑ *a programmer's workbench,*
- ❑ *an AMR (Agent Message Router) facilitator,*
- ❑ *an execution and debugging environment,*
- ❑ *a KQML based communication facility.*

Although, originally designed to facilitate creation, testing, debugging and tuning of *Agents capable of communicating using the KQML*, the *Stanford JAT model* can be easily adapted for building any *ACL (Agent Communication Language)* based agents. Like the *Aglet Development Tool-kit*, the *JAT Lite* can run on any *JDK / SDK* supporting platform. *Agents* created by this tool are capable of utilizing *TCP / IP* and *Socket-based communication*.

As shown in Fig. 4.1, the *JAT Lite Architecture* is a *five-layer architecture*:

- The *Abstract Layer* is responsible for making the *abstract classes* required for implementation.
- The *Base Layer* caters to the *communication-specific requirements* as imposed by the *protocol model* supported (TCP / IP) and as supported by the *Abstract Layer*.
- The *KQML Layer* is responsible for providing support for *KQML-specific message generation, parsing, validation, storage and retrieval*. Using the *extended KQML* provides *registration support* and additional features.
- The *Router Layer*, as the name rightly suggests, is primarily responsible for *routing Agent Messages*. Additional functionality of this layer includes *Name Registration, Crash Recovery* and providing expected support to the *Protocol Layer* sitting atop it.
- The *Protocol Layer* is primarily responsible for providing *higher-level protocol support* as may be necessary for any given *Inter-Agent Communication over the*

Internet, intranets or extranets. Protocols like SMTP, HTTP and FTP are already supported by the current version of the JAT Lite.

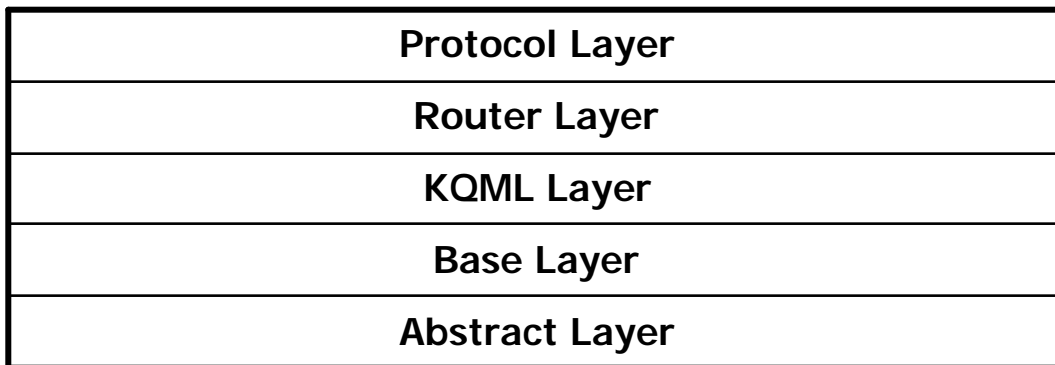


Fig. 4.1: The Stanford's JAT Architecture

4.12 The JAFMAS Technology Architecture

The acronym *JAFMAS* stands for the *Java-based Agent Framework for Multi-Agent Systems*. Unlike the *Stanford JAT*, it is particularly suitable for the Carl Searle's '*Speech-Act-based Multi-Agent Systems*'. This environment aids the development of *scalable, autoconfigurable and fault-tolerant MAS*. The *JAFMA*, like the *JAT*, provides a *general purpose Agent development support*.

This scheme enforces a *five-stage development methodology*:

- *Identification of agents,*
- *Identification of dialogues / conversations to take place amongst agents,*
- *Identification of rules and conventions to be followed during conversations,*
- *Analysis of the resulting Conversation Model &*
- *Implementation of the Multi-Agent System.*

4.13 Summary

Agents have been used in networking applications since ages. Due to recent impact of the Business AI, a renewed interest in internetworking agents of ordinary as well as intelligent classes is being witnessed at present. There do exist several competing technologies to build stationary as well as mobile agents of both categories. Java-based agents are fast becoming developers' choice. KQML-Java combine offers yet another interesting feature-rich set of capabilities in development and communication of agents over the internetworks.

An *Intelligent System* is *often* a knowledge-based multi-agent system. Simple *Gopher Agents* act as an intelligent or non-intelligent *interface agent* that may, for instance, act

as appointment tracking and alerting system. *Service Agents* are explicitly designed to perform specialized services. *Data Mining Agents* and *Intelligent Monitoring Agents* are the examples of the *Predictive and Proactive Agents*.

Principal *Intelligent Agent architectures* can be roughly classified as *Deliberative, Reactive* and *Hybrid Architectures*.

The deliberative architecture based agents may involve *behavioural traits and their representations*. Such agents often use *logical / symbolic reasoning* schemes.

Capability of reasoning in a *heterogeneous environment* is a major reason behind adoption of *Intelligent Agents*. *Intelligent Agents* are the integral parts of *Distributed Reasoning Architectures*.

Although, originally designed to facilitate creation, testing, debugging and tuning of *Agents capable of communicating using the KQML*, the *Stanford JAT model* can be easily adapted for building any *ACL (Agent Communication Language)* based agents. The acronym *JAFMAS* stands for the *Java-based Agent Framework for Multi-Agent Systems*. The *JAFMA*, like the *JAT*, provides a *general purpose Agent development support*.

Agents have been used in networking applications since ages. Java-based agents are fast becoming developers' choice.

4.14 Recommended Readings

1. A. Amandi and A. Price: **Object Agent Programming through Brainstorm System**, Proceedings of the PAAM '97 Conference, London, April 1997.
2. B. Grosz: **Building Commercial Agents: An IBM Perspective**, Invited Paper, PAAM '97 Conference, 1997.
3. B. Hayes-Roth: **Architecture for Adaptive Intelligent Agents**, Artificial Intelligence, Vol. 72, No., 1995, pp. 329-365.
28. B. Pell: **Agent Architectures for Autonomous Control Systems**, Tutorial at the PAAM '97 Conference, London, 1997.
4. B. Quendt: **An Agent-Based Resource Control of the Signaling System for the Open Telecommunication Market**, Proceedings of the PAAM '97 Conference, April 1997.
5. C. Leckie et al: **A Multi-Agent System for Distributed Fault Diagnosis**, Proceedings of the PAAM '97 Conference, London, April 1997.
6. D. Martin et al: **Information Brokering in an Agent Architecture**, Proceedings of the PAAM '97 Conference, London, April 1997.
7. D. Pinnard, M. Wiess and T. Gray: **Issues in Using an Agent Framework for Converged Voice / Data Applications**, Proceedings of the PAAM '97 Conference, London, April 1997.
8. Donald Steiner: **Issues in Agent Interaction**, Invited Paper, Proceedings of the PAAM '97 Conference, London, April 1997.
9. H. Baumgartel, S. Bussmann and M. Kolsterberg: **Multi-Agent Coordination**

- of Material Flow in a Car Plant**, Proceedings of the PAAM '97 Conference, London, April 1997.
10. J. Arthursson et al: **A Platform for Secure Mobile Agents**, Proceedings of the PAAM '97 Conference, London, April 1997.
 29. J. Bradshaw: **Software Agents: The Next Generation**, Tutorial at the PAAM '97 Conference, London, 1997.
 11. J. Dospisil, E. Kendall and T. Polgar: **Multimedia Presentation Scheduling with ILOG**, PAP '97 Conference, London, April 1997.
 12. J. E. Whatley and P. J. A. Scown: **Simultaneous Multiple Agents Working in Real-Time: From Interaction Framework to Prolog**, PAP '97 Conference, London, April 1997.
 13. John Fox: **Intelligent Agents Which Reason About Beliefs, Decisions and Plans: Logical Foundations and Practical Applications**, Invited Paper, PAP '97 Conference, London, April 1997.
 14. Marvin Minsky: **Society of Mind**, Simon & Shuster, New York, 1980.
 15. N. Bensaid and Ph. Mathieu: **A Hybrid and Hierarchical Multi-Agent Architecture Model**, Proceedings of the PAAM '97 Conference, London, April 1997.
 16. N. R. Jennings: **Agent Software**, Preprint, QMWC, Univ. of London, 1995. (Subsequently published.)
 17. P. Charlton et al: **An Open Architecture Supporting Multimedia Services on Public Information Kiosks**, Proceedings of the PAAM '97 Conference, London, April 1997.
 18. P. Ciancarini, D. Rossi, F. Vitali, A. Knoche and R. Tolksdorf: **Coordinating Java Agents for Financial Applications on the WWW**, Proceedings of the PAAM '97 Conference, London, April 1997.
 30. Pattie Maes: **User-facing Software Agents**, Tutorial at the PAAM '97 Conference, London, 1997. <http://www.demon.co.uk/ar/PAAM99/>
 31. Rahul Banerjee: **An Intelligent System for Behavioral Analysis**, Ph. D. Thesis, AU, Amt., 2001.
 19. Raj R. Reddy: **Challenges in the AI**, ACM Computing Surveys, Vol. 27, No. 3, 1995, pp. 301-303.
 20. Richard Murch & Tony Johnson: **Intelligent Software Agents**, Prentice-Hall PTR, New Jersey, 1999.
 21. S. Mitaim and Bart Kosko: **Profile Learning with Neural Fuzzy Agents**, Proceedings of the PAAM '97 Conference, London, April 1997.
 22. T. S. Dahl, S. Pearson and C. W. Priest: **An Agent Communication Platform in Object Oriented Prolog**, PAP '97 Conference, London, April 1997.
 32. Tim Finin: **Agent Communication Languages: KQML, KIF and the Knowledge Sharing Approach**, Tutorial at the PAAM '97 Conference, London, 1997.
 23. V. Braun, B. Steffen and H. Wendler: **Service Definition of Intelligent Networks: Experience in a Leading-edge Technological Project Based on Constraint Techniques**, PAP '97 Conference, London, April 1997.
 24. V. V. S. Sarma: **Intelligent Agents**, Journal of IETE, Vol. 42, No. 3, 1996, pp. 105-109.
 25. Y. Han et al: **Agents for Citation Finding on the World Wide Web**, Proceedings of the PAAM '97 Conference, London, April 1997.

4.15 Exercises

1. Design and implement a User Agent for any well-defined Data Mining purpose.
2. Design and implement a Mobile Agent that could monitor network congestion and warn about possible / impending failure.
3. What are the situations in which you would use both: IBM's Aglets and Stanford's JATLite-based Agents and why? You may consider any Internet-based solution that shall benefit from their co-existence. Please clearly explain how these two types of Agents shall communicate with each other / one-another?
4. What is the primary strength of the Stanford JAT Lite? Can it be used for the purpose of developing an Internetwork Information Management software system? Please discuss in brief
5. Suggest a suitable internetworking application in the real world that could warrant use of the MAS technology. In which manner shall the JAFMAS technology be applied herein?
6. Explain the architectural details of the IBM's Aglet technology in contrast to that of the JAFMAS. Which one would you use for the development net-centric Agents and why?
7. If you need to design an Agent-based Security Alert System for your Intranet, which factors would you consider on a priority basis and what influence they will have on your choice of an appropriate Agent Architecture? Discuss in detail.

Chapter-5

The TCP/IPv6 Internetworking Architecture

Interaction Goals

Interaction Goals of this chapter include developing an understanding of the internals of the TCP/IPv6 Architecture involving the Transmission Control Protocol (TCP) and Internet Protocol Version 6 (IPv6) from a designer's perspective. We look at the IPv6 from the application developer's viewpoint and brief analysis of the Plug-and-Play support in IPv6. We shall also take a quick look at the accepted industry practices and evolving trends.

At the end of this chapter, you should be able to:

- Understand the internals of the TCP/IP Architecture,
- Identify functionalities, design goals and issues related to the IP Layer and IP Header Structure,
- Choose one of the TCP and UDP Protocols based on application's requirements.
- Compare the TCP/IPv4 and TCP/IPv6 stacks.
- Differentiate between the characteristics and capabilities of IPv4 and IPv6,
- Identify the areas where IPv6 may be a preferred technology,
- Identify the gray areas of IPv6 implementation;
- Understand the pros and cons of automatic address configuration (plug-and-play) support and
- Use IPv6 for use in soft-real-time internetwork designs.

The prerequisites are some exposure to networking environments and components.

5.1 Introduction

Gone are the days when the term Internet referred to an internetwork connecting Computer Science departments of select privileged universities and the Advanced Research Project Agency of the U. S. Department of Defense. The ARPA Internet, which later evolved into the worldwide Internet of today, no longer exists but its legacy survives. The TCP / IP architecture owes its architecture to this rich heritage. Despite its several weaknesses and vulnerabilities, the TCP/IP protocol family has survived the test of the time and played a very important role in making the Internet what it is today. Yes, even as you read this piece, IP or IPv4 in its three-decade old form (the Internet Protocol Version 4 is often called IPv4 or simply IP as described in the RFCs 760 and 791), rules the world. Its successor has arrived in 1998 and has been specified in the RFC 2460. Though proposed originally much earlier, by different people in different forms, it may take quite some time before the world really talks IPv6. Why then we focus here on this

new and yet unproven technology? This is due to several reasons, starting with a large (and inexhaustible-in-near-future) address space (128-bit) to its thoughtful support for time-sensitive traffic and provision for greater security. The list is long. Moreover, it has the potential to support the hybrid (multimedia included) traffic. Not many people view the ATM technology as a threat for the IP technology. Each of these technologies has shown potential to complement each other in terms of efficiency, cost and reliability.

5.2 The TCP/IPv6 Architecture: An Introduction

As shown below, the TCP/IP architecture is a four-layer system defined by the erstwhile ARPANET and the IAB. Each of these layers has its well-defined set of functionalities. The layers in the TCP/IP Architecture are (bottom up):

- Host-to-Network Interface
- Internet Protocol (IP) Layer
- Transmission Control Protocol (TCP) / UDP Layer
- Application Layer (AL)

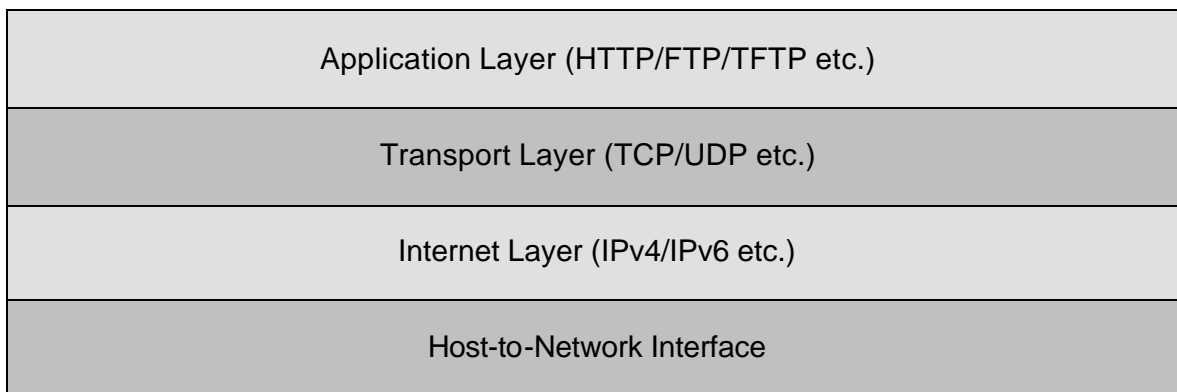


Fig. 5.1 The TCP/IP Architecture

5.2.1 The Application Layer

The Application Layer is responsible for a set of functions commonly required by various applications. It uses a set of protocols for carrying out these functions. Examples of some application layer protocols include protocols like HTTP, FTP, Telnet and TFTP etc. Virtual Terminal Emulation and similar functionality are included in the list of responsibilities of the Application Layer. It can offer both, connection-oriented as well as connectionless services.

5.2.2 The TCP/UDP Layer

The TCP / UDP Layer is actually a transport layer. It is responsible for receiving the data from the upper layer (AL) and, if so needed, dividing it into manageable chunks for the purpose of further processing and onward transmission via the IP Layer after prefixing its own header to the processed data. *At the other end, this operation is reversed when this*

layer receives data from the IP Layer and after due processing passes it on to its upper layer (AL). Other activities of this layer include creation of network connections as per the transport connection requests by the upper layer. In this scheme, TCP (Transmission Control Protocol) offers connection-oriented service whereas the UDP (User Datagram Protocol) offers connectionless service. (Fig. 5.2 and Fig. 5.3 show the header structures of the respective protocols.)

The TCP Header Structure

Source Port Number (16-bit)	Destination Port Number (16-bit)
Sequence Number (32-bit)	
Acknowledgement Number (32-bit)	
HLEN + Reserved + Code	Sliding Window
TCP Checksum (16-bit)	Urgent Pointer (16-bit)
Options (if present) + Padding)if needed)	
Payload Data	

Fig. 5.2: The TCP Header Structure

The UDP Header Structure

Source Port Number (16-bit)	Destination Port Number (16-bit)
Message Length (16-bit)	UDP Checksum (16-bit: Optional)
Payload Data	

Fig. 5.3: The UDP Header Structure

5.2.3 Internet Layer:

Internet Layer is a layer of the TCP/IP Architecture that is primarily concerned with getting Packets from the source node and delivering it to the intended destination node (through any number of intermediate nodes). In IP terminology, a Network Layer Data Unit (NLDU) called a Packet.

The Internet Layer primarily deals with:

- Accepting TPDU's from the Transport Layer through the Service Access Points (SAPs),
- Deciding route (for further transmitting this TPDU after encapsulating it within Packet),
- Passing this packet through the Service Access Points (SAPs) to the lower layer,
- Accepting NLDU / Packets from the lower layer through the SAP and Processing these NLDU / Packets,
- Removing the encapsulation and passing the TPDU through the SAP to the upper layer,
- Providing support for connection-oriented / connectionless services as the case may be (depending upon the protocol stack and need) and
- Providing diagnostic support for network monitoring, configuration, management and trouble-shooting.

In a nutshell, packet handling, packet management, Routing are the major responsibilities of the Internet Layer. In the context of packet routing, this layer's structural design goals include ensuring the shortest possible delay and thereby the highest throughput at the least possible cost, ensuring acceptably reliable packet delivery (may be optional in some cases), ensuring secure packet delivery (may be optional in some cases)

Major Issues Related to the Design of the Internet Layer include:

- Choice of the Services to be provided by the Internet Layer to the Transport Layer and their nature: The primary issue here, in the context of the OSI Reference Model, is the choice of any one of the Connection-oriented or Connectionless types of service. However, in the TCP/IP Architecture, connectionless services have been mandated for the IP Layer and this choice has been shifted to the upper layer.
- Services to be provided by the Internet Layer to the lower layer (Data Link Layer in the OSI terminology) and their nature: The primary issue here is the format and size of the Packet in which the data is to be passed to the lower Layer with or without encapsulation and additional information.
- Architecture and internal organization of the Internet Layer to be able to meet the design goals: In this case, the real issue is choice of the mechanism that

would satisfy the primary design goals.

- Choice of Interior and Exterior Routing and Protocol Translation schemes: Here the role of the Router, its location in the Autonomous System hierarchy and level of trust with the adjacent nodes influence the choice.
- Choice of Security to be provided at the subnet level: Choices here may include optional or compulsory support for encryption and authentication.
- Choice of conventional / mobile / hybrid routing support: The choice in this case is primarily need-based as this affects the scope as well as the cost.
- Choice of support for Quality-of-Service (QoS) and Faster Than Real-Time (FTRT) processing: These choices are to be made based on the intended class of applications.

5.2.4 Host to Network Interface

This acts as an interface between the Network Interface Card (NIC) and the IP-Layer. For instance, this may be seen as an interface that sits between the IP and underlying IEEE 802.x compliant NIC controller (like IEEE 802.3, IEEE 802.4, IEEE 802.5 and IEEE 802.11 etc.)

The Physical Layer inside the NIC is responsible for receiving the data from the Data Link Layer, converting it into equivalent signal (representing the data in bits) and transmitting these signals in the desired manner over a shared or dedicated transmission link. This layer is also concerned with the mechanical issues like connector dimensions, inter-pin distance, mechanical strength needed etc. Issues like physical connection establishment, direction of transmission, frequency usage and other procedural matters are under its purview.

The Data Link Layer inside the NIC is responsible for receiving the data from the Network Layer, process it, insert the processed data into Data Frames, add control information to it by prefixing a header and suffixing a trailer to the processed data; and, finally pass it on to the Physical Layer for actual transmission in signal form. *This operation is reversed at the receiving end.*

MAC-sub layer of the DLL implements the Access Control Policy for coordinated usage of the Medium of Data Transfer and ensures acceptably error-free transmission, flow control, traffic direction regulation etc. in case of shared media systems.

5.3 The Internet Protocol

The Internet Protocol (IP) Layer is responsible for receiving the data from the TCP / UDP Layer, process it for finding out the required resources, if required-- divide the data into fragmented units, decide the route to be taken by the respective data units and passing the data to the underlying Host-to-Network Interface after prefixing its own header to it. In the other role, *at the receiving end for instance, this operation is reversed.* Routing decision can be based on a fixed / static routing policy or a dynamic (situation dependent) routing policy. Other functions of this layer include congestion control, assistance in address resolution, protocol translation and resource usage accounting. IP Layer offers only a connectionless service. The currently prevalent version of the IP is IP version 4 or IPv4 and therefore, when used today without any suffix, most people relate

IP to IPv4. The IP has been a case of immensely successful protocol that has gradually matured to deliver support for a variety of features, like support for optional encrypted payload, optional IP-level authentication, mobility-extension, stateless address autoconfiguration (in the latest version of the IP) etc. which were not intended when the protocol was first conceptualized. In fact, in the latest version of the IP, better known as IP version 6 or simply IPv6, provision has been made to permit optional quality-of-service by inclusion of flow-specification field (although, as of this writing, standard specification of the FL usage is yet to emerge).

The IP Header Structure

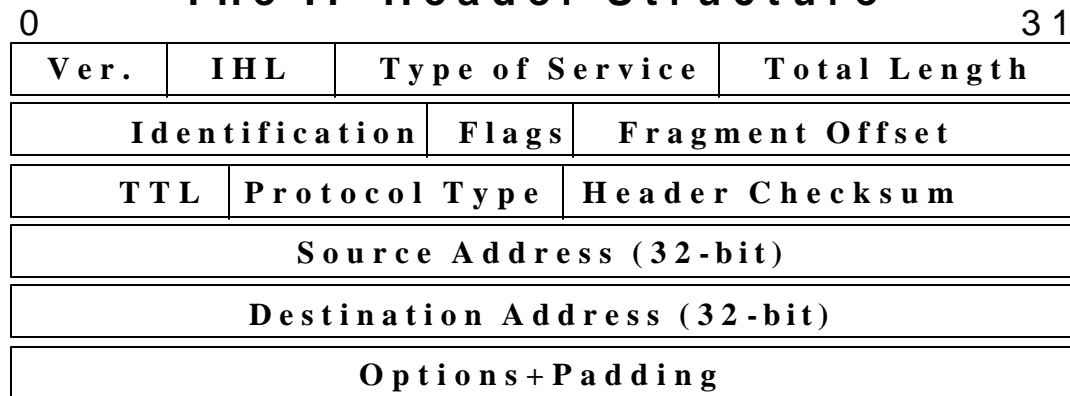


Fig. 5.4: The IPv4 Header Structure

IPv4 Field Name	Length in bits	Purpose
Version	4	It contains the Version Number, here, 4.
IHL / HLEN	4	Header length= IHL*4 <IHL is expressed in terms of 4-byte words.>
Type of Service (TOS) / DS <Often, this field is ignored by the routers.>	8	<i>Earlier:</i> [(3+1+1+1+(2))] bits referred to Precedence, Throughput-maximization bit, Reliability-maximization bit and unused bits. <i>Now:</i> [6+1+(1)] bits referred to Differentiated Service Classes, Cost-minimization bit and unused bit as sub-fields.
Total Length	16	Total length= [(IHL*4)+Payload Length]
Identification	16	IP Packet Sequence Number
Flags (unused, DF, MF)	3	Two flag bits are used as 'Don't fragment flag and 'More fragments flag' bits.
Fragment Offset	13	For packets > 64 K, this offset helps in reassembly.
Time to Live (TTL)	8	Meant to specify lifetime of a packet.
Protocol Type	8	Specifies about the next protocol header immediately following this header.
Header Checksum	16	Helps in identifying error in the header at each hop.

Table-5.1: A Summary of the Length and Purpose of IP Header Fields

5.3.1 IPv4 Options

The IPv4 header, as shown in the Fig. 5.4, is a variable-sized header primarily due to incorporation of the Options field. As shown in Fig. 5.5, the Option sub-field is divided into three parts: the 8-bit Code field (a 1-bit Copy subfield, a 2-bit Class subfield and a 5-bit number subfield), an 8-bit Length field and a variable-length Data field. Some of the legal options and their 5-bit codes as defined for the IPv4 are:

- No Operation (Code: 00001): a 1-byte option meant to be employed as filler,
- Security and Handling Option (Code:): meant to be used to specify security parameters,
- Strict Source Routing Option (Code: 01001): meant to specify complete route involving all routers that **MUST** be traversed to reach the destination,
- Loose Source Routing Option (Code: 00011): meant to specify one or more routers (not the complete route) that **MUST** be traversed to reach the destination,
- Record Route Option (Code: 00111): meant to record the entire route (done by allowing the every intermediate router along the route to add its 32-bit IP Address in the option-data area of the IPv4 header),
- Timestamp Option (Code: 00100): meant to allow every intermediate router to put its 32-bit IP address and a 32-bit timestamp in the options-data area of the header (often used for debugging purposes),
- End of Option (Code: 00000).

5.3.2 IPv4 and the World of Classes:

In the IPv4, any address is 32-bit long and is represented in four parts of one byte each separated by decimal points or dots. (This convention of addressing in IPv4 terminology is known as the Dot-Decimal Notation.) There exist two ways of looking at the IPv4 world: Class-based or Classful view (Classes: A, B, C, D, E) and Classless view.

The 32-bit address comprises of two parts: Network address / identifier and Host address / identifier. In the class-based version, the classes are designated based on the first few bits of the Network Address portion of the IP address. For instance, as shown below, if the first bit in this field is 0 (zero), it is a Class-A IP address; if the first two bits in this field are 10, it is a Class-B IP address; if the first three bits in this field are 110, it is a Class-C IP address and if first four bits are 1 110, it is a Class-D (Multicast) address.

0	Network Address	Host Address (3-octet)
10	Network Address	Host Address (2-octet)
110	Network Address	Host Address (1-octet)
1110	Multicast Address	

*Fig. 5.5: Four Major Classes of IPv4 Addresses
(Class-E, with prefix 1111: reserved, not shown here)*

In Class-A address, the first byte constitutes the Network Address and remaining three bytes constitute Host Addresses. In Class-B address, the first two bytes constitute the Network Address and the remaining two bytes constitute Host Addresses. In Class-C address, the first three bytes constitute the Network Address and the remaining byte represents the Host Addresses.

Example of a Class-A address: 12.0.0.3, (Network Address:12.0.0.0; Host Address: 0.0.0.3).

Example of a Class-B address: 180.16.0.1

Example of a Class-C address: 192.12.7.8

Class-A Address Range:

1.0.0.0 - 127.255.255.255

Class-B Address Range:

128.0.0.0 - 191.255.255.255

Class-C Address Range:

192.0.0.0 - 223.255.255.255

Class-D Address Range:

224.0.0.0 - 239.255.255.255

Class-E Address Range:

240.0.0.0 - 247.255.255.255

5.3.3 Concept of Subnetting and Supernetting

An IPv4 Subnetwork is often referred to mean a subset of one of the three IPv4 classes (A, B and C). A Subnet Mask is a sequence of bits that is used to separate Network and Host Addresses from each other. This mask divides the Address portion into another set of Network-Host Addresses.

Types of Subnetting:

- Fixed-Length Subnetting / Basic Subnetting
- Variable-Length Subnetting

Natural Masks

- Natural Mask for Class-A: 255.0.0.0
- Natural Mask for Class-B: 255.255.0.0
- Natural Mask for Class-C: 255.255.255.0

Mask help in separating network address from the host address, make subnetting

possible that in turn helps in fighting the IPv4 Address Depletion problem in some limited but effective way. Every LAN segment is usually associated with at least one network number (more are possible) and if no subnetting is done, only one segment can use a given network address.

Variable Length Subnet Masking (VLSM)

In this scheme, a given network can be masked with masks of different lengths thereby providing required flexibility of having network segments as required (instead of just dividing a given network into 'n' number of networks of equal sizes ---- as is the case with the fixed-length subnetting). All masks have a string of '1's to the left and string of '0's to the right.

Primary Objective of the Classless Inter-Domain Routing (CIDR) in IPv4 Subnets includes finding a temporary solution to the IPv4 Address space depletion. The basic idea behind the CIDR are two: (a)- Allocation of the unallocated set of Class-C IPv4 network addresses in variable-sized address blocks; and, (b)- While allowing 'a', in effect allocating contiguous Class-C IPv4 network addresses.

As per the RFC 1519 allocation rules, the whole world was suggested to be divided into four zones each of which could use nearly 32 Million Addresses:

- Asia-Pacific:
- Central-Southern America
- Europe
- North America

A set of nearly 320000000 addresses were suggested to set aside for future use. If a router 'X' get a packet that belongs to the IPv4 addresses of one these four zones, the packet is simply forwarded to the zonal gateway.

The Supernetting:

In principle, 'a network whose prefix-boundary has lesser number of bits than the natural mask of the network itself, is called a Supernet'.

Two ways to represent the same CIDR address are :

- 199.28.0.0/16
- 199.28.0.0 255.255.0.0

Terms 'Aggregation', 'CIDR Block allocation', 'Supernetting' etc. are often used interchangeably in the IPv4-CIDR literature.

In addition to the IPv4 addresses discussed above, there do exist a few reserved addresses. These *IPv4 Addresses having special meaning* include 0.0.0.0 denoting local

host (used during booting), 255.255.255.255 denoting the local network broadcast address, 127.x.y.z denoting local loopback testing addresses, 'Net-id followed by all binary '1's' denotes broadcast to a remote network address.

5.3.4 On the Internet Control Message Protocol (ICMP):

During the normal operation of the Internet, many a times, errors, crashes and some other unexpected events may occur. The protocol that reports these problems to the Routers is called the Internet Control Message Protocol (ICMP). ICMP is a protocol that is practically inseparable from the IP as both go together. ICMP messages are transmitted as payload of the IP header and this is why in the Fig. 5.6, ICMP has been shown placed in the same layer as IP but has been placed just above it. If an IP header is followed by an ICMP Message as its payload, the Protocol Type field of the IP header shall carry a code of '1'.

Some of the common ICMP messages include:

- Echo Request,
- Echo Reply,
- Timestamp Request,
- Timestamp Reply,
- Redirect, Parameter Problem,
- Source Quench,
- Destination Unreachable,
- Time Expired / Exceeded

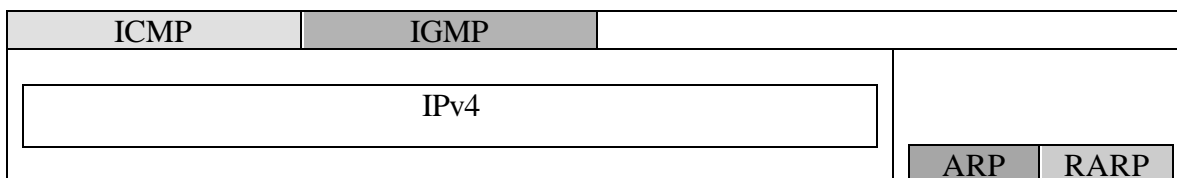


Fig.5.6: IPv4 and Its Accompanying Protocols: ICMP, IGMP, ARP and RARP (DHCP not shown here due to its location at a higher layer though its effect is typically visible at the same level as that of the ARP/RARP)

ICMP Messages are typically classified into two categories of ICMP Query (and / or Response) Messages and ICMP Error Messages.

5.3.5 On the Internet Group Management Protocol (IGMP):

Like the ICMP, until the version 4 of the IP, another companion protocol known as Internet Group Management Protocol (IGMP) was designed to support group-based

applications at the network level. It manages the groups locally and helps a multicast-capable router in creation and update of a list of group members for a relevant group contacting through a relevant interface. By itself, IGMP is *not* a Multicast Routing protocol. Its placement has been shown in the Fig. 5.6. IGMP messages are transmitted as payload of the IP header. If an IP header is followed by an IGMP Message as its payload, the Protocol Type field shall carry a code of '2'.

In the latest version of the IP, the accompanying ICMP has been extended to provide these functionalities and thus IPv6 and ICMPv6 put together take care of the functions that were handled by IP, ICMP and IGMP.

5.3.6 The Address Resolution Protocol (ARP)

All computers in the IP world must be associated with one IP address or the other. For the purpose of actual delivery of a packet, the packet has to be sent through the Host-to-Network layer which means, for actual transmission the association of a given IP address to the lower layer address (say an Ethernet address / MAC Sub-layer Address) is required. The protocol that permits a machine desiring to deliver a message at a particular IP address over a local link to enquire about the corresponding IP-address holder's MAC address (i.e. lower layer address) is called the 'Address Resolution Protocol' (ARP). This is a stateless protocol and does not require any address servers. Here the query can be responded simply through a *query-broadcast over the LAN and reply-unicast cycle*. Fig. 5.7 below shows the format of an ARP data unit.

Type of Hardware (HTYPE) 16-bit		Protocol Type (PTYPE) 16-bit
Length of Hardware (HLEN) 8-bit	Protocol Length (PLEN) 8-bit	Operation (OPER) <Request / Reply> 16-bit
Sender's Hardware <MAC> Address (SHA) Variable-length (e.g. 48-bit for IEEE 802.3)		
Sender's Protocol <Upper Layer> Address (SPA) Variable-length (e.g. Sender's IP Address)		
Target Hardware <Intended Recipient's MAC> Address (THA) Variable-length (e.g. 48-bit for IEEE 802.3) (Used in Reply, Not used in Request)		
Recipient's Upper Layer Address Variable-length (e.g. Receiver's IP Address)		

Fig. 5.7: Format of an ARP Data Unit

Once generated, an ARP data unit has to be encapsulated inside a lower-layer frame (e.g. an IEEE 802.x frame) and transmitted over the LAN.

5.3.7 The Reverse Address Resolution Protocol (RARP)

As indicated earlier, all computers in the IP world must be associated with at least one IP address that is associated with a MAC Sub-Layer address for the purpose of communication; therefore, a machine that knows just its hardware address will need to

learn / discover about the associated IP Address as well. The protocol that permits a machine holding its lower layer address (say its Ethernet Address) to enquire about its associated IP address is called the 'Reverse Address Resolution Protocol' (RARP). RARP is a stateful protocol and requires the presence of an RARP Address Server that could reply to any query by a node that owns and therefore knows a MAC Address and wants to learn the corresponding IPv4 address that it could use for building an IP packet where this newfound address could be used a source address. Like the ARP, here too, the query can be responded simply through a *query-broadcast over the LAN and reply-unicast cycle*.

ARP, RARP and at the higher layer, the DHCP, are intended to support the Dynamic Mapping of Hardware Address to IP Address and IP Address to Hardware Address. In case, there exists a statically configured setup, these protocols are not used.

5.3.8 Mobile IP

The variant of the Internet Protocol that has been specifically designed for providing support for Mobile Hosts willing to communicate over the Internet is called as the Mobile IP. It is the result of deliberations of a special IETF workgroup and has been described in the RFC.

The basic IP has an Addressing Scheme that comprises of Class Id., Network Number and Host Number. Therefore, any packet intended for a given Mobile Host shall have no problem as long as the MH stays on the Home LAN; since Routers all over the world can continue to use the Class plus Network Address information to route any information to it. The problem of discontinuation of service would arise as soon as the MH moves out of its Home Zone; since now, the Routers would still continue to send traffic meant for this MH to the Home LAN address they have in the know of!

One solution to this problem could have been assigning a new IP address to this MH once it moved away from its Home Zone. However, this is a non-solution primarily because such an assignment would require this information to be specifically sent to a large number of Routers, Databases and of course the intending communicators / collaborators, every time such a transition takes place. Given the large amount of transactions and inconvenience involved in implementing this solution and increasing number of people using the MHs, this would translate into a huge network traffic / bandwidth requirement by itself.

Yet another possible solution could have been requiring the Routers to take routing decisions on the complete IPv4 address, instead of the customary Class-Id. plus Network Address. This, again, is a non-solution since the this requirement would translate into the requirement of huge Routing Table space, which in turn would mean the unacceptably high cost of transmission over the Internet.

Clearly, any acceptable solution had to avoid these traps and at the same time should have provided the required mobility, along with the continuity of communication at an acceptably low cost and without forcing the existing software to undergo any major change; and, thus the Mobile IP was born!

Now, just because the Mobile Hosts are to be accommodated, the Stationary Hosts

should not be required to make any change in their local software. Routers, all over the world, should not be required to alter their Routing Software as well as Routing Table structures or entries merely for this purpose. Databases and other collaborating entities should not need to be explicitly informed of the changed identity of the MH. No extra cost should be required to be added to the transactions while an MH was in its home zone. As a consequence, an important goal of not assigning a new IP address to the MH was added to the IETF-WG list.

One of the several proposed solutions was that each of the site supporting Mobile IP should create an Agent called 'Home Agent' / 'Home Address Agent'. This HA / HAA should be in charge of keeping track of which MHs of its home network are currently visiting a foreign network zone; and providing support services to these MHs as per need. Each site supporting the visiting MHs should create its own Agent called 'Foreign Agent' / 'Care-of Agent'. This FA / COA should be responsible for identifying the visiting MHs, keeping their track, authenticating their credentials by communicating with the corresponding HA / HAA and providing support services as per need. Whenever anyone sends packets for a MH that is currently visiting a foreign zone, the Router at the home zone attempts to resolve the address of the intended MH in the usual way of employing the ARP. The response to this ARP broadcast then comes from the HA / HAA, which supplies its address to the enquiring Router. A technique called Gratuitous ARP (G-ARP) is used to take care of invalid cache-entry in the Router. Once the packet is received by the HA / HAA, it encapsulates it and passes it to the IP address of the COA / FA, who on receipt, decapsulates and sends the packet to the visiting MH. This is immediately followed by sending the IP address of the current COA / FA to the original sender, so that any subsequent communication could use this new address thereafter.

5.3.9 The Internet Protocol Version 6 (IPv6)

IPv6 stands for the Internet Protocol Version 6. It is successor of the IPv4, which was designed in keeping with the technologies of the early Seventies. It does away with some features of the IPv4 while adding many new features. One basic advantage is the enlarged address space (adequate for a reasonably long time). To many scientists, this by itself is a good reason to consider the IPv6. There exist many practical problems before the world could finally switchover from IPv4 to the IPv6.

This specification is still evolving and an experimental Research and Development Tested called 6-Bone (IPv6 Backbone inspired by the success of the Multicast Backbone (M-Bone) experiment) exists. This is a collaborative test-bed involving universities, companies, research laboratories and individuals the world over. A wealth of information about this and related initiatives is available at the project site: <http://www.6bone.net/>.

5.3.9.1 Major Goals of IPv6 Design

Simplification of the basic protocol was the first goal of the design team. This was primarily achieved by three measures: providing a common format for all the headers, eliminating the IPv4 procedure used for 'hop-by-hop' segmentation handling and doing away with the header checksum, padding, header length and options fields. Reduction in

the packet processing time at the routers was another primary design goal. Simplification, as attempted by the designers, helped in achieving this goal.

Providing support for a very large number of addresses was probably the most immediate goal. Providing 128-bit source / destination address took care of this objective. It remains; however, debatable whether 128-bit was the best choice possible.

Providing support for flow specification and priority for the time-sensitive applications was a capability that the designers wanted to provide to this new version. Introducing the Flow Label and Priority fields made it possible.

Improving security of the packets transmitted over the IP was an objective necessitated by the potential the Internet Commerce demonstrated. This was made possible by introducing Authentication Header and Encrypted Security Payload Header.

Other major goals of the IPv6 design included reduction in the size of Routing Tables, providing support for new as well as older versions of the IP, providing reasonable support for multicasting, allowing smooth extensibility and modifiability in the years ahead and providing for a single, unique address assignment to mobile hosts.

5.3.9.2 On the EUI-64 Addresses and the Link Local Addresses:

As most parts of the Internetwork comprised of IEEE 802 LANs, the IEEE later evolved a common 64-bit Address Format for all the IEEE 802 LANs. Irrespective of whether a LAN interface is Ethernet, Token Ring, FDDI, CDDI or Radio based, this common format called IEEE EUI-64 Address Format would ensure that worldwide uniqueness could be guaranteed. In fact, this is a superset of the traditional 48-bit Ethernet Address, which was thought to be inadequate for providing unique identification numbers in projected quantum.

Significance of these unique addresses is that they are used / to be used as tokens for assignment of DLL level addresses called Link Local Addresses. Link Local Addresses can be used only locally. This is, however, important to note here that for being part of a network or an internetwork having a specific address format is not a prerequisite as long as some mapping mechanism exists!

5.3.9.3 How to convert a 48-bit Ethernet Address into the IEEE EUI-64 Address?

- Take a 48-bit Ethernet / IEEE 802.3 Address.
- Divide it into two parts, each of 24-bit length.
- Express it in Hexadecimal Format.
- Insert a 16-bit hexadecimal number FFFE in between the two parts obtained in an earlier step. The resultant address shall be the EUI-64 equivalent of the original 4b-bit address.

5.3.9.4 What about the networks for which no IEEE 802 address is available?

Interestingly, some networks exist (or might exist) for which no manufacturer-provided unique address is available. Still, a Link Local Address can be generated in a roundabout manner. Random number pick-up scheme is one such scheme that may be

used here. There may exist several ways to take care of this situation and decide upon a unique address on the basis of different associated parameters or factors.

In one simple yet workable scheme, if there be 'N' hosts in such a network, then each of these 'N' stations may randomly pick up a 64-bit number for itself. The scheme works because of the simple reason that number of likely address-collisions is quite low for a large number of connected hosts.

5.3.9.5 The IPv6 Base Header Design

The IPv6 main header is often called its Base Header. Structure of this header has been depicted below. As shown here, this header is simpler than that of the IPv4 and at the same time has capability to support, on 'as required' basis, several types of optional headers called Extension Headers.

The 4-bit Version field is used for indication of the protocol version. The next two fields Traffic Class and Flow Label collectively use 28 bits (8+20) and are primarily provided for time-sensitive traffic support. Different flow labels can now be assigned as a result of this enhancement. In a way, these fields collectively help in some form of QoS specification. The payload length uses 16 bits to represent the length of the IPv6 payload. Header length is not included in this computation.

The Next Header field is an 8-bit field that contains a standard code representing the type of the optional extension header or transport protocol header following the Base Header. This code might represent any one of the Extension headers as shown in subsequent figures or could simply indicate that the next header is a TCP or UDP header for instance.

The Hop Limit is an 8-bit unsigned integer that is decremented by 1 by each node that forwards the packet. The packet is dropped when this value becomes zero.

Source Address is the 128-bit address of the node originating the packet. Destination Address is another 128-bit address of the intended final or intermediate recipient of the packet.

Destination Field represents the final destination if the extension header called the Routing Header is not present; otherwise, it contains the address of the first intermediate node through which the packet is necessarily expected to pass on its way to the final destination. In such a case, the Routing Header contains the other chosen intermediate node addresses (essentially Router Addresses) and the final destination address.

As for the choice of the 128 bits long addressing scheme is concerned it was the result of consensus building efforts between two extreme camps supporting 64-bit addresses and 256-bit addresses respectively. The basic objective here was to avoid the address-space depletion problem in near future while ensuring the lower wastage of routing table space.

4-bit Version Field	8-bit Traffic Class Field	20-bit Flow Label Field		
16-bit Payload Length Field		8-bit Next Header Field	8-bit Hop Limit Field	
128-bit Source Address				
128-bit Destination Address				

Fig. 5.8: The IPv6 Base Header Structure

The IPv6 Addresses can be of three basic types:

- **Unicast:** (One station sends a packet to another single station / interface.)
- **Multicast:** (One station sends a packet to every member station / interface belonging to a designated group.)
- **Anycast:** (A form of packet transfer in which the packet is delivered to the nearest member of a designated group instead of sending to each group member individually.)

Clearly, the header accommodates each of these addressing provisions as per situation-specific requirements.

5.3.9.6 The IPv6 Extension Header Structure

There may exist six different types of Extension Headers, as per current provision in the IPv6 specification RFC. These have been shown below. These headers are appended to the Main Header as per need. There may be instances where several such headers may be required to be the part of a single IPv6 packet.

These headers (containing Internet Layer information) may be placed between the IPv6 header and the Transport Layer header in an IPv6 packet. An IPv6 packet may carry zero or more extension headers, each of them identified by the Next Header field of the preceding header:

Hop-by-Hop Extension Header
Routing Header
Destination Options Header
Fragment Header
Authentication Header
Encrypted Security Payload Header

Fig. 5.9 The IPv6 Extension Headers

Normally, any node along the route of a packet does not examine extension headers; until the packet reaches the node (or each of the set of nodes, in the case of multicast) identified the IPv6 header. At this stage, demultiplexing is performed on the Next Header field, which triggers the processing of the first extension header, or the transport-layer header in case no extension header is present. The extension headers must be processed strictly in the order of their appearance in the packet.

Certain situations like those requiring selective *debugging, management and network monitoring* etc. the Destination Option Header option may not prove adequate. The Hop-by-Hop Option Header is used in such cases (like multicast routing management, RSVP etc.) so that the necessary information could be communicated to all the intermediate routers, who would not bother to process the Destination Option Header by default. It is identified by the extension header code '0' (ze ro). Its format is as shown below:

Next Header	Header Extn. Length (8-bit)	Option Type	Option Data Length
Option Data (and Optional Padding, If needed)			

Fig.5.10: The Destination Options Header: Basic structure

There exist a Jumbo Payload option, as shown below. The Option Type Header in this case is set to the code '194'. This option is used in the cases wherein the length of a packet of larger than 64 Kbytes size is to be used. It may be interesting to note that for using the Jumbo Payload option, the Length Field of the IPv6 Base Header is set to zero. Naturally, there has to be an alternative mechanism for determination of the actual length of such packets! And, this alternative mechanism involves decoding of the 'Jumbo Payload Length' field for computing this size.

Alignment of Jumbo Payload Length (32-bit) field is another point to notice! Length field is necessarily required to start on a 32-bit boundary; and to make it possible, the Option Type Field (194) is set to $4n+2$ boundary.

A Router Alert Option, as proposed by D. Katz and R. Atkinson, has been suggested as well. The primary purpose of such an option is to alert / notify the intermediate routers that the packet-in-question does have some substantial information that demands a careful examination.

Just like the Destination Option Header case, in case of Hop-by-Hop Option Header also, exactly the same padding scheme (of one or more pads) is employable.

Next Header	Header Extn. Length	Option Type	Option Data Length
Option Data (and Optional Padding, If needed)			

Fig.5.11: The Hop-by-Hop Options Header: The basic structure

Next Header	Header Extn. Length	Option Type = 194	Option Data Length
Jumbo Payload Length			

Fig. 5.12: The Hop-by-Hop Options Header: As used for Jumbograms

The IPv6 Destination Options Header is identified by the Header Type code '60'. It is used as a general purpose Destination Option-based Header that may specify one or more options in its Option Type field (uniquely identified by an appropriate code) to be processed by the designated destination node. The Header Extension Length field carries an 8-bit number that represents exactly how many 64-bit words, excluding the first 64-bit word, do exist in the Destination Option Header. Option Type field is an 8-bit field that species the type of designated option; the first two higher-order bits of which specify an explicit desired action to be taken in the event of misinterpretation / ignorance of the options code by the destination node, a single bit 'C' flag specifies whether this specified option may be modified en route the destination and the remaining five bits specify a number such that the LSB encodes this option code itself. Not all options have an associated action! For instance, non-critical / additional information-based options do not warrant an action in the event of failure of the destination node in recognizing them. If the option containing critical information is not recognized or is ignored, the corresponding packet has to be discarded immediately and normally, an appropriate designated ICMP error message should be generated and sent back to the Source Node.

As for padding, if there are two options to be specified in the Option Data field, they are to be separated by null bytes (Pad1s) such that the options are at the two farthest ends of the Option Data field. Any number of null bytes may be used if needed. In case the number of bytes / octets to be skipped exceeds one, preferably the other padding option should be used (Pad2).

Next Header	Header Extn.	Option Type (2+1+5) bits	Option Data Length
Option Data (and Optional Padding, If needed)			

Fig. 5.13: Inside the Option Type field of the Destination Options Header

The IPv6 Routing Header plays the same role as the Source Routing Option of the IPv4; i.e. it contains the list of designated intermediate Router Addresses, which should be traversed by the packet-in-question (depending upon the loose / strict source routing option).

Fig. 5.14 shows the structure of the IPv6 Routing Extension Header that, as explained in the IPv4 section, can be used to specify loose or strict Source Routing behaviour.

Next Header	Header Extn. Length	Routing Type = 0	Segments Left
Reserved			
Router Address-1			
Router Address-N			

Fig. 5.14: The IPv6 Routing Extension Header

5.3.10 IPv6 Versus IPv4: A Brief Comparison

Let us briefly look at the list comparing the IPv6 and IPv4 header structures and indicating the implications of differences.

As vouched by the respective header structures, the major observations would lead to the following:

- In the IPv6, the IPv4 Options were replaced by Extension Headers.
- IPv6 has a Flow Label field in its header primarily meant for supporting the real-time applications.
- Traffic Class field was introduced in the IPv6 header that supports priority (mainly for real-time applications). The IPv4 had a field called Service Type in its header that has been replaced in the IPv6 header by the TC field.
- The IPv6 header has Payload Length field in place of the Total Length field of IPv4.
- The IPv6 has a Next Header Type field in place of the Protocol Type field of the IPv4.
- The IPv6 has Hop Limit field instead of the Time-To-Live field of the IPv4.
- The IPv6 provides Stateless Address Autoconfiguration capability which was not possible with IPv4.
- In contrast to the IPv4, which does not have any explicit provision for aiding privacy and security, the IPv6 does have built-in provisions for these requirements.
- Unlike IPv4, the IPv6 provides support for large Datagrams called Jumbograms.
- Both permit Fragmentation, but the IPv6 format keeps it in an extension header specifically meant for the job unlike the IPv4 format in which this information was to be maintained in a fixed field within the IP header.
- In IPv6, the multi-purpose Next Header field is used to indicate the type of protocol whereas IPv4 had a Protocol Type field for this purpose.
- The IPv6 header does away with the Header Checksum field of the IPv4.
- In IPv6, all addresses starting with eighty (80) 'zeros' followed by sixteen (16) bits of all 'ones' or all 'zeros' are considered as IPv4 addresses.
- In IPv4, there were five address classes (A to C of Network / Host combination types, D for Multicasting and E reserved for future use). In IPv6, the IPv4 Classes have been replaced with Types. Unlike the IPv4, that permits a two-level hierarchy of network and host prefixes, the IPv6 proposes to offer multi-level

hierarchy or even multiple hierarchies of prefixes. In IPv6, the first byte of the address refers to the type of address.

5.3.11 The IPv6 Address Notations:

Unlike IPv4 address notation, in which a 4-part IP address was expressed in Decimal Number System with a '.' used as a separator between every two parts; an IPv6 address is expressed as an 8-part IP address expressed in Hexadecimal Number System with a ':' used as a separator.

Example: ABCD:CA74:120A:4567:BDEA:FA3B:BB4C:1963

IPv6 also permits Address Abbreviation / Shorthand Notation.

Example: The IPv6 address ABCD:0000:120A:0000:0000:0000:BB4C:1963 can be denoted as: ABCD:0:120A:0:0:0:BB4C:1963 -- a case of replacing leading zeros by a single zero. Similarly, this address can be further abbreviated as: ABCD:0:120A::BB4C:1963 -- a case of eliminating an all-zero part of the address.

In the second example above, there are two consecutive colons. This notation is called the Double Colon notation and has the restriction that it can be used only once within a single IPv6 address. The primary reason behind this restriction is the Alignment Problem.

An IPv4 address, by prepending 96 zeros may form a valid IPv6 address. Such addresses are often written using a hybrid notation with the last 32-bits expressed in the Dot Decimal notation.

Example: ::0A00:0003 may be written as ::10.0.0.3

The IPv4 had a Prefix Notation that has been retained by the IPv6 as well. This involves using a normal IP address followed by a slash (/) followed by a number that represents Length of the Prefix. This Prefix Notation is useful to indicate that in any given IP address, how many bits (starting with the leftmost bit) belong to the Network-in-question.

Example: The notation A127:0:8:a123::/64 refers to a 64-bit Network Prefix in an IPv6 environment.

5.3.12 Address Issues in IPv6

The assigned IPv6 addresses do have a limited lifetime. However, it is possible to set this lifetime to infinity, as of now. There may exist two cases corresponding to the IPv6 Modes of operation:

- Stateless case: No Address Servers are required in stateless mode.
- Stateful case: This mode requires use of Address Servers.

5.3.12.1 Valid Address-Lifetime

This is often defined as the lifetime as assigned by the Address Server in the Stateful case; and lifetime as computed on the basis of Address-Prefix Lifetime (contained in the Router Advertisement Message) in the stateless case. An IPv6 address, whose valid lifetime has expired, must not be used.

5.3.12.2 Preferred Address-Lifetime

An IPv6 address whose preferred lifetime has expired is called an Invalid Address. Such addresses can be used for the current transaction; however, these cannot be used for initiating a new connection by the TCP.

5.3.13 Address Autoconfiguration / Plug-and-Play Support in IPv6

Ideally, Autoconfiguration refers to the capability of the system-in-question to be ready for being used as soon as it is connected to its compatible neighboring components without any manual or pre-programmed configuration. Theoretically, such systems (hardware or software) are expected to deliver performance that is comparable to those requiring manual configuration. In practice, however, this is seldom the case.

A major feature of such Plug-and-Play systems, as they are often called, is that they are capable of discovering required details / parameters pertaining to their operating environment. It is primarily this ability that helps them to configure themselves. One of the initial goals of the IPv6 design was to make it autoconfigurable.

5.3.13.1 Associated Factors of Autoconfiguration

There are several issues and factors affecting the autoconfiguration and manual configuration choices. These are:

- To which extent should this facility be available?
- Should the autoconfigurability be the only available choice or should it be one of the choices?
- How to use this capability and when?
- What should be the degree of control and security that would normally suffice?
- One of the reasons that the IPv6 supports automatic as well as manual configuration was the concern of the network Managers / Administrators with respect to the possibilities like reduced throughput, potential security problem associated with any Plug-and-Play support at the Network Layer level, Lower overall control, which might be detrimental at times.

The Road to Stateless Autoconfiguration involves adoption of the Stateless Model, discovery of the lower level (I.e. at the DLL level) addresses, Address Resolution in case of non-availability of IEEE 802 Addresses in any given environment, Address Configuration, periodically flushing and updating configuration table, Neighbour Discovery and Dynamic Address Allocation.

It is important to note here that the IPv6 is autoconfigurable in the Stateless Mode as well as in the Stateful Mode. In case of the Stateful Mode, it is possible for the network administrators to switch between autoconfiguration and manual modes.

5.3.13.2 Stateless Autoconfiguration

Initialization of IPv6 nodes is the primary concern here. All the nodes using the IPv6 in the Stateless Mode initialize themselves by doing the following:

- They use the pre-programmed capability of their interfaces for receiving every such packet that be transmitted by anyone using the Address: FF02::1 which is an 'All nodes address' for multicast.
- Next, they send a Solicitation message to routers (ICMP 133) with Hop Limit of 255 [Repetitions <=3]. This message may include options for type, length etc. Typically, a Source Address Option might include Type, Length and DLL Address. This message is sent at the address: FF02::2 All routers address for multicast.
- In the process, a Router Advertisement message (ICMP 134) is sent periodically / in response to the Solicitation message-- whichever takes place earlier.

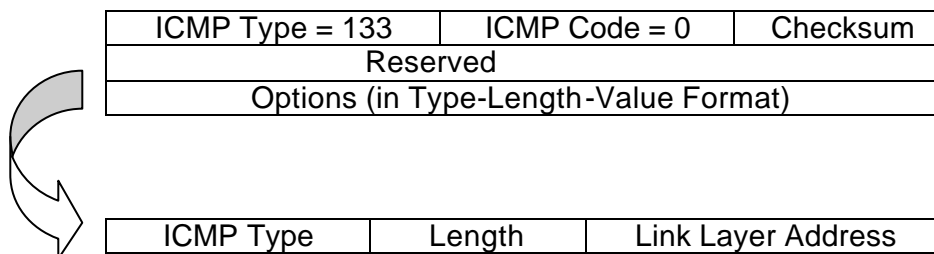


Fig. 5.15: The ICMP Neighbour Solicitation Message (ICMP 133)

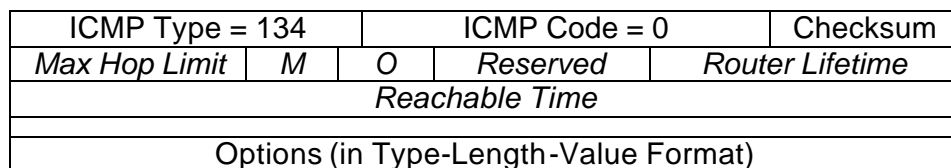


Fig. 5.16: The ICMP Router Advertisement Message (ICMP 134)

5. 3.13.3 The Stateful Autoconfiguration

Stateful Configuration can be of automatic or manual type. It takes care of inefficient utilization of address space, permits more than merely the address management, requires Servers and offers greater security (particularly in the manual mode). Both types of configurations could exist simultaneously in a network / internetwork. This

requires support from the DHCPv6. The following section focuses on such autoconfiguration in detail.

The IPv6 version of the DHCP (DHCPv6) is a Stateful configuration protocol. It is based on the traditional Client / Server model. In this scheme, clients query all the DHCP Servers and Relay Agents about their (i.e. clients') configuration parameters by sending the DHCP Solicitation Messages (Address: FF02:1:2). Solicitation messages are sent by making use of the UDP at the UDP Port 547 of the Servers. In return, Clients expect a response from Servers at their (clients') UDP Port 546. Servers send their response in the form of an Advertise Messages either to Clients or their Relay Address depending upon the information content of the Solicitation Message.

The IPv6 version of the DHCP makes use of the IPv6 capabilities of letting the Hosts to generate their Link Local Addresses and Multicasting. A Host may receive several Advertisement Messages, in which case, it chooses a Relay Agent and a Server for further transactions. Once this choice is over, the Host sends the chosen DHCP Server a Request Message for obtaining details of its configuration parameters and the DHCP Server responds with a Reply Message. (All these message formats are shown below for greater clarity.)

Format of the Dynamic Host Configuration Protocol (DHCP) Solicitation Message is given below.

Message-Type	C-bit	A-bit	Reserved bits
Link Local Address of Client (128-bit)			
(Optional) Relav Agent's Address (128-bit)			

Fig. 5.17 DHCP Solicitation Message Format

Format of the DHCP Advertise Message is shown below.

Message-Type	S-bit	Reserved bits
(Server / Relay) Agent's Address (128-bit)		
Link Local Address of the Client (128-bit)		
[Optional] Server Address (128-bit)		
Configuration Parameters / Extensions		

Fig. 5.18 DHCP Advertise Message Format

Format of the DHCP Request Message is as shown below.

Message-Type	S-bit	C-bit	Reserved	Transaction ID
[Optional] Server Address (128-bit)				
Agent's Address (128-bit)				
Link Local Address (128-bit)				
Configuration Parameters / Extensions				

Fig. 5.19 DHCP Request Message Format

Format of the DHCP Reply Message shall be as provided below.

Message-Type	L-bit	Error Code	Transaction ID
[Optional] Link Local Address (128-bit)			
Configuration Parameters / Extensions			

Fig. 5.20 DHCP Reply Message Format

Format of the DHCP Release Message is given below.

Message-Type	D-bit	Reserved	Transaction ID
Agent's Address (128-bit)			
Client's Link Local Address (128-bit)			
(Optional) Client Address (128-bit)			
Configuration Parameters / Extensions			

Fig. 5.21 DHCP Release Message Format

Finally, format of the DHCP Reconfigure Message is as shown in the Fig. 5.22.

Message-Type	Reserved	Transaction ID
Server Address (128-bit)		
Configuration Parameters / Extensions		

Fig. 5.22 DHCP Reconfigure Message Format

5.3.14 Time-sensitive IPv6 MM Traffic Over the Ethernet

Multimedia traffic is time-sensitive and hence synchronization of one data type with the other(s) is often a requirement. One way to solve the problem of synchronization is by assigning a quality of service (QoS). The million Dollar question here is that where do we enforce this and exactly how --- so as to have a cost-effective solution!

Setups comprising of the Ethernet LANs, need an economical method of accommodating multimedia-networking applications. It is here that the IEEE 802.9a: Isochronous Ethernet (also known as ISLAN-16T, or isoEthernet) comes into picture. A LAN is called Isochronous if it operates in real time. (The standard 8-kHz clock defines this time.) The IEEE 802.3 / Ethernet networks are asynchronous and have no clocking signal that may be needed by the multimedia components (e.g. voice). Clearly, the isochronous scheme symbolizes a departure from the original Ethernet scheme. This scheme (IEEE 802.9a) considers voice as the critical component. The Isochronous Ethernet symbolizes the marriage of multirate ISDN to the IEEE 802.3 / Ethernet. This combination of the two technologies makes it possible to offer guaranteed QoS for voice in MMI applications.

The Isochronous Ethernet is compatible with the common videoconferencing, and video-distribution standards.

Isochronous Ethernet is a hybrid network that integrates the standard 10 Mbps Ethernet technology with the 6.144 Mbps of Isochronous (ISDN) technology. A total of 16 Mbps is thus made available to any application. This is why it is sometimes called as Integrated Services Local Area Network, or ISLAN 16-T. The well-known 4B:5B-encoding scheme allows the total bandwidth to be 16 Mbps while still using the same good old 20 MHz clock that provides only 10 Mbps with the Manchester encoding. Isochronous Ethernet provides ninety six 64-Kbps ISDN B channels over the CAT-3 / CAT-5 cable. Such a good bandwidth may be adequate for a multi-point videoconference with over five participants (operating at the 384-Kbps). Yet adequate bandwidth remains available for the Whiteboard-based shared fora, E- FAX (Group-4), E-mail and Voice-mail etc. over independent frequency-bands / channels.

The Isochronous Ethernet solution is one of the most cost-effective multimedia networking solutions currently available. For integrating this solution to an existing IEEE 802.3 environment, only the following steps are required:

1. Replace the existing 10BaseT hub with an Isochronous Ethernet hub.
2. Replace the existing Ethernet Adapter cards of the identified Multimedia systems with Isochronous Ethernet Adapters so as to connect these systems to the above-referred hub.
3. Wherever required, older hubs can still be used for non-MM traffic handling. For this purpose, an Attachment Unit Interface (AUI) cable is used to connect the Isochronous Ethernet hub and existing / older Ethernet hubs.

National Semiconductor, the original developer of Isochronous Ethernet, provides hardware level solutions for the OEMs like Ericsson, Business Networks, Ascom Nexion etc.

An alternative and more cost-effective but propriety solution is offered by the 3Com. Termed as the Priority Access Control Enabled (PACE), this technology involves simply adding a specially designed Workgroup Switch for connecting the group of identified MM systems / workstations. In the 3Com PACE solution, neither existing Ethernet Adapters nor the cables need to be replaced. This feature has proven a boon for the PACE solution, since big brands like Silicon Graphics, Apple, Dell, Novell and Sun Microsystems have decided to provide support for it, sensing the market potential of the solution. The PACE solution concentrates upon the frame delivery timing and priority aspects. The basic problems in an Ethernet environment are the unbounded worst-case delivery time and lack of a priority scheme. Since multimedia applications need consistent, jitter-free and predictable data delivery, this problem required fixing right in the beginning. This solution overcomes reduces the frame delay by using star-wired switching and select propriety enhancements -- while maintaining the backward compatibility with existing Ethernet adapters. PACE employs specifically designed traffic-control algorithms for providing predictable delivery delay by flow regulation (and thereby minimization of jitter) and thereby forcing each Ethernet LAN segment to operate at a very high efficiency (nearly 98%, as per claims).

A method of prioritizing traffic over Ethernet to deliver QoS has been employed by the 3Com in the PACE solution. This propriety prioritization scheme offers two levels of service: high and low. One of the 3Com Workgroup Switches -- the LinkSwitch 1000 is available in the market that supports this technology, more or less on an experimental basis. As of now, however, the PACE technology has not been able to actually provide the QoS guarantee often required for high-quality MMI interactions.

5.3.15 A Quick Note on Mobile IPv6

In contrast to the Mobile IP (i.e. Mobile IPv4) discussed earlier, the Mobile IPv6 represents a complete design that separates identity from location. It can also be used to solve part of the host multi-homing problem. A multi-homed host, in principle, can have as many addresses as interfaces. At the start of any TCP connection or a UDP association, one such address can be selected as the home address for the connection or association. Later, as per the need, binding updates can be used to move the connection or the association to a different interface. It has been suggested that site multi-homing may be seen as a variation of host multi-homing, assigning as many addresses to each host as many providers to the site. As one of the discussion threads at the IETF suggests: "Use a variation of MobileIPv6 to improve host multi-homing; solve site-multi-homing by treating it as a variation of host-multi-homing". (*Christian Huitema quoted from the IETF mailing archives.*)

5.3.16 On the Current State of IPv6 Research, Development and Deployment Around the World

In Asia, as of this writing, Japan has done the most visible test-deployments of the IPv6 technology. There exists a Japanese national programme supporting IPv6 as the country has pledged itself to achieve the national-wide IPv6-capability by the year 2005. Project WIDE and numerous other projects have already made sizeable contribution towards this and other related goals. In India, Centre for Advanced Software Technologies (CASTLE): a research wing of the Centre for Software Development of the Birla Institute of Technology & Science at Pilani (BITS-Pilani), through its early initiative in form of the "Project IPv6@BITS" had begun an ambitious IPv6 research and development project in the year 1998. BITS was the first Indian entity to be connected to the 6-Bone as *IPv6-BITS-IN* and was subsequently granted a pTLA (pseudo Top Level Aggregator) status. BITS operates several international IPv6 tunnels including those connecting Canada, The Netherlands, Singapore, Korea, China and USA as well as some inland tunnels to Indian Research & Development organizations. Most of the groundwork in this direction is being done in form of structured term projects, undergraduate thesis work, graduate dissertation work and international collaborative research projects. As of this writing, this project has already brought numerous firsts to India in the area of next generation network research and development. The project website (<http://ipv6.bits-pilani.ac.in>) may be consulted for more details and technology transfer requests. Incidentally, this site was the first Indian website to become fully IPv6-compliant and accessible through native IPv6-capable web-browsers. The list of technologies, products, IETF documents, projects and papers produced by this research initiative is available at this site. Later, in the early 2002, many more organizations including Samsung India Research Centre at Bangalore, FutureSoft at Chennai, Wipro

Technologies at Bangalore and the Indian Institute of Technology at Kanpur (IIT -Kanpur) were connected to the 6 -Bone. As of now, several IPv6 interest groups exist and most of them not only have regular meetings but also have active discussion lists. Apart from Japan and India, among others in Asia, South Korea too has an ambitious national programme and so does China (through CERNET). Similarly, in Singapore, National University of Singapore and the SingAREN have joined the international efforts for testing and deployment of IPv6. Taiwan, Thailand and Malaysia have also begun select efforts recently.

In Europe, IPv6 has already received a big boost from practically all quarters including the European Commission. Italy (University of Sofia, Telecom Lab Italia, University of Rome etc.), France (INRIA, IRISA, University of Haute Alsace etc.), Switzerland (University of Berne, Telscom etc.), Denmark (Ericsson), Spain (University Polytechnic of Madrid, University of Vigo, ConsullIntel, Versaware etc.), Germany (Eurescom), Ireland (WIT) and many others – all have already made significant investment in future by contributing to the research, development, early testing and deployment. In UK, several universities, Laboratories and other organizations are contributing to the project. The list includes Lancaster University (that also maintains a vast and well-managed site on IPv6), British Telecom, University College of London and many others. Several applications have been ported to the IPv6 and a few applications have been built with dual-stack (IPv4 as well as IPv6) support. Early this year, Europe has launched the Euro6IX collaborative project, which, within a short period, has drawn considerable attention across Europe. Recently, during the Global IPv6 Summit at Madrid, an interesting demonstration involving use of IPv6 over IEEE 802.11x wireless LANs and a large number of portable computers had attempted to drive home the point the readiness of the IPv6-aware systems for real-life use.

In USA, the list of contributing organizations including the best-known companies, Laboratories and a few universities is quite big and seems to grow forever. Almost all hardware and software majors including IBM, DEC, HP, Compaq, Cisco, AT & T, Bell Laboratories, Microsoft etc. and almost all mobile communication giants including Nokia and Ericsson have their research groups on IPv6 and are participating in the 6-Bone initiative. Similarly, in Canada, a number of organizations have already begun their work in this area. The NSF-supported Internet2 initiative (see <http://www.internet2.edu/>) has some of its members contributing to the IPv6-based research and testing.

Worldwide numerous research, development, deployment and testing initiatives have been supported by an increasingly growing number of ISPs and international experimental network initiatives (like STARTAP, 6TAP, Euro6IX etc.) The IPv6 Forum (<http://www.ipv6forum.org/>) has its presence worldwide and by the way of generating awareness of the IPv6 and presenting business case for IPv6 to the decision makers. Currently headed by Latif Ladid of Ericsson Telebit, this forum has been able to generate general interest in the technical as well as business circles, particularly in Asia, Europe and North America. Adoption of the IPv6 as the base technology by the 3GPP has been a major boost for the IPv6 in the mobile world and due to the traditional lead of Europe in the area of mobile telecommunication / networking, IPv6 has been able to make more headway in Europe than in the North America which has been a leader in fixed telecommunication / networking. The NGN project in the USA (<http://www.ngn.org/>) and the NGNI project of the Europe (<http://www.ngni.org/>) are other projects working towards

exploring the best combination of technologies that could prove a winning combination in the near future.

The major hub of IPv6 activities in terms of proposals and drafts for various improvements, modifications and standards remains the corresponding working groups at the IETF. Steve Deering of Cisco and Bob Hinden of Nokia, currently co-chair the IPv6 Working Group at the IETF whereas Brian Carpenter of IBM moderates the DiffServ-specific discussions. Of late, a host of potential solutions have been proposed by individuals as well as working groups in the areas like Quality-of-Service (general mood at the IETF and elsewhere continues to be skeptical in this matter with several issues left open by most of the proposed modifications to the RFC 2460 in terms of the QoS), support for multi-homing for mobility etc. Similarly, a certain degree of disagreement (leading to a relatively slow pace of acceptance of the IPv6) within the IETF itself and initial reluctance on part of commercial ISPs towards large-scale deployment of IPv6 is likely to diminish in near future. In view of this author, the IPv6 is no longer the Next Generation Internet Protocol; it has quietly arrived and even though it may be some time before the said outstanding issues may be sorted out, it is clear that after passing through the Dual-Stack (and other) Migration stages, it is destined to play a major role in the world of Internetworking: both fixed and mobile.

5.4 On the Congestion Control in Internetworks

Congestion Control can be of two types; open loop and closed loop. Open Loop Congestion Control Schemes include the Traffic Filtering Schemes (use accept / reject rules) and the Traffic Scheduling Schemes; whereas the Closed Loop Congestion Control Schemes include the Uni-Variable Feedback based schemes and the Multi-Variable Feedback based schemes.

Congestion Metrics may include Average / Mean Queue Length, Average number or percentage of lost / discarded packets, Number of retransmitted packets those had to be sent again because of Transmitter's Time-out and Average / Mean Delay in Packet Delivery

5.4.1 Congestion Control Strategies:

- Congestion control by regulating admission of Packets / Cells
- Congestion control by regulating traffic based on traffic-type / traffic-rate (packet rate / cell rate / bit rate etc.) analysis
- Congestion control by admission-time resource reservation
- Congestion control by threshold monitoring and message passing
- Congestion control by preferential restraint (in research stage)
- Congestion control by Ostrich algorithm (debatable)
- Congestion control by supervised blocking / rerouting (under investigation)

5.4.1.1 The Anticipatory Buffer Allocation Scheme:

In this scheme, which is particularly suitable for Virtual Circuit Subnets, congestion can be effectively controlled / avoided by estimating the optimal buffering needs of the

Switches and allocating this buffer capacity to Virtual Circuits on anticipatory / pro-active basis. It is a variation of pre-allocation scheme since it allocates estimated capacity in advance. This scheme differs from the standard VC establishment scheme in the way that in the latter no buffer-space allocation is reserved at the Switches by the call-request packet. Also, no permanent buffer allocation is done a-priori, in the latter scheme. This scheme may be implemented using many different protocols including the Sliding Window and Stop-and-Wait protocols. Choice of a protocol, in any case depends on the desired throughput, available buffer capacity and the associated price. However, for the VCs that may not, at an average, have adequate traffic so as to effectively use a sizeable chunk of such pre-allocated buffer-space, the economics may not be favourable. Moreover, this is, in effect, a Congestion Avoidance Scheme rather than an adaptable Congestion Control Scheme.

A possible variation of this scheme could be, as suggested in the beginning, a dynamic allocation scheme that is proactive by nature and that, by using some adaptive / statistical buffering need-determination algorithm, estimates / anticipates the required buffer size and if available, allocates the VC in question. The primary difference here is that the call request packet need not ask for any buffer reservation. Moreover, this allocation may be done after the establishment of the VC. This scheme duals as an Avoidance as well as a Control scheme since if invoked during VC establishment, it provides avoidance whereas if triggered by anticipation of congestion, could simply reduce the chances of its building up. However, this solution is relatively complex to implement and has a potential of occasional misfire.

Both of the discussed solutions, therefore, do not prove attractive.

5.4.1.2 'Arbitrary Packet Rejection-based' / 'Reject-on-Getting-Full' Congestion Control Scheme

This scheme is the simplest of all congestion control schemes. It controls further building up of congestion just by dropping any further packets reaching the node in question, entirely arbitrarily, without any learned analysis. As a result, even ACKs might get rejected and cause a series of unwarranted problems. This scheme requires absolutely no buffer reservation / advance allocation, in complete contrast to the earlier scheme. A variation of this scheme is called the Leaky Bucket Algorithm. This too is not an attractive solution because of obvious potential for creating deadlocks.

5.4.1.3 Selective Packet Rejection based Congestion Control Scheme

This scheme is the modified version of the previous congestion control schemes. It controls further building up of congestion by selective dropping of packets reaching the node in question. The choice of selective acceptance / rejection is governed by a set of rules. This scheme, like its predecessor, requires absolutely no buffer reservation / advance allocation.

5.4.1.4 Permit-based / Token-based / Isarithmic Congestion Control Scheme:

As the name itself suggests, this algorithm uses a Permit / Token based admission control with respect to entry to a node. Any sender node willing to transmit 'n' packets to

a receiving node is first required to capture 'n' Tokens / 'Permit for sending 'n' packets'. If only one Token is captured, only one packet can be transmitted. The number of total Tokens available is usually kept constant; and as result, this scheme ensures a predictable constant traffic, without any loss of packets. A variation of this algorithm is known as the Token Bucket Algorithm.

5.4.1.5 The Choke Packet Scheme of Congestion Control:

One of the possible ways to control congestion is to cut down the incoming traffic to a node by informing the originator of the traffic that a state of congestion has occurred and the originator should cut down its packet transmission rate intended to reach / pass through this receiver. This scheme uses just that! It makes use of what is termed as 'Choke Packet' for indicating to the originator about the congestion and expects it to cut down its transmission rate by a pre-defined percentage.

What exactly happens is this! The various Routing nodes periodically run a routine for estimating the state of utilization of their one or more output lines and compute an index that could, on crossing a certain threshold value, normally suggest that a state of congestion is about to arrive or has arrived. Whenever this threshold value is reached, the congestion control routine gets fired. Once this routine swings into action, any packet other than an ACK that arrives at this node intending to be forwarded on any one of the congested output lines is blocked and a special packet called the "Choke Packet" is constructed by extracting the originating node's address from the Sender's Address field of the packet that has been blocked. The original packet itself is tagged / included as payload (to the generated header with a bit set) so as to help the originator learn so that it does not generate traffic any further / more than the default cut-down rate thereafter for a stipulated period of time.

Many variations of the Choke Packet-based scheme exist. However, most of them have potential to generate further network congestion due to a lot of possible choke-packet traffic. One such possible solution is the Hop-by-Hop Choke Packet-based scheme of congestion control. This scheme has a special feature of helping in cutting down the incoming traffic systematically and gradually by informing every intermediate Router along the way of the Choke Packet. Thus, in effect, at every hop, the scheme succeeds in immediately initiating reduction in traffic towards the congested node; rather than allowing the flow to continue until the Choke Packet reaches its destination and an action is taken.

5.4.2 Deadlock due to congestion

There exists an extreme effect of failure to timely control of congestion! That's the Transmission Deadlock / Lock-up State. Such deadlocks can be of several types including Direct Store-and-Forward Deadlock / Lockup and Indirect Store-and-Forward Deadlock / Lockup.

A well-known solution to such deadlocks was suggested long back by Merlin and Schwietzer that involved use of a specially constructed directed graph showing Buffers as nodes and arcs connecting a pair of buffers in the same or adjacent router. Several other solutions have been proposed since then.

5.5 More on the Generic Transport Layer Concepts

Transport Layer, as seen earlier, is a layer of the Network Architecture that is primarily concerned with getting TPDU from the upper layer (usually Application Layer) and delivering it to the same layer at the intended destination node (through the underlying Network Layer). The reverse is also the responsibility of this layer.

5.5.1 Transport Layer Responsibilities

Transport Layer, in general, primarily deals with:

- Accepting APDU from the Application Layer through the SAP
- Processing these APDU
- Deciding transport connection requirements (for further transmitting this DU after encapsulating it within a TPDU)
- Passing this packet through the SAP to the lower layer (NL)
- Accepting TPDU from the lower layer through the SAP
- Processing the TPDU
- Removing the encapsulation and passing the APDU through the SAP to the upper layer (Application Layer)
- Providing support for connection-oriented / connectionless services as the case may be (depending upon the protocol stack and need)
- Provide diagnostic support for network monitoring, configuration, management and trouble-shooting at the Transport Layer or higher layer.

5.5.2 Generic Transport Service Primitives:

A possible set of generic Transport Service Primitives include the following primitives:

- Create / Identify and Assign / Bind
- Listen / Wait
- Accept
- Connect
- Send / Transmit
- Receive
- Disconnect / Close / Terminate

5.5.3 Generic Transport Service Primitives:

A less flexible set of the Transport Service Primitives may include the following primitives:

- Listen
- Connect
- Send
- Receive
- Disconnect

5.5.4 Transport Service Primitives: The Berkeley Sockets Set for the TCP

In the specific case of the TCP Transport Services, following primitives have been defined in the standard Berkeley Socket API:

- Socket
- Bind
- Listen
- Accept
- Connect
- Send
- Receive
- Close

5.5.5 The Transport Service Access Point (TSAP) and the Network Service Access Point (NSAP)

In an IP network / internetwork, the NSAP refers to the IP Address of the node. In a TCP / UDP over IP stack, the TSAP refers to the pair: {IP Address, Local Port Number} *It is at the TSAP, at which a peer process listens / contacts. Both TSAPs and NSAPs could be one or more per node / host.*

5.5.6 QoS Considerations in the TL As Used During the Option Negotiation Process

The Quality-of-Service considerations in the Transport Layer, may include the following factors:

- Required Priority of Service
- Ceiling for Residual Error Ratio
- Maximum Acceptable Connection Establishment & Transit Delays
- Minimum Acceptable Throughput
- Maximum Acceptable Probability of Connection-Establishment-Failure
- Maximum Acceptable TL-initiated Abnormal Termination of Connection
- Security and Protection Specifications

5.5.7 Inside the TCP

TCP stands for Transmission Control Protocol. An RFC (RFC 793) by John Postel was the first description of the TCP as seen today. It offers a Connection-oriented Transport Service. It assumes the underlying IP-subnet as unreliable and therefore takes care of reliability, flow control and reordering of data units as per requirement and sends data to

the IP Layer in MSS-sized blocks or smaller, after prefixing a TCP header to each such segment. Default value of the MSS is 536, in case the peer at the other end does not specify a smaller value to be used with it. MSS is normally of lesser than or equal to the size of the MTU (for IPv4: 40, for IPv6: 60). TCP requires that a TCP Client establishes a Full-Duplex connection with a TCP Server (before any real data could be exchanged between them). After the data exchange is over, this connection has to be explicitly Terminated.

As the TCP provides a reliable service, it expects an ACK to be received for the data transmitted by an TCP-entity (Client or Server). If the ACK does not arrive within a Time-out period, it retransmits the data and waits for a longer period of time to receive an ACK. Even if after a certain number of such attempts the data cannot be successfully transmitted, it gives up further attempts and informs the Application Layer. (Intermediate failures are not reported to the Application, however!) The maximum period for such retransmission-attempts and associated wait-periods for a single data unit, put together, may be anywhere between 4 Minutes to 10 Minutes, depending upon the TCP implementation and Stack Configuration.

Round-Trip Time (RTT) is automatically, dynamically, computed between a Client-Server pair by a routine internal to the TCP implementation. RTTs are always more for WANs than for LANs. TCP recognizes byte-boundaries and is thus a byte-stream-oriented protocol. As it provides each of its Segments a Serial Number, reordering, rejection of duplicate segments etc. becomes possible. It uses Sliding Window Protocol for the purpose of data transmission / reception / flow-control.

5.5.7.1 About the TCP Ports

TCP Ports are 16-bit numbers. They are of three types: Well-known Ports (0-1023: Controlled by the IANA), Registered Ports (1024-49159) and Ephemeral / Dynamic Ports (49152-65535). (RFC 1700 shows a list suggested initially. This applies to the UDP as well.) FTP over TCP uses 21 whereas TFTP over UDP uses 69 for instance. X-Windows Server uses 6000-6063 Registered Ports. For BSD, the Well-Known Ports are: 1-1023, Reserved Ports: 1024-5000 and the Unprivileged Server Ports are: 5001-65535.

5.5.7.2 The 3-Way Handshake in TCP

TCP requires a Three-Way Handshake for the Connection-Establishment. This is called so since a minimum of three data-units need to be exchanged between a TCP Client and a TCP Server for establishing a TCP connection. These packets may be SYN+Seq-No (C-to-S), SYN+Ini-Seq-No (S-to-C) on which ACK+I+1-Seq-No piggybacks (S-to-C) and lastly, ACK+Ini+1-Seq-No (C-to-S). Here, SYN stands for Synchronize segment. It takes just 1 -byte of Sequence Number Space. In a similar way, Connection-Termination takes four data-units. It takes place using FIN (Final Segment) and associated ACK. Both sides send one FIN and one ACK to each-other, in this case. SYN contains TCP options of MSS, Sliding Window Scaling (Left-Shifting by 0 to 14 bits allows window-sizes of 64K to 1 GB) [RFC 1323 by Jacobson et al], Timestamp (for High-speed connections) etc.

5.5.7.3 Of the Crashes and Crash Recovery Mechanisms and Strategies applicable to the TCP/IP Architecture

In any connectionless packet delivery system, including the IP based systems, there is always a possibility, however small, that a Transport Protocol Data Unit (TPDU) may be lost on its way to the destination Host's Transport Layer. This simply means that, a TL mechanism must exist in such situations that could continually monitor the transfer / receipt status, identify any missing TPDU and take corrective measures for getting it, if protocol so enforces / permits. These are the very mechanisms that may handle Subnet Crash Recovery, where so possible. However, the real challenge lies in Recovering from the Host Crashes. This is primarily because of the fact that certain amount of data loss (apart from the connection-loss, in case of Connection-oriented Transport Services) in the crashed host is bound to create problems in the recovery process. A little thought over the complexity forced by the commonly used Sliding Window Protocol in such a situation would make the magnitude of the problem clear.

Crash-Classification One:

- Host Crashes
 - Client Crashes
 - Server Crashes
- Subnet-Device Crashes
 - Router Crashes
 - Bridge Crashes
 - Repeater Crashes
- Link Crashes

Crash-Classification Two:

- Temporary Crashes / Non-Fatal Crashes
- Permanent / Fatal Crashes
- Intermittent / Unanticipated Repeated Crashes

5.5.7.4 Client Crash Recovery Strategies:

Consider a case in which a Client Host that is downloading a large file from a Server that is remotely located crashes temporarily and then comes back. Three things would happen in such a situation:

- Any User / Control Data currently in the local buffer which is yet to be written to the disk / storage medium shall be lost.
- Any Control Signal, say an ACK, generated but yet not transmitted shall be lost.
- The data under processing at the time of crash, at the local host shall be lost.

Depending upon the protocol in use these situations may lead to different inferences, and hence may require different recovery policies. Let us assume that our protocol requires that the downloading Client sends an ACK to the sender only after successfully writing to the disk (in case of a simple Stop-and-Wait protocol). In such a case, if the data is written to the disk but before the ACK could be sent, the host crashes, it would mean the loss of the ACK as well as the Connection. Let us see what could happen when this host comes back to operation! When, in this case, the Client Host comes back, it has already lost the transport connection and any data in its memory.

Clearly, unless there exists a semi-permanent status-of-progress record that is updated after every successful operation, the host may be unable to track exactly what it was doing and the associated details. The first question is that who should maintain such a record and where? One possibility is to query the Server about the status, but this is a non-starter given the sheer resource requirements servers shall be forced to have if they are to retain all such status data even if for a short duration of say five minutes. Moreover, how does the Client know which Server to contact, if there is no record at its disposal locally. These points suggest that it may be preferable to expect each Client Host to maintain these records and utilize them in the event of a crash. Assuming that our Client Host in question does have such a record, the next question is what steps should it follow for Recovery and Resumption of the file-download. One possibility is that soon after the Client Host comes back, it seeks to learn about its network status and thereafter, retrieves its stored records which have an 'Incomplete so far' type of tag / flag.

The operation, previously interrupted due to the crash, could be resumed only after a fresh connection is requested to the designated Server and is obtained. Once such a connection is setup, the Client may inform the Server that it needs the file-download to resume from a specific point onward (as per the local record) instead of a fresh full-retransmission.

5.5.7.5 Server Crash Recovery Strategies:

Just like the cases we considered for Client Host Crash and a possible Recovery, many possible situations may arise, though very rarely, in which a Server Host crashes during a operation, say involving download or upload of a large file. One possibility could be to require the Server (after it quickly reboots) sending a Broadcast Message to all other visible Hosts querying about the respective transfer-status (like if they were interacting with it when it crashed and if yes, what was the pre-crash status?).

Clearly, this scheme assumes that the Crash was of very short duration and that various Host Clients have yet not 'closed' their open connections.

Depending upon whether 'First Acknowledge then Write to the Output Stream' or 'First Write then Acknowledge' this scheme could invite Loss of a TPDU or generate a Duplicate TPDU respectively. This suggests that both the Client and the Server need to maintain their respective records for a possible recovery; but still there may be situations which may be difficult to be handled at the TL itself. This gives an important message loud and clear! The higher layer cannot be ignorant of a crash if an acceptable recovery is to be done.

5.6 About Application Client and Application Server Processes

A process that provides any set of predefined services to one or more requesting clients is called a Server Process. Types of Application Server Processes include:

Concurrent Server Process: A process that simultaneously provides any set of predefined services to one or more requesting clients is called a Concurrent Server Process.

Iterative Server Process: A process that provides any set of predefined services to only one requesting client at any point of time is called an Iterative Server Process.

A process that solicits any specific service from any designated server is called as a Client Process.

5.7 Summary

The TCP/IP is a Network Architecture comprising of four layers namely, Application Layer, TCP/UDP Layer, IP Layer and Host-to-Network Interface. In this architecture, Connectionless service is a Store-and-Forward scheme whereas the Connection-oriented service is a logical connection-based scheme. TCP offers connection-oriented services whereas UDP offers connectionless services. IP is a connectionless protocol. IP and ICMP protocols are inseparable in the practical world. A bridge is used to interconnect two devices at MAC-level. A Router is an IP-level device whose primary function is to decide the optimal routes between any two network nodes.

Internet Protocol has evolved with the evolution of the Internet. Although, IPv4 has served extremely well over the years, some of the recent developments, particularly in the world of multimedia internetworking, have found themselves in problems to address which this protocol was not really designed (for the simple reason that in those days such applications were relatively rare and nobody anticipated such rapid growth and acceptance of the internetworking technologies). Although workable solutions have been suggested and are already being used (like the RSVP over the IPv4), these often prove short-term measures because of the inherent deficiencies of the protocol itself. For instance, lack of support for flow-specification, smaller address-space, option-handling overheads, lack of adequate security features etc. call for an improved protocol design so as to take care of these issues as well as provide certain other desirable features like support for larger packet size where so required, better support for autoconfiguration, adequate degree of backward as well as forward compatibility etc. The IPv6 was designed to do just that!

As you have witnessed throughout the chapter, the new protocol has a very good potential for cost-effective multimedia internetworking over existing as well as upgraded infrastructures. In particular, the successful commercial solutions like Isochronous Ethernet based IPv6 show a definite direction the MMIs are likely to take leveraging on the capability of IPv6 and continued improvement in the low-cost networking technologies.

5.8 Recommended Readings

1. A. S. Tanenbaum: **Computer Networks**, Fourth Edition, Prentice-Hall, Upper Saddle River, 2002.
2. B. A. Forouzan & C. H. Fegan: **TCP/IP Protocol Suite**, Second Edition, Tata-McGraw-Hill Publishing Co. Ltd., New Delhi, 2002.
3. Cisco staff: **Internetwork Design Guide**, 1999 available at: <http://www.Cisco.com/univercd/cc/td/doc/cisintwk/idg4/nd2002.htm#xtocid29276>
4. Cisco staff: **Internetworking Case Studies**, Cisco Press, 1996.
5. Cormac Long: **IP Network Design**, Osborne-McGraw-Hill, Berkeley, 2001.
6. D. Comer & D. L. Stevens: **Internetworking with TCP /IP**, Vols. 2-3, PHI, 1994, 1993.
7. D. Comer: **Internetworking with TCP /IP**, Vol. -1, Fourth Edition, Pearson Education, New Delhi, 2001.
8. Dave Koiur: **IP Multicasting: The Complete Guide to Interactive Corporate Networks**, John Wiley & Sons, New York, 1998.
9. Garry R. McClain (Ed.): **Handbook of Networking and Connectivity**, AP Professional, 1994.
10. James Kurose & Keith W. Ross: **Computer Networking**, Second Edition, Pearson Education, New Delhi, 2002.
11. K. Downes, Marilee Ford, H. K. Liu, Steve Spanier & T. Stevenson : **Internetworking Technologies Handbook**, Second Edition, Cisco Press / Techmedia, 1999.
12. Paul T. Ammann: **Managing Dynamic IP Networks**, Tata-McGraw-Hill Publishing Co. Ltd., New Delhi, 2001.
13. Rahul Banerjee: **Lecture Notes on Computer Networks**, Nov. 2002, available on-line at: <http://www.bits-pilani.ac.in/~rahul/CN/index.html/>
14. **RFC 1009** (Requirements for Internet Gateways)
15. **RFC 1009** (Requirements for Internet Gateways)
16. **RFC 1011** (Official IP)
17. **RFC 1042** (IP over IEEE 802.3)
18. **RFC 1124** (Policy Issues in Interconnecting Networks)
19. **RFC 1124** (Policy Issues in Interconnecting Networks)
20. **RFC 1125** (Policy Requirements for Inter-Administrative Domain Routing)
21. **RFC 1147** (FYI: A list of Network Management Tools)
22. **RFC 1175** (FYI: A very useful reference-list on Internetworking related information)
23. **RFC 1208** (Glossary of Networking Terms)
24. **RFC 1209** (IP over SMDS)
25. **RFC 1254** (Gateway Congestion Control)
26. **RFC 1360** (Official Protocol Standards of the Internet Architecture Board)

27. **RFC 1360** (Official Protocol Standards of the Internet Architecture Board)
28. **RFC 1630** (Universal Resource Identifiers in the WWW)
29. **RFC 1738** (Uniform Resource Locators)
30. **RFC 1809** (IPv6 Flow Labels: An Informational RFC)
31. **RFC 1825** (IP Security Architecture)
32. **RFC 1826** (IP Authentication Header)
33. **RFC 1827** (IP Encapsulation Security Payload)
34. **RFC 1828** (IP Authentication using MD5)
35. **RFC 1883** (Older IPv6 Specification)
36. **RFC 1884** (IPv6 Addressing)
37. **RFC 1886** (IPv6 DNS Extensions)
38. **RFC 1887** (IPv6 Unicast Addressing)
39. **RFC 1971** (IPv6 Address Autoconfiguration)
40. **RFC 1972** (IPv6 over Ethernet)
41. **RFC 2019** (IPv6 over FDDI)
42. **RFC 2023** (IPv6 over PPP)
43. **RFC 781** (IP Timestamp)
44. **RFC 791** (IP version 4)
45. **RFC 815** (IP Datagram Reassembly)
46. S. Keshav: **An Engineering Approach to Computer Networking**, Addison-Wesley, Reading, 1997.
47. Smoot Carl-Mitchell & John S. Quarterman: **Practical Internetworking with TCP / IP and UNIX**, Addison-Wesley, Reading, 1993. (This book does not really discuss the IPv6. This however, helps the reader to take a look at the pre-IPv6 days and realize the wisdom of evolution of the IP.)
48. Uyles D. Black: **TCP / IP & Related Protocols**, Second Edition, McGraw-Hill, N. Y., 1995.
49. W. Buchanan: **Advanced Data Communication and Networks**, Chapman & Hall, London, 1997.
50. Y. Zheng & S. Akhtar: **Networks for Computer Scientists and Engineers**, Oxford University Press, New York, 2001.

5.9 Exercises

1. Consider an organization, which has multiple locations spread over length and breadth of a country. If this organization frequently requires arranging Quality Circle meetings involving key people at various levels working at different locations, travel expenses and associated support expenditure might prove prohibitively high. Multimedia Internetworking based video-conferencing might prove a far more cost-effective solution in such circumstances, provided that the designer takes wise design decisions with respect to choice of hardware, software, quality of multimedia content, subnet protocol stack and host protocol stack. Given such an assignment, what shall be your approach towards providing a viable technology solution to your client?
2. Study the LAN infrastructure of any medium or large sized commercial software development organization. If such LANs need to be interconnected into an intranet such that select managers and project leaders could participate in a desktop videoconference and if the organization is reluctant to invest huge sums in upgrading the

- infrastructure to support high quality videoconferencing, can you suggest a viable approach to selectively and incrementally solve this problem?
3. Consider the existing IPv4-friendly setup of the Internet. What problems *other than those of the Address Space* could you identify in the existing setup and exactly how do you suggest these to be fully or partially addressed by the IPv6 and associated technologies? (Hint: Identify any five major problems and discuss their solutions, if any.)
 4. A university currently uses a combination of Optical Fibre backbone, Gigabit Ethernet LAN, Fast Ethernet LANs having IP version-6 atop them. If it is required to offer Desktop Multiparty Video-conferencing at the intranet level, shall this set up require any modification / tuning? What solution shall you recommend under these circumstances and why? Explain in detail with the help of diagrams, if necessary.
 5. What are the weaknesses of the IPv6 specifications with respect to MMI traffic handling? *Could you suggest any innovative and workable solution to these problems that would not require any immediate change in the IPv6 specification itself?* This solution may use complementary capabilities and may work atop the IPv6 layer in the Routers or in any higher layer on the Hosts. Please justify your solution in brief
 6. DiffServ and IntServ refer to differentiated and integrated types of quality of service respectively. What are the possible ways of their deployment in the TCP/IPv4 and TCP/IPv6 WANs?
 7. As of now, the IPv6 Flow-Label specification is incomplete. However, the Traffic Class specification has been defined reasonably well. Can you enlist the QoS parameters that, in your opinion, should be defined by the Flow Label field? Please justify your response in each case.
 8. What are the shortcomings of the current IPv6 specification in terms of the support for IP-level Quality of Service and what is the best solution proposed so far at the IETF and Why? (Hint: Refer to the IETF workgroups on IPv6 and DiffServ. Also see the Appendices A-1 and A-2.)
 9. Refer to the Cisco IP/TV architecture described in Chapter-9. If this architecture is to be ported to the IPv6-only WANs, then shall it require any changes? If yes, exactly what changes would you suggest to be made and why? If no changes are to be made, then please explain exactly how the existing architecture would be able to work over such internetworks?
 10. What is an IPv6 ESP Header and what is its significance with respect to WANs?
 11. Enumerate any two major advantages of the DHCP (IPv6 version)? Enumerate any two major issues concerning the use of the DHCP (IPv6 version)?
 12. Suggest a suitable set of Flow Label parameters for an IPv6-based network configuration that could permit QoS parameters to be taken into account.
 13. Consider a large multi-campus (eight locations) university IPv6-capable intranet that comprises of over 6400 computers spread over numerous separate networks interconnected in a hierarchical manner. In this case, the university had chosen the RIPv6 as its routing protocol. As the network performance statistics showed a marked deterioration in terms of performance with the growth in network traffic and increased inter-campus collaboration trends. If, you were asked to suggest a detailed

solution to this problem, what design solution / solutions you would have taken and why? Please explicitly mention all your assumptions and give brief justification in favour of each of these. You are expected to present a detailed, step-by-step solution alongwith necessary logic / data to substantiate your choice(s). Diagrams, where required, are to be provided.

14. ISLAN based networking has its own quota of problems. What are these problems? Suggest an effective solution to these.
15. Compare QoS support features of IPv6 technology with those of IPv4 technology. Comment on the Flow Label usage of IPv6-based internetworks. Does it replace the RSVP? Explain in brief.
16. What is a Routing Header and what is its significance with respect to large internetworks? (Please answer with respect to IPv6 alone.)
17. Consider a large MNC organization having over 2048 computers spread over 12 countries (having one establishment per country) running a range of operating systems like MS Windows 2000 / XP, Linux 2.4.x, SCO Unix OpenServer, IBM's AIX, HP-UX and Sun Solaris etc. spread over numerous separate networks. If these networks are to be interconnected in a hierarchical manner and you were asked to suggest a cost-effective internetworking solution with native IPv6 support to this problem, what design choices you would make and why? Please explicitly mention all your assumptions and give brief justification in favour of each of these. You are expected to present a detailed, step-by-step solution alongwith necessary logic / data to substantiate your choice(s). Diagrams, where required, are to be provided.
18. Discuss the differences between Mobile IP and Mobile IPv6.

Chapter-6

The Internetwork Routing Architectures

Interaction Goals

Interaction Goals of this chapter include developing an understanding of the internals of the major Routing Architecture involving the Internet and large Intranets from a designer's perspective. We shall also take a quick look at the accepted industry practices and evolving trends.

At the end of this chapter, you should be able to:

- Understand the internals of the major Internet Routing Architectures,
- Identify functionalities, design goals and issues related to the Router Design,
- Choose one of the Routing strategies based on system requirements.
- Compare the routing algorithms.
- Differentiate between the characteristics and capabilities of Core and Edge Routers, and
- Use an appropriate Routing Architecture for use in soft-real-time internetwork designs.

The prerequisites are some exposure to networking basics, queuing theory and graph theory.

6.1 Introduction

Effective and timely decision about choice of an appropriate route is key to successful routing. Design and choice of appropriate Routing Architectures, therefore, are extremely important for attaining the desired performance from the internetworks.

In general, there exist two primary classes of routing strategies: Centralized Routing and Distributed Routing. Naturally, each category has a set of Routing Architectures belonging to it.

In the Centralized Routing, routes are computed centrally by a designated central router and thereafter periodically distributed to all routers in a given subnet. This class of routing is suitable for small networks, has concerns about control-traffic volume, single point of failure, and delayed adaptation to changes in topology. Thus, it may be preferable in small, centrally administered subnetworks. Its availability and performance are not impressive.

In Distributed Routing, routing choices are made locally, in a collaborative / pre-planned manner. Due to distributed nature, processing too is distributed and unlike the central model, it does not tax a single router's processor(s).

6.2 About Routing Terminology

It shall be in order to be introduced to select terms relevant to Routing, before moving ahead.

- *Traffic Multiplication Effect (TME)*: This refers to multiplication of the same copy of data by way of replication, redundancy and circulation. (All flooding schemes suffer from this effect, for instance.)
- *Node Bypass / Route-around*: This refers to the strategy of bypassing a failed node if an alternate route exists.
- *Network Layer Data Unit / Packet Die-out*: This refers to the state of termination / death / lifetime expiry of a packet.
- *Principle of Optimality*: If a node B falls along the optimal route between node A and node E, then the optimal route from node B to node E also falls along the same path.
- *Sink Tree*: The optimal routes between any number of given Source nodes and a certain Destination node, as discussed above, invariably form a tree that might be seen as starting at the Destination node and ending at the Source node. This tree is called a Sink Tree.
- *Routing Directory / Table*: This is a table containing computed routes between various nodes of a subnet. Types of Routing Directory / Routing Table include:
 - Fixed / Static Directory / Table
 - Session-specific Directory / Table
 - Adaptive Dynamic Directory / TableAll of these can further classified based on support for Full / Partial Paths.
- *Packet Routing Problems*: Problems in packet routing arise from one or more of the following areas concerning routing, management and maintenance:
 - Loss of packets
 - Receipt and circulation of duplicate packets
 - Packet Choking / Network Congestion
 - Network Cleansing
 - Worst-case upper bound problem
 - QoS negotiation
 - Failure Handling
 - Quick Recovery Requirement
 - Route Tracing
 - Mobility Support
 - Tunnelling Support
 - Network Management Support

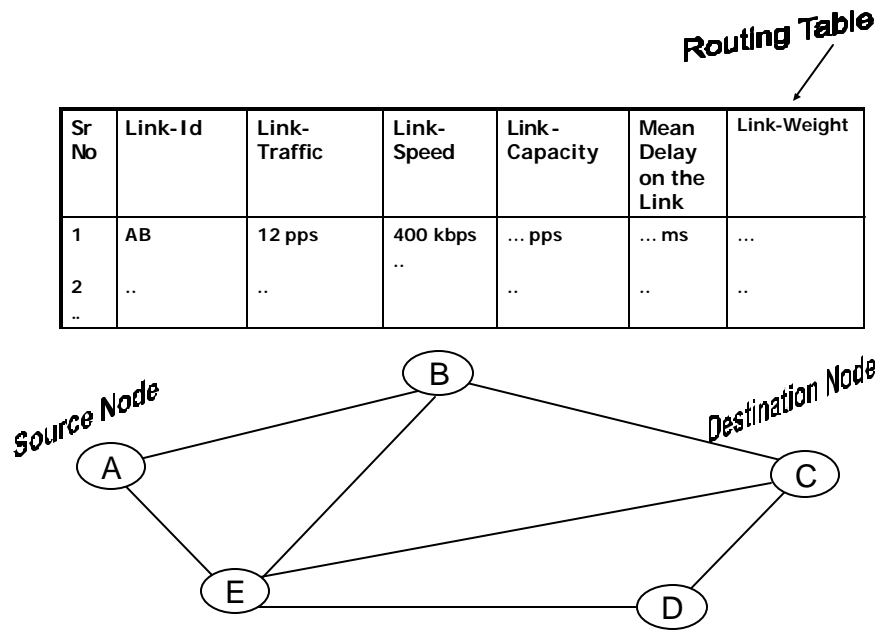


Fig. 6.1: A Sample Subnet and an Example Routing Table Structure

6.3 Classification of Routing Architectures

Algorithm-based Routing Architectures may be broadly divided into two classes, namely, Static Routing Architectures and Dynamic Routing Architectures. Major entries belonging to both of these classes include:

- Packet Flooding
- Random Routing
- Shortest Path Routing
- Flow-based Routing
- Distance Vector Routing
- Link State Routing

Organization-based Routing Architectures could have its genesis in the organization and functionality-based classification. Thus, these Routing Architectures could be broadly divided into the following categories, each of which could have optimality requirement:

- Hierarchical Routing
- Directory Routing
- Broadcast Routing
- Multicast Routing

Policy-based Routing Architectures are classified based on policies leading to the following classes:

- Policy-based QoS Routing Architectures
- Policy-based Security Routing Architectures
- Policy-based Hierarchical Routing Architectures
- Policy-based Session Routing Architectures

6.4 Shortest Path Routing:

This is one of the simplest routing schemes and the primary technique involved here is the determination of the shortest available path between a source and a destination.

The term shortest path may be interpreted in a variety of ways including:

- path of the least geographical distance
- path of the least congestion
- path of the least number of Hops
- path of the least mean queuing delay
- path of the least propagation / transmission delay
- Any weighted average based metric can be yet another choice for employing this scheme.

6.4.1 Dijkstra's Algorithm:

One of the best known algorithms that may be employed for determination of the shortest path is the one suggested by E. W. Dijkstra in as early as 1959. The gist of this strategy is given below.

1. Each node is labeled with the name of the source node and its distance from the current node. Normally, the labeling is done in the reverse order, i.e. the label (9, A) represents distance of the current node from the source node (9) followed by the name of the source node (A). <A label may be permanent or tentative.>

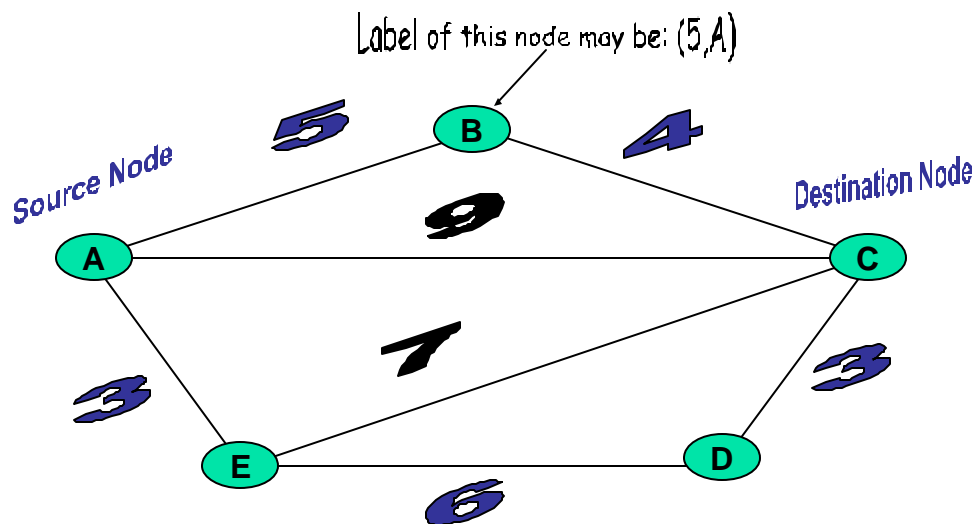


Fig. 6.2: Dijkstra's Routing Algorithm

2. At the start of the algorithm, all nodes are labelled tentatively.
3. As the algorithm progresses, the labels may change.

4. *At any stage, when it becomes clear that the current label represents the smallest distance / shortest path between a node and the source node, former's label is marked as a permanent label.*
5. *As the algorithm progresses, more and more nodes acquire permanent labels.*
6. *The algorithm terminates when the destination node gets a permanent label.*

6.5 Flooding Based Routing

Flooding-based Routing, as the name suggests uses redundant replication of incoming packets / NLDUs on available outgoing links. It has three major variants:

- Pure / Unconstrained Flooding (default flooding behaviour)
- Hop-Count Based / Constrained Flooding
- Selective / Direction-Constrained Flooding

6.5.1 Pure Flooding Algorithm

This algorithm is one of the simplest algorithms available to date that has a simple logic that suggests that if a packet arrives at a node that is member of the Flooding-based routing architecture, simply copy it (by replicating the original) on all outgoing links other than the link going back to the node wherefrom the packet has just arrived. Although under extreme unpredictability, this algorithm demonstrates consistent robustness and guaranteed delivery as long as at least one path leading to the destination is available, it is inherently an inefficient algorithm due to the possibility of indefinite circulation of packets / NLDUs.

6.5.2 Hop Count based Flooding Algorithm

This algorithm may be expressed as follows:

- *At any originating node 's', structure a packet such that its header contains a 'hop count' that be initialized to length of the path (if known) or full diameter of the subnet.*
- *At every intermediate node 'i' examine the incoming queue of packets, take the packet at the head of the queue and note the packet-id, line on which it arrived on, its hop count and destination address.*
- *Decrement the hop count by one (1).*
- *If the count becomes zero, discard / drop the packet and flush the corresponding entries in the local table.*
- *Otherwise, generate (n-1) replicas of the packet (where 'n' is number of arcs converging at this node) and transmit one replica on all arcs / lines except the one this packet arrived on.*

- *Examine the incoming queue and if it is non-empty, repeat steps 2 to 5 else wait until a new packet arrives and then repeat steps 2 to 5.*

6.5.3 Selective / Direction-Constrained Flooding Algorithm

It is a variant of the basic Flooding Algorithm with the constraint of direction thrown in for the purpose of improved efficiency. In this scheme, packets are selectively flooded by the routers in such a way that they move approximately in the right direction (i.e. leading towards the Destination).

6.6 Flow-based Routing Algorithm

This is yet another Static Routing Algorithm; but unlike the Shortest Path based Routing Algorithm and the Flooding based Routing Algorithm, which primarily consider the Subnet Topology alone, it considers Subnet Topology as well as Load (Traffic). This is particularly suitable for the subnets characterized by nearly stable average data transfer rate / mean data flow rate. In other words, this scheme may not prove to be effective if the mean inter-node data flow in a given subnet cannot be reliably predicted / estimated. This algorithm, unlike the other algorithms discussed so far, has several pre-requisites including the following:

- Subnet topology must be known in advance.
- Link / Line Capacity Matrix must be known in advance.
- Traffic Matrix must be available a priori.
- Mean packet-size must be known.
- Some preliminary Routing Algorithm must be available.

The scheme makes use of the fact that under the above stated circumstances, for each of the links, if the link-capacity, average rate of data-flow and topology are known and if the traffic -matrix and subnet topology is available in advance, then it is possible to:

1. Compute the mean delay in packet-delivery per link,
2. Compute the mean (overall) delay in packet-delivery over the given subnet,
3. Compute the most appropriate route between any pair of Source and Destination.

6.7 Distance Vector Routing Algorithm

This (DVR) is also known as the *Bellman-Ford* or *Ford-Fulkerson Routing Algorithm*. It is the original *Dynamic Routing Algorithm* used in the erstwhile ARPANET. For quite some time, it was popular over the Internet where a variant of it called *Routing Internet Protocol* (RIP) was used. Many Routers still use one or other variation of this algorithm. In brief, this scheme may be expressed as:

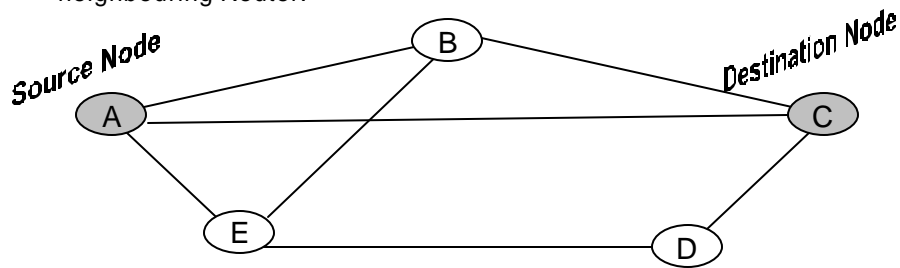
- Each Router knows / discovers its distance from its neighbours.
- Each Router locally maintains a Routing Table indexed by an entry for every other Router in the subnet and identification of a preferred neighbour / link leading to that Router.
- Metric of estimation may vary. For instance, it may be any one of Physical Distance, Hops, Delay etc.

- Periodically, each Router sends a Vector to its neighbouring Routers. As this vector contains estimated distances, it is called a Distance Vector.
- On receipt of such Vectors from its neighbours, every Router revises its estimates and updates its local routing table.

For the given subnet

2	B
13	C
5	E

where, the first column indicates Current Estimates and the second column refers to Identification Symbol for the corresponding neighbouring Router.



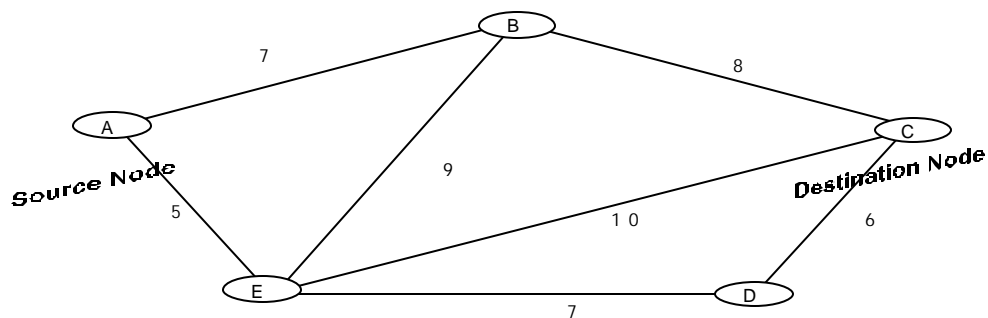
Destination	Distance	Next Hop Via Router
A	0	A
B	2	B
C	6	D
D	7	D
E	3	E

Fig. 6.3: Structure of a Distance Vector and Routing Table at the Router 'A'

Novell's well-known IPX used this scheme for quite a while. The primary drawback of this algorithm is its vulnerability to the 'Count-to-Infinity' problem. Solutions to this problem, though proposed from time to time, had little or no success. (Examples include the Split Horizon, Split Horizon with Poisoned Reverse etc.) Another drawback of this scheme is that it does not take into account Link Bandwidth and it takes appreciably long time for convergence. The default behaviour of the original DVR Algorithm about the requirement for transmitting a vector for each update brings yet another problem: instability as well as control-traffic overheads. (Here, Controlled or Triggered Updates of Vectors often eases the situation.) Due these inadequacies, the erstwhile ARPANET that was using this algorithm until 1979, had to switch over to the Link-State Routing Algorithms discussed in the following section. (Solutions like Source Tracing and Path Vectors led to new algorithms. The latter is used in the BGP.)

6.8 Link-State Routing Algorithm

As the name suggests, in this algorithm (LSA) exchange of the Link-State Packets over the subnet hold key to facilitating the routing process. In this algorithm, network topology and link costs are estimated by making each node broadcast what is referred as 'Link State Packets' carrying 'Identities of Neighbours' and 'Corresponding Link Costs' as shown in the Fig. 6.4. The basic idea involves the computation of the Local Routing Table by each Router on the basis of its own estimates and the similar Link State Broadcasts received from other routers in the subnet.



A	
11...001	
60	
B	7
E	5

Destination (Router)	Link-Cost	Next Hop (Router)	Hop Count
A	0	A	1
B	7	B	1
C	15	B	2
D	12	E	2
E	5	E	1

Source	Sequence No.	Age	Send Flags	Acknowledgement Flags	Data
--------	--------------	-----	------------	-----------------------	------

Fig. 6.4: Structure of a Link-State Packet, Routing Table and Packet Buffer (at Router A)

In a simple version, following this algorithm, each router:

- Discovers its neighbours and their Network Addresses by sending special packets called 'Hello' packets.
- Estimates delay / cost or any other metric for reaching its neighbours by sending another special packet called 'Echo' packets.

- Immediately applies its recent knowledge to form Link-state packet, which encapsulate this estimate; and, sends (broadcasts) the packet to all the discovered routers.
- Computes the shortest path to every other router using the Shortest Path Algorithm and updates the local Routing Table.
- Immediately forms fresh *Link-State Packets (LSPs)* and executes *link state broadcast*. (This is sometimes called '*Controlled Flooding*').

In general, fresh link-state packets are built either periodically or upon occurrence of an event like node-failure / link-failure / addition of a node or link / revival of a failed node or link. An appropriate choice of this strategy, in practice, has a bearing on the performance of the algorithm.

In a typical Link-state packet, the first row indicates the *Originating Router*, the second row refers to the Sequence Number of the link-state packet, third row shows the Age of the packet, the fourth and subsequent rows indicate estimated metrics for each of the neighbouring routers (B and E in this case).

Upon the termination of this algorithm, each router would have acquired knowledge about its neighbour along with associated least-cost path from the source which itself might have acquired such information from its neighbours. Thus, gradually, each Router in the subnet learns about the rest of the routers. The algorithm requires that names / identifiers representing the routers be unique (globally).

Two of the problems associated with this algorithm include the sequence number problem (including the wrap around problems and router-crash-reboot-reinitialise based sequence number problems) and the oscillation problem (due to oscillating costs). While the first problem can be effectively handled by using the so-called Lollipop Sequence Numbers (-N to +N-1), the second could be handled by using randomisation-based varied-periodicity for execution of the Link-State Algorithm at individual Routers. This randomisation also avoids the probability of self-synchronization of instances of LSA's execution at different routers in a subnet. An important aspect of the LSRP (Link State Routing Protocol) is that one or more corrupt LSP / LSPs can make the routing process unstable. Thus, a variety of techniques / schemes have been suggested in the literature to handle accidental or malicious corruption of LSPs. Use of the solution involving a combination of LSP-checksums and password-based authentication (passwords being known only to the administrators of the routing domain) takes care of this aspect as well.

Examples of some of the well-known implementations of this scheme include the *Open Shortest Path First (OSPF)* scheme and the *Intermediate System-Intermediate System (IS-IS)* scheme.

6.9 Hierarchical Routing Architectures

Any Routing Architecture that may support flat architectural model (i.e. a non-hierarchical architectural model) cannot be scalable as the number of routers in the internetworks continues to grow beyond a particular value. This is so because beyond this threshold level, the required Routing Table Space as well as the Routing Processing Time become so enormous that no physical router can really deliver the

requisite storage and processing capacities and expected performance. (Routing Table needs $O(n)$ space for an 'n'-router subnet.) The principle behind the Hierarchical Routing Architectures lies in partitioning the given large internetworks into multiple levels of hierarchy. (Please refer to Chapter-1 for recapitulation of some related information.)

6.9.1 The Interior Gateway Protocol (IGP)

Historically, in the ARPANET set-up, a 'Gateway' was a device that connected the campus networks to the ARPANET. The protocol that these gateways used to communicate within the campus networks was called the Interior Gateway Protocol.

6.9.2 The Interior Gateway Routing Protocol (IGRP)

This is a protocol based the Intra-AS Routing Architecture. Cisco Systems originally developed the Interior Gateway Routing Protocol (IGRP) in the mid-1980s. This protocol did not support VLSM scheme. Its successor, EIGRP, supports VLSM. Basic objective of the IGRP was to provide a robust protocol for routing within an Autonomous System (AS). The most commonly used AS-AS routing protocol prior to the advent of the IGRP was the Routing Information Protocol (RIP). As mentioned earlier, very small hop limit (only 16 hops) restricted the size of RIP based internetworks. Moreover, most of the problems discussed in the section describing the DVR, apply to the IGRP as well.

In the extended version of the IGRP, known as EIGRP (Extended Interior Gateway Routing Protocol), a technique based on an algorithm known as Distributed Update Algorithm (DUAL) has been employed. DUAL guarantees loop-free routing table generation irrespective of the frequency of changes in the subnet topology. The basic idea lies in the fact if a Router updates the existing Next Hop Router entry to contain a new Next Hop Router entry; then even if a loop develops, it would generate a cost that would be definitely higher than the direct cost, and consequently due to algorithm's behaviour to report the least cost-based Next Hop, the loop and its effect automatically become irrelevant. In other words, it may be said that in effect, this update that was caused by a reduction in the cost leading to any Router no loop can be formed.

6.9.3 The Exterior Gateway Protocol (EGP)

The Exterior Gateway Protocol (EGP) is an inter-domain (inter-AS) connectivity / reachability protocol. It is important to note here that in an Inter-AS scenario, though the Edge Routers / Border Routers / Gateways have to cooperate with one-another, the involved Autonomous Systems may not necessarily have the same degree of trust in one-another and therefore all Inter-AS Routing Protocols have to provide methods and means of configuration and enforcement of suitable checks without adding to the network overhead beyond acceptable limits.

The original version of the EGP that enjoyed quite a bit of popularity is gradually giving way to other competing exterior gateway routing protocols (like the BGP and the IDRP). This is because of certain weaknesses that came to light with the exponential growth of the Internet over the years.

6.9.4 The Border Gateway Protocol (BGP)

Currently, the most well known Inter-Autonomous System exterior Routing Protocol is the Border Gateway Protocol Version 4. Incidentally, BGP4 also happens to be the first version that is capable of handling the CIDR and Supernetting. BGP uses the TCP as its transport protocol of choice. (Advantage of this approach is that BGP can relieve itself of reliability specific concerns.)

BGP is a *Path Vector Protocol* (PVP) which is based on a major improvement in the Distance Vector Routing Protocol (DVRP). (Since, the routing information used by the BGP consists of a vector of Autonomous System ID Nos., which actually maps to a traversed path / route, it is called as a path vector protocol. Unlike the DVRP, the PVP does not suffer from the Count-to-Infinity problem) It is primarily used for exchange of information between autonomous systems.

6.10 Issues in Hierarchical Routing Architectures

Primary issues in the design of the Hierarchical Routing Architectures include determination of the optimal number of levels of routing hierarchies (as discussed briefly in Chapter-1), trust-mapping, QoS-mapping, cost-mapping, translation of data between different interior and exterior routing protocols (the Border Routers face this problem as they mediate between the interior and exterior routing domains) and choice of parameters and mechanisms for collaborative network monitoring.

Currently, OSPF is the preferred Intra-AS Routing Protocol in the Internet whereas the preferred Inter-AS Routing Protocol is the BGP. (Earlier, these positions were held by the RIP and EGP respectively.)

6.11 Summary

There are two primary classes of routing strategies: Centralized Routing and Distributed Routing. In Distributed Routing, routing choices are made locally, in a collaborative / pre-planned manner. Problems in packet routing arise from Loss of packets, Receipt and circulation of duplicate packets, Packet Choking / Network Congestion, Network Cleansing, Route Tracing, Network Management Support etc. *Algorithm-based Routing Architectures* may be broadly divided into two classes, namely, Static Routing Architectures and Dynamic Routing Architectures. Sub-categories include: Packet Flooding, Random Routing, Shortest Path Routing, Flow-based Routing, Distance Vector Routing, Link State Routing, Hierarchical Routing, Directory Routing, Broadcast Routing and Multicast Routing. Policy-based Routing Architectures provide yet another specialized classes of routing architectures.

Distance Vector Routing Algorithm (DVR) is also known as the *Bellman-Ford* or *Ford-Fulkerson Routing Algorithm*. It is the original *Dynamic Routing Algorithm* used in the erstwhile ARPANET. In the DVR, each Router locally maintains a Routing Table indexed by an entry for every other Router in the subnet and identification of a preferred neighbour / link leading to that Router. Periodically, each Router sends a Vector to its neighbouring Routers. On receipt of such Vectors from its neighbours, every Router revises its estimates and updates its local routing table. (Solutions like Source Tracing and Path Vectors led to new algorithms.

Link-State Routing Algorithm (LSA) exchange of the Link-State Packets over the subnet hold key to facilitating the routing process. In a simple version, each router discovers its neighbours and their Network Addresses by sending special packets called 'Hello' packets, immediately applies its recent knowledge to form Link-state packet, which encapsulate this estimate; and, sends (broadcasts) the packet to all the discovered routers, computes the shortest path to every other router using the Shortest Path Algorithm and updates the local Routing Table and immediately forms fresh *Link-State Packets (LSPs)* and executes *link state broadcast*. In general, fresh link-state packets are built either periodically or upon occurrence of an event like node-failure / link-failure / addition of a node or link / revival of a failed node or link. The algorithm requires that names / identifiers representing the routers be unique (globally).

The Interior Gateway Routing Protocol (IGRP) is a protocol based the Intra-AS Routing Architecture. Cisco Systems originally developed the Interior Gateway Routing Protocol (IGRP) in the mid-1980s. This protocol did not support VLSM scheme. The most commonly used AS-AS routing protocol prior to the advent of the IGRP was the Routing Information Protocol (RIP). 6.9.2 The Exterior Gateway Routing Protocol (EGRP).

The Exterior Gateway Routing Protocol (EGRP) or the Exterior Gateway Protocol (EGP) is an inter-domain (inter-AS) connectivity / reachability protocol. Currently, the most well known Exterior Gateway Routing Protocol is the Border Gateway Protocol Version 4. BGP is a *Path Vector Protocol (PVP)* which is based on a major improvement in the Distance Vector Routing Protocol (DVRP).

6.12 Recommended Readings

1. A. S. Tanenbaum: **Computer Networks**, Fourth Edition, Prentice-Hall, Upper Saddle River, 2002.
2. B. A. Forouzan & C. H. Fegan: **TCP/IP Protocol Suite**, Second Edition, Tata-McGraw-Hill Publishing Co. Ltd., New Delhi, 2002.
3. Cisco staff: **Internetwork Design Guide**, 1999 available at: <http://www.Cisco.com/univercd/cc/td/doc/cisintwk/idg4/nd2002.htm#xtocid229276>
4. Cisco staff: **Internetworking Case Studies**, Cisco Press, 1996.
5. Cormac Long: **IP Network Design**, Osborne-McGraw-Hill, Berkeley, 2001.
6. D. Comer & D. L. Stevens: **Internetworking with TCP /IP**, Vols. 2-3, PHI, 1994, 1993.
7. D. Comer: **Internetworking with TCP / IP**, Vol. -1, Fourth Edition, Pearson Education, New Delhi, 2001.
8. Dave Koiur: **IP Multicasting: The Complete Guide to Interactive Corporate Networks**, John Wiley & Sons, New York, 1998.
9. Garry R. McClain (Ed.): **Handbook of Networking and Connectivity**, AP Professional, 1994.
10. James Kurose & Keith W. Ross: **Computer Networking**, Second Edition, Pearson Education, New Delhi, 2002.

11. K. Downes, Marilee Ford, H. K. Liu, Steve Spanier & T. Stevenson : **Internetworking Technologies Handbook**, Second Edition, Cisco Press / Techmedia, 1999.
12. Paul T. Ammann: **Managing Dynamic IP Networks**, Tata-McGraw-Hill Publishing Co. Ltd., New Delhi, 2001.
13. Rahul Banerjee: **Lecture Notes on Computer Networks**, Nov. 2002, available on-line at: <http://www.bits-pilani.ac.in/~rahul/CN/index.html/>
14. **RFC 1009** (Requirements for Internet Gateways)
15. **RFC 1011** (Official IP)
16. **RFC 1124** (Policy Issues in Interconnecting Networks)
17. **RFC 1125** (Policy Requirements for Inter-Administrative Domain Routing)
18. **RFC 1147** (FYI: A list of Network Management Tools)
19. **RFC 1175** (FYI: A very useful reference-list on Internetworking related information)
20. **RFC 1208** (Glossary of Networking Terms)
21. **RFC 1360** (Official Protocol Standards of the Internet Architecture Board)
22. **RFC 1825** (IP Security Architecture)
23. **RFC 1826** (IP Authentication Header)
24. **RFC 1827** (IP Encapsulation Security Payload)
25. **RFC 1887** (IPv6 Unicast Addressing)
26. **RFC 1971** (IPv6 Address Autoconfiguration)
27. **RFC 781** (IP Timestamp)
28. **RFC 791** (IP version 4)
29. **RFC 815** (IP Datagram Reassembly)
30. S. Keshav: **An Engineering Approach to Computer Networking**, Addison-Wesley, Reading, 1997.
31. Smoot Carl-Mitchell & John S. Quarterman: **Practical Internetworking with TCP / IP and UNIX**, Addison-Wesley, Reading, 1993. (This book does not really discuss the IPv6. This however, helps the reader to take a look at the pre-IPv6 days and realize the wisdom of evolution of the IP.)
32. Uyless D. Black: **TCP / IP & Related Protocols**, Second Edition, McGraw-Hill, N. Y., 1995.
33. W. Buchanan: **Advanced Data Communication and Networks**, Chapman & Hall, London, 1997.
34. Y. Zheng & S. Akhtar: **Networks for Computer Scientists and Engineers**, Oxford University Press, New York, 2001.

6.11 Exercises

1. Compare the routing strategies applicable to VC-switched routing to the packet routing strategies.
2. What are the strengths and weaknesses of the Distance Vector Routing scheme?
3. What is the difference between the Distance vector Routing and Path Vector Routing? Why is there no count-to-infinity problem in the latter case?
4. Compare Link State Routing scheme with the Distance Vector Routing scheme and comment on their suitability of application in hierarchical routing, if any.
5. What is the primary difference between the Intra-AS and Inter-AS Routing Architectures?

6. Why does the Open Shortest Path First Protocol provide a password-based authentication scheme? Who holds this password and how does the scheme work?
7. How can a malicious user introduce routing instability in the Link State Routing scenario and what can be done for preventing such attacks from affecting the subnet?
8. Why does the BGP make use of TCP's capabilities?
9. What is meant by BGP4+? Explain with the help of an example.

Chapter -7

Internetwork Management Architectures

Interaction Goals

Objectives of this chapter are to introduce the fundamental concepts of network and internetwork management including local as well as remote networks.

At the end of this chapter, you should be able to:

- Identify the basic elements of an Internetwork Management Architecture (IMA),
- Tailor any combination of network management services,
- Evolve your own IMA as per requirements of a situation.
- Analyse the correctness of the IMA design approach,
- Tell about how to extend an existing IMA design without throwing away existing set-up; and,
- Differentiate between various direct and indirect / hidden design constraints and their consequences.

The treatment assumes the working knowledge of Computer Networks and some exposure to Operating Systems as well as Data Communication areas.

7.1 Introduction

As the networks grow into internetworks and the internetworks interconnect to form bigger internetworks, need for being able to design sound management architecture becomes obvious. The primary objective of the Internetwork Management Architecture (IMA) is therefore to provide a flexible and robust framework using which effective internetwork management could be ensured.

Due to increasing complexities of the networks, managing them has become a complex job by itself and a host of protocol stacks in the real world do not serve to simplify the problem. Emergence of Network Operations Centres (NOCs) that monitor the global network status, maintain the health of the network and assist in an early restoration of the network operations in case of abnormal network behaviour or intermittent network crash etc. has its roots both in the referred complexity of the tasks as well as the plain economics of scale.

It is in this context, that this chapter attempts to briefly introduce you to this increasingly challenging area of internetwork management. Fig. 7.1 depicts a simple extensible Internetworking Management Architecture that shows the core constituents of object-oriented organization-based agent-driven management systems.

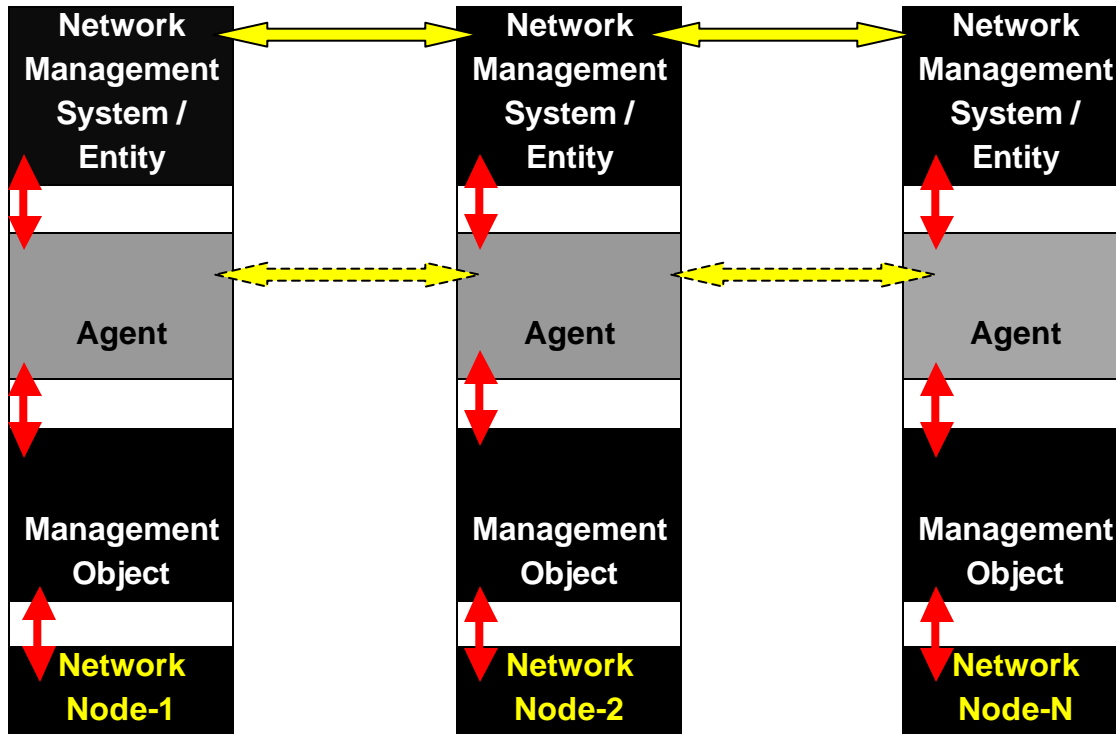


Fig. 7.1: Architecture of an Internetwork Management Architecture: Functional View

Fig. 7.2, on the other hand, depicts the Protocol view of the same IMA. It may be interesting to notice that these two depictions are simply two different perspectives of the same system of internetwork management but have two completely different purposes from the viewpoint of an architect / engineer. While the former view is meant to separate functions that could in effect help in modularising the different component-code design elements; the latter is meant to separate sets of rules and conventions used in the interaction of one or more of the entities involved in network operations, administration / monitoring / control, provisioning / allocation and behavioural / performance analysis related activities: those, put together, form the complete set of network management functionalities.

Security is one other aspect that is implicit in these representations but does not, in any way, affect the functionality of the overall system except for the overheads added by its implementation.

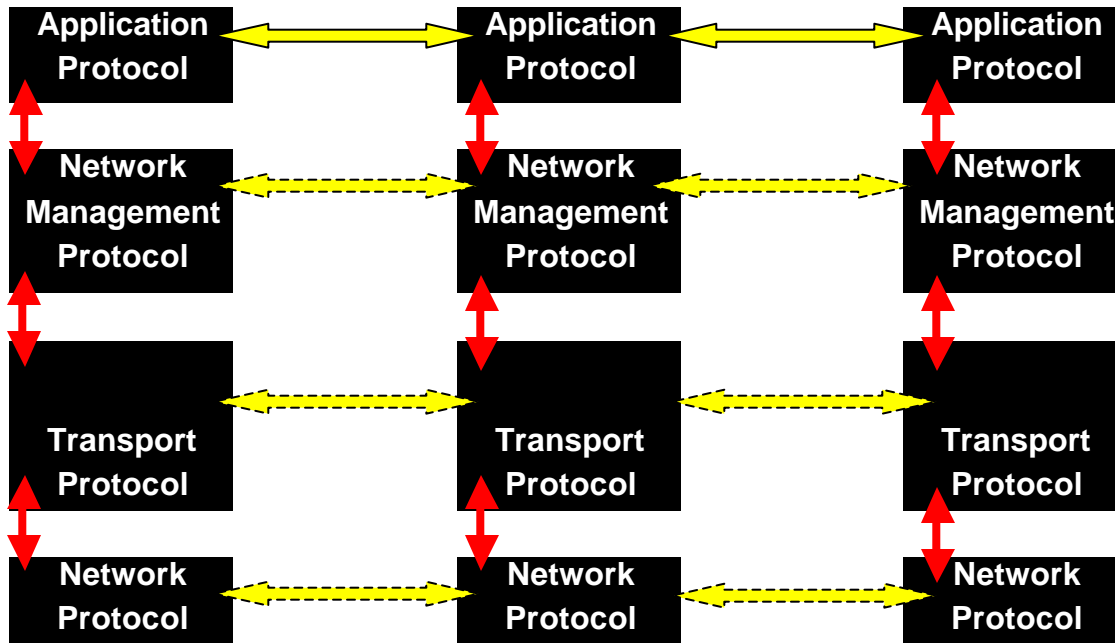


Fig. 7.2: Architecture of an Internetwork Management Architecture: Protocol View

A look at the Table 7.1 indicates that a network / internetwork can make use of early symptoms for effectively reducing the chances of network malfunction / under-performance / failure but also assist in effecting an early automated / manual / hybrid recovery and thus help in maximizing the network uptime and productivity at remarkably low cost.

Type	Problem	Symptom
1	Improper Address Management	Network access failure due to erroneous IP / IPX / NL Address binding to the relevant MAC Address
2	Improper Fault Management	Node failure (most common cause that includes the fault in the node processor, memory, fabric or interface: the last being the commonest cause), Link Failure, Connection failure preceded by marked deterioration in network performance
3	Improper Media Management (partly overlaps with the <i>Type Two</i> problem)	Recurring but intermittent failure that is not due to fault in the node or breakage in the medium
4	Improper Power Management (partly overlaps <i>Type Two</i> problem)	Access Failure due to unintended configuration change
5	Improper Security Management	Intermittent or frequent intrusions without trace, Fatal access errors, Prolonged / Repeated data delivery

Table 7.1: Symptoms of Internetwork Troubles: A Diagnostic View

These are these five principal internetwork management problem classes that play dominant roles in different internetwork management scenarios and as such all IMAs need to provide support (internal or external) relevant management services. The sole exception to this rule, in some cases, can be the *Type Five* problem, support services for which may be at times provided by a full-fledged security architecture (see Chapter-8).

Naturally, it is only logical that a variety of IMAs have evolved over a period of time in view of complexities involved in the internetwork management. Most of these are built around a few network / internetwork management standards including:

- OSI-CMIP: The Common Management Information Protocol (The OSI standard for the network management),
- IEEE 802.x Standards for LAN / MAN,
- JMX: The Java Management Extension etc.
- SNMP: The Simple Network Management Protocol (meant for the TCP/IP internetworks like the Internet and proposed at the IETF),
- TMN: The Telecommunications Management Network (The ITU-T standard for the network management),
- WBEM: The Web-based Network Management standards,

Out of these, the SNMP is the most common protocol due to its relatively simple and distributed architecture. However, out of the Network Management Models / Architectures, the most well structured model is the one derived from the OSI Reference Model shown in Fig. 7.3. The ISO-OSI Network Management Model 10xxx and the ITU X.7xx Network Management Framework are functionally equivalent.

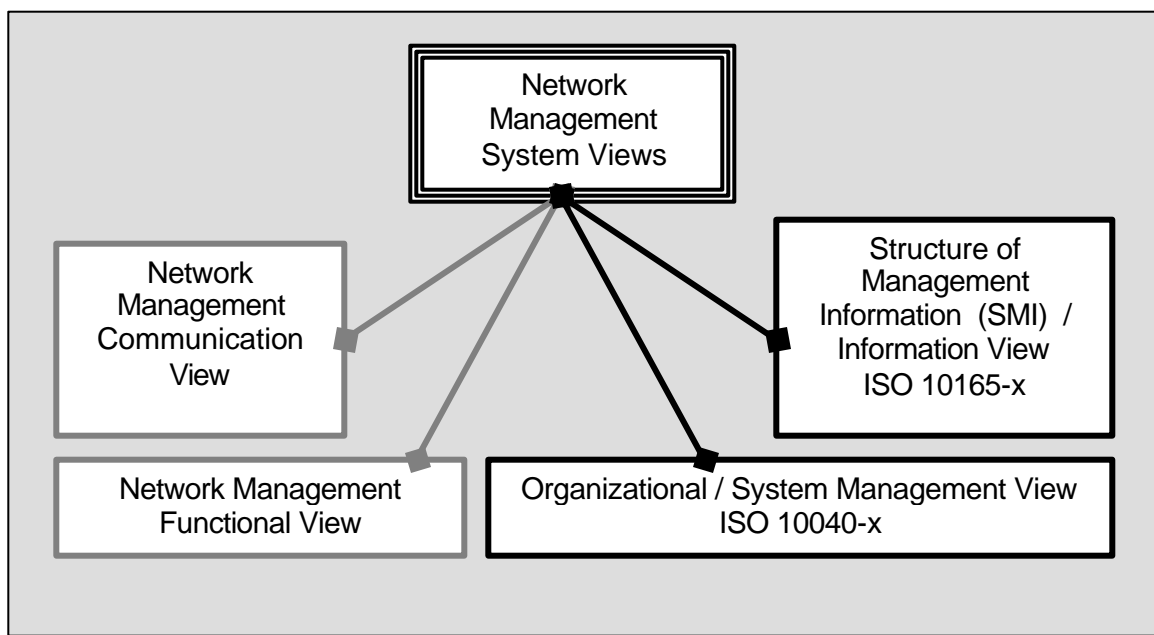


Fig. 7.3: The ISO's OSI Network Management Reference Model / Framework

7.2 The Simple Network Management Protocol

As shown in the Fig. 7.4, the Simple Network Management Protocol (SNMP) has been continuously evolving and it may be difficult for the designers to claim that their design is necessarily up-to-date. Fortunately, to make the things simple, all the older versions have been accommodated in the newer versions of the SNMP. SNMPv3 Architecture, for instance, has included the SNMPv1 and SNMPv2 compatibility.

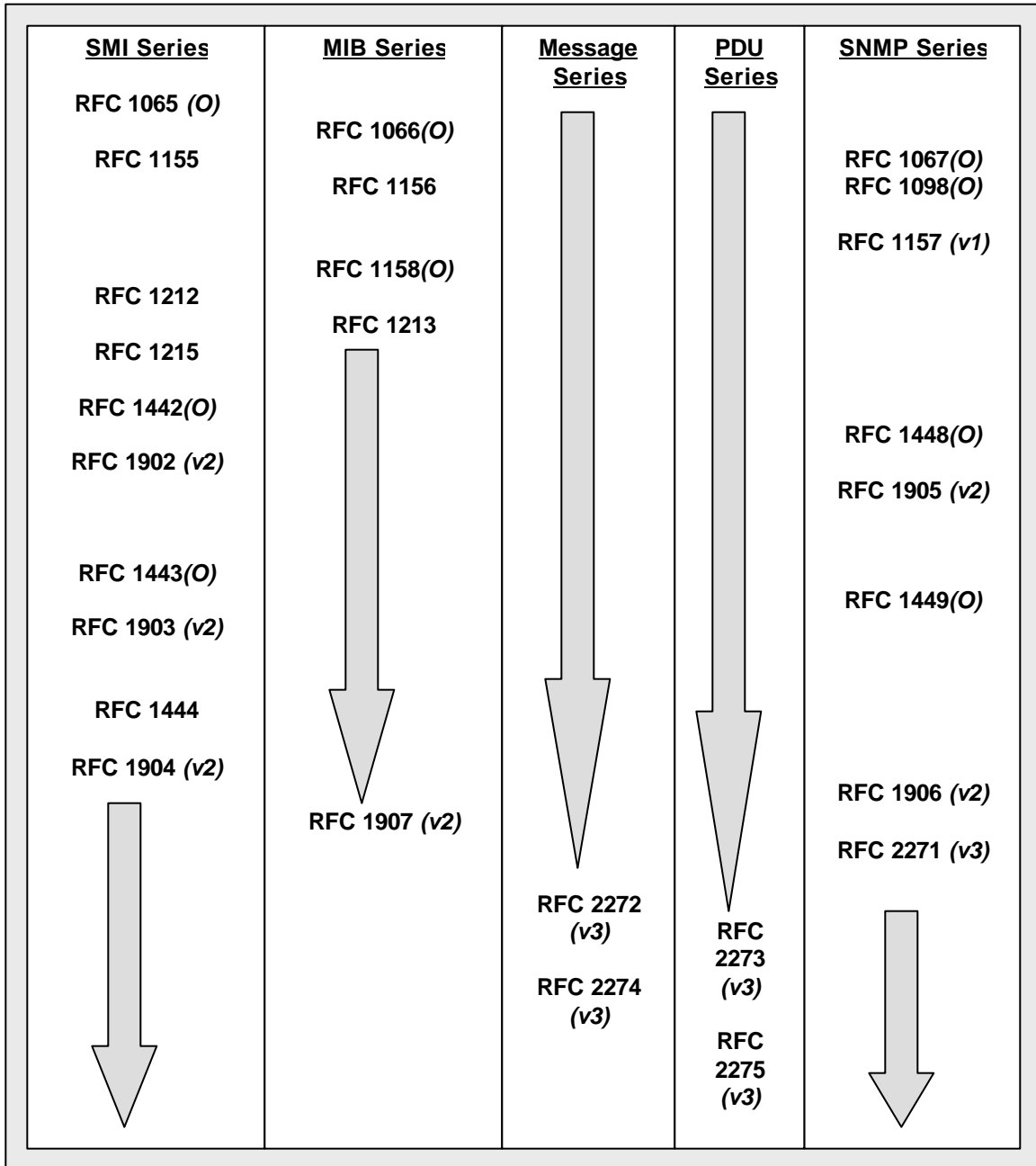


Fig. 7.4: The IETF Journey of SNMP: SNMPv1, SNMPv2 and SNMPv3

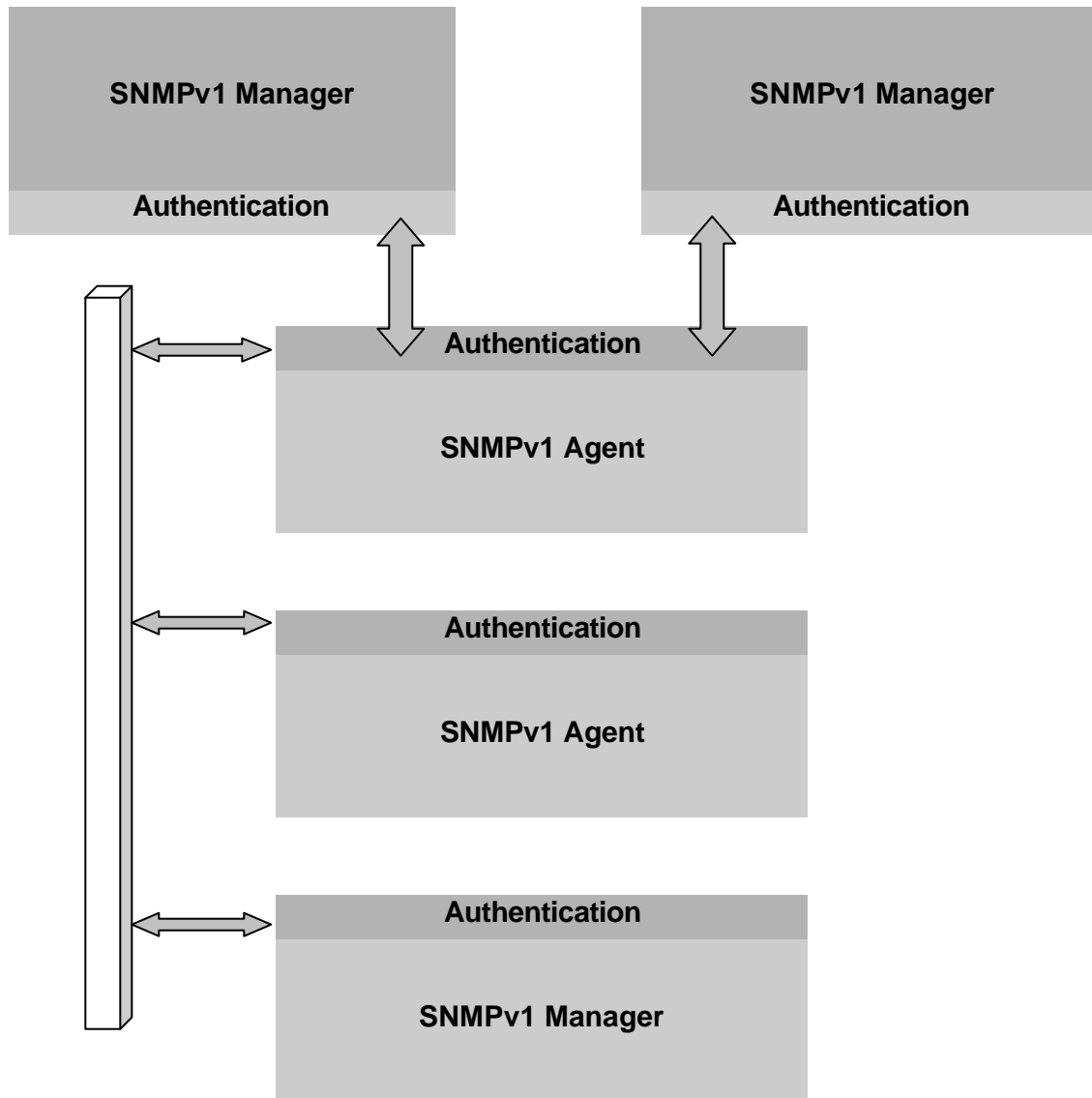


Fig. 7.5: The SNMP Architecture: Perspective of the SNMPv1

A typical encapsulated SNMP Protocol Data Unit (called as SNMP PDU) comprises of an Application Header, a Version Identifier, an SNMP Community Descriptor, and the SNMP Data. At the Transport Layer, this is taken as the payload of the UDP forming the UDP PDU or the Transport PDU. Its further encapsulation in IP and subsequent encapsulation in the lower layer PDU are the next two stages following which the physical layer takes over and at the receiver end the process is reversed as usual.

The SNMP Network Management System, as discussed earlier, has an element called the SNMP Manager. This element maintains a database comprising of two sets of data; one that is static in nature, contains information about the objects and is known as the

Management Information Base (MIB) and the other that is dynamic in nature and contains the measured values of the objects. MIB, as is obvious, is not a true database by definition. However, while the SNMP Manager has both of these the MIB and the dynamic object-value database, the SNMP Agent has just the MIB.

Fig. 7.6 shows the abstract view of the SNMPv1 Architecture and the Fig. 7.6 shows the mechanism of SNMP Proxy using which the SNMP Community can communicate with a non-SNMP Managed Community.

A major initial attempted enhancement in the SNMPv2 was the provision of the security features that was not included in final specification due to lack of agreement within the SNMP developer community. However, the SNMPv2 specification had brought in several important changes in form of the addition of messages (Bulk Data Transfer Message and Manager-Manager Message), addition of compliance testing metric (through the Conformance Statements), addition of new object in the Object Table, evolution of the SMIv2, addition of two new MIB sub-groups (the SNMPv2 and Security sub-groups) and enhancement in the list of allowable Transport Level Protocols (in addition to the default UDP). One major problem in the initial acceptance of the SNMPv2 was its lack of backward compatibility with the SNMPv1.

In the latest version of the SNMP: the SNMPv3, principal features include a structural formally defined *Network Management Architecture* that encompasses the entire range of SNMP versions (1,2 and 3) as well as a formalized *Management Security Model / Framework* that encompasses factors like verification / authentication of the origin of data, data-integrity, confidentiality, timeliness and a certain degree of message replay protection. It employs an extended Access Control Model known as the View-based Access Control Model (VACM) that features greater control and flexibility of configuration.

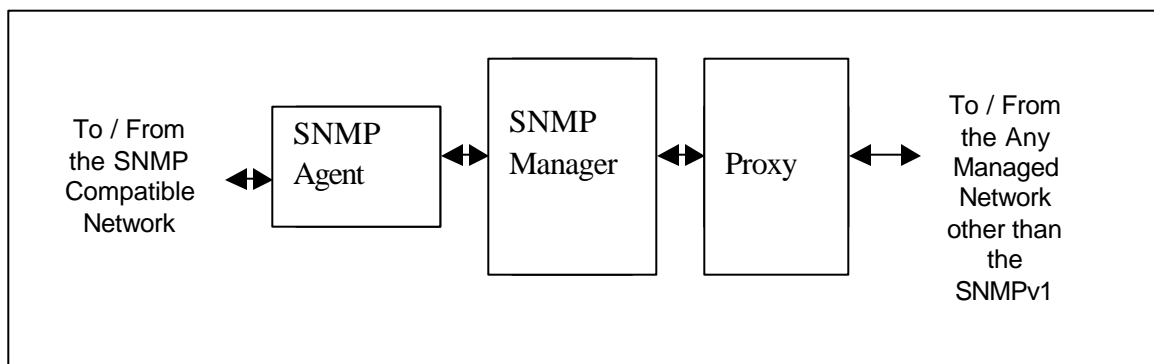


Fig. 7.6: Role of the SNMP Proxy Server

One form of SNMP and Web-interface based NMS is pretty common these days that makes use of the SNMP queries and that internally functions as a polling-oriented

scheme. This form of NMS should not be confused with a true Web-based NMS that necessarily uses the Web-Client and Web-Server combination with the latter having a built-in Agent meant to monitor, control and thus manage as requested by the Web-Client (typically a browser). A well-known standard for the Web-based Network Management is known as the Desktop Management Interface (DMI). DMI, SNMP both were meant to complement each other but in some way have proved to be competitors. This situation has led to the formation of the Desktop Management Task Force that has brought out the framework to integrate various network management standards under the name Web-Based Enterprise Management (WBEM). Sun's Java Management Extensions (JMX) is a Java-based framework for making Java Applet-based management tools as Web-based management extensions with Java-embedding.

7.2.1 The SNMPv3 Architecture

The SNMPv3 Architecture can be seen as a two-part architecture: the *SNMP Base Architecture* that forms the *SNMPv3 Management Engine*; and, the *SNMP Management Command Interface Architecture* that defines the *SNMP-specific applications* like Command Handler, Alerter, Forwarder etc. The SNMP version 3, as shown in the Fig. 7.4, has been described in the RFCs 2271-2275. These five specifications collectively describe the *SNMPv3 Document Architecture*.

A typical view of the SNMPv3, in this case, may depict a multi-node network/internetwork in each node of which an SNMPv3 entity resides such that constituents of each SNMPv3 entity (Management Engine and Application Interface) and their respective attributes assist these nodes in exchange, interpretation, monitoring and managing the configured network/internetwork, as the case may be.

In SNMPv3, an SNMP Entity has a minimum of nine types of desired control hooks, each hook leading to a functional sub-system within every SNMP Entity:

- | | | |
|---|---|--|
| <i>Core /
Kernel of the
SNMPv3
Entity</i> | { | <ul style="list-style-type: none"> • Access Control • Dispatch Control • Security Control • Message Processing and Control |
| <i>Extra-Kernel
elements of
the SNMPv3
Entity</i> | { | <ul style="list-style-type: none"> • Command Control • Command-Response Control • Notification Control • Notification-Response Control • Proxy Forwarding Control |

First four of the controls are implemented through four individual functional subsystems called as Access Controller, Dispatcher, Security Controller and Message Processor.

Collectively, these four sub-systems form the core/kernel of the SNMPv3 Entity and known as the SNMPv3 Management Engine often simply called as SNMPv3 Engine. In the SNMP syntax, *snmpEngineID* refers to the specific unique identifier associated with an SNMPv3 Engine. An SNMPv3 Engine ID is distinguished from its earlier versions with the help of the first bit (1' in case of v3) of the *snmpEngineID*. It has provisions for indicating IPv4 as well as IPv6 address type in its 'Format Indicator' field. An SNMPv3 Entity is named by its unique SNMPv3 Engine ID and there exist two types of names that are associated with any ID as per the SNMPv3 syntax; namely, *principal* (the service requesting user's or application's name) and *securityName* (the pronounceable string associated with the first name) respectively.

Rest of the controls (defined as SNMP Applications in the RFC 2273) are implemented through five other separate functional subsystems called as SNMP Command Generator, SNMP Command Responder, SNMP Notification Originator / Generator, SNMP Notification Responder / Receiver, SNMP Proxy Forwarder respectively; and collectively, they form the extra-kernel (not to be confused with the Operating System Kernel that sits at a lower level).

The SNMP Command Generator is used to generate a variety of single / bulk *get* (e.g. *get-request* and *get-bulk*) as well as *set* (*set request*) messages. In a typical scenario, an NMS sends a *get-request* to an *SNMP Agent* and it is this *Agent* that coordinates the next step forward. The SNMP Command Responder is used to react to the requests generated by the SNMP Command Generator application invoked by a remote SNMP Entity. The SNMP Notification Originator / Generator is used to originate / generate an alert / notification message (sometimes called trap) along with version and security parameters on occurrence of certain events. The SNMP Notification Responder / Receiver first gets an explicit registration at the SNMPv3 Engine and thereafter receives the alert / notification messages and reacts as defined. The SNMP Proxy Forwarder is positioned such that with explicit restrictions it could be used to forward permitted SNMP traffic including various SNMP requests, responses and alerts. (In principle, SNMPv3 Proxy Forwarder is functionally the same as the SNMP Proxy Server of the earlier version.)

The SNMPv3 specification permits extensibility by the way of permitting addition of any number of appropriate extra-kernel functional blocks in addition to the ones described here. (In fact, already some such blocks have been defined.)

In the SNMPv3, various subsystems comprising an SNMPv3 Entity communicate with each other through a simple structured interface. The specification also presents the conceptual abstraction of such an interface that is application independent and offers a range of general services to the requesting subsystems. This interface is known as the *Abstract Service Interface (ASI)*.

A major enhancement visible in the SNMPv3 is the provision of Security Management in SNMP transactions in form of measures for authentication, privacy management, authorization / access control and flexibility of choice of mutually agreed security protocols / algorithms. In the SNMPv3, the User-based Security framework has been advocated and a set of security protocols / algorithms and privacy protocols / algorithms has been recommended. During the interaction between two SNMPv3 Engines, at any given point of time, based on specified governing rules, one acts as an Authoritative SNMP Engine while the other acts as the Non-Authoritative SNMP Engine.

7.3 The Remote Monitoring Protocol

It was the unprecedented success of the SNMP that led to the chain of developments each building onto the success of its predecessor. This chain included the evolution of remote network operation through the network operations centre, network fault management methods, network configuration management methods, statistical parametric measurement methods and eventually the evolution of the full-fledged specifications for the Remote Network Monitoring (RMON).

The basic principle behind the RMON is simple: collect / probe and analyse the network monitoring information locally and then communicate the extracted information in a pre-defined format and manner to a remotely located network node functioning as a designated Network Management Station (NMS). The local probes that serve to reliably provide specific monitoring information to the remote NMS are called as RMON devices. Use of these RMON devices helps in reducing the SNMP-specific traffic overhead in the WAN-subnet, requires lesser use of active network agents (instead of continuous visibility requirement of such agents to the network management entities), ensuring relatively accurate solicitation and interpretation of the results of the monitoring responses; and, finally, allows effective continuous monitoring of LAN segments. All these features effectively translate into assurance of higher network reliability as well as enhanced availability without any significant addition to the operational costs.

The RMON-MIB defines RMON groups. RFC 1271 first described the RMON-1 in the year 1991 that was replaced by the version described in the RFC 1757 (Ethernet-specific version) in 1995 and the subsequently the latest stable implementations have been built around the RFC 2021 (appeared in 1997) that describes the RMON-2.

7.4 Role of Agents in Internetwork Management

When it comes to the IMAs, the internetwork management makes some of the best-known uses of the Agent Technology. In principle, the network management agents are typically designed to be small, low resource hungry, efficient, object oriented and robust: though not necessarily intelligent.

In specific context of the SNMP, out of the five simple protocol messages (*get-request*, *get-next-request*, *set-request*, *get-response* and *trap*) only the first three are generated by the SNMP Applications whereas the remaining two are generated by the SNMP Agents.

7.5 Summary

The basic goal of the Internetwork Management Architecture (IMA) is to provide a flexible and robust framework using which effective internetwork management could be ensured.

Emergence of Network Operations Centres (NOCs) that monitor the global network status, maintain the health of the network and assist in an early restoration of the network operations in case of abnormal network behaviour or intermittent network crash etc. has its roots both in the referred complexity of the tasks as well as the plain economics of scale.

OSI-CMIP is the Common Management Information Protocol (The OSI standard for the network management), JMX is the Java Management Extension, TMN is the Telecommunications Management Network (the ITU-T standard for the network management) and WBEM is the Web-based Network Management standards. The ISO-OSI Network Management Model 10xxx and the ITU X.7xx Network Management Framework are functionally equivalent.

SNMPv3 Architecture has included the SNMPv1 and SNMPv2 compatibility. It can be seen as a two-part architecture: the *SNMP Base Architecture* that forms the *SNMPv3 Management Engine*; and, the *SNMP Management Command Interface Architecture* that defines the *SNMP-specific applications* like Command Handler, Alerter, Forwarder etc. Collectively, these four sub-systems form the core/kernel of the SNMPv3 Entity and known as the SNMPv3 Management Engine often simply called as SNMPv3 Engine. In the SNMP syntax, *snmpEngineID* refers to the specific unique identifier associated with an SNMPv3 Engine. Rest of the controls (defined as SNMP Applications in the RFC 2273) are implemented through five other separate functional subsystems called as SNMP Command Generator, SNMP Command Responder, SNMP Notification Originator / Generator, SNMP Notification Responder / Receiver, SNMP Proxy Forwarder respectively; and collectively, they form the extra-kernel (not to be confused with the Operating System Kernel that sits at a lower level).

The SNMP Command Responder is used to react to the requests generated by the SNMP Command Generator application invoked by a remote SNMP Entity. The SNMP Proxy Forwarder is positioned such that with explicit restrictions it could be used to forward permitted SNMP traffic including various SNMP requests, responses and alerts. (In principle, SNMPv3 Proxy Forwarder is functionally the same as the SNMP Proxy Server of the earlier version.)

In the SNMPv3, various subsystems comprising an SNMPv3 Entity communicate with each other through a simple structured interface. This chain included the evolution of remote network operation through the network operations centre, network fault management methods, network configuration management methods, statistical parametric measurement methods and eventually the evolution of the full-fledged specifications for the Remote Network Monitoring (RMON).

This chain included the evolution of remote network operation through the network operations centre, network fault management methods, network configuration management methods, statistical parametric measurement methods and eventually the evolution of the full-fledged specifications for the Remote Network Monitoring (RMON).

7.6 Recommended Readings

1. Information processing systems - Open Systems Interconnection - **Specification of Abstract Syntax Notation One (ASN.1)**, International Organization for Standardization. International Standard 8824, December 1987.
2. J. Case, Fedor, M., Schoffstall, M., Davin, J., Simple **Network Management Protocol**, STD 15, RFC 1157, May 1990.
3. J. Case, McCloghrie, K., Rose, M., and Waldbusser, S., Coexistence **between version 1 and version 2 of the Internet-standard Network Management Framework**, RFC 1452, April 1993.
4. J. Case, McCloghrie, K., Rose, M., and Waldbusser, S., **Manager-to-Manager Management Information Base**, RFC 1451, April 1993.
5. J. Case, McCloghrie, K., Rose, M., and Waldbusser, S., **Structure of Management Information for version 2 of the Simple Network Management Protocol (SNMPv2)**, RFC 1442, April 1993.
6. J. Case, McCloghrie, K., Rose, M., and Waldbusser, S., **Textual Conventions for version 2 of the Simple Network Management Protocol (SNMPv2)**, RFC 1443, April 1993.
7. J. Case, McCloghrie, K., Rose, M., and Waldbusser, S., **Conformance Statements for version 2 of the Simple Network Management Protocol (SNMPv2)**, RFC 1444, April 1993.
8. J. Case, McCloghrie, K., Rose, M., and Waldbusser, S., **Management Information Base for version 2 of the Simple Network Management Protocol (SNMPv2)**, RFC 1450, April 1993.

9. J. Case, McCloghrie, K., Rose, M., and Waldbusser, S., **Protocol Operations for version 2 of the Simple Network Management Protocol (SNMPv2)**, RFC 1448, April 1993.
10. J. Case, McCloghrie, K., Rose, M., and Waldbusser, S., **Transport Mappings for version 2 of the Simple Network Management Protocol (SNMPv2)**, RFC 1449, April 1993.
11. J. Case, M. Fedor, M. Schoffstall and J. Davin: **A Simple Network Management Protocol (SNMP)**, RFC 1157, May 1990.
12. J. Galvin and McCloghrie, K., **Administrative Model for version 2 of the Simple Network Management Protocol (SNMPv2)**, RFC 1445, April 1993.
13. J. Galvin and McCloghrie, K., **Security Protocols for version 2 of the Simple Network Management Protocol (SNMPv2)**, RFC 1446, April 1993.
14. K. McCloghrie & M. Rose: **Management Information Base for Network Management of TCP/IP-based internets**, RFC 1156, May 1990.
15. M. Rose and McCloghrie, K., **Concise MIB Definitions**, RFC 1212, March 1991.
16. M. Rose and McCloghrie, K., **Structure and Identification of Management Information for TCP/IP-based internets**, RFC 1155, May 1990.
17. McCloghrie, K., and J. Galvin, **Party MIB for version 2 of the Simple Network Management Protocol (SNMPv2)**, RFC 1447, April 1993.

7.7 Exercises

1. Study the referred RFCs on various versions of the SNMP and compare the management capabilities, security features and performance constraints of the major versions.
2. Microsoft's Common Information Model (CIM) is based on the WBEM. Are these two frameworks identical? Comment in term of architecture, capabilities and interoperability.
3. It is said that the MIB is not a true dynamic database. Then, why is it used at all and that too in the SNMP Manager as well as the SNMP Agent?
4. What are principal features of the SMIv2 and how do they relate to the SMI, if at all?
5. Why is it said that a Web-Interface-based NMS is different from a true Web-based NMS?
6. What are the security features available in the SNMPv3 and what are the associated performance trade-offs, if any?
7. Compaq was one of the first few companies that had adopted the DMI. Study the Compaq DMI and compare it with the Microsoft's CIM-based solution.

Chapter 8

Internetwork Security Architectures

Interaction Goals

Objectives of this chapter are to introduce the basic concepts related to the security aspects of all varieties of internetworks and the applications that run atop them.

At the end of this chapter, you should be able to:

- Identify the basic elements of an Internetwork Security Architecture (ISA),
- Tailor any combination of network security services,
- Evolve your own ISA as per requirements of a situation.
- Analyse the correctness of the ISA design approach,
- Tell about how to extend an existing ISA design without throwing away existing set-up; and
- Differentiate between various direct and indirect / hidden design constraints and their consequences.

The treatment assumes the working knowledge of Computer Networks and some exposure to Operating Systems and Data Communication areas.

8.1 Introduction

Traditionally, the Internet has been popular primarily due to its practically unrestricted access to the majority of its resources those are so conveniently linked, stored, searched and retrieved. It is this ease of access that is the cause of concern for the providers and users of the Internet-based services. Security considerations assume significance as the Internet comprises of an extremely large number of networks of individual networks most of which are managed by different organizations and agencies and use a variety of protocols, policies and technologies the problem of securing the Internet-based transactions as well as the problem of connecting private networks / intranets to the Internet adds another dimension of complexity. Internet Security Architectures (ISAs) attempt to provide acceptably secure (not absolutely secure) access to Internet resources based on the content-owner / site-owner / service-owner's configured choices as well as attempt to protect the average users of the Internet and Internet-based services by allowing a set of measurable degree of privacy and security.

It may be important to realize that there is no such thing as absolute security or one hundred percent security as this is simply not possible in any shared resource based system including the Internet. Therefore, when the experts suggest that a specific ISA is secure they only mean that this ISA offers a measurable degree of security that is possibly adequate in the context of a specific set of requirements of an operating environment.

Like all good things in life, Internet Security too comes with a price of its own --- the price to be paid in terms of need for increased processing resources, need for increased processing time and resultant delays in data-delivery. This added delay / latency may serve to deteriorate the performance of the time-sensitive services (due to compromised QoS) and may not be acceptable in certain cases. The very nature of best-effort delivery framework of the Internet Protocol (IP) does not make the problem any simpler! Naturally, as the security requirements become more stringent, the network / internetwork performance further droops and it becomes extremely difficult to strike an acceptable balance between the security needs, desired performance index and the targeted cost of service. Most of the ISAs, in effect, seek to do just that but often require careful use and design-level support for maximising the benefits as per the targets of the owner of service / content / infrastructure / user.

Internet Security Architecture (ICA), in general, and Cryptography, in particular, is expected to provide Confidentiality, Integrity, Authentication and Non-repudiation mechanisms. While the first of these is the primary expectation from a Cryptographic solution, often the real-life applications dictate that the ISAs provide a set of cryptographic measures that provide one or more of the last three features. A good example of a broad class in which a very large number of applications warrant all the provision for all of the four features is the Internet Commerce (briefly explored in the Chapter-11).

8.2 Security Issues in the Intranets and the Internet

There exist some common issues pertaining to the security of any kind of internetwork. Certain security issues, however, apply uniquely to respective class of internetworks (for instance, security issues applicable to the Internet, Intranet and Extranet may not be necessarily the same).

Primary issues relevant to the internetwork security include choice of degree of security and privacy targeted, choice of proper encryption algorithm / encryption protocol, choice of proper key-length / key-type (public versus private key for instance!), choice of authentication algorithm / authentication protocol, choice of access control policy, choice of number of levels / tiers of access-control, choice of form of data-encapsulation etc.

Classically, the security problems related to all categories of internetworks can be classified in terms of some form of access violations or misuse (since no security breach / attack is possible without an access) and therefore all the internetwork security breaches can be categorised as follows:

- Intention-based classification
 - Unintentional access breaches
 - Intentional access breaches
- Origin-based classification
 - Internally originated access breaches
 - Externally originated access breaches
- Instantiation-based classification
 - Centralised access breaches
 - Distributed access breaches
- Portability-based classification
 - Platform-dependent access breaches
 - Platform-Independent access breaches
- Service-based classification
 - Access breach leading to blocking of the service
 - Access breach leading to overwhelming the service
 - Access breach leading to redirection of the service
 - Access breach leading to abuse of service
 - Access breach leading to modification of service
 - Access breach leading to termination of service
- Periodicity-based classification
 - Periodic (pattern-directed) access breaches
 - Aperiodic (random / one-time) access breaches
- Event-based classification
 - Event-driven access breaches
 - Event-independent access breaches
- Storage-based classification
 - Memory-based access breaches
 - Secondary Storage-based (Disk I/O included) access breaches
- Data-based classification
 - Access breaches leading to corruption of data (control / actual data)
 - Access breaches leading to leakage / unauthorised forwarding of data
 - Access breaches leading to unauthorised storage of data
- Damage-based classification
 - Control-data damage-based access breaches
 - User-data damage-based access breaches
 - Software damage-based access breaches
 - Firmware damage-based access breaches

- Hardware damage-based access breaches

The list is incomplete but broadly representative in nature and all possible attacks or access violations can be mapped onto one or more of these security violation classes. All of these classes have their corresponding security issues in terms of scope, economics, effectiveness, detection / tracking mechanism, security metric etc.

Two of the most common forms of Internetwork Security Architectures use the combination of appropriate encryption algorithms / protocols and selective control / data flow.

8.3 Encryption and Authentication-based Security Solutions

A text message in its original unencrypted form is called a plaintext or cleartext message. A message that has its real meaning encoded such that even if it becomes available to an unintended receiver the receiving node could not extract the hidden meaning is called encrypted message. If the encrypted message is in the text format, it is often called ciphertext.

Mathematically, encryption and decryption may be defined as follows:

If the Original (unencrypted) message is denoted by $M_{original}$, Encrypted version of the same message is denoted by $M_{encrypted}$, Decrypted version of this message is denoted by $M_{decrypted}$, Encryption Function is denoted by E and Decryption Function is denoted by D , then the entire process may be expressed as:

$$E(M_{original}) = M_{encrypted}$$

$$D(M_{encrypted}) = M_{original}$$

$$D(M_{encrypted}) = D(E(M_{original})) = M_{original}$$

The process of encoding a message such that its meaning could be securely hidden from the unauthorized or unintended recipients is called Encryption or Enciphering. The process of retrieval of the original plaintext message or the process of extraction of the encoded meaning from an encrypted message is called Decryption or Deciphering.

The real challenge here is in carrying out the entire process of encryption and decryption in a secure manner (often over interconnected topologies like networks and internetworks of all types and sizes). Cryptography is the name given to the area of knowledge that deals with the study and practice of such secure encryption processes whereas Cryptanalysis is the name given to the theory and practice of the process that involves decoding or decrypting an encrypted message.

The term Cryptology refers to the branch of theoretical computer science or formal mathematics that deals with the theoretical (essentially mathematical) aspects of both Cryptography and Cryptanalysis.

8.3.1 Classification of ISA Cryptographic Components

On the basis of algorithmic categories, applications, targets, schemes and protocol complexities, various cryptographic components of the Internet Security Architectures can be classified as shown below.

- Cryptographic Algorithms (Hardware / Software / Hybrid encryptions possible in most cases)
 - Asymmetric Cryptographic Algorithms / Public-Key Cryptographic Algorithms
 - Channel Encryption Algorithms
 - Link / Link-to-Link (L2L) Encryption Algorithms
 - End-to-End (E2E) Encryption Algorithms
 - L2L and E2E Combination Encryption Algorithms
 - Digital Signature Algorithms
 - Key-Agreement Algorithms
 - Message Authentication Algorithms
 - Symmetric Cryptographic Algorithms / Private-Key Cryptographic Algorithms
- Steganographic Algorithms (Hardware / Software / Hybrid encryptions possible in most cases)
- Internet Security Protocols (Hardware / Software / Hybrid implementations possible in most cases)
 - Level-1 Protocols: Simple / Preliminary
 - Private / Symmetric Key-Exchange-based Internet Security Protocols
 - Key-Distribution Centre-based Protocols
 - Encryption Protocols
 - Authentication Protocols
 - One-way Function-based Authentication Protocols
 - Two-way Function-based Authentication Protocols
 -
 - Public / Asymmetric Key-Exchange-based Internet Security Protocols
 - Rivest & Shamir's Interlock Protocol
 - Digital Signature-based Session-Key-Exchange Protocols

- Public-Key and Signed Message Unicast Protocols
 - Encryption Protocols
 - Authentication Protocols
 - Public-Key and Signed Message Broadcast Protocols
 - Encryption Protocols
 - Authentication Protocols
- Level-2 Protocols: Moderate / Intermediate
 - Timestamped variants of Level-1 Protocols (e.g. Surety Technologies' Digital Notary System <patented>)
 - Subliminal Communication Channel Protocols (e.g. Simmon's Subliminal Digital Signature Scheme)
 - Non-Repudiable / Undeniable Digital Signature Protocols (Trust-based and Plain variations)
 - Proxy (Digital) Signature Protocols (proxying without sharing signatories private key)
 - Group (Digital) Signature Protocols (Trust-based and simple variations)
 - Fail-Stop Digital Signature Protocols (e.g. Pfitzmann & Waidner's Fail-Stop Digital Signature Scheme)
 - Public-Key Fair Coin Flip Protocol
 - Poker Protocols (Anonymous Key Distribution-based and plain variations)
 - One-Way Accumulator Protocols
 - ANDOS Protocols (ANDOS stands for All-or-Nothing-Disclosure-Of-Secrets)
 - Key-Escrow Protocols (e.g. NSA's Escrowed Encryption Standard Scheme)
- Level-3 Protocols: Complex / Advanced
 - BZK (Basic Zero-Knowledge Proof) Protocols (Interactive and Non-Interactive variations)
 - PZKP (Parallel Zero-Knowledge Proof) Protocols
 - MDP (Minimum-Disclosure Proof) Protocols
 - Blind Signature Protocols
 - Identity-based Public-Key Cryptographic Protocols
 - Contract Signing Protocols (Normal and Simultaneous Signing variations with / without Arbitrator)
 - Simultaneous Oblivious Transfer and Signature Protocols (e.g. Digitally Certified Mailing Schemes)
- Level-4 Protocols: Impenetrable
 - Electronic Voting Protocols (EVPs)
 - Simple Electronic Voting Protocols

- Blind Signatures-based Electronic Voting Protocols
- Single-CTF Electronic Voting Protocols (CTF stands for Central Tabulation Facility)
- Twin-CTF Electronic Voting Protocols
- CTF-Less Electronic Voting Protocols
- Multi-Key Cipher Electronic Voting Protocols
- Receipt-Free Electronic Voting Protocols
- Secure Multi-Party Computation Protocols (SMPCPs)
 - Conditionally Secure Multi-Party Computation Protocols
 - Unconditionally Secure Multi-Party Computation Protocols
 - Secure Circuit Evaluation-based Multi-Party Computation Protocols
- Secure Anonymous Message Broadcast Protocols
- Digital Money Protocols / Digital Cash Protocols (e.g. the Digital Cash Protocols from the Dutch major: DigiCash)
 - One-time Digital Cash Protocols
 - Multiple-Use Digital Cash Protocols
 - E-Coin Protocols (Fixed amount)
 - E-Cheque Protocols (Variable amount)
 - Anonymous Multi-Party-Supported Credit Card Protocols
 - Anonymous Multi-Party-Supported Bank Transfer Protocols

It is increasingly common to find the combination of Encoding, Encryption and Data / Storage Compression techniques used in ISAs. Often the order is the opposite; i.e. compression of data in order to reduce the redundancies is followed by encryption of the resultant compressed data and finally encoding of this compressed and encrypted data for subsequent transmission for providing optional features like error detection / correction / recovery.

8.3.2 Examples of Select Applications based on the Layer-based Classification of ISAs

As explained earlier, the ISA may involve one or more layer of internetwork protocols. Some examples given below demonstrate that in practice several approaches have evolved and survived the Litmus Test. The most-effective solution, as usual, remains a multi-tier ISA employing two or more layers (often three layers in the TCP/IP scenario).

- The IPSec Architecture (a Network Layer ISA-based Solution)

- The IPv6 ESP (Extension) Header Architecture (a Network Layer ISA-based Solution)
- The SSL Architecture (a Transport Layer ISA-based Solution)
- The SHTTP Architecture (an Application Layer ISA-based Solution)

As shown in the Chapter-11, in real life, several purpose-specific ISAs have been employed some of which are explicitly designed for secure transactions over the Internet. The well-known Secure Electronic Transaction (SET) developed jointly by two major Credit Card businesses, VISA and Master Card, is an example in case.

8.3.2 A Brief Note on Commonly Deployed Authentication-based Solutions

In the context of the earlier classifications, some of the commonly debated authentication-based ISAs (simple to moderate) can be listed as follows:

- IPv6 Authentication (Extension) Header Architecture
- Kerberos-based Authentication Architectures
- The Challenge-Response Architectures
- Mobility-specific Authentication Architecture
- Multi-Level Ticket-driven Authentication Architectures

This is not an ordered list nor is it exhaustive. It is, however, interesting to note here that despite better mechanisms being available, in a sizeable number of real-life applications, relatively primitive authentication architectures are commonly used. Some start-up and even a few well-known Internet Commerce sites are no exceptions to this phenomenon.

8.4 Firewall Architectures

Typically, a Firewall is defined as an integral component of the Internetwork Security System that is designed to safeguard the parameterised interests of a managed network / internetwork (owners, users all included) in a policy / configuration-driven manner.

Classically, Firewalls can be used for one or more of the following functions:

- Security policy implementation
- Choice of policy enforcement points
- Traffic segmentation
- Traffic Isolation

However, Firewalls cannot be used for detecting the internal intrusion originating from the trusted users operating at trusted network nodes and certain unforeseen penetration (virus / worm etc).

Firewalls themselves, by definition, should be non-corruptible and virtually impenetrable in character. The security cover provided by a Firewall requires that the system to be protected from external threats as well as from internal leakage should strictly route all incoming as well as outgoing traffic through the Firewall and the policy enforcement points be defined unambiguously.

Firewall components include the Application Gateways and Packet Filtering Routers amongst others. Firewalls are typically located between the untrusted network / internetwork and the designated trusted network / internetwork. Based on the location and functioning of a Firewall, it may be classified as belonging to one of the following classes belonging to what is often known as the Location-based Firewall Classification Scheme (LBFCS):

- Intranet Firewalls
- Sub-Intranet Firewalls
- Internet Firewalls
- Extranet Firewalls

It may be instructive to note here that firewalls form only a part of a complete Internet Security Architecture and as such they should not be expected to deliver a complete impenetrable security framework in isolation.

8.5 The ISA Design Goals and Issues

Any Internetwork Security Architecture has the single goal of providing a *certain degree of guaranteed protection* from one or more classes of security threats. The term 'a certain degree' has been used here for underlining the very fact that no real-life ISA can ever prove that it is absolutely secure / impenetrable. The Total Network Security is more of a myth than reality. Therefore, in practice, the researchers and implementers often try to evolve / deploy an ISA solution that is believed to be *reasonably secure* in the sense that for all practical purposes the chosen solution shall provide an acceptably good quality of protection from external threats and penetrations while retaining the security overhead to the lowest possible level.

Primary design issues specific to the ISAs, therefore, are concerned with the various trade-offs that a designer may have to arrive at in the process of attaining this design goal. The list of issues is long and includes issues like levels of protection, local security policy templates, choice of graded security threats, performance degradation monitoring, cost factor, delay-factor and location-dependence amongst others.

8.6 Summary

Primary issues relevant to the internetwork security include choice of degree of security and privacy targeted, choice of proper encryption algorithm / encryption protocol, choice of proper key-length / key-type (public versus private key for instance!), choice of authentication algorithm / authentication protocol, choice of access control policy, choice of number of levels / tiers of access-control, choice of form of data-encapsulation etc.

Unintentional access breaches, Intentional access breaches, Internally originated access breaches, Externally originated access breaches, Centralised access breaches, Distributed access breaches, Platform-dependent access breaches, Platform-Independent access breaches, Access breach leading to overwhelming the service, Access breach leading to abuse of service, Access breach leading to modification of service, Access breach leading to termination of service, Periodic (pattern-directed) access breaches, Aperiodic (random / one-time) access breaches, Event-driven access breaches, Event-independent access breaches, Memory-based access breaches, Access breaches leading to unauthorised storage of data, Control-data damage-based access breaches, User-data damage-based access breaches, Software damage-based access breaches, Firmware damage-based access breaches, Hardware damage-based access breaches etc. are various form of access-based security breaches.

Network Layer Security Solutions to these breaches involves schemes like: The IPSec Architecture, The IPv6 ESP (Extension) Header Architecture etc. whereas the Transport-Layer Security Solutions include the SSL Architecture. Application-Layer Security Solutions include the SHTTP Architecture. An example of a Hybrid Security Solution is the SET Architecture advanced by Master Card and Visa.

Firewall can be of several types including: Intranet Firewalls, Internet Firewalls, Extranet Firewalls and their location and purpose decides choice of their design parameters.

8.6 Recommended Readings

1. Bruce Schneier: **Applied Cryptography**, Second Edition, John Wiley & Sons, New York, 2001.
2. C. Kaufman, R. Perlman and M. Spenser: **Network Security**, Second Edition, Prentice-Hall, Englewood Cliffs, 2002.
3. D. Chapman and E. Zwicky: **Building Internet Firewalls**, O'Reilly, Sebastpool, 1995.
- 4.D. Maughan, M. Schertler, M. Schneider and J. Turner, **Internet Security Association and Key Management Protocol (ISAKMP)**, RFC 2408, November 1998.
- 5.D. Piper: **The Internet IP Security Domain of Interpretation for ISAKMP** RFC 2407, November 1998.

6. Grady N. Drew: **Using SET for Electronic Commerce**, Prentice-Hall PTR, 1998.
7. H. Orman: **The OAKLEY Key Determination Protocol**, RFC 2412, November 1998.
8. Harkins and D. Carrel: **The Internet Key Exchange (IKE)**, RFC 2409, November 1998.
9. ISO/IEC 9594-2, **Information Technology - Open Systems Interconnection - The Directory: Models**, CCITT/ITU Recommendation X.501, 1993.
10. ISO/IEC 9594-8, **Information Technology - Open Systems Interconnection - The Directory: Authentication Framework**, CCITT/ITU Recommendation X.509, 1993.
11. K. Bajaj & D. Nag: **E-Commerce: The Cutting Edge of Business**, Tata McGraw-Hill, New Delhi, 1999.
12. M. Pistoia, D. F. Reller, D. Gupta, M. Nagnur and A. Ramani: **Java2 Network Security**, Second Edition, Person Education, New Delhi, 1999.
13. Madson and N. Doraswamy: **The ESP DES-CBC Cipher Algorithm With Explicit IV**, RFC 2405, November 1998.
14. Madson and R. Glenn: **The Use of HMAC-MD5 within ESP and AH**, RFC 2403, November 1998.
15. Madson and R. Glenn: **The Use of HMAC-SHA-1-96 within ESP and AH**, RFC 2404, November 1998.
16. Mani Subramanian: **Network Management: Principles and Practice**, Pearson Education, New Delhi, 2000.
17. Microsoft Research: **Microsoft's Guideon Firewall Design**, available at the URL: <http://www.microsoft.com/technet/treeview/default.asp?url=/technet/itsolutions/idc/rag/ragc03.asp>.
18. R. Friend and R. Monsour: **IP Payload Compression Using LZS**, RFC 2395, August 1998.
19. R. Glenn and S. Kent: **The NULL Encryption Algorithm and Its Use With IPsec**, RFC 2410, November 1998.
20. R. Pereira and R. Adams: **The ESP CBC-Mode Cipher Algorithms**, RFC 2451, November 1998.
21. R. Pereira: **IP Payload Compression Using DEFLATE**, RFC 2394, August 1998.

22. R. Thayer and N. R. Glenn: **Request for Comments: 2411, IP Security Document Roadmap, November 1998.**
23. S. Bellovin and W. Chesvick: **Internet Security and Firewalls**, Second Edition, Addison-Wesley, Reading, 1998.
24. S. Kent and R. Atkinson: **IP Authentication Header**, RFC 2402, November 1998.
25. S. Kent and R. Atkinson: **IP Encapsulating Security Payload (ESP)**, RFC 2406, November 1998.
26. S. Kent and R. Atkinson: **Security Architecture for the Internet Protocol**, RFC 2401, November 1998.
27. Shacham, R. Monsour, R. Pereira and M. Thomas: **IP Payload Compression Protocol (IPComp)**, RFC 2393, August 1998.
28. Uyless Black: **Internet Security Protocols: Protecting IP traffic**, Pearson Education, New Delhi, 2000.
29. William Stallings: **Cryptography and Network Security**, Second Edition, Prentice-Hall, Upper Saddle River, 1999.

8.7 Exercises

1. What are the various types of Access Violations that may lead to the possible attacks, security breaches or information corruption over an Internetwork?
2. Why can't the Firewalls necessarily prevent all kinds of attacks on the Internetwork?
3. Compare various Internet Security Architectures in terms of acceptance-rate in the industry, respective strengths and weaknesses, performance overheads and consistency.

Chapter-9

Internetwork-based Video-on-Demand Architectures

Interaction Goals

Objectives of this chapter include providing a brief introduction to the Video-on-Demand system architecture and related design issues with the help of some established practices and brief case studies.

At the end of this chapter, you should be able to:

1. Find the common elements of the ADDL and Desktop Video-on-Demand systems,
2. Identify the basic elements of a VoD system,
3. Tailor any combination of on-demand service sets over the internetwork in a way to balance the load; and,
4. Evolve your own VoD architecture as per requirements of a situation.

The treatment presupposes a background in network programming and operating systems in addition to some exposure to continuous media fundamentals.

9.1 Introduction

Basically, *Video-on-Demand (VoD)* may be defined as a *service* that could provide the 'required' Video Data, at an acceptable transmission rate, of an acceptable quality and at an affordable price as per the explicit demand / requirement / need of the client.

Incidentally, at the present state of technology, these basic requirements are best met by the MMI Technology (though it is certainly not the only technology capable of offering the VoD services). Here, we shall focus only on the MMI-based VoD technologies.

9.2 Types of Video-on-Demand Technologies

Basically, there do exist two classes of Video-on-Demand systems:

- Full *Video-on-Demand* Technologies (Unicasting-based)
- *Near-Video-on-Demand* Technologies (Multicasting and Unicasting combinations with the former having greater share)

9.3 The Video-on-Demand System

In view of the current state of technology, the VoD System may be defined as an Interactive Multimedia System that allows one or more clients to choose and view any movie (or a video clip of it) out of a large database of movies.

VoD's primary advantage lies in the possibility of providing video services related to different video data to different clients simultaneously.

Constantly drooping prices of multimedia PCs, digital storage devices and associated hardware in conjunction with readily available user-friendly software have made the MMI-based VoD System a technology of choice for the majority of people.

9.4 The VoD Architecture

Primary technique used in MMI-based VoD Systems is the Streaming Technique. Video Streaming as well as Audio Streaming technologies coexist. Unlike the traditional bursty nature of the normal network / internetwork traffic, the Video Streaming permits continuous (non-bursty) video-traffic and thereby provides better performance than the traditional non-streaming variants.

Unicasting capability holds key to the VoD; however, it is complemented by Anycasting and Multicasting / Broadcasting capabilities of the associated services. IPv6 is likely to give this technology a further boost to this technology due its support for faster processing, flow-specifications and associated flexibilities.

A Simple VoD Architecture

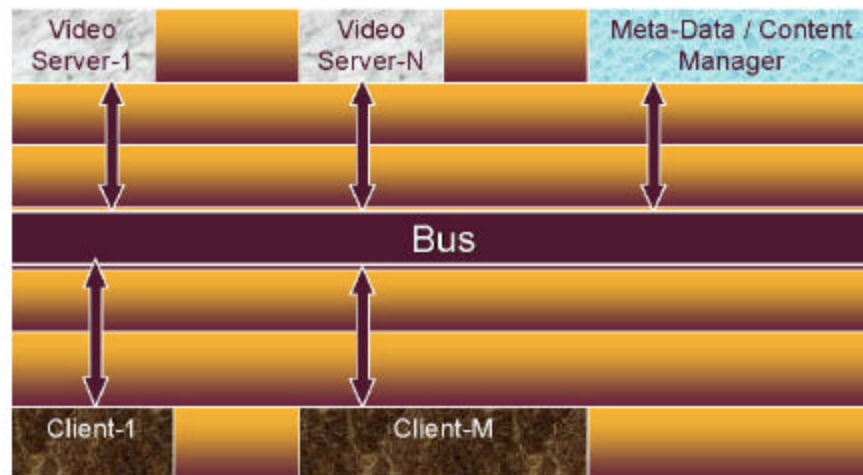


Fig. 9.1: A simple VoD Architecture

9.5 Basic Issues in VoD Design

There exist twelve basic issues related to the design of a good Video-on-Demand (over the Net) System. These are:

- Extent and type target Interoperability:
- Interoperability with respect to content form
- Interoperability with respect to presentation format
- Interoperability with respect to software platform / support
- Interoperability with respect to internetwork communication technologies
- Choice of a Meta Data management strategy
- Choice of a Location Transparency mechanism
- Choice of Acquisition / Organization / Manipulation mechanisms
- Choice of User-Level Service-set vis-à-vis the User Authorization and Access status
- Choice of Authentication, Security and Protection policies and mechanisms
- Choice of Resource Location and Management mechanism

- Constitution of a VoD Content Management System

9.6 Constituents of a VoD System

A Video-on-Demand System may be seen as comprising of at least five major constituent components. These are:

1. Resource Manager
2. Meta-Data Manager
3. Application Enabler
4. A Multi-Agent System comprising of many Agents including a set of Service Agents and a Monitoring Agent
5. Security Manger

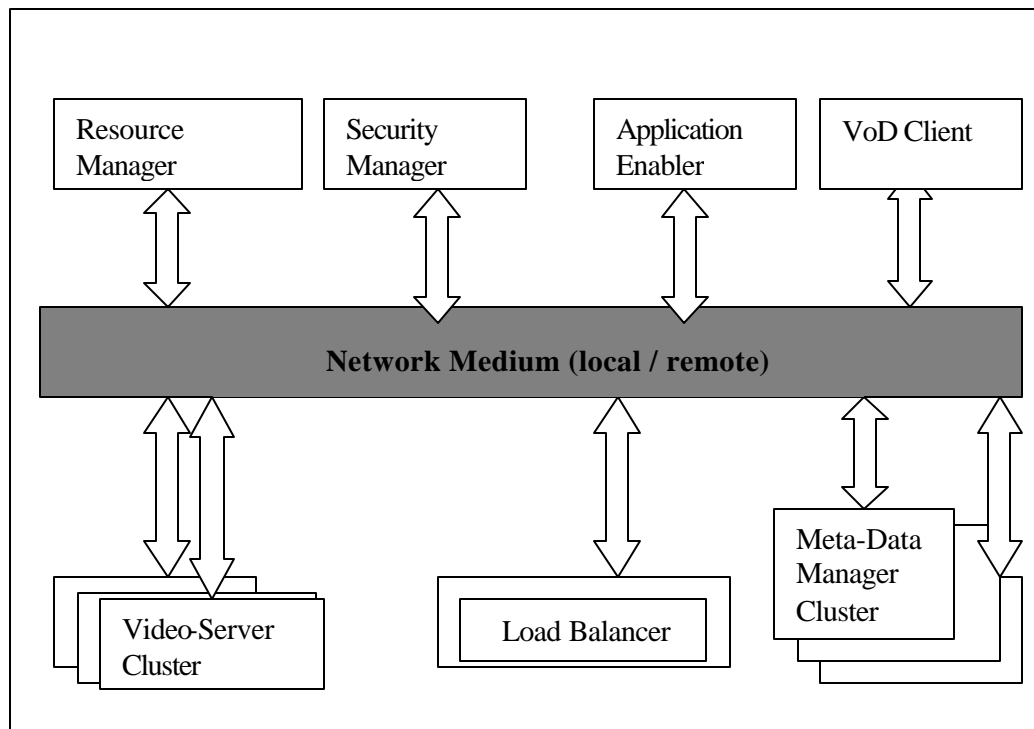


Fig. 9.2: Constituents of the Improved VoD Architecture

Resource Manager may further be seen as comprising of many sub-modules including:

- *Authentication Module* (overlaps the functionality provided by the first-level Security Manager)
- *Network / Internetwork Directory Service Module*
- *File System Module*
- *Database System Module*
- *Streaming Service Module*

Meta-data Manager is primarily concerned about keeping track of data on *content, load, location* and *control matters*; and, provides a *transparent interface* to the other modules including the Client Module.

Role of the Application Enabler is the same as in conventional enablers while the MAS has exactly the role that has been discussed in the Chapter-4. Security Manger is primarily responsible for a multi-level Authentication, Authorization and Access Control framework that could allow a reasonable degree of customizability at the administrator's end.

9.7 Internetworking Aspects of Video-on-Demand Technology

As may be evident from the earlier discussion, several aspects of the network-based on-demand technologies have aspects that assume more prominence while the delivery is targeted over the Wide-Area Internetworks. The following aspects and factors influence design and performance requirements and constraints of a VoD System.

- Media Server Design Constraints
- Server Cluster Design Constraints
- Server Location / Content Location Scheme
- Bandwidth Optimization Goals
- Content Synchronization Strategy
- Content Distribution Strategy
- Load Balancing Mechanism
- Auditing Strategy
- Interoperability Constraints
- Video-Database Design
- Ranges of Allowable Quality-dependent / Media-dependent Compression Ratio
- Client Design Constraints
- QoS Assurance
- Privacy and Security Requirements

Some of the protocols relevant to the VoD over the Internetworks include Internet Protocol versions 4 and 6, RSVP: Resource ReserVation Protocol, TCP: Transmission Control Protocol, UDP: User Datagram Protocol, RTSP: Real-Time Streaming Protocol, RTCP: Real-Time Control Protocol, RTP: Real-time Transport Protocol, HTTP: Hyper Text Transfer Protocol. In certain cases, for mobile devices, protocols have been proposed for video (lower resolution video) delivery over a limited area. Work is going on at the IETF and elsewhere to extend this functionality to the longer distances without disrupting the normal mobile traffic. Similarly, a lot of good work is going on around the world to solve the quality of service issues posed by the on-demand services like the VoD. BITS-Pilani maintains a huge searchable repository of the relevant documents produced worldwide in the direction of solving the longstanding QoS problems. This site is accessible at the URL: <http://www.bits-pilani.ac.in/~ngni/>.

9.8 Case Study of the Cisco's IP/TV Solution

Cisco Corporation has developed a cost-effective IP based VoD technology under the trade name of Cisco IP/TV. The following diagram broadly explains the IP/TV architecture over a local LAN. The solution, however, is not limited to the LAN technology alone. It, in fact, has a very good ready applicability in the areas like Educational Content Distribution, Limited VoD and Interactive Distance Education etc.

Cisco's IP/TV is primarily a Video-on-Demand technology for the IP-based internetworks. It is easy to be integrated with the common web-based technologies. One major advantage of the IP/TV is its scalable distributed architecture that offers acceptably good bandwidth-utilization factor through the use of the IP Multicasting for scheduled broadcasts. It is compatible with many popular / common networking standards. It can be used with a variety of communication media, Codecs and packet / cell switching systems. It allows Remote Server Administration.

There exist three basic functional elements / building-blocks in the IP/TV:

1. IP/TV Server
2. IP/TV Content Manager
3. IP/TV Viewer

These three components provide video storage / transmission, video-content management / directory-response and video-content display / client-request-initiation services respectively.

IP/TV Server is responsible for storage, retrieval and transmittal of the video data / software / programs. It forms the basis of the good quality Audio/Video Synchronization by integrating three functionalities of the live audio/video-capture, data encoder and file server within itself. Currently, it runs on only Microsoft Windows 95, 98, 2000 and NT Workstation based platforms. It has the capability to capture and transmit live video data from the connected Video Cameras. In addition, it can capture data from DVDs, VCRs and several other devices and media including the Satellite Microwave media and 75-Ohm CATV Cables.

IP/TV Content Manager is a multi-module system that decides the protocol / specifications to be followed by the IP/TV Server, provides the solicited content-specific service-location information to the IP/TV Viewer, provides content management services, helps in load-balancing and optimal utilization of the available bandwidth, optional security services and content-distribution services (to Servers). Currently, it requires Microsoft NT based platform (requires certain hardware support for acceptable performance). It has been developed in Java and therefore may not be too difficult to be ported to other OS based platforms if situation so requires.

IP/TV Viewer, as mentioned earlier, is primarily a client software that provides an intuitive User Interface, negotiates with the Content Manager and Server as per requirements, displays a structured 'availability list' (of videos), initiates a video-content request on behalf of the user, permits the user to choose between live broadcast, scheduled broadcast and video-on-demand services and displays the requested video on screen. Currently, it runs on Microsoft Windows 95, 98, 2000 and NT Workstation platforms and requires some hardware support for an acceptable performance.

Fig. 9.3 shows the Cisco's IP/TV Architecture. This architecture has proved to be generally scalable and exhibits acceptable performance over homogenous internetworks.

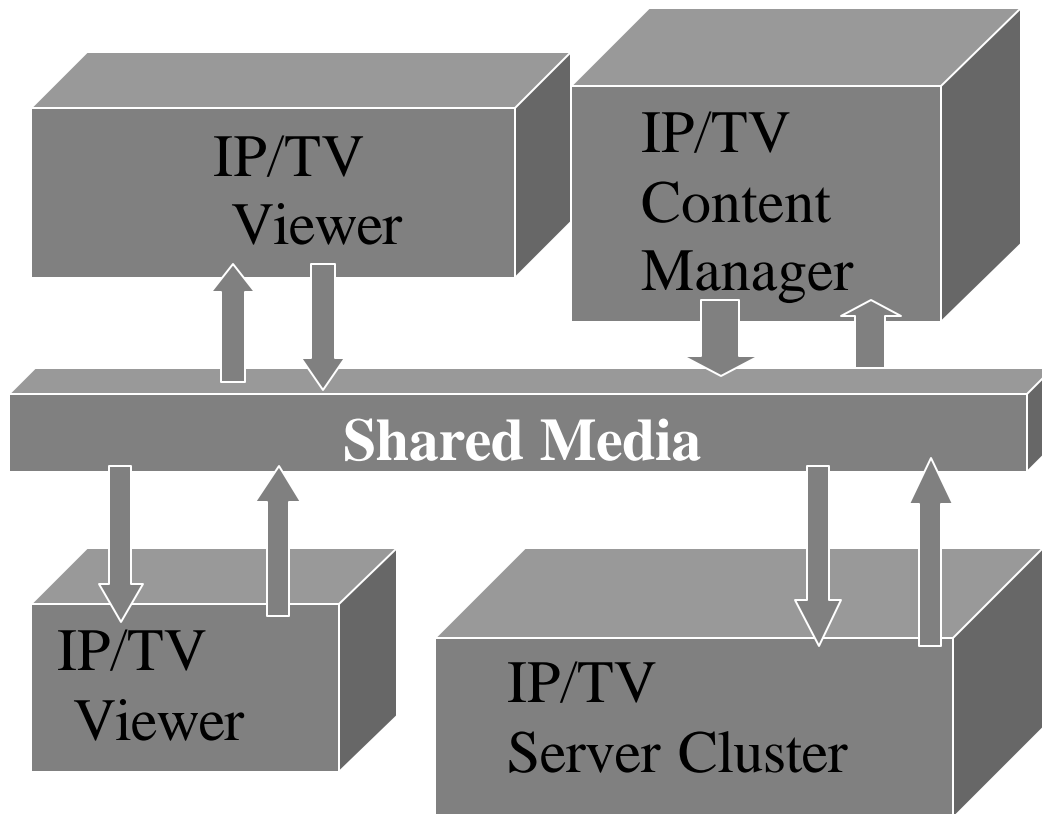


Fig. 9.3: The CISCO IP/ TV Architecture

9.9 Case Study of the Ichcha-Drishti: Case Study of the World's First Native IPv6-capable VoD System (VoDv6)

Through its acclaimed "Project IPv6@BITS", a group of researchers involving students and faculty at the Birla Institute of Technology & Science, Pilani have successfully built and demonstrated one of the world's first Desktop Video-on-Demand system for native IPv6-based Internetworks. A detailed case study of this project along with entire design is available in public domain at the project website that can be reached at the URL: <http://www.bits-pilani.ac.in/~rahul/>.

The following components have been identified as the three main sub-systems of the Video on Demand System.

- Server Side (Control Server, Video Server)
- Intermediate Components (Load Balancing Tool, Meta Data Manager, Request Evaluator)
- Client (Player, Decoder)
- Transport Protocols.

The Server side, in order to provide reliability is sub-divided into server components that lead to redundancy in the system. In order to arbitrate between the various server components, the Load Balancing Tool (LBT) is used that determines the least loaded component and passes on any incoming requests to the least loaded server.

"Meta Data" essentially means Data about Data. Hence the Meta Data Manager (MDM) maintains the Database of all the presentations served by the various Video

Servers. So once a particular presentation is updated in the Meta Data Manager, the Meta Data Manager updates this information with its current Database. The client side is concerned with requesting, receiving, decoding and displaying of media presentation files. The Transport Protocols used for the System include RTSP (Real Time Streaming Protocol), RTP/RTCP (Real Time Protocol/Real Time Control Protocol) and RSVP (Resource Reservation Protocol).

The control server provides for administrative capabilities to the entire system. It acts as a single point of entry to the entire system. The Video Servers are assumed to be in a distributed cluster with selective mirroring at various places and it is the control server acts as a single point of entry to the entire system. Only the control servers do any change that is to be made to the video servers. Only the control server Administrator can perform any addition, deletion or update of the contents on servers in the video-server cluster. All in all the control server has the following primary functions:

- Support for administration and remote access facilities,
- Providing Control and Data Access Channel between the Control server and the Administration modules; and,
- Providing the Control and Data Access Channel between the Control and the Video Servers.

The video Server initially registers with the Control Server and the Load balancing Tool. It also guarantees continuous playback of a stored video to the client.

9.10 Summary

VoD may be defined as a service that could provide the 'required' Video Data, at an acceptable transmission rate, of an acceptable quality and at an affordable price as per the explicit demand / requirement / need of the client. Incidentally, at the present state of technology, these basic requirements are best met by the MMI Technology (though it is certainly not the only technology capable of offering the VoD services). The *MMI-based Service-on-on-Demand* technologies may include Video-on-Demand Technologies, Audio-on-Demand Technologies, FAX-on-Demand Technologies, Game-on-Demand Technologies, News-on-Demand Technologies, Education-on-Demand Technologies and Finance -on-Demand Technologies. Although there exist numerous commercial as well as experimental VoD solutions, the IP/TV Technology Solution offered by Cisco is by far one of the best and the most flexible ready-to-use VoD solution that offers a choice of VoD, Live Broadcast and Scheduled Broadcast services.

9.11 Recommended Readings

1. Cisco Corporation: IOS Enabling Services: IP/TV Q&A, available at <http://www.cisco.com/>
2. Judy Estrin and Stephen Casner: Multimedia over IP: Specs Show the Way, Percept Software, August 1996.
3. Rahul Banerjee: Architecture of the BITS-MOS: The BITS Multimedia Operating System, available at the URL: <http://www.bits-pilani.ac.in/~rahul/papers/BITS-MOS/index.html/>
4. Rahul Banerjee: Design of an Innovative Video-on-Demand System, available at the URL: <http://www.bits-pilani.ac.in/~rahul/papers/BITS-VoDv6/index.html/>
5. RFC 1112 (IP Multicasting)
6. RFC 1889 (RTP)

7. RFC 1890 (Audio and Video over RTP)

9.12 Exercises

1. Design and implement a simple VoD system for your intranet using off-the-shelf hardware and custom-built software. The software Codecs may be preferred and common audio/video formats like Apple QuickTime, Real Video, MPEG etc. should be readily supported. Also, estimate its implementation as well as running cost.
2. Compare the Cisco IP/TV technology with any other leading-edge commercial-strength VoD over IP technology in terms of the following:
 - Scalability
 - QoS
 - Availability
 - Compression Strategy
 - Channel Utilization Efficiency
 - Ease of Use
 - Ease of Administration
 - Compatibility with respect to popular video formats
 - Involved Protocols
 - Price-Performance Ratio.
3. Why is there a basic difference between the designs of the Video-on-Demand and Near-VoD (desktop in both cases and with IPv6 support)? Also explain your viewpoint with the help of the relative architectural differences between the two applications.
4. There exist two basic types of Video-on-Demand Systems for network-oriented deliveries. These are primarily classified on the basis of one of the numerous parameters of the QoS. Similarly, there do exist many other individual factors that affect the design of any IP-based VoD System. Identify any such five components and briefly discuss them
5. Consider the Cisco IP/TV architecture studied by you. If this architecture is to be ported to the IPv6-only WAIs, then shall it require any changes? If yes, exactly what changes would you suggest to be made and why? If no changes are to be made, then please explain exactly
6. Compare the Cisco's IP/TV Architecture with the BITS Ichhadrishti VoD Architecture. What are the respective strengths and weaknesses of each of these?
7. Cisco's IP/TV solution works well for TV-quality video to be delivered over the enterprise intranets using IP as their basic network-layer protocol. It can combine streaming video with the application management features. If your university decides to conduct a multi-location training programme for a laboratory course in such a way that the instructor chooses to remain in his / her chamber while the students chose to be anywhere in the institute, can this solution be still used economically? If yes, please explain why and how shall this be possible? Assume that the Webcams are available in all places mentioned above and the PCs are MPC-II compliant.
8. Various architectures based on ATM, TCP/IP etc. are used for Video-on-Demand services. Which of these technologies has the potential for being a cost-effective Internetwork-based Distance Learning Technology given the ground realities of the country of your residence and why? You may project it for five years from now. Please write your assumptions clearly

Chapter-10 Internetwork-based Digital Library Architectures

Interaction Goals

Objectives of this chapter are to introduce internetwork-based Digital Library Service-on-Demand Architectures, discuss their basic constituents, learn about the challenges they offer, realize the relevant design problems posed by them, understand relevant design concepts and appreciate the wide spectrum of applications they may be closely associated with.

At the end of this chapter, you should be able to:

- Identify the basic elements of a Digital Library system,
- Identify common elements of the ADDL and CDDL systems,
- Tailor any combination of Digital Library service sets over the internetwork in a way to balance the load,
- Evolve your own Digital Library architecture as per requirements of a situation.
- Analyze the correctness of the internetwork design approach,
- Tell about how to extend an existing design without throwing away existing setup; and
- Distinguish between various forms of such libraries over the Internet.

The treatment, like the related treatment in Chapter 9, presupposes the working knowledge of Computer Networks and some exposure to Operating Systems and Data Communication areas.

10.1 Introduction

The term Digital Library is often interpreted in a variety of ways. One possible definition states: "A Digital Library" is a set of diverse information structures, normally built within / atop an internetwork, which provide access on demand to one or more authorized category of entities (people / systems) and wherein most of the information is encoded in multimedia formats.

Interestingly, not all Digital Libraries fit into this definition, at least as on today. For instance, there do exist a few Digital Library architectures that do provide, by design, only asynchronous services primarily due to the reasons of economics involved. In fact, some experts feel that the term Digital Library is, probably, a misnomer.

A Digital Library is also visualized as a coordinated cluster of object-oriented user-level / application-level services all / some of which may be owned / controlled by one or more agencies.

In a way, a Digital Library can be seen as a technology solution involving Digital Computing Machinery, Multi-Level Storage Mechanism, Distributed Multi-Media Content and associated Software required to manipulate, extract, interact, emulate, retrieve, collect, catalogue, customize or find any part of this content and selectively add / modify any relevant content (if appropriately authorized).

Unlike the popular misconception, a Digital Library is not just another data warehouse comprising of metadata.

10.2 Classification of Digital Library Architectures

There may exist several categories of Digital Libraries including the following:

- Academic Document Digital Library
 - Asynchronous Media Distribution class of ADDL
 - Synchronous Media Distribution class of ADDL
 - Hybrid ADDL
- Corporate Document Digital Library
 - Asynchronous Media Distribution class of CDDL
 - Synchronous Media Distribution class of CDDL
 - Hybrid CDDL
- Virtual Digital Library

The term 'Document' has been used here in the widest possible sense including all types of storable, retrievable and distributable media objects.

10.3 Major Digital Library Architectures

All over the world, various government agencies have been busy formulating their National Digital Library Initiatives. The US led the move by establishing an early Digital Library Initiative (DLI) handled as a major component of its national information infrastructure and setting the pace by designating the National Science Foundation as the primary funding agency for this purpose. The NSF invited the research proposals from the leading universities and out of an overwhelming seventy-plus such proposals, six proposals were granted sizeable research grants. UCB, UIUC, UCSB, UC, SU and UM were the early leader thus! Germany, UK, China, Japan, India, Netherlands and several other countries soon followed the suit, mostly having their own priorities and model architectures. Except for the linguistic (translation-related issues included!), content-access and controllability issues that were generally different in almost each of these cases, all other major issues were common to all these initiatives. Interestingly, a few commercial organizations like the IBM Corporation and the Sun Microsystems had also realized the need of the hour and came up with their own characteristic solutions. In most of the cases, the participants in the various national initiatives have chosen to focus on select research areas, though in each case other associated technologies are also being explored. Another common aspect of each of these project-sites is the effort taken / being taken by the respective site towards establishing a large-scale / medium-large-scale test-bed for their pilot tests and debugging purposes. Majority of the Asian, Australian and European enterprises, except those in a few countries like Germany and Japan have been relatively unorganized efforts in comparison to those in the USA.

Some of the major efforts / architectures / implementations, which have made a major contribution towards the Digital Library Technology include the following:

- The ACM Digital Library Architecture-- developed by the ACM (USA).

- The Berkeley Digital Library Architecture -- being developed at the University of California, Berkeley (USA), initially received boost from the SunSITE architecture.
- The BITS DigiLib Architecture – a Digital Library being developed at the BITS, Pilani (India).
- The CMU Digital Library Architecture – being developed at the Carnegie-Mellon University, Pittsburgh (USA).
- The German MeDoc Architecture – being developed at the FRG as the national initiative.
- The IBM DB2 Digital Library Architecture -- developed by the IBM Corporation (USA).
- The Stanford Digital Library Architecture -- developed at the Stanford University, Stanford (USA).
- The SunSITE Architecture -- developed by the Sun Microsystems (USA) and employed at over 56 sites all over the world.
- The UCSB Digital Library Architecture – being developed at the University of California, Santa Barbara (USA).
- The UIUC Digital Library Architecture – being developed at the University of Illinois, Urbana Champaign (USA).
- The University of Chicago Digital Library Architecture – being developed at the University of Chicago, Chicago (USA).
- The US Library of Congress Digital Library Architecture-- employed at the Library of Congress, Washington (USA).

10.4 Basic Issues in Digital Library Design: Internetworking Viewpoint

There do exist several factors that influence the design of a Digital Library design for a Wide Area Internetworking scenario. These factors often lead to a number of design issues including the following:

- Extent and type of Interoperability with respect to content form
- Extent and type of Interoperability with respect to presentation format
- Extent and type of Interoperability with respect to software platform / support
- Interoperability with respect to internetwork communication technologies
- Choice of a Meta Data management strategy
- Choice of a Location Transparency mechanism
- Choice of Acquisition / Organization / Manipulation mechanisms
- Choice of User-Level Service-set vis-à-vis the User Authorization and Access status
- Choice of Authentication, Security and Protection policies and mechanisms
- Choice of Resource Location and Management mechanism

10.5 Constitution of a Digital Library

A Digital Library may be seen as comprising of at least following major constituents:

- Digital Library Resource Manager
- Digital Metadata / Content Manager
- Application Enabler (optional)
- A Multi-Agent System comprising of many Agents including a set of Service Agents and a Monitoring Agent
- Digital Library Security Manger

A Digital Library Resource Manager comprises of several sub-systems including an Authentication Module (overlaps the functionality provided by the first-level Security Manager), a Network or an Internetwork Directory Service Module, a File System Module, a Database (or Support-support) Module and a Digital Library Service Module amongst others. At the heart of every Digital Library remains an excellent architecture for metadata collection, validation, organization and search support.

10.6 Internetworking Aspects of Digital Libraries: Multimedia Object Handling

Primarily, the Digital Library research concentrates upon Internetwork-based Distributed Multimedia Information Systems. Organization, Storage, Search, Retrieval, Customized Document Services, Intra-document Media-Object Search and Transparency across a variety of platforms are the primary research issues. Choice of Asynchronous versus Synchronous services forms a major research issue, in addition to those mentioned above. For quite sometime now, the need for a cost-effective and secure Object-Oriented Multimedia Database technology has been more or less felt by the SDDL and SCDDL architects but due to lack of any common agreement no universally acceptable standard has been possible to be evolved till date. This lack of a standard has created interoperability problems between synchronous segments of cooperating Digital Libraries. Certain Architecture Neutral Format-based proposals have been around of late and it appears to be a possible approach towards a more interoperable approach.

10.7 Case Study of the Stanford Digital Library Architecture

The Stanford University Digital Library Technologies revolve around an innovative DL information handling architecture called InfoBus. The term InfoBus is an abbreviation for the Information Bus. This architecture, currently, focuses on Asynchronous Information.

Primary content or holding of this Digital Library like that of any other library of its tribe is the Meta Data. Apart from the traditional services provided by the conventional libraries like content-acquisition, content-storage, content-retrieval, content-manipulation and content-organization etc. the InfoBus aims to provide certain value-added services like bibliography generation, citation analysis etc. Other services offered by it include Billing, Payment, Certification, Intellectual Property Rights monitoring, security and authentication.

It may be worth noting here that the InfoBus architecture is being used as a component of the Stanford Digital Library Project Test-bed; and is not a mature design as of this writing. However, it is significant because of its innovative

architectural design and lower service-cost. Fig. 11.1 depicts the principal components of this framework.

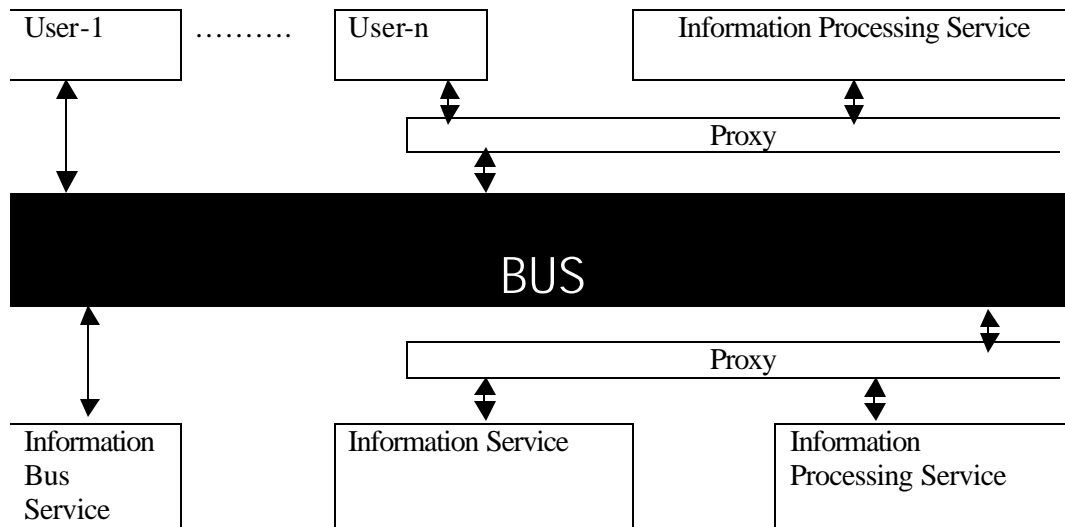


Fig. 11.1: The Stanford Digital Library Architecture

In principle, Content of the DL may be of any form: text, audio, video, graphics, image, and three-dimensional models.

The term DILITE stands for Digital Library Integrated Task Environment. It is primarily a user interface model-based architecture to support the Stanford Digital Library Architecture. In DILITE, Task-context based control is with the user and the interface model can accommodate user-services using a variety of service / communication speeds to be able to cater to the needs of a large number of users from varying backgrounds. In addition, it offers seamless service integration and exhibits support for sharing, reuse and persistence. In addition, Drag-and-Drop based object manipulation capabilities (using graphics-enabled compliant browsers / clients) and direct manipulation support through Windows-style GUIs are featured by DILITE. Yet another interesting aspect is the use of CORBA (Common Object Request Broker Architecture) implemented by Xerox Inter-Language Unification (ILU) System. The system is available in two versions: one developed using Java AWT for Microsoft Windows platforms and the other using Python and Tk for X-Windows platforms. Current version has been optimized for the Netscape Navigator, although the other popular browsers like Microsoft Internet Explorer is also supported. DILITE has a concept of DILITE Workcenters. Here, a workcenter is used to refer to a location wherein a set of pre-defined operations can be carried out with the aid of locally available tools. Each of these tools in a workcenter is called a Component in DILITE terminology. Examples of a few DILITE Components include Documents, Collections, Queries, InfoBus, and User-Representation etc. Since in this architecture, the Web-Browsers are remotely controllable, in terms of their behaviour, the DILITE makes use of this ability for marked display etc. A wealth of public-domain information on the internal design of this architecture is available at the project site.

10.8 Case Study of the CMU Digital Library Architecture

The CMU Digital Library architecture, like the Stanford initiative, is outcome of a DLI-supported (NSF funded) project. However, this architecture is, in its current state, more scalable but less function-rich than that of the Stanford except for its video-object support which is one of the best in its category. This architecture, due to its relative elegance and scalability is being considered a potential architecture for a media-rich distributed model based Digital Library Architecture.

As per the CMU's Informedia status statement, the core technology developed under Informedia-I supports speech, image and natural language processing and can be used to automatically transcribe, segment and index searchable video and image retrieval; whereas the Info rmedia-II aims at the dynamic extraction, summarization, visualization, and presentation of distributed video in addition to providing the abstraction service.

There is likelihood that a consortium of select government funded leading institutes of technology and management of India would soon adopt this architecture from the CMU for an ambitious Distance Learning initiative by the consortium. A detailed case study of this project is available at the CMU's Informedia Digital Video Library site ([http:// www.informedia.cs.cmu.edu/](http://www.informedia.cs.cmu.edu/)) as well as at the DLI site.

10.9 Case Study of the JournalServerSM Virtual Digital Library Architecture

About four years ago, when some Rhodes Scholars went to join the Oxford University in UK they carried with them the desire to see that the good research done in any part of the world carries its due respect not necessarily by getting published in the established research publications in the developed world but also if they get published in their less known counterparts in the developing countries. One of these scholars, Sheshadri Vasani, had an initial proposal of bringing some of the representative quality-journals in any field of knowledge to a more visible platform via the Internet such that these journals and their contents could have a greater reach and therefore greater readership ultimately resulting in a chain reaction that could bring more respect to the journal which in turn would attract greater number of good researchers for publishing their works / results in these. (The basic idea was that as a consequence, the respectable but less-known refereed journals of this category could come out of the current vicious cycle of lower visibility leading to lower respectability leading to continued lack of desire amongst the brightest local researchers in publishing in these journals.) Fortunately for Vasani, he could muster the support from some other like minded people then residing mostly in Oxford and London who took the baton further and discussed the idea with their contacts in academia and industry both of which showed certain degree of definite interest. This initial success led to the formation of the JournalServer Trust at Oxford that on a non-profit basis conceptualized the JournalServer Project. With the backing of the Oxford University's Bodleian Library and support from the OUCS, the Trust approached a few leading journal publishers, organizations and universities and soon had its advisory support coming from many continents from Asia and Africa to Europe and North America. In India, when Vasani approached his Alma Mater Birla Institute of Technology & Science at Pilani (better known as BITS, Pilani) for technical collaboration, BITS suggested that this objective could be best met if the collaboration-model was made to function like a Virtual Digital Library using a simple and low-cost architecture (low-cost was an essential requirement since the project aimed in 'charge-free' dissemination of the holdings of the VDL). The idea appealed

to the group and the JournalServer Virtual Digital Library project gradually took off. As of this writing, the project has spread its wings to UK, USA, India, Norway, Italy, Taiwan, Bangladesh and Pakistan and has four mirror sites at Oxford University, BITS-Pilani, Purdue University and National Science Council of Taiwan. The project home page is reachable at the URL: <http://www.journalserver.org/> and already over one hundred journals have committed their support to it.

Technically, the project is now in its second phase that is likely to be completed by the year-end, which is the time around which it proposes to go public. Metadata architecture has been developed and a security framework is under development. The project already has a set of end user tools (aimed at journal editors, publishers, researchers and the development team that is spread over UK, India and USA with the project director working from London, Webmaster from Houston and the bulk of the architecture and software development team working from Pilani.

Unlike all other ADDLs discussed in this chapter, this VDL shall offer only a subset of a true ADDL functionalities limited to Journals (catalogue, abstract, full-text search facilities already in place although currently limited to the HTML and PDF versions only). In order to take care of the synchronization needs of the meta-data sites as well as to ensure the respective security measures, the JS architecture has been planned as a push-pull-oriented content synchronization mechanism that proposes to use the power of the PKI for inter-domain exchanges. More details are available at the project site.

10.10 Summary

As per the IITA (apex body of the US-NII) report entitled "The Grand Challenge of Digital Libraries" the final or ultimate goal is the "deep semantic interoperability - the ability of a user to access, consistently and coherently, similar (though autonomously defined and managed) classes of digital objects and services, distributed across heterogeneous repositories, with federating or mediating software compensating for site-by-site variations. Achieving this will require breakthroughs in description as well as retrieval, object interchange and object retrieval protocols. Issues here include the definition and use of metadata and its capture or computation from objects (both textual and multimedia), the use of computed descriptions of objects, federation and integration of heterogeneous repositories with disparate semantics, clustering and automatic hierarchical organization of information, and algorithms for automatic rating, ranking, and evaluation of information quality, genre, and other properties." (Copyright: IITA) The term "federating", used here, means mapping various identical / similar objects from different locations in such a way that could give the illusion of a single organized collection of objects.

10.11 Recommended Readings

1. Bruce Schatz & H. Chen: **Building Large-Scale Digital Libraries**, IEEE Computer, May 1996, available at the URL: <http://computer.org/computer/dli/index.html>.
2. Bruce Schatz et al: **Building the Interspace**, 1996, available at the URL: <http://csl.ncsa.uiuc.edu/interspace.html>.
3. Bruce Schatz et al: **Federating Diverse Collections of Scientific Literature**, IEEE Computer, May 1996, pp. 28-36.

4. **Communications of the ACM, Special Issue on Digital Libraries**, April 1995.
5. DLI Staff: **Home page of the DLI National Synchronization Effort**, available at the URL: <http://www.grainger.uiuc.edu/dli/national.htm>.
6. DLI Staff: **The Digital Library Forum home page**, accessible at the URL: <http://www.dlib.org/>.
7. G. Taubes, "Indexing the Internet," *Science*, Sept. 8, 1995, pp. 1,354-1,356.
8. H. Chen, **Collaborative Systems: Solving the Vocabulary Problem**, *IEEE Computer*, May 1994, pp. 58-66.
9. **IEEE Internet Computing, Special Issue on Digital Libraries**, April 1998.
10. IITA Staff: **Interoperability, Scaling, and the Digital Library Research Agenda**, IITA report, 1995, available at the URL: <http://www-diglib.stanford.edu/diglib/pub/reports/iita-dlw/main.html>.
11. R. Pool, **Turning an Info-Glut into a Library**, *Science*, Oct. 7, 1994, pp. 20-22.
12. The US Digital Library Initiative: **Agency perspectives**, available at the URL: <http://computer.org/computer/dli/r50022/agencies.htm>.

10.12 Exercises

1. Suggest an architectural framework, keeping internetworking as well as economic aspects in focus, using which, in your opinion, any Corporate Digital Library should evolve. Please note that the Stanford or Berkeley model may not suit your company given your different financial and infrastructure requirements. Your scheme should permit private as well as collaborative business paradigms to co-exist in a cost-effective way. Please mention all your assumptions clearly before proposing your solution.
2. Study the JournalServer Virtual Digital Library Architecture and compare it with the usual ADDL Architectures in terms of features, functionalities and services.
3. Study the SunSITE Architecture and comment on its suitability as a full-fledged Digital Library Architecture in terms of the functionalities and services.
4. Study the IBM's DB2-based Digital Library Architecture for Synchronous Media and compare it with the attributes of any other SCDDL.
5. Examine the Stanford Digital Library Architecture and comment on its suitability of application to any large software company in terms of the following:
 - Content Location / Discovery
 - Content Classification
 - Storage
 - Retrieval
 - Content Distribution
 - Meta-data Architecture
 - Choice of Middleware
 - Choice of Scripting Languages
6. Suggest an architectural framework, keeping internetworking as well as economic aspects in focus, using which, in your opinion, a multi-campus university should evolve its Digital Library Architecture. Please note that the

Stanford InfoBus model may not necessarily suit us given our different financial and infrastructure requirements. Your scheme should permit on-campus, distance learning and collaborative learning paradigms to co-exist in a cost-effective way. Please note the University would be required to use this facility in a seamless manner. Please mention all your assumptions clearly before proposing your solution.

7. The Stanford Digital Library Architecture and the IBM DB2 Digital Library Architecture represent two possible digital library architectures of ADDL and CDDL type respectively. Focusing on the internetworking aspects of these two representative cases, it may be easily pointed out that high availability, interoperability, support for high compression ratio, load-balancing, synchronization and multiple-communication-speed-support are some of the major concerns in any quality multi-purpose digital library design. *If you were to design a DL for your organization, how would you address each of these internetworking concerns?* Support your answer with logic / computation.

Chapter-11

Internet Commerce Architectures

Interaction Goals

Objectives of this chapter include introduction to the basics of the Internet-based Commerce, its prevalent modes, associated architectures, protocols, services and role of security mechanisms involved.

At the end of this chapter, you should be able to:

- Find the common elements of the I-Commerce Systems,
- Identify the basic design issues related to an I-Com system,
- Differentiate between E-Commerce and I-Commerce as well as find their common elements,
- Recognize the trade-off between various I-Commerce technologies and choose the one most appropriate to your specific requirements,
- Tailor any combination of service sets over the internetwork in a way to ensure secure transactions over the Internet,
- Evolve your own I-Com-based business architecture as per requirements of a situation; and
- Analyze the correctness of the secure transaction system design,
- Tell about how to extend an existing design without throwing away existing setup.

The treatment assumes knowledge of simple cryptographic techniques and associated mathematics.

11.1 Introduction

Traditionally, the economics has been calling shots as far as majority of engineering design decisions are concerned. With the traditional commerce getting along the Internet-based cousin, this is once again a very clear indication. When commerce comes in, can there be privacy and security issues left behind? Obviously not! That's precisely why the Internet Commerce concentrates not only on the commerce aspects but also privacy and security issues related to the Net-based transactions. Authenticity of the signed / certified documents and their near-impenetrability are naturally basic concerns in such cases.

The primary idea behind the Internet commerce is the flexible yet secure way of customizable services and payments thereof in a verifiable and mutually beneficial manner to all the involved parties, without compromising on the privacy of the participating agencies.

Predecessors of the Internet Commerce or 'I-Commerce', as it is popularly known as, have been traditional and pre-Internet variant of Electronic Commerce (i.e. pre-Internet E-Commerce of the EDI and EFT over telephone company's voice-grade

and data networks respectively). Offshoots of the ICommerce frameworks include Mobile-Commerce (better known as M-Commerce) and the so-called D-Commerce.

11.2 Principal Objectives of Internet Commerce

Primary objectives of the Internet Commerce include greater market penetration, identification of new ways of doing business over the Internet, evolving an easy yet verifiable transaction tracking system, implementing secure fund and document transfer mechanisms to attract potential customers as well as secure organization's own interests and integrating existing relevant technologies for profit maximization at a low establishment cost.

It is important to note here that like any other business framework, existence of a sound revenue model is essential for the long-term success of any business based on ICommerce as well. In deed, it has its own set of potential risk factors and illusory peaks those need to be tackled with caution. It was neglect of this factor in the late 90's and early 2000 AD that lead to a few phenomenal successes followed by a virtual collapse of several Dot Com ventures leading to almost two years of global economic depression that gradually reflected in the slow-down of the non-I-Commerce segments as well.

11.3 Fundamental Components of Internet Commerce Frameworks

As of now, there exist the following principal instruments of the I-commerce System (mostly, over the Web) that can be used as a set of competing or even complementary (at times) components of scalable ICommerce Frameworks:

- Electronic Data Interchange (EDI)
- Electronic Funds Transfer (EFT)
- Secure Electronic Transactions (SET)
- Corporate Digital Library Systems (CDLS)
- Secure Electronic Messaging Systems (SEMS)

In the following sections, we would explore the EDI, EFT, SET and SEMS only since the CDLS basics have been already studied in the Chapter-10.

11.4 Electronic Data Interchange (EDI)

Electronic Data Interchange (EDI) is a standard syntax based encoded transmission scheme that permits unambiguous exchange of information of economic / strategic relevance between autonomous / independent collaborating agencies / partners (usually, over the Internet). It was originally proposed in the 1960s.

EDI is a technique and a convention of structured document interchange. In the Internet Commerce applications, the primary role of the EDI is to facilitate such exchanges in a controlled way decided by common agreement between collaborating business partners / agencies. These considerations include the identification of entities eligible for the document disclosure, decision about the authentication and authorization schemes and levels, decision about the timing of disclosure and definition of the nature and extent of interaction.

11.5 The EDI Architecture

The EDI architecture, like most contemporary architectures, is a layered architecture. As shown in the Fig. 11.1, apart from the Physical Layer, it has three other layers namely, EDI Transport Layer, EDI Standard Layer and EDI Semantic Layer.

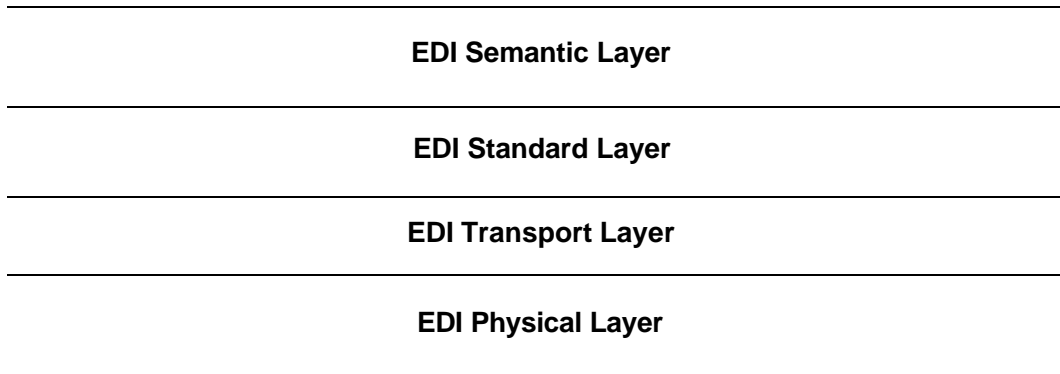


Fig. 11.1: The Layered EDI Architecture

The EDI Semantic Layer is the layer that is concerned about application layer functionality's like description of a business application that must drive the EDI in a given case, format translations, acknowledgements etc.

The EDI Standard Layer is usually composed of two individual competing standards:

1. EDIFACT Business Form Standards (UN/ECE) and,
2. ANSI X.12 Business Form Standards (US).

It is important to note here, that just like many other architectures, even here, the application layer is not really truly transparent to its immediate lower layer.

The EDI Transport Layer is concerned primarily with the exact means / mechanism that may be appropriate for actual transfer of such business documents. Such means may include: HTTP transfers, E-mail transfers, FTP transfers etc. Naturally, the layer utilizes the services that either alone or in combination serve the interest of its higher layers best.

11.6 Electronic Funds Transfer (EFT)

The Electronic Funds Transfers (EFTs) are primarily credit transfers between collaborating agencies like Banks, Financial Institutions, Credit Card Companies etc. However, they may take place in many ways involving use of Automatic Teller Machines, Computers, POS Terminals, and Smart Cards etc. Like the EDI, EFT too might exist with or without the Internet although the latter is more common of the two.

Some of the early implementations of the EFT have been at The Federal Reserve's System at Fedwire, Clearing House Interbank Payment System (CHIPS) at New York, Master Card and VISA (all in the USA). In India, the ICICI Bank in Mumbai was one of the earliest such I-Commerce initiative. In China, the HSBC was amongst the early adopters of the technology.

11.7 Secure Electronic Transactions (SET)

It is a joint Visa and MasterCard scheme that combines encryption services and merchant and consumer authorization services. When a credit card purchase is placed, the merchant / agency receives an ITU X.509 certificate.

The ITU X.509 Certificate Format contains:

- Version
- Serial Number
- Algorithm Identifier
- Issuer's Name
- Validity Dates
- Subject Name
- Subject Public-Key Information
- Issuer-Unique Identifier
- Subject-Unique Identifier
- The ITU X.509 Certificate Extension Format contains:
- Authority-Key Id
- Key Usage
- Private-Key Usage Period
- Certificate Policies
- Subject Alternate Name
- Basic Constraints
- Issuer's Alternate Name
- Private Extensions

SET Certificates: Broad Categories

- Digital Signature Certificates
- Key Encryption Certificates
- Certificate and Certificate Revocation List Signing Certificates

The Thumbs / Thumbprints in the SET

Making use of certain specified type of data to Hash Functions generates these. This data comprises of Certificates, Certificate Revocation Lists (CRLs) and Brand CRL Identifiers (BRIs). These Thumbprints are thereafter used to determine exactly which certificates are to be sent to the sender for successful completion of an on-line business transaction using the SET. Inclusion of Thumbprints in SET messages is optional. It insists that consumers necessarily register their accounts with the issuing financial institution so it can provide the authentic digital certificate.

SET does have several issues that may need careful addressing / considerations for providing a flexible. Easy yet secure design.

11.8 The SET Architecture

As of now, very few E-Commerce sites have really adopted the SET; the situation is beginning to change, however. Many companies are considering use of specialized hardware / software systems for greater control and security. Like the IBM-Equifax initiative and the VeriSign's services, the SET initiative may be seen as an effort to

assure prospective ICommerce agencies and customers that business over the Net can be secure and has marked financial benefits.

11.9 The X.400 Standard-based Solution

X.400 (1984, 1988, 1992): It is a store-and-forward messaging standard from the ISO.

The X.400 User Agent: It is an X.400 agent software that interacts in an X.400 setup on behalf of the user / application.

The X.400 Message Transfer Agent: The MTA is X.400 agent software that behaves as if it is a store-and-forward node in an X.400 setup.

The X.400 Message Transfer System: This is a set of the collaborating / interconnected X.400 Message Transfer Agents.

The X.400 Message Store: It is an entity that is responsible for temporarily storing X.400 messages until they are demanded and retrieved by the User Agent(s).

The X.400 Message Handling System: The complete infrastructure comprising of various components of the X.400 setup like UAs, MTAs, MSs etc. is called the X.400 MHS.

X.435: This is an ISO standard that specifies details of EDI Message Transmission over X.400 networks.

Double Bagging: Double Bagging is a technique in which an X.400 header is prefixed to the EDI message header for the purpose of sending an EDI message over an X.400 network / internetwork. As this method involves duplication of some information, it has potential to deteriorate internetwork performance in case of heavy EDI traffic.

An EDI message comprises of a message header and a message body. In case of the X.435, unlike the Double Bagging approach, a special 'EDI Identifier Field' is inserted in the X.400 Envelope. As can be seen, compared to the former method, this method offers better performance since duplication of information is not required.

Although in the market X.435 has not received warm response by any standard, it does have certain useful capabilities and features including:

- It can reliably exchange / manipulate a wide variety of body parts (in a single packaging entity).
- it is capable of exploiting the full power of the X.400, not just the EDI-specific functions.
- it can offer encryption facility on demand.

11.10 The MIME-based Solution

Who, then, is the candidate for the EDI (if X.435 is yet not 'hot')? Interestingly, it is the MIME standard that is getting greater attention. As of now, it is very difficult to predict the probable winner since both standards and related technologies have been around for quite sometime now and also because they have their own set of 'supporters'. S-MIME has only made the battle more interesting!

MIME (Multipurpose Internet Mail Extension) is popular because of several reasons including:

- Support for several mail formats and a large variety of attachments (including the Multimedia, Spreadsheet, Word-processing, HTML, Plain Text, EDI etc.) is readily available.

- Optional encryption and security is available.
- all this can be done just via the good old SMTP itself.

It is relevant to note here that neither the SMTP nor the plain vanilla MIME offers any encryption or authentication services.

Inside MIME: A Quick Look

MIME specifies use of:

1. MIME-version Header
2. MIME-Content-type Header
3. MIME-Content-Transfer-Encoding Header
4. MIME-Content-Identification Header (Optional)
5. MIME-Content-Description Header

Like the X.400, the MIME has:

1. MIME User Agent (comprises of a Parser Module and a post-parsing Dispatcher Module that also invokes a designated viewer as may have been configured by the user for viewing different types of contents / formats)
2. MIME Message Transfer Agent
3. MIME Message System
4. MIME also has provision for Directory Services.

11.11 Smart Cards and other Solutions

A Smart Card, also known as the ICC (Integrated Circuit Card) has a microchip with a small amount of memory (say 8K or 16K) for holding encoded data embedded into it. It resembles the traditional Credit Cards. The primary idea behind the Smart Card Technology is to strengthen the Cardholder's security simultaneously with increased ease of access / use.

11.12 On the Digital Signature and Digital Certificates

A computer network / internetwork that is primarily used for EDI transactions, is a typical example of a VAN. EDI-VANs are relatively slow because of the EDI-specific concerns. EDI-VANs cost more than generic networks / internetworks (in terms of 'operational costs'). VANs do prove cost-effective if designed well.

The Life Without VANs

It is perfectly possible to use the Electronic Data Interchange over the Internet even without using VANs provided there exists a mutual agreement on the rules, conventions and mechanisms of the game between the concerned agencies. These modalities may include:

- encapsulation format, if any,
- addressing rules and formats,
- encryption rules and formats,
- certification and signature formats and trust metrics
- message formats.

The I-Commerce Gateways

- Strong Encryption (subject to legal restrictions)
- Server Gated Cryptography
- Enterprise PKI Technology
- Standalone PKI Solutions
- Integrated PKI Solutions
- Digital ID Technology
- Global Server Ids (subject to legal restrictions)
- Secure Site IDs
- Individual Digital Ids
- Digital Signature and Digital Certificate Technologies
- The Security Solutions: Encryption Basics
- Encryption: It is an action that performs the transformation of a given data into a form that cannot be easily interpreted / understood / used by an unauthorized entity.
- Two basic types of encryption policies do exist:
- Symmetric Encryption Policy / Secret-Key Cryptography &
- Asymmetric Encryption Policy / Public-Key Cryptography

Further, an encryption may be categorized as Strong and Weak Encryption.

A Digitally-signed Communication is a message that has been processed by a computer in such a manner that ties the message to the individual that signed the message.

Criteria for Digital Signatures Technology

An acceptable technology must be capable of creating signatures that conform to requirements:

- It is unique to the person using it;
- It is capable of verification;
- It is under the sole control of the person using it;
- it is linked to data in such a manner that if the data are changed, the digital signature is invalidated.

The technology known as Public Key Cryptography is an acceptable technology for use by public entities.

Asymmetric Cryptosystem: It refers to a computer algorithm or series of algorithms that utilize two different keys with the following characteristics:

- one key signs a given message;
- one key verifies a given message; and,
- the keys have the property that, knowing one key; it is computationally infeasible to discover the other key.

Certificate: It refers to a computer-based record which:

1. Identifies the certification authority issuing it.
2. Names or identifies its subscriber;
3. Contains the subscriber's public key; and
4. Is digitally signed by the certification authority issuing or amending it&
5. Conforms to widely used standards.

Certification Authority. This refers to an entity that issues a certificate, or in the case of certain certification processes, certifies amendments to an existing certificate.

Key Pair: This refers to a private key and its corresponding public key in an asymmetric cryptosystem. The keys have the property that the public key can verify a digital signature that the private key creates.

Private key: It refers to the key of a key pair used to create a digital signature.

The Signature Dynamics Technology

It is an acceptable technology for use by public entities that uses as the means the metrics of the shapes, speeds and/or other distinguishing features of a signature as the person writes it by hand.

It involves binding the measurements to a message through the use of cryptographic techniques. Signature Digest is the resulting bit-string produced when a signature is tied to a document using Signature Dynamics.

Digital Certificate: One of the simplest ways to describe the function of a Digital Certificate is to treat it as a means to verify the genuineness of the Public-Key. This is treated as one method of easy authentication. Just as the individuals / groups are normally assigned Digital Signatures, the corporate merchants and E-Commerce / Commerce Gateways are issued Digital Certificates for proving their authenticity to others.

Certificate Expiry: Most of the certificates have their period of legal validity as marked by the issuing entity / authority, after which it is considered as invalid or expired.

Certificate Revocation: If the Certificate is found to be compromised, it may be explicitly revoked by the Certificate Authority (CA) and included in the subsequently published Certificate Revocation List.

Certificate Validation: It refers to the verification of the Certificate Chain.

Certificate Authorities

As per the SET, the following CAs may exist:

- The Root Certificate Authority (RCA)
- The Brand Certificate Authority (BCA)
- The Geo-Political Certificate Authority (GCA) <optional>
- The Merchant Certificate Authority (MCA)
- The Payment Gateway Certificate Authority (PGCA)
- The Cardholder Certificate Authority (CCA)

Certificate Categories:

- ?Merchant Certificates
- ?Cardholder Certificates

The SET Security Solutions

The SET Protocol, discussed earlier, does use a combination of both Secret-Key as well as Public-key Cryptography. For instance, before transmission of information,

the SET uses the former whereas after the transmission of information, the latter is used.

The Secret-Key encryption scheme used by the SET is the Data Encryption Standard (DES) scheme developed by the IBM Corporation.

The Public-Key encryption scheme used by the SET is the Public-Key Cryptography Standard #7 (PKCS #7) developed by the RSA Data Security company. In near future, the Elliptic Curve Cryptography may coexist with the PKCS in the SET.

The very reason of using such a combination is due to the Encryption Speed of the DES and the superior security of the PKCS (commonly called as RSA).

11.13 The I-Commerce Gateways

Each application-service component of the ICommerce may have its own gateway. Alternatively, integrated gateways may exist. An example of a Service Gateway is an EDI Gateway. One basic function of a gateway is protocol / format translation. Recommending a strategy of solving an ICommerce oriented problem may often involve the following steps:

1. Analyze the company / organization's profile
2. Analyze the customer / audience satisfaction data
3. Suggest an I-commerce Architecture that could provide benefits of the technology with the least possible financial requirements in phases
4. Build / evolve a structural design
5. Carry out simulation studies and if required modify the design
6. Implement a prototype, if it seems a high-risk venture. Else, implement the design.

11.14 Summary

Predecessors of the Internet Commerce or 'I-Commerce', as it is popularly known as, have been traditional and pre-Internet variant of Electronic Commerce (i.e. pre-Internet E-Commerce of the EDI and EFT over telephone company's voice-grade and data networks respectively). Offshoots of the ICommerce frameworks include Mobile-Commerce (better known as MCommerce) and the so-called D-Commerce. Fundamental Components of Internet Commerce Frameworks include: Electronic Data Interchange (EDI), Electronic Funds Transfer (EFT), Secure Electronic Transactions (SET), Corporate Digital Library Systems (CDLS) and Secure Electronic Messaging Systems (SEMS).

The EDI architecture apart from the Physical Layer, has three other layers namely, EDI Transport Layer, EDI Standard Layer and EDI Semantic Layer. The Electronic Funds Transfers (EFTs) are primarily credit transfers between collaborating agencies like Banks, Financial Institutions, Credit Card Companies etc. Digital Signature Certificates, Key Encryption Certificates, Certificate and Certificate Revocation List Signing Certificates are the principal certificate types.

EDI-VANs are relatively slow because of the EDI-specific concerns. VANs do prove cost-effective if designed well. The technology known as Public Key Cryptography is an acceptable technology for use by public entities. The keys have the property that the public key can verify a digital signature that the private key creates. Just as the individuals / groups are normally assigned Digital Signatures, the corporate

merchants and E-Commerce / I-Commerce Gateways are issued Digital Certificates for proving their authenticity to others. The Secret-Key encryption scheme used by the SET is the Data Encryption Standard (DES) scheme developed by the IBM Corporation. The Public-Key encryption scheme used by the SET is the Public-Key Cryptography Standard #7 (PKCS #7) developed by the RSA Data Security company.

11.15 Recommended Readings

1. Grady N. Drew: **Using SET for Electronic Commerce**, Prentice-Hall PTR, 1998.
2. Ravi Kalakota & Andrew B. Whinston: **Frontiers of Electronic Commerce**, Addison-Wesley Longman, Inc., Reading, 1996 (IE: 1999).
3. K. Bajaj & D. Nag: E-Commerce: **The Cutting Edge of Business**, Tata McGraw-Hill, New Delhi, 1999.
4. William Stallings: **Cryptography and Network Security**, Second Edition, Prentice-Hall, Upper Saddle River, 1999.

11.16 Exercises

1. Suggest an I-Commerce architectural framework, keeping internetworking as well as economic aspects in focus, using which, in your opinion, a Digital Library should evolve. Your scheme should permit on-campus, distance learning, practice school and collaborative learning paradigms to co-exist in a cost-effective way. Please mention all your assumptions clearly before proposing your solution.
2. Can the Public Key Cryptography system be also used as an electronic signature? Please justify your answer in brief.
3. Compare the provisions of the EDI, SET and EFT in terms of transfer formats, security-support, support for open standards and primary applications.
4. What role can biometric methods play in I-Commerce frameworks? Explain with the help of a possible scenario.

Chapter-12

Internet Programming

Interaction Goals

Objectives of this chapter are to identify the programming needs of various internetwork applications, discuss their basic constituents, learn about the associated issues, realize the design problems they pose and appreciate the wide spectrum of applications they may be closely associated with. Consequently, this chapter includes a brief revisit of Internet Programming basics, Linux Network Programming and Application Programming Issues for the Web.

At the end of this chapter, you should be able to:

- Realize the employment requirements and usage of the Hyper Text Transfer Protocol (HTTP) and web-based FTP,
- Know the right places of deployment of the Dynamic HTML, XML, Java Technologies, VB Script, PERL and CGI Scripting,
- Realize the relevant Internet Program Design Issues.

12.1 Introduction

This chapter introduces you to the simple basics of the software aspects of the Internet-based Protocols and gives an overview of select Internet Programming Tools in common use. By no means, the treatment given here is either complete or in-depth since the idea here is just to let you have a feel of several basic issues and aspects of Internet Programming. Pre-requisites are minimal. You need only to know a bit of Network Programming (knowledge of programming through Sockets and RPC is enough) and have an elementary knowledge of popular Operating Systems like Linux, Unix, Windows NT/2000 and Solaris.

12.1.1 Linux Network Programming Basics Revisited

Every network protocol has its own definition of *Network Address*. In C, a protocol implementation provides a **struct sockaddr** as the elementary form of a *Network Address*.

A sample definition of **struct sockaddr** may be given as:

```
#include <sys/socket.h>
struct sockaddr {
    unsigned short sa_family;
    char sa_data [MAXSOCKADDRDATA]
}
```

In Linux, sockets are created by the **socket()** system call. This call returns a *file descriptor* for the socket that is yet to be initialized. Then the socket is initialized by

binding it to a particular protocol and address using the **bind()** system call. Use of this provision has been demonstrated below:

```
#include <sys/socket.h>
int socket(int domain, int type, int protocol);
```

Here, the parameter **domain** specifies the **PF**, parameter **type** usually specifies either of **SOCK_STREAM** or **SOCK_DGRAM** and parameter **protocol** specifies the protocol to be used (0=> default protocol associated).

```
int bind (int sock, struct sockaddr * my_addr, int addrlen);
```

Here, the parameter **sock** is the socket-in-question, parameter **sockaddr** is the address of protocol and parameter **addrlen** is the length of the address for the local end-point.

Next, **listen()** system call is executed for informing the system that the process is now ready to allow other processes establish a connection to this socket at the specified end-point. This step does not really establish a connection by itself, however! Now, the **accept ()** system call is made for accepting the connection requests. **accept ()** is a blocking call as it blocks until a process requests a connection. In case, the socket has been marked as 'non-blocking' by the **fcntl()** call, **accept()** would return an error if no process is requesting it for a connection.

```
#include <sys/socket.h>
int listen (int sock, int backlog);
```

Here, the parameter **sock** is the socket-in-question, parameter **backlog** is the number of connection requests that may be pending on the socket before any further connection requests are explicitly refused.

```
int accept (int sock, struct sockaddr * addr, int * addrlen);
```

The **select ()** system call can also be made for determining if any connection request is currently pending to a socket.

A Client attempts to connect to a Server by creating a socket, binding it to an address (optionally) and making the **connect()** call to the Server at the known address.

12.1.2 A Subset of Address Families Used in Linux Environment

Unix / Linux Domain:	AF_UNIX
TCP/IPv4 Domain	AF_INET
TCP/IPv6 Domain	AF_INET6
Novell NetWare Domain:	AF_IPX
AppleTalk Domain:	AF_APPLETALK

12.1.3 A Subset of Protocol Families Used in Linux Environment

Unix / Linux Domain:	PF_UNIX
TCP/IPv4 Domain	PF_INET
TCP/IPv6 Domain	PF_INET6

Novell NetWare Domain:
AppleTalk Domain:

PF_IPX
PF_APPLETALK

12.1.4 Socket Errors (ERRNO VALUES)

The various errors returned by **Socket**-specific operations may include the following:

ENOTSOCK	ETIMEDOUT
EDESTADDRREQ	ECONNREFUSED
EPROTOTYPE	EADDRINUSE
ENOPROTOOPT	EADDRNOTAVAIL
EPROTONOSUPPORT	ENETDOWN
ESOCKTNOSUPPORT	ENETUNREACH
EPFNOSUPPORT	ENETRESET
EAFNOSUPPORT	ECONNABORTED
ENOTCONN	ECONNRESET
EHOSTDOWN	ENOBUFS
EHOSTUNREAD	EISCONN

12.2 The World Wide Web and the Hypertext Transfer Protocol

A Physicist by profession, Tim Berners Lee originally conceived the World Wide Web while doing research at the European Particle Physics Laboratory at the CERN, Geneva in the year 1989. Naturally, world's first Web Server was set up at the CERN naturally and was known as **info.cern.ch**.

Dr. Lee writes in a brief recount of events that unfolded over the years: "I wrote in 1990 a program called "**WorldWideWeb**", a point and click hypertext editor which ran on the "NeXT" machine. This, together with the first Web server, I released to the High Energy Physics community at first, and to the hypertext and NeXT communities in the summer of 1991. Also available was a "line mode" browser by student Nicola Pellow, which could be run on almost any computer. The specifications of UDIs (now URIs), HyperText Markup Language (HTML) and HyperText Transfer Protocol (HTTP) published on the first server in order to promote wide adoption and discussion.*Between the summers of 1991 and 1994, the load on the first Web server ("info.cern.ch") rose steadily by a factor of 10 every year. In 1992 academia, and in 1993 industry, was taking notice. I was under pressure to define the future evolution. After much discussion I decided to form the World Wide Web Consortium in September 1994, with a base at MIT in the USA, INRIA in France, and now also at Keio University in Japan.*"

The WWW is not the Internet, as sometimes believed by the uninitiated. In fact, it is a hypermedia based web / infrastructure of information which made the already existing Internet so user-friendly and hence extremely popular. It is primarily a Client / Server oriented technology which has contributed a lot in the rapid growth of the use of the Internet. In the WWW terminology, a hypermedia document is often called a Page and several such related pages (hosted anywhere within the Web and served by a Web Server) constitute a Website. Often, the entry page, which has an index to further hyperlinks to parts of itself or other hyperdocuments, is called a Home Page.

Web Servers and Web Clients (like Web Browsers) may be located on the same network, on various networks over a local area network or even at the farthest ends of the world over the Internet.

One of the greatest advantages of the WWW technology is the capability of accessing and transfer of information across the interconnected computer networks. Ease of use, connectivity and compatibility with wide-ranging technologies (both hardware and software) is another plus point of this technology that opened up new areas of education, training, commerce, counseling, advertisement, telecollaboration and maintenance.

12.3 The World Wide Web and Uniform Resource Locators (WWW & URLs)

The scheme of resource location uses URLs. The URLs may consist of two or more components depending upon the location of the resource (often a file). These components may include:

- Name of the protocol / scheme using which a resource can be located, like: HTTP, FTP, TELNET, File, News, and Mailto etc.
- Name of the Server / Domain on / in which the resource is located, like: www.bits-pilani.ac.in, web.mit.edu, vu.nl, www.ietf.org etc.
- Path of the file to be located, like: /xyz/abc/hello.html , /asdf/erp.asp etc.
- Filename (as shown above as last part of a path).

12.4 The World Wide Web and File Transfer Protocol (WWW & FTP)

The FTP URL may be expressed in many forms including the following:

ftp://<Server Name>/<Directory Name>/<Filename>

or

ftp://<User ID>:<Password>@<Server Name>:<Port No.>/<Change Directory command with Sub-Directory Name>/<Filename>

The Server Name field is sometimes called as Host Name field.

12.5 The Common Gateway Interface (CGI)

A CGI program / script provides a way for manipulating data on the server side (i.e. the CGI program / script is executed at the end where the Web Server is running). CGI programs / scripts, unless carefully written may invite Security Problems. These security issues may become of greater importance if the machine on which the Web Server is running also has DNS, Mail Server, File Server etc. running on it. One of the most common CGI scripting language is PERL. Steps towards writing a Secure CGI Script may include:

- Before enabling CGI, it may be ensured that a proper policy of access and ownership of various server processes exists.
- If it is a Linux or UNIX or similar environment, web server process should not be normally run as 'root'.
- For Linux / UNIX systems, choice of associating a unique GID and UID may help. (For Apache Servers, normally, it is the httpd.conf file where such configuration information may reside.)
- It may be preferable, in cases like above, that the CGI scripts be located in such a way that monitoring of these becomes simple. Also, limiting visibility of the server only to the files under its own root may add to the security.
- At a given point of time, preferably, there should be only one 'cgi-bin' directory in active or accessible state.
- The options in the access file, often called 'access.conf' in such environments, should be carefully configured so that normal users don't get to see 'cgi-bin' directory index.
- Another file called 'srm.conf' may be looked into for finding out if the SSIs are permitted. Commenting out one more lines starting with 'Addtype' and having 'text/server-parsed-html' may lead to restricting use of SSIs.
- Also, limiting the Server Side Includes (SSI) helps to reduce the security threats to the system.

12.5.1 The Common Gateway Interface (CGI) and PERL

'PERL' stands for Practical Extraction and Report Language. It is used in Web Server scripting, UNIX Domain System Administration Scripting (this was the original purpose of development of PERL when Larry Wall presented the language) etc.

PERL is popular primarily due to its suitability for rapid development, ease of use and maintenance, efficiency and flexibility. Although, PERL scripts may exist on their own, they are often embedded within the HTML code for many Web-oriented applications. PERL 5 provides object-oriented capabilities. It has a module called pppperl (Pretty Good Privacy PERL) that supports public key encryption and thereby enables additional privacy and security to sensitive data, wherever desirable.

Traditionally, PERL has been used to query Databases in UNIX environments and this capability of the language now finds support in many other popular environments as well.

12.5.2 Invoking the PERL

In UNIX / Linux: `perl <optional switch> filename.pl`

(For embedded scripts, in UNIX / Linux systems, a line like `#!/usr/local/bin/perl` may have to be placed at the start of the PERL script.)

In Windows NT Server: `perl <optional switch> filename.pl`

A pathname may have to be provided in both of the cases.

12.5.3 Select command-line switches and options:

Switch	Option	Action / Meaning
-a	None	Automatically split records
-c	None	Check syntax
-e	Command	Pass a command to the PERL from the command line
-l	File extension	Replace the original file with result of script execution
-n	None	Execute script by taking each of the specified files as arguments
-s	None	Allow arbitrary switch(es) to be passed to the PERL
-w	None	Print syntax error warnings
-D	Flags (p,s,l,t,o,c,p,m,f,r,x,u,L,X,H,D)	Debugging behaviour specification
-F	A regular expression (default is white space)	Split records by using the said expression as a separator
-I	Directory name with path	Include file(s) from this directory
-P	None	First, pass the script through the C preprocessor, then compilation by the PERL compiler

12.5.4 Data Types in PERL

Scalar (numbers, strings; *\$You* is an example of scalar variable)

Array (a list of scalars indexed by integer subscripts; *@Boyis* an example of array variable)

Associative Array (a list of values indexed by strings (called 'keys'); *%Faculty* is an example)

12.5.5 File Handles in PERL

STDIN / STDOUT / STDERR / filename

12.5.6 File Access Symbols

(other than the 'pipe' |)

< means Open to read (default)

> means Open to write

>> means Open to append
+< or +> means Open for reading & writing

12.5.7 Relational Operators

> gt
< lt
>= ge and <= le
== eq and != ne
<==> cmp

12.5.8 Logical Operators

|| OR
&& AND
! NOT

12.5.9 Conditional Operators

IF / WHILE
UNLESS / UNTIL

12.6 The Server Side Includes: An Example

As explained earlier, although there do exist quite a few situations that benefit by the SSIs, a careless use of these may give hackers an ample opportunity of attacking the system. An example code having SSI with potential risk factor is given below.

```
<HTML><body>
...
<Thanks for visiting this website!>
<h1>This page was last modified on</h1>
<!--#echo var="LAST MODIFIED"-->
<h2>See you soon!</h2>
<!--#exec cmd="..." -->
...
```

```
<h1>Response Form from the client</h1>

<form action="http://www.bits-pilani.ac.in/CGI-bin/rb.pl" method="get">

<input type="hidden" name="address" value="rahul@bits-
pilani.ac.in">

...

</form>

</body></HTML>
```

12.7 Java Technologies

12.7.1 The Concept of the Java Threads

A Java *thread* is a program's path of execution. Most programs written today run as a single thread. They, therefore, cause problems when multiple actions need to be carried out at the same time. Multithreaded applications are capable of running many concurrent threads within a single program. Multithreading refers to multiple lines of a single program those can be executed at the same time.

In Java, a thread shares the original data area of the parent. The thread that should be run is chosen on the basis of its priority number (ranging from 1 to 10). The default priority of a thread (Thread.NORM_PRIORITY) is set to 5. (Thread.MIN_PRIORITY is set to 1, and Thread.MAX_PRIORITY is set to 10.) The getPriority() method can be used to find the current value of the priority of a thread. Daemon threads (service threads) are those threads that normally run at a low priority and provide a basic service to a program / programs.

12.7.1.1 Creating threads

Java provides two ways of creating threads:

- implementing an interface and
- extending a class.

Extending a class is the way Java inherits methods and variables from a parent class. In this case, one can only extend or inherit from a single parent class.

Interfaces provide a way for programmers to lay the groundwork of a class. They are used to design the requirements for a set of classes to implement.

There are a few differences between a class and an interface:

- An interface can only contain abstract methods and/or static final variables (constants); Classes can implement methods and contain variables that are not constants.

- An interface cannot implement any methods; a class that implements an interface must implement all methods defined in that interface.
- An interface has the ability to extend from other interfaces, and (unlike classes) can extend from multiple interfaces.
- Furthermore, an interface cannot be instantiated with the new operator; for example, `Runnable a=new Runnable();` is not allowed.

12.7.2 The Java Script: A Scripting Language

Java Script is a scripting language that, like the VB Script, is embedded within an HTML script and is interpreted by the Browser. Like Java, it supports Objects; but unlike Java that is compiled on the Server prior to its execution Java Script (JS) is interpreted by the Client. Also, in contrast to Java, the JS supports loose / weak typing (for variable declarations). Similarly, unlike Java, the JS features Dynamic Binding. (This means that in this case Object are verified at the Run Time rather than at the Compile Time.)

12.7.2.1 Java Script, HTML and Frames

Frames allow creation of multiple document windows within one browser. Each frame appears to act like a separate browser windows, displaying multiple information sources simultaneously. Within each frame you can scroll up and down, and perform all the things that you would normally do within a single browser window. Additionally, the links in a frame can control what is displayed in other frames or windows. With JavaScript, this control becomes even more useful. In order to maintain compatibility with the browsers that do not support Frames, there is a *NOFRAMES* tag pair that displays alternative pages on the screen of such browsers. A multi-frame document should not have a *BODY* tag pair in the FRAMESET HTML file. We can also specify the size in a fixed number of pixels. Alternatively, we may place an asterisk for one of the rows for indicating that the row in question will consume the rest of the space.

Several other options are available for frames' support. Each frame can also access the objects of its parent and indirectly of other frames in the document. Each given object has a parent, whether implicitly or explicitly identified. The parent of any variable that you create is the JavaScript list.

In frames' world, if we wish to have a link to a previously created page, we may insert a single line like:

```
passfr.location = "http://www.bits-pilani.ac.in/~rahul/";
```

In JavaScript, it is possible to specify functions to be executed in an event-driven manner.

12.7.2.2 Java Script: A Partial Event List

`onChange` when a selection, text, or textarea field is modified *or* loses "focus"

`onClick` when a button, checkbox, link, or radio object is clicked or selected

onLoad	when a window or a FRAME is loaded
onMouseOver	when the mouse is moved over a hyperlink
onSelect	when text in a text or textarea field is selected
onSubmit	when a FORM is submitted

These event handlers are identified as additional parameters to specific HTML tags: BODY, A HREF, INPUT (all forms), FORM, TEXTAREA, and SELECT.

12.7.2.3 The Visual Basic Script and its Position vis-à-vis Java Script

The Visual Basic Script (VBScript), like the Java Script, is embedded within an HTML script. As of now, the primary use of the VB Script is the Web Page Enhancement. The VB Script has born out of the Microsoft Visual Basic language. Thus, syntactically, VB Script is not really different from the VB.

Unlike the JS, the VB Script is not case-sensitive but like the JS, the Visual Basic Script too requires the HTML tag `<SCRIPT>`.

12.8 The ActiveX Scripting Services

The ActiveX class of Microsoft technologies was originally designed for being used as the backbone of the parent company's objective to evolve a long-term Internet-based strategy.

Just as the Java Script of Netscape obviates use of CGI scripting for certain situations like the cases where conditional logic is to be used or wherein some event driven (say user response driven) output / action is desirable; the ActiveX Scripting Services aim to provide similar functionality (and a few more!) for the compatible browsers.

12.8.1 Classes of ActiveX Scripting Components

Basically, there exist two classes of ActiveX Scripting components:

1. ActiveX Scripting Hosts and
2. ActiveX Scripting Engines.

The host / client platform (often a compatible browser like the *IE 5* of Microsoft, a Web Authoring Tool or a Web Server) provides facilities / services for the execution of the scripting engine. In other words, the ActiveX Scripting Engine requires an ActiveX Scripting Host to execute.

A well-known example of such an engine is the VB Script, which is a Web-oriented scripting language. Other possible ActiveX Scripting Engine environments include Borland's Delphi, Scheme and PERL.

ActiveX Scripting like the ActiveX controls is based on the OLE specification (2.0 and later). As a result, in the compatible environments, it helps to add the automation capability and capability to interact with other existing applications. On the flip side, it is rarely available for non-MS Windows environments.

12.8.2 The VB Script and the Visual Basic

The VB Script needs a compatible browser whereas the VB programs can execute at his or her own. Current version of the VB Script does not support Array Handling in the manner VB supports. The VB Script does not support features like Conditional Compilation, DDE, Static Variables and Collection.

As of now, several control statements like For Each...Next, GoTo, On Error GoTo, GoSub...Return, With....End With etc. are not available to the VB Script writer.

12.9 XML: A Quick Look

XML is the abbreviation for the eXtensible Markup Language. It is a successor to the HTML and derived from the SGML. This offers a universally recognizable and hopefully portable syntax for describing and structuring data. This data description is independent of the application logic. XML supports data interchange in a networked environment and document publishing to multiple media and devices. The World Wide Web Consortium (W3C) has developed XML. *The XML development effort started in 1996 led by Jon Bosak of Sun Microsystems.* The basic idea was to develop a simplified version of SGML (Standard Generalized Markup Language) for the Web. In February 1998, XML 1.0 specification became a recommendation by the W3C.

XML promises to simplify and lower the cost for data exchange and publishing in a Web-based environment. XML has a text-based syntax that is portable and reusable across different platforms and devices. An XML document is also flexible and extensible, allowing new tags to be added without breaking an existing document structure. Through Unicode support, XML also provides global language support.

Applications of the XML include e-commerce, supply-chain integration, media-independent publishing (XML does so by allowing documents to be written once and published in different formats and for different devices.) etc.

12.9.1 XML and Java: A Quick Look

The Java and XML technologies are complementary. The Java technology provides the platform-independent, maintainable code that is needed to process platform-independent XML data. In addition, the Java technology offers a substantial productivity boost for software developers compared to programming languages such as C or C++.

Using XML and Java together, developers can build powerful web-based applications and platform-independent web-based applications more quickly and at a lower cost. Java APIs are under development to support XML. These APIs will be developed completely in the Java and will support, and fully conform, to widely accepted XML standards, including XML 1.0, SAX, DOM Level 1 Core, XML Namespaces, Emerging XML standards likely to be supported include: XSLT and XQL.

XML is an essential component in the Java 2 Enterprise Edition and will be supported throughout J2EE as a means for enabling business-to-business information interchange using XML. For robust, synchronous data messaging, Enterprise JavaBeans technology can be used to create a business service object and the XML content can be sent over the wire using JSP. Currently, JavaServer Pages can be used to generate and consume XML between n-tier servers or between server and client. Java Messaging Service provides a means for asynchronous XML data messaging. In addition, Enterprise JavaBeans uses XML to describe its deployment properties, giving Enterprise JavaBeans data portability in addition to its code portability.

12.10 Summary

Every network protocol has its own definition of *Network Address*. In C, a protocol implementation provides a **struct sockaddr** as the elementary form of a *Network Address*. A Client attempts to connect to a Server by creating a socket, binding it to an address (optionally) and making the **connect()** call to the Server at the known address.

The WWW is not the Internet; in fact, it is a hypermedia based web / infrastructure of information. In the WWW terminology, a hypermedia document is often called a Page and several such related pages (hosted anywhere within the Web and served by a Web Server) constitute a Website. Web Servers and Web Clients may be located even at the farthest ends of the world over the Internet. One of the greatest advantages of the WWW technology is the capability of accessing and transfer of information across the interconnected computer networks.

A CGI program / script provides a way for manipulating data on the server side (i.e. the CGI program / script is executed at the end where the Web Server is running). CGI programs / scripts, unless carefully written may invite Security Problems. 'PERL' stands for Practical Extraction and Report Language. It is used in Web Server scripting, UNIX Domain System Administration Scripting. Although there do exist quite a few situations that benefit by the SSIs, a careless use of these may give hackers an ample opportunity of attacking the system.

A Java *thread* is a program's path of execution. Most programs written today run as a single thread. Multithreaded applications are capable of running many concurrent threads within a single program. Multithreading refers to multiple lines of a single program those can be executed at the same time. : Java provides two ways of creating threads: implementing an interface and extending a class. Extending a class is the way Java inherits methods and variables from a parent class. In this case, one can only extend or inherit from a single parent class. Interfaces provide a way for programmers to lay the groundwork of a class. They are used to design the requirements for a set of classes to implement.

Java Script is a scripting language that, like the VB Script, is embedded within an HTML script and is interpreted by the Browser. Like Java, it supports Objects; but unlike Java that is compiled on the Server prior to its execution Java Script (JS) is interpreted by the Client.

The VB Script needs a compatible browser. Current version of the VB Script does not support Array Handling in the manner VB supports. The VB Script does not support features like Conditional Compilation, DDE, Static Variables and Collection. Just as the Java Script of Netscape obviates use of CGI scripting for certain situations like

the cases where conditional logic is to be used or wherein some event driven (say user response driven) output / action is desirable; the ActiveX Scripting Services aim to provide similar functionality (and a few more!) for the compatible browsers.

XML is the abbreviation for the eXtensible Markup Language. It is a successor to the HTML and derived from the SGML. This offers a universally recognizable and hopefully portable syntax for describing and structuring data. This data description is independent of the application logic.

The Java and XML technologies are complementary. The Java technology provides the platform-independent, maintainable code that is needed to process platform-independent XML data.

12.11 Recommended Readings

1. Michael Afergan et al: **Web Programming Desktop Reference**, Prentice Hall of India, New Delhi, 1998.
2. William Buchanan: **Advanced Communications and Networks**, Chapman & Hall, London, 1997.
3. Dr. Tim Berners Lee: **The World Wide Web: A very short personal history**, available at the URL: <http://www.w3.org/People/Berners-Lee/ShortHistory.html>

12.12 Exercises

1. Compare VB Script capabilities with the Java Script capabilities with respect to the Client-side Scripting support.
2. You are designing a distributed Portal for a commercial software company. If this company has multiple offices spread all over the country and you are required to provide seamless information transfer in an easy to use but secure manner using heterogeneous hardware and software platforms in different locations, which set of technologies would you prefer for the following list of services, in which manner and why?
 - i. Server-side scripting and execution support
 - ii. Server-to-Server communication support
 - iii. Client-side scripting support
 - iv. Control over the client's browser
 - v. Monitoring the user-behaviour

Please discuss each of your choices very briefly.

3. Compare the strengths and weaknesses of the VB Script with that of the Java Script.
4. Why could it be dangerous to indiscriminately allow the SSIs?
5. What is a Socket API and how does it aid in network programming?

Appendix-1

Extracts from a Suggested IPv6 Flow Label Specification from an ID Published by the IETF

IPv6 Working Group
Internet Draft
draft-banerjee-flowlabel-ipv6-qos-03.txt

Rahul Banerjee
Sumeshwar Paul Malhotra
Mahaveer M
BITS, Pilani (India)

(This is a Modified Specification for use of the IPv6 Flow Label for providing an efficient Quality of Service using a hybrid approach. Obsoletes 00, 01, 02 versions of this draft.)

Status of This Memo

This document is an Internet Draft and is subject to all provisions of Section 10 of RFC 2026. Internet Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet Drafts.

Internet Drafts are draft documents valid for a maximum of 6 months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet Drafts as reference material or to cite them other than as a "work in progress".

The list of current Internet Drafts can be accessed at <http://www.ietf.org/lid-abstracts.html>

The list of Internet Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>

Copyright(C) The Internet Society (2002). All Rights Reserved.

Abstract

This memo suggests a pragmatic specification for defining the 20-bit Flow Label field using a hybrid approach that includes options to provide IntServ as well as DiffServ based support for IPv6 Quality of Service. It also compares various suggested approaches for defining the 20-bit Flow Label field in IPv6 Base Header based on RFC 2460 (December 1998) and few other drafts. Addressing the IPv6-Multicast-QoS issues also becomes possible as a consequence. This draft clearly specifies exactly when and how various options are to be used; and in case of the MFC, exactly how a specific action might be taken by the suggested implementation. Thus the resultant mechanism is fully implementable and unambiguous as even the lower level details have been worked out as may be required for actual implementations. The draft also has a pointer to an experimental QoS scheme called MultServ.

A-1.1. Introduction

This draft addresses the design and implementation-specific issues pertaining to the Quality of Service (QoS) support in the Flow Label field of the IPv6 Base Header. It provides support for IntServ and DiffServ Quality-of-Service. Though the IPv6 Base Header has a 20-bit Flow Label field for QoS implementation purposes, it has not yet been exploited. Very few Internet Drafts address these long-standing issues and attempt to present solutions in the form of a clear specification of the 20-bit Flow Label in IPv6. This work attempts to provide an analysis of these definitions and subsequently suggests a modified IPv6 Flow Label specification, which in view of the authors can provide an efficient Quality-of-Service.

A-1.2. IPv6 Flow Labels

The IPv6 Flow Label [RFC 2460] is defined as a 20-bit field in the IPv6 header which may be used by a source to label sequences of packets for which it requests special handling by the IPv6 routers, such as non-default quality of service or "real-time" service. The nature of that special handling might be conveyed to the routers by a control protocol, such as RSVP, or by information within the flow's packets themselves, e.g., in a hop-by-hop option.

The characteristics of IPv6 flows and Flow Labels are given in the Appendix A.1

A-1.3. Issues related with IPv6 Flow Label

According to RFC 1809, the IPv6 specification originally left open a number of issues, of which the following are important.

A-1.3.1. What should a router do with Flow Labels for which it has no state?

[RFC 1809] and the author's view suggest that the default rule should be that if a router receives a datagram with an unknown Flow Label, it treats the datagram as if the Flow Label is zero. Unknown flow labels may also occur if a router crashes and loses its state. As part of forwarding, the router will examine any hop-by-hop options and learn if the datagram requires special handling. The options could include simply the information that the datagram is to be dropped if the Flow Label is unknown or could contain the flow state the router should have.

A-1.3.2. How does an internetwork flush old Flow Labels?

Stale Flow Labels can occur in a number of ways, even if we assume that the source always sends a message deleting a Flow Label when the source finishes using a Flow.

1. The deletion message may be lost before reaching all routers.
2. Furthermore, the source may crash before it can send out a Flow Label deletion message.

The authors of the document suggest the following approach as a solution to this problem:

1. The MRU (Most Recently Used) algorithm should be used for maintaining the Flow Labels. At any point of time, the most recently used Labels alone will be kept and the remaining should be flushed.
2. Before flushing a label, the router should send an ICMP message to the source saying that the particular label is going to be flushed. So the source should send a KEEPALIVE Message to the router saying not to flush the Flow Label in case the source requires the Flow Label to be used again. On the other hand, if the source agrees with the router to delete the Flow Label, it should send a GOAHEAD Message to the router. On receiving the GOAHEAD Message, the router immediately deletes the label for that particular source. These messages are also sent to all the intermediate routers, so that, those routers can as well flush the Flow Labels for that particular source.
3. In case, the router does not receive any consent from the source, it will re-send the ICMP message for at most two or three times. If the router does not receive any reply from the source, it can flush the particular Label assuming that the Flow Label was not important for the source or any other intermediate router. The intermediate routers will also delete that Flow Label as they didn't receive any message from the source. The policy of sending the ICMP message to the source two or three times ensures the proper behavior of the method of flushing Flow Labels in case of packet loss. This method assumes that the ICMP message would not be lost all the three times. Hence, if the router doesn't receive any reply from the source even after sending the ICMP message three times, it deletes the label.

A-1.3.1. Which datagrams should carry non-zero Flow Labels?

According to RFC 1809, following were some points of basic agreement.

1. Small exchanges of data should have a zero Flow Label since it is not worth creating a flow for a few datagrams.
2. Real-time flows must always have a Flow Label.

One option specified in [RFC 1809] is to use Flow Labels for all long-term TCP connections. The option is not feasible in the view of the authors, as it will force all the applications on that particular connection to use the Flow Labels, which in turn will force routing vendors to deal with cache explosion issue.

A-1.3.2. Mutable/Non-mutable IPv6 Flow Label

The Flow Labels should be non-mutable because of the following reasons:

1. Using mutable Flow Labels would require certain negotiation mechanism between neighboring routers, or a certain setup through router management or configuration, to make sure that the values or the changes made to the Flow Label are known to all the routers on the path of the packets, in which the Flow Label changes. On the other

hand, the non-mutable Flow Labels certainly have the advantage of the simplicity implied by such a characteristic.

2. A mutable Flow Label characteristic goes against the IPv6 specification of the Flow Label explained in section 2 and the IPv6 Flow Label characteristics explained in the coming sections.

A-1.3.1. Filtering using Flow Label

If, at all, any filtering has to be done based on the Flow Label field in the IPv6 header, the expectation is that the IPv6 Flow Label field carries a predictable or well-determined value. This is not the case if the Flow Label has randomly chosen values.

Supporting the arguments given in [draft-conta-ipv6-flow-label-02.txt], the authors of this document suggest that the problem of not being able to configure load-filtering rules, which are based or are including the Flow Label, can be resolved by relaxing IPv6 specification of having a random number in the Flow Label field. Exactly how can it be done has been suggested later.

A-1.4. A modified specification for the IPv6 Flow Label and related implementation mechanism: A hybrid approach suggested by this work

A-1.4.1. Overview

Appendix A.2 gives a comparison on various approaches suggested in [draft-conta-ipv6-flow-label-02.txt] on defining the 20-bit Flow Label. This section specifies a modified Flow Label for IPv6 for providing efficient Quality of Service that utilizes the results of some of the works referred in Appendix A.2, extends some of these suggested mechanisms and finally presents an integrated hybrid approach.

A-1.4.2. Definition of first three bits of the Flow Label

The hybrid approach suggested in this section includes various approaches, which are mentioned in Appendix A.2. The 20-bits of the Flow Label should be defined in an appropriate manner so that various approaches can be included to produce a more efficient hybrid solution. Hence, for this purpose, the first three bits of the IPv6 Flow Label are used to define the approach used and the next 17 bits are used to define the format used in a particular approach.

Following is the bit pattern for the first 3 bits of Flow Label that defines the type of the approach used:

- 0 0 0 Default.
- 0 0 1 A random number is used to define the Flow Label.
- 0 1 0 The value given in the Hop-by-Hop extension header is used instead of the Flow Label.

- 0 1 1 PHB ID.
- 1 0 0 A format that includes the port number and the protocol in the Flow Label is used.
- 1 0 1 A new definition explained later in this section is used.
- 1 1 0 Reserved for future use.
- 1 1 1 Reserved for future use.

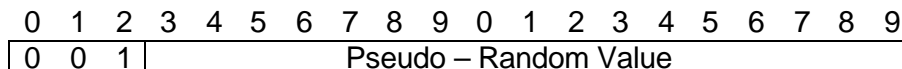
This definition of Flow Label includes IntServ, DiffServ and other approaches for defining the Flow Label. A further explanation of these options is provided in the remaining part of this section. The default value specifies that the datagram does not need any special Quality of Service.

A-1.4.3. Defining the remaining 17 bits of the IPv6 Flow Label

The remaining 17 bits of the IPv6 Flow Label are defined based on the approach defined in the first three bits of the Flow Label.

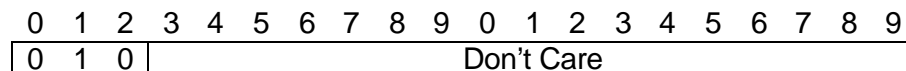
A-1.4.3.1 Random Number

As specified in IPv6 specification, a random number can be used to define the Flow Label. Here a 17-bit random number can be used. The random numbers can be generated in the range from 1 to 1FFFF. Keeping the IPv6 specifications in mind, the authors of this document believe that the random number can be used as one of the approaches. As other approaches are defined in the Flow Label, this random number approach may not be used whenever not feasible or efficient to do so.



A-1.4.3.2 Using Hop-by-Hop extension header

As defined in [draft-banerjee-ipv6-quality-service-02.txt], Hop-by-Hop extension header can be used for defining the Flow Label in case IntServ is used. In this case the value in the 20-bit Flow Label is ignored. The modified Hop-by-Hop extension has been suggested and defined in the reference [draft-banerjee-ipv6-quality-service-02.txt]. In that draft, the Hop-by-Hop extension header has been defined to be used with IntServ. This mechanism applies to define for DiffServ as well.



A-1.4.3.3 Using PHB ID

This defines the DiffServ with MF classifier. In that case the format of the Flow Label will be as shown below:

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
0	1	1	DiffServ IPv6 Flow Label																

As suggested in [draft-conta-ipv6-flow-label-02.txt], this Flow Label can be a PHB ID (Per Hop Behavior Identification Code). In this case, 16-bit PHB ID will be used and the remaining 1 bit is reserved for future use.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
0	1	1	Per Hop Behavior Ident. Code																R

'R' is reserved.

Packets coming into the provider network can be policed based on the Flow Label. The provider, based on the SLAs, SLSS, TCAs, TCSs agreed with the client, configures MF classifiers. This draft specifies the classifier which is little different from the one suggested in the [draft-conta-ipv6-flow-label-02.txt]. The classifier looks like:

C = (SA/SAPrefix, DA/DAPrefix, Flow-Label).

or

C` = (SA/SAPrefix, DA/DAPrefix, Flow-Label-Min: Range).

The range here specifies the difference between the maximum and the minimum Flow Label. The significance of using the range instead of Maximum Flow Label is the reduced number of bits. Definitely the difference between the two values can be specified in a lesser number of bits as compared to the value itself.

Flow-Label-Classifer:

IPv6SourceAddressValue/Prefix : 10:11:12:13:14:15:16:17:18::1/128
IPv6DestAddressValue/Prefix : 1:2:3:4:5:6:7:8::2/128
IPv6 Flow Label : 50

or

IPv6SourceAddressValue/Prefix : 10:11:12:13:14:15:16:17:18::1/128
IPv6DestAddressValue/Prefix : 1:2:3:4:5:6:7:8::2/128
IPv6 Flow Label:Range : 10:20

Incoming Packet header (SA, DA, Flow Label) is matched against classification rules table entry (C or C`).

A-1.4.3.4 Using the Port Number and the Protocol

This approach defines Flow Label by including the server port number and the host-to-host protocol. The "Server Port Number" is the port number assigned to the server side of the client/server applications. As specified in [draft-conta-ipv6-flow-label-02.txt], this approach reserves 16 bits for the port number and 1 bit for the protocol with the remaining bits reserved for the future use.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
0	0	0	TCP Server port number																0

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
1	0	0	UDP Server port number																1

But this approach puts the restriction on the protocol to be used by any application.

As most of the application seeking Real-time service use TCP or UDP as the transport layer protocol, this approach would work fine in most of the cases. In case the application requires to use any other host-to-host protocol, the other methods for specifying the Flow Label, discussed in this section can be used. Anyhow, this method for specifying the port number and the protocol can be exploited further in the future to remove any limitations.

A-1.4.3.5 A new structure and mechanism for the use of the Flow Label

This section describes an innovative approach to define the 20-bit Flow Label field in IPv6 header. By the optimal use of the bits in the Flow Label, this approach includes various Quality of Service parameters in the IPv6 Flow Label that may be requested by any application. The various Quality of Service parameters are:

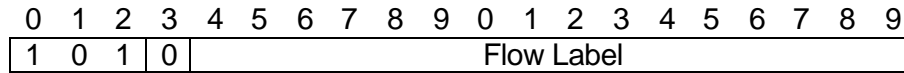
1. Bandwidth
2. Delay or Latency
3. Jitter
4. Packet Loss
5. Buffer Requirements

As packet loss and the jitter are often desired to be of minimum value by any application, these two parameters may not be defined in the Flow Label field itself. Instead, if needed, the Hop-by-Hop EH space can be effectively used to specify these parameters. Bits thus saved in the Flow Label can be effectively used for more demanding purposes. The Quality of Service parameters that are to be included in the Flow Label are:

1. Bandwidth (to be expressed in multiples of kbps).
2. Delay (to be expressed in nanoseconds).
3. Buffer requirements (to be expressed in bytes).

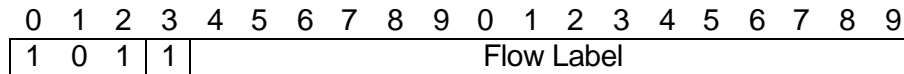
As there are only 17 bits left, the optimal use of the bits is very important so as to obtain the maximum information out of those 17 bits. The first bit out of these 17 bits is used to differentiate between the hard real time and soft real time applications. This bit is set to 0 for soft real time applications and it is set to 1 for hard real time applications.

Soft Real time applications:



This service is meant for RTT (Real Time Tolerant) or soft real time applications, which have an average bandwidth requirement and an intermediate end-to-end delay for an arbitrary packet. Even if the minimum or maximum values specified in the Flow Label are not exactly met, the application can afford to manage with the QoS provided.

Hard Real time applications:



This service is meant for RTI (Real Time Intolerant) or hard real time applications, which demand minimal latency and jitter. For example, a multicast real time application (videoconferencing). Delay is unacceptable and ends should be brought as close as possible.

For this videoconference (DTVC) case, the required resource reservations are

- a. Constant bandwidth for the application traffic.
- b. Deterministic Minimum delay that can be tolerated.

These types of applications can decrease delay by increasing demands for bandwidth. The minimum or maximum values specified in the Flow Label have to be exactly met for these kind of applications.

After keeping one bit for Hard/Soft real time applications, we are left with 16 bits for defining the Flow Label. The remaining part of this section discusses how to represent the values of bandwidth, delay and buffer requirements.

1. Bandwidth

This definition specifies 6 bits out of the 16 bits to be used for specifying the bandwidth value.

Each value in these six bits corresponds to a pre-defined value for bandwidth. Further explanation about this is given at the end of this section.

2. Buffer Requirements

This definition specifies next 5 bits out of the 16 bits to be used for specifying the buffer value.

Each value in these six bits corresponds to a pre-defined value for buffer requirement. Further explanation about this is given at the end of this section.

3. Delay

This definition specifies last 5 bits out of the 16 bits to be used for specifying the delay value.

Each value in these six bits corresponds to a pre-defined value for delay.

The approach described here is a DiffServ based mechanism for providing the QoS as any packet received by any router is classified based on the MF Classifier which is a triplet consisting of the source address, destination address and (bandwidth, buffer and delay). The packet that arrives at the router is examined for the values specified in bandwidth, buffer and delay fields and is matched with the classifiers corresponding to which the packet is provided with the QoS. The classifier looks like:

C = (src address, dest address, flow label);
Where flow label = (bandwidth, buffer, delay)

MF Classifier	Bandwidth	Buffer	Delay
0, 0, 0	32 kbps	512 bytes	4 ns
0, 0, 1	32 kbps	512 bytes	8 ns
63, 31, 31	64 tbps	1 tbytes	8 sec

A-1.5. A possible mechanism for the implementation of the above design.

This section describes one possible mechanism that will allow immediate and practicable implementation of the above design.

A-1.5.1 Data structures required (at the router).

The data structures are specific to the implementations. Different implementations can choose their own data structures that will be required to implement the above design.

Any router that tries to implement QoS maintains a QoS routing table and keeps track of the QoS available to each destination through the required number of hops [RFC 2676]. Apart from this table, the router needs to keep track of the allotted QoS to each and every flow.

This table is the ALLOTTED_QOS_TABLE.

1. Defining the different approaches.

```
enum MODEL_ID {  
    RANDNUM=1,    // the random number method
```

```

HOPBYHOP=2, // the hop-by-hop extension header method
PHB_ID=3, // the multi-field classifier
PORT_PROT=4, // port/protocol method
HYBRID=5 // the hybrid approach
};

```

2. Defining the different Resource Identifiers.

```

enum RES_ID {
    BANDWIDTH=0, // bandwidth requirement
    DELAY=1, // delay requirement
    BUFFER=2, // buffer requirement
};

```

3. Defining the value of the resource.

```

typedef unsigned int RES_VAL;

struct RESOURCE {
    RES_ID res_identifier; // identifier of the resource
    RES_VAL res_value; // 32-bit value of the resource
};

```

4. Defining the Quality of Service.

```

struct QOS_INFO {
    MODEL model_id;
    RESOURCE resource;
};

```

5. Defining the port/protocol and the flow label.

```

struct port_protocol {
    unsigned port; // port number
    unsigned protocol; // protocol
};

union format {
    unsigned flowlabel; // 20-bit Flow Label value
    struct port_protocol port_prot;
};

```

6. Defining the packet information.

```

struct PACKET_INFO {
    struct sockaddr_in6 src_addr;
    struct sockaddr_in6 dest_addr;
    union format format_value;
};

```

7. Defining the Alloted QoS table.

```
struct ALLOTTED_QOS_TABLE {  
    struct PACKET_INFO packet;  
    struct QOS_INFO qos;  
};
```

A-1.5.2 Function of the Source

The application specifies the desired QoS and the Flow Label field in the IPv6 header is filled based on the QoS asked by the application. The application has the flexibility of specifying which format it wants to use for getting the desired QoS. It can specify any of the formats described in this document. The packet is then put on the network and it reaches the intermediate routers

A-1.5.3 Function of each relevant intermediate router

A-1.5.3.1 Initial Processing (Checks for default service)

It gets the format used by the packet by reading the first three bits of the Flow Label. In case the first three bits are 000 or 110 or 111, it represents the default service. No specific treatment is required for this particular packet. In this case, no further processing of the packet is required and the default QoS is provided to the packet. If the value given in the first three bits is 010, no further processing is done and the router knows that the required QoS is specified in the hop-by-hop extension header.

A-1.5.3.2 Searching for the entry (In case of non-default service)

1. The ALLOTTED_QOS_TABLE table is searched based on the source address.
2. If an entry is found, then for that particular source, a search is made based on the PACKET_INFO structure defined above. If all the information stored exactly matches with the information contained in the incoming packet, the IPv6 packet is processed so that the reserved QoS is met.

A-1.5.3.3 New Entry

1. If an entry is not found, a new entry is made in the ALLOTTED_QOS_TABLE table for the source and further processing of this new entry is done as follows.
2. All the relevant structures defined above are filled based on the information contained in the packet. Information about the packet is stored in the PACKET_INFO structure.
3. It reads the desired QoS from the packet's header. If the format specifies that a random number is used in the Flow Label field, it reads the RANDOM_NUMBER table. It reads the specified QoS from the table and

maintains that in the QOS_INFO structure after updating the RESOURCE structure. It then moves onto step 7.

4. If the format specifies that PHB ID is used in the Flow Label field, it reads the Flow Label and the packet is classified based on the MF classifier described in the previous section and it moves on to the step 7.
5. If the value in the Flow Label field specifies that the PORT/PROTOCOL field is used in defining the QoS required by the packet, it fills the RESOURCE structure and the QOS_INFO structure and moves onto step 7.
6. If the value in the Flow Label field specifies that the hybrid approach is used where the packet specifies the values of the bandwidth, delay and buffer requirement. The packet is classified based on the MF classifier described in the previous section and it moves on to the step 7.
7. It then checks with the QoS Routing table, to find out if the desired QoS is possible to be provided to the packet. If yes, it updates the new entry in the ALLOTTED_QOS_TABLE table in the memory or else this entry is removed.
8. If any relevant router en-route is not able to guarantee the requested QoS, an ICMPv6 message is sent to the source and the other routers (that had guaranteed the QoS) are also notified of the same so that they delete the corresponding entry from their QoS tables.

This process executes at all the intermediate routers between the source and the destination.

A-1.6. When to use which approach?

1. Random Number: This approach supports the pure IntServ based model. So if the network uses only IntServ model for QoS, using random numbers in Flow Label is a valid option. But in some conditions it is not desirable to use random numbers in Flow Label. If the network is required to have a deterministic behavior, using random numbers is not a good option as it increases the unpredictability. Again, if any load filtering rules have to be designed based on or using the Flow Label, random numbers should not be used as the value in the Flow Label can not be predicted.
2. PHB ID: This approach supports the pure DiffServ based model. So if the network is designed so as to support DiffServ model for QoS, using PHB ID in flow label and using MF classifier as described in the previous sections is a valid option.
3. Hybrid: Again, if the network supports DiffServ model for QoS, using this approach is a valid option. Here the application should be capable of providing the exact values of bandwidth, delay and buffer requirement it needs.
4. Hop-by-Hop: For using this approach, the application should be capable of specifying the values of QoS parameters. So if the application has these

details and the values asked by the application are not supported by the hybrid approach, this approach should be used.

5. Port-Protocol method: If the network is designed so as to perform some load filtering based on the port number or the protocol, this approach is a valid option.

A-1.7. Where other approaches differ in defining the Flow Label from the proposed approach?

Few internet drafts have differentiated between the control and Forwarding plane. [draft-ietf-ipv6-flow-label-00.txt] defines the Control plane as part of an IP node taking care of control functions, such as routing protocols and flow establishment protocols and Forwarding plane as part of an IP node receiving and forwarding IP packets; also known as the "datapath". Having a separation of control plane and forwarding plane does have an advantage as explained in that draft. But it may not be completely beneficial as the TCP/IP architecture itself is not fully layered. Moreover, this approach might require some changes in the existing architecture as opposed to the proposed solution given in this draft.

A-1.8. Security Considerations

The specifications of this draft do not raise any new security issues. The Flow Label field in the IPv6 header cannot be encrypted because of the known reasons. If encrypted, each in between router has to decrypt the header for providing the required QoS to the packet. As the QoS specification requires minimum delay for the packet, decrypting each packet's header at each router will not be a good idea because of the time required in processing the packet.

A-1.9. Conclusion

This report has dealt extensively with all the suggested formats for defining the 20-bit IPv6 Flow Label and finally has suggested a hybrid approach for efficiently defining the 20-bit IPv6 Flow Label.

One of the major reasons why the current solution proposed in this Draft provides choice for IntServ/DiffServ based quality of service is the fact that a few representative research experiments in many places including those in Europe (www.bits-pilani.ac.in/ngni) have shown that while DiffServ is definitely an attractive solution due to its scalability, IntServ has been found to be fair and reasonably efficient under a real life situation constraints that were stimulated in these experiments.

In the meanwhile, yet another Quality of Service approach is gradually evolving (Appendix A.3) that aims to provide a seamless application transparency based solution to provide end-to-end quality of service support. Inspired from the initiative in the distributed operating system research and policy-based QoS mechanisms, this approach is still evolving and refined. It is hoped that once this approach becomes

verifiable and viable, an alternate protocol independent quality of service strategy shall be possible to be implemented in the near future.

The emphasis of this work is to result into a practically acceptable specification that could be effectively used for a reasonably long period of time for implementing IPv6 Quality of Service that so far has been elusive in absence of a clear, verifiable and complete specification. A separate ID is under preparation specifically building upon these specifications so as to explicitly address the scalability issues related to the IPv6-Multicast-QoS.

A-1.1-A.1. Characteristics of IPv6 flows and Flow Labels

The characteristics of IPv6 flows and Flow Labels as given in RFC 2460 are rearranged as follows:

- (a). A flow is uniquely identified by the combination of a source address and a non-zero Flow Label.
- (b). Packets that do not belong to a flow carry a Flow Label of zero.
- (c). A Flow Label is assigned to a flow by the Flow's source node.
- (d). New Flow Labels must be chosen (pseudo) randomly and uniformly from the range 1 to FFFFF hex. The purpose of the random allocation is to make any set of bits within the Flow Label field suitable for use as a hash key by routers, for looking up the state associated with the flow.
- (e). All packets belonging to the same flow must be sent with the same source address, destination address, and Flow Label.
- (f). If packets of flow include a Hop-by-Hop options header, then they all must be originated with the same Hop-by-Hop options
- (g). If packets of a flow include a routing header, then they all must be originated with the same contents in all extension headers up to and including the routing header. header contents.
- (h). The maximum's lifetime of any flow-handling state established along a flow's path must be specified as part of the description of the state-establishment mechanism, e.g., the resource reservation protocol or the flow-setup hop-by-hop option.
- (i). The source must not reuse a Flow Label for a new flow within the maximum lifetime of any flow-handling state that might have been established for the prior use of that Flow Label.

A-1.A.2. Comparison of already suggested approaches in defining the IPv6 Flow Label format

This section discusses the already suggested approaches in [draft-conta-ipv6-flow-label-02.txt] for defining the 20-bit Flow Label. It discusses the advantages

and disadvantages of these approaches. Finally it tells about accepting or not preferring these approaches and includes the accepted approaches (with modifications wherever required) in the final definition of the Flow Label discussed in the next section.

A-1.A.2.1 First approach

Following format can be used for the Flow Label:

0	Pseudo – Random Value
---	-----------------------

1	DiffServ IPv6 Flow Label
---	--------------------------

The DiffServ IPv6 Flow Label is a number that is constructed based on the Differentiated services "Per Hop Behavior Identification Code".

1	Per Hop Behavior Ident. Code	Res.
---	------------------------------	------

The "Res" bits are reserved.

The PHB ID is either directly derived from a standard differentiated services code point, or it is an "IANA Assigned Value".

Advantages

Preserves compatibility with the random number method of selecting a Flow Label value defined in IPv6 specification.

Captures the differentiated services treatment intended to be applied to the packet.

Unlike the value of the traffic class field, it is not locally mapped and hence suitable for use in an end-to-end header field.

Disadvantages

It captures less information than the port number and protocol number normally used in multi field classifier.

A-1.A.2.2 Second Approach

DiffServ with multi field classifier can be used in a more efficient and practical manner as an alternative to IntServ and RSVP. The Flow Label classifier is basically a 3-element tuple-source and destination address and IPv6 Flow Label.

The classifier can be defined in any of the following two ways:

$C = (SA, SAPrefix, DA, DAPrefix, Flow\ Label)$.

$C' = (SA, SAPrefix, DA, DAPrefix, Flow\ Label\ min: Flow\ Label\ max)$.

Incoming packet header (SA, DA, Flow Label) is matched with classification rules table entry C or C`.

Advantages

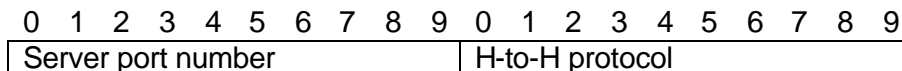
Helps the IPv6 Flow Label to achieve, as it is supposed, in a more efficient processing of packets in QoS engines in IPv6 forwarding devices.

Disadvantages

When packets are transmitted, the end nodes have to force the correct Flow Label in the IPv6 headers of outgoing packets or the first hop routers have to do this job. To accomplish these rules, these routers will be configured with MF classifiers. This puts extra computations to be done by the routers.

A-1.A.2.3 Third approach

Includes the algorithmic mapping of the port numbers and protocol into the Flow Label. It reserves 12 bits for the port number and 8 bits for the protocol.



Advantages

Classification rule is 5 or 6 element tuple format of a DiffServ MF classifier, containing the source and the destination address, the source and the destination ports, the host-to-host protocol. So no new classification rule format is needed.

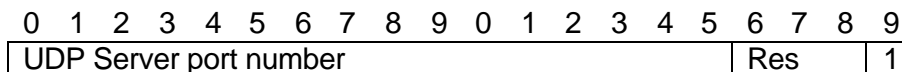
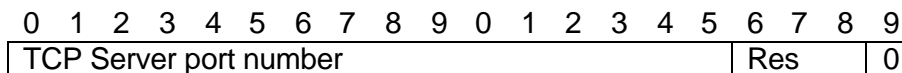
Disadvantages

It cannot differentiate among multiple instances of the same application running on the same two communication end nodes.

The reduced number of bits (12 out of 16) limits the value of ports. 12 bits can represent only the "IANA well-known ports", that is from 1 to 1023 and a subset of "IANA registered ports", that is from 1024 to 4095. Registered ports have values between 1024 and 65535. 1-A.2.4

A-1.A.2.4 Fourth approach

The field occupied by host-to-host protocol could be reduced to 1, as TCP and UDP are the only well known protocols.



The "Res" bits are reserved.

The "TCP Server Port Number" or "UDP Server Port Number" is the 16-bit port number assigned to the server side of the client/server application.

Advantages

Again the classification field is a 5 or 6 element tuple. So no new classification rule is needed.

This approach keeps 16 bits for the port number so that all the "IANA well-known ports" and "IANA registered ports" can be accommodated in these 16 bits.

Disadvantages

This approach, too, cannot differentiate among multiple instances of the same application running on the same two communication end nodes.

Reserving only 1 bit for the protocol field in the Flow Label restricts the use of any protocol other than TCP and UDP.

A-1.A.2.5 Fifth approach

Header length format

Another possible solution is to store the length of IPv6 headers length that is the length of the IPv6 Base Headers and IPv6 extension headers preceding the host-to-host or transport header. The length of IPv6 headers in the Flow Label value would provide the information, which a DiffServ QoS engine classifier could use to locate and fetch the source and destination ports and apply those along with the source and destination address and host-to-host protocol from the Flow Label, to match the source and destination address, the source and destination ports and the protocol identifier elements of a DiffServ MF classifier.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
Length of IPv6 headers										H-to-H protocol									

Advantages

"Length of IPv6 headers" allows skipping the IPv6 headers to access directly the host-by-host header for other purposes. This format is useful for classifying packets that are not TCP or UDP, and have no source and destination ports.

Disadvantages

IPv6 header does not include "Total Headers Length" field. So introducing this new field in the Flow Label puts extra computation to be done that may result in the processing delays. Including "Length of IPv6 headers" in the Flow Label does not carry any significance in case ESP is used for IP Security.

This approach is not preferred because of the reasons given above. Again, it does not carry any direct advantage in keeping the "Length of IPv6 headers" in the Flow Label.

A-1.A.3. Recent works in progress

An emerging packet switched QoS approach for providing end-to-end quality of service transparent to the application programs is in the verge of becoming a realistic solution for the IPv6 based WAN-QoS requirements. Known as MultServ, this approach finds its inspiration from the initiatives and the results of the distributed operating system research. Some fundamental initial work has been done by the IPv6-QoS research group at the Center for Software Development, BITS, Pilani (India).(<http://ipv6.bits-pilani.ac.in/ngni/NGNI-MMI-QoS-D4-v1.3-secure.pdf>). It is expected that an IETF document shall soon be submitted to the QoS community for their inputs and review of the emergent approach.

A-1.A.4. QoS through policy based protocol implementation

For quite sometime now, an interesting and promising approach that is generic in nature has been suggested and even implemented in parts in terms of quality of service. This approach called policy based control protocol has already one standardized protocol known as Common Open Policy Service (COPS). COPS implementation has been available in several newer routers. This policy based quality of service framework permits the network administrators to define QoS Policies that explicitly define rules pertaining to handling aggregated flows at a network node known as the Policy Enforcement Point (PEP). The policy servers known as the Policy Decision Point (PDP) computes or determine the exact QoS enforcement action to be taken on the policy-classified packets to be executed at the PEPs. Although very useful, this approach exhibits certain basic flaws. For instance, PDPs could be the point of failures and building redundancy by providing more PDPs may lead to network degradation (due to possible overheads and synchronization issues) unless it is very carefully designed. [Qos_pol113]

Actually this policy based QoS solution augments the DiffServ approach, since in this case the PDPs are expected to map the flow information to specific DiffServ traffic conditioning action meta data which is communicated back to PEP; which thereafter uses this information for future processing. However this approach has one advantage that qualifies for an honourable slot in the QoS strategies and that is because such a mechanism does not require the application themselves to be QoS aware. This also happens to be the strong point of the MultServ approach, but it does not operate on the client-server methodology.

The Quality of Service has one aspect called C&A (Charging and Accounting) which the commercial providers of the service require to support in case they have to charge their customers on the basis of QoS requirements. As of now, most of these service providers either do not provide QoS or provide certain flat tariff rates based on the explicit choices made by the customers that requires the customers to be QoS aware. All this is due to the fact that there is no C&A provision in the majority of the proposed mechanisms pertaining to QoS.

The management of the QoS capable networks (QoS WANs) is yet another area that has not been adequately addressed by most of the existing proposed QoS mechanisms (with or without IPv6). The key problem here is that since the routers do offer a variety of packet handling mechanisms, the operator has to specifically select and combine the required traffic conditioning components at the Edge Routers and even at the Core Routers at the service provider's end. Although the aggregated end-to-end flow can be implemented in such cases, the task to define the exact router configuration remains an increasing complex job particularly in wide area heterogeneous networks. A related issue is scalability of management of such QoS-capable networks.

The abovementioned issues are the two areas that are specifically being attempted to be addressed as built-in features of the MultServ quality of service mechanism, which may eventually be implemented in IPv6 WANs and which will not require any major change in the basic protocol itself.

Acknowledgements

Authors acknowledge technical inputs and support from the members of the "Project IPv6@BITS" as well as the graduate students registered in EA C451 Internetworking Technology course at the Birla Institute of Technology & Science, Pilani, India, Dr. Latif Ladid of Ericsson Telebit, (Luxembourg); Dr. Torsten Braun of University of Bern (Switzerland); Dr. Pascal Lorenz of I.U.T. at the University of Haute Alsace, Colmar (France); Dr. S. Rao of Telscom A.G. (Switzerland); Dr. Bernardo Martinez of Versaware Inc. (Spain); Dr. Juan Quemada of UPM, Madrid (Spain); Dr. Merce and Dr. Paulo Desousa at the EC; Dr. Zoubir Mammeri of IRIT (France) and Dr. Brian Carpenter of IBM. The IPv6-QoS team wishes to explicitly acknowledge the support from Dr. S.Venkateswaran of BITS, Pilani (India).

Authors gratefully acknowledge the works of many dedicated brains at the IETF, ETSI and elsewhere, sections or extracts of which have helped us to shape this document.

References

- [RFC 2460] S. Deering and Bob Hinden, "The Internet Protocol Specification", RFC 2460, Internet Protocol version 6 Specification.
- [RFC 1809] C. Partridge, RFC 1809, "Using the Flow Label Field in IPv6".
- [RFC 2676] RFC 2676, QoS Routing Mechanisms and OSPF Extensions.
- [RFC 1633] RFC 1633, Integrated Services in the Internet Architecture: an overview.
- [RFC 2475] RFC 2475, An Architecture for Differentiated Services.
- [RFC 2676] RFC 2676, QoS Routing Mechanisms and OSPF Extensions.
- [Qos_pol113] QoS Forum: "Whitepaper in QoS Policy", available at the URL: http://www.gt-er.cg.org.br/sgt-qos/documents/qospol_v11.pdf

References to the works in progress

[draft-banerjee-ipv6-quality-service-02.txt] Rahul Banerjee, N.Preethi, M. Sethuraman, "Design and Implementation of the Quality-of-Service in IPv6 using the modified Hop-by-Hop Extension header - A Practicable Mechanism".

[draft-conta-ipv6-flow-label-02.txt] A. Conta, B. Carpenter, "A proposal for the IPv6 Flow Label".

[draft-rajahalm-ipv6-flow-label-00.txt] J. Rajahalm, A. Conta, "An IPv6 Flow Label Specification".

[draft-banerjee-flowlabel-ipv6-qos-02.txt] Rahul Banerjee, Sumeshwar Paul Malhotra, Mahaveer M, "A Modified Specification for use of the IPv6 Flow Label for providing an efficient Quality of Service using a hybrid approach".

[draft-jagadeesan-rad-approach-service-01.txt] Harshavardhan Jagadeesan, Tuhina Singh, "A Radical Approach in providing Quality-of-Service over the Internet using the 20-bit IPv6 Flow Label field".

[NGNI-MMI-QoS: D1] Rahul Banerjee (BITS), Juan Quemada (UPM), P. Lorenz (UHA), Torsten Braun (UoB), Bernardo Martinez (Versaware): "Use of Various Parameters for Attaining QoS in IPv6-based Multimedia Internetworks", Feb. 2002 readily available at the URL: <http://ipv6.bits-pilani.ac.in/ngni/>.

[NGNI-MMI-QoS: D3] Rahul Banerjee (BITS), Juan Quemada (UPM), P. Lorenz (UHA), Torsten Braun (UoB), Bernardo Martinez (Versaware): "Quality of Service Directions, Bench Marking and Roadmaps for IPv6 Oriented NGN Multimedia Internetworks". <http://ipv6.bits-pilani.ac.in/ngni/>.

[NGNI-MMI-QoS: D4] Rahul Banerjee (BITS), Juan Quemada (UPM), P. Lorenz (UHA), Torsten Braun (UoB), Bernardo Martinez (Versaware): <http://ipv6.bits-pilani.ac.in/ngni/NGNI-MMI-QoS-D4-v1.3-secure.pdf>

Disclaimer

The views and specification here are those of the authors and are not necessarily those of their employers. The authors and their employers specifically disclaim responsibility for any problems arising from correct or incorrect implementation or use of this specification.

Authors Information

Rahul Banerjee / Sumeshwar Paul Malhotra / Mahaveer M
3256, Center for Software Development
BITS, Pilani – 333031, Rajasthan, India.
Phone: +91-159-7645073 Ext. 335
Email: rahul@bits-pilani.ac.in

Full Copyright Statement

Copyright (C) The Internet Society (2002). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Appendix-2

Extracts from a Suggested Modified Specification of the IPv6 Hop-by-Hop Header ID Published by the IETF

IPv6 Working Group
Internet Draft
draft-banerjee-ipv6-quality-service-02.txt

Rahul Banerjee
N. Preethi
M. Sethuraman
BITS, Pilani (India)
March 2002

Design and Implementation of the Quality-of-Service in IPv6 using the modified Hop-by-Hop Extension header -A Practicable Mechanism

Status of This Memo

This document is an Internet Draft and is subject to all provisions of Section 10 of RFC 2026. Internet Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet Drafts.

Internet Drafts are draft documents valid for a maximum of 6 months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet Drafts as reference material or to cite them other than as a "work in progress".

The list of current Internet Drafts can be accessed at <http://www.ietf.org/lid-abstracts.html>

The list of Internet Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>

Copyright (C) The Internet Society (2002). All Rights Reserved.

Abstract

This paper proposes a practicable solution to the QoS implementation in IPv6, the design of which uses the Hop-by-Hop Extension header and not the 20-bit flow label field in the IPv6 Base Header. This paper deals extensively with Integrated Services type of QoS model (like the one supported by RSVP) and gives the definition of the important TLV options that will be needed to specify the Type of QoS and the corresponding resource requirements in the Hop-by-Hop Extension Header. This design can also support the Differentiated Services type of QoS model, which has been dealt in brief. The work also elaborates on the data structures that will be required at the routers and provides the algorithm that the source and the router should follow while trying to implement this design.

A-2.1. Introduction

This paper suggests a possible design as well as gives an overview of the implementation details of Quality of Service (QoS) in IPv6. Though the IPv6 Base Header has a 20-bit flow label field for QoS implementation purposes, it has not yet been exploited. This work explores the possibility of using the hop-by-hop extension header for implementing QoS at the IPv6-layer. This design (mechanism included) is based on the Integrated Services model and can also act as an effective transitional solution till the specification to use the 20-bit flow label field in the IPv6 base header is developed acceptably.

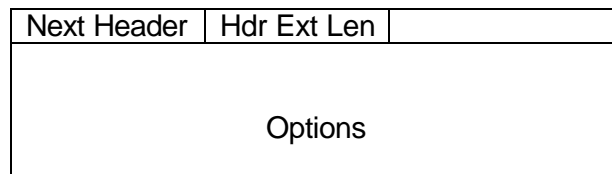
A-2.2. Motivation for using the hop-by-hop extension header implementing QoS

To implement any model of QoS, all the routers en-route have to be requested for the particular resources required and it is important that they give their consent on the same. The hop-by-hop extension header is one that will be processed by all the routers en-route to the destination. So all the routers in the path will see any information that is embedded in this header.

The TLV options in the hop-by-hop extension header have not yet been fully exploited. By exploiting those options to our convenience, it is possible to specify the requisite information for each flow (i.e. the type and the resources required) to all the intermediate routers. The individual routers can send appropriate messages to the source if it cannot meet the resource requirements.

A-2.3. The Hop-by-Hop Extension header

According to RFC 2460 - the formal specification for IPv6, the Hop-by-Hop Extension Header is used to carry optional information that must be examined by every node along a packet's delivery path. It is identified by a Next Header value of 0 (Zero) in the IPv6 header, and has the following format:



Next Header: It's an 8-bit field that identifies the type of header immediately following the Hop-by-Hop Options header. Hdr Ext Len: It's an 8-bit unsigned integer field, which tells the length of the Hop-by-Hop Options header in 8-octet units, not including the first 8 octets.

Options: It's a variable-length field, of length such that the complete Hop-by-Hop Options header is an integer multiple of 8 octets long.

A-2.4. Type - length - value (TLV) options

A-2.4.1 Introduction

The hop-by-hop options header can carry a variable number of TLV encoded "options", of the following format [RFC 2460]:

Option Type	Opt Data Len	Option Data
-------------	--------------	-------------

Option Type : 8-bit identifier of the type of option.

Opt Data Len : 8-bit unsigned integer. Length of the Option Data field of this option, in octets.

Option Data : Variable-length field. Option-Type-specific data.

The Option Type identifiers as defined in RFC 2460 are internally encoded such that their highest-order two bits specify the action that must be taken if the processing IPv6 node does not recognize the Option Type.

The third-highest-order bit of the Option Type specifies whether or not the Option Data of that option can change en-route to the packet's final destination. A full 8-bit Option Type, not just the low-order 5 bits of an Option Type, identifies a particular option.

A-2.4.2 The Already defined TLV options

The only hop-by-hop options defined in RFC 2460 (IPv6 Specification) are the Pad1 and PadN options specified as follows:

A-2.4.2.1 Pad1 option

0

The Pad1 option is used to insert one octet of padding into the Options area of a header [RFC 2460]. It does not have length and value fields.

A-2.4.2.2 PadN option

1	Opt Data Len	Option Data
---	--------------	-------------

The PadN option is used to insert two or more octets of padding into the Options area of a header. For N octets of padding, the Opt Data Len field contains the value N-2, and the Option Data consists of N-2 zero-valued octets. [RFC 2460]

A-2.4.2.3 The router alert option

This option has been defined in RFC 2711 and has the following format:

Type	Length = 2	Value
0 0 0 0 0 1	0 1 0 0 0 0 0 0 0 0 1 0	Value (2 octets)

The first three bits of the first byte are zero and the value 5 in the remaining five bits is the Hop-by-Hop Option Type number. By zeroing all three, this specification requires that, nodes not recognizing this option type should skip over this option and continue processing the header and that the option must not change en-route.

The above 3 are the options that have been defined in RFCs. The rest of the values for the option type of the hop-by-hop options header haven't been defined yet. [RFC 2711]

A-2.5. Using the TLV options to implement QoS

This design hopes to exploit the remaining non-defined and possible values of the option type in the Hop-by-Hop options header, (after leaving some values for future use) to indicate some important QoS types.

A-2.5.1 QoS Models and their representation in the options field

Since this work focuses to provide a complementary mechanism for providing QoS-support (by complementing the 20-bit flow control field in the IPv6 base header), it deals with a Integrated Services (IntServ) model like that supported by RSVP [Paul et al.], wherein each and every flow needs to specify its TYPE and the RESOURCES that it needs en-route. (This design can also support the Differentiated Services (DiffServ) model of QoS, in which, each flow is aggregated to a particular class of traffic. This design can then act as a substitute for the concept behind the Traffic Class bits (8-bit field in the IPv6 base header.)

The source tells the routers that it is using the Integrated Services model by setting the nineteenth bit of the first 32 bits.

0	8	16	Type	24	Length
Next Header	Hdr Ext Len	0	0	0	1
		0	0	0	0
Options data					

The Differentiated Services (DiffServ) feature can be mentioned by setting the twentieth bit of the first 32 bits.

0	8	16	Type	24	Length
Next Header	Hdr Ext Len	0	0	0	0
		0	0	1	0
		0	0	0	0
Options data					

This report deals with the IntServ model only and only indicates use of the DiffServ model.

A-2.5.2 The IntServ Model

The two main Types of flows in the IntServ model are [Paul et al]
 Guaranteed flow service
 Controlled Load Service

The last three bits of the Type field i.e. the bits numbered 21, 22,23 are used to represent one of these types.

0	8	16	Type	24	Length
Next Header	Hdr Ext Len	0	0	0	0
Options data					

There are a total of 8 possible combinations of which the IntServ model uses two. The rest can be can be exploited by the DiffServ model and for future use.

A-2.5.2.1 The QoS Identifier

This is an 8-bit identifier and occupies the first byte in the options data field as shown in the figure below. There might be many applications from the same source wherein each one has its own flow specifications. So there arises a need to uniquely identify each such flow. The QoS identifier does this job. A particular source can establish a maximum of 256 connections that need QoS guarantee.

0	8	16	Type	24	Length
Next Header	Hdr Ext Len	0	0	0	0
QoS Identifier					

A-2.5.2.2 Resource Identifier

This is a 4-bit identifier that specifies the type of the resource needed by a particular flow. The different types of resources needed are indicated using these identifiers in a list. This list follows the QoS Identifier in the option data field, which in turn is followed by a list of 32 bit values that specify the amount of resource required for each of the resource types. Some of the identified resource types are:

0000 - End of List Identifier

This is a special identifier that specifies the end of the resource-required list (brief explanation in section 5.2.3).

0001 - Constant Data Transfer Rate

This identifies the Constant Bandwidth required and the value is given in a 32-bit field specified in Kbps (Kilo bits per second). (Max value = 512 GBps)

0010 - Average Data Transfer Rate

This identifies the Average Bandwidth required and the value is given in a 32-bit field in Kbps (Kilo bits per second).

0011 - Maximum Data Transfer Rate

This identifies the Maximum Bandwidth required and the value is given in a 32-bit field specified in Kilobits per second (Kbps).

0100 - Minimum Delay Requirement

This identifies the Minimum Delay that the application demands and the required value is given in a 32-bit field specified in nanoseconds. (Max value = 4.3 sec)

0101 - Average Delay Requirement

This identifies the Average end-to-end delay that the application can tolerate and the value is given in a 32-bit field specified in nanoseconds.

0110 - Buffer Requirement

This identifies the Buffer Requirement by the flow at each router and the amount required is expressed as a 32-bit quantity specified in bytes. (Max value = 4 GB)

A-2.5.2.3 Resources Required List

The Type of flow (Guaranteed/Controlled Load, briefly explained in section 5.3) is specified in the option type bits of the Hop-by-Hop Extension header. The resources needed by this flow at each router are specified in the bits following the 8-bit QoS identifier in the options data field. The resource identifiers (4 bits each) are specified one after the other and the list ends with the 0000 End of List Identifier (as mentioned above). The corresponding amount of resource required (a 32 bit quantity only) for all the resource types listed is specified in the same order as that of the resource types, starting from the next aligned 32 bits.

A-2.5.3 The Different TYPES OF FLOW in the IntServ Model

A-2.5.3.1 Guaranteed Flow Service

This service is meant for RTI (Real Time Intolerant) or hard Real Time applications, which demand minimal latency and jitter. For example, consider a multicast real time application (video conferencing). Delay is unacceptable and ends should be brought as close as possible. [Paul et al] The whole application should simulate each person talking face to face.

For this case, the required resource reservations are

- a. Constant bandwidth for the application traffic
- b. Deterministic Minimum delay that can be tolerated.

These types of applications can decrease delay by increasing demands for bandwidth. A further explanation is given in Appendix A.

A-2.5.3.2 Controlled Load Service

This service is meant for RTT (Real Time Tolerant) or soft Real Time applications, which have an average bandwidth requirement and an indeterminate end-to-end delay for an arbitrary packet. [Paul et al] These RTI applications demand weak bounds on the maximum delay over the network. Occasional packet loss is acceptable. For example, consider video applications which use buffering.

The required resource reservations can be

- a. Average bandwidth for the application traffic
- b. Buffer requirement at each relevant intermediate router

A further explanation is provided in Appendix A.

A-2.5.4 Overview of some important facts.

There are two Types of Flows (Guaranteed/Controlled Load). Under each type, the Resource Requirement List can vary for each and every application that needs QoS. The application has to specify the following.

- Whether it requires Guaranteed/Controlled Load treatment.
- The List of Resources it requires.
- The required amount of these resources.

For applications that do not need QoS, it can specify the No Flow Control option defined as defined in the next section.

A-2.5.5 No Flow Management

This type indicates that the source requires no QoS and will be content with a 'best effort' treatment.

A-2.5.5.1 Option Definition

The option type is defined as

0	8	16	Type	24 Length														
Next Header	Hdr Ext Len	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

The value of 0 in the least significant 5 bits numbered 19,20,21,22,23 signifies the type for No QoS required at the intermediate routers. The numeric decimal value specifying this type is 0. But it is different from the Pad1 option in the following way. The Pad1 option doesn't have a length and data field. But the No flow control option has a value of 1 in the length field and no data field.

A-2.6. At the Router

Any router that tries to implement QoS maintains a QoS routing table and keeps track of the QoS available to each destination through the required number of hops. [RFC 2676]. Apart from this table, the router needs to keep track of the allotted QoS to each and every flow. This table is the AllottedQoS table.

A-2.6.1 The AllottedQoS table

It has the following entries:

1. Source address
2. QoS identifier for that particular flow from the source.
1. Information regarding whether it is the IntServ Model or the Diffserv Model.

```
enum MODEL_ID {  
    INTSERV=0, // the IntServ Model  
    DIFFSERV=1 // the DiffServ Model  
};
```

2. List of resources allotted to that entry (i.e.) an array of values like the following.

```
struct RESOURCE_ALLOCATED {
    short int Res_identifier; //the 4 bit identifier of the resource
    int Res_allocated; //the 32 bit value of the allocated resource
};
```

A-2.6.2 Resource Required List

The list of resources will be an array of pointers to the structure RESOURCE_ALLOCATED as declared below. Struct RESOURCE_ALLOCATED *res_allocated[MAX]; This array will be maintained for each source address. The QoS Identifier will be the array subscript for each source. The pointer value stored acts as the head of the list of the resources allotted for that particular QoS identifier.

A-2.6.3 Defining the different Resource Identifiers

```
enum RES_ID{
    ENDOFLIST =0, // End of List Identifier
    CONSTBW   =1, // Constant Data Transfer Rate
    AVBW      =2, // Average Data Transfer Rate
    MAXBW     =3, // Maximum Data Transfer Rate
    MINDELAY  =4, // Minimum Delay Requirement

    AVDELAY   =5, // Average Delay Requirement
    BUFFREQ   =6 // Buffer Requirement
};
```

A-2.6.4 Template for the AllottedQos table entry

```
#define MAX 256 //maximum of 256 QoS Ids for every source
typedef struct {
    struct sockaddr_in6 *srcaddr; //the source IPv6 address
    struct RESOURCE_ALLOCATED *res_allocated[MAX]; //a pointer which
//acts as the head for each of the lists i.e. for each of the
//0..MAX QoS Identifiers for the particular source address.
    MODEL_ID model; // IntServ or DiffServ
}ALLOTTEDQOS_TABLE;
```

A-2.7. Overview of the whole design.

This section describes the whole process by taking an example. Consider any application (like Videoconferencing or Video/Audio on Demand) that needs some specified QoS.

A-2.7.1 Function of the Source

It gets a unique QoS Identifier for that particular flow and fills it in the Hop-by-Hop header. Next, it specifies the IntServ model by setting the appropriate bit. The source application then fills in the resource-required list and the corresponding 32

bit values (the amount of each resource needed) in the options data part of the Hop-by-Hop header. Finally, this packet is put on the network and it reaches the intermediate routers.

A-2.7.2 Function of each relevant intermediate router

A-2.7.2.1 Initial Processing

It gets the option type value from the header. Checks if its the default (no QoS required) which is indicated by a value of all bits being 0 in the 5 bits numbered 19,20,21,22,23. If it is not the default QoS, it gets the QoS identifier from the first byte of the options data field.

A-2.7.2.2 Searching for the entry

1. The ALLOTTED_QOS table is searched based on the source address.
2. If an entry is found, then for that particular source, a search is made based on the QoS Identifier got during the Initial Processing stage. (the array index for the res_allocated structure is the corresponding QoS Identifier and this pointer is NULL if its a new entry).
3. If the entry already exists, the IPv6 packet is processed so that the reserved QoS is met.
4. If the entry is not found, a new entry is made in the ALLOTTED_QOS table for the source and the QoS Identifier and further processing of this new entry is done as follows.

A-2.7.2.3 New Entry

1. The router now checks if it is the IntServ Model or the Diffserv Model by checking the appropriate bits in the options type field and stores this information in the model variable of type MODEL_ID in the ALLOTTED_QOS table.
2. The router then gets the Resources required list and their corresponding values from the options data field and updates the res_allocated array structure.
3. It then checks with the QoS Routing table, to find out if this reservation is possible. If yes, it updates the new entry in the ALLOTTED_QOS table in the memory or else this entry is removed.
4. If any relevant router en-route is not able to guarantee the requested QoS, an ICMPv6 message is sent to the source and the other routers (that had guaranteed the QoS) are also notified of the same so that they delete the corresponding entry from their QoS tables.

This process repeats at all the intermediate routers between the source and the destination.

A-2.8. Security Considerations

The specifications of this draft don't raise any new security issues as hop-by-hop extension header is used in this draft, which according to RFC 2460, can not be encrypted due to the possibility of increasing the overhead in the router's processing these headers. If encrypted, each intermediate router has to decrypt the header for

providing the required QoS to the packet. As the QoS specification requires minimum delay for the packet, decrypting each packet's header at each router will not be a good idea because of the time required for that packet to be processed.

A-2.9. Conclusion

This work has dealt extensively with the design of the Integrated Services model of Quality of Service in IPv6 using the Hop-by-Hop Extensions Header. It is being suggested initially as a transitional mechanism / solution although it has a definite potential to qualify as an effective QoS support measure.

A-2.2-Appendix A. Examples

A-2.2-A.1 Guaranteed Flow Service Example

The example of a multi-party videoconferencing cited in section 5, which is a Guaranteed Type of Service, can be defined in the following way.

0	8	16	Type	24	Length
Next header	Hdr Ext Len		1 0 0	1 0 0 1 0	
QoS Identifier	0 0 0 1	0 1 0 0	0 0 0		
32-bit value – constant bandwidth in kbps					
32 bit value – min delay in nanoseconds					

Explanation

The first 3 bits numbered 16,17,18 being 1,0,0 say that if the router is not able to recognize the option type, it should discard the packet and, regardless of whether or not the packet's Destination Address was a multicast address, send an ICMP Parameter Problem, Code 2, message to the packet's Source Address, pointing to the unrecognised Option Type and the value of the option data field should not be changed en route by any routers [RFC 2460].

The value of 18 in the 5 bits numbered 19,20,21,22,23 defines this QoS type of IntServ and Guaranteed Service. The numeric decimal value specifying this type is 146.

The Resource Required List and its Specification

- Constant Bandwidth Requirement: The bit value of 0001 after the QoS identifier is the identifier for this and the first 32-bit value gives the amount of bandwidth in kbps to be reserved.
- Minimum delay Requirement: The deterministic minimal delay in nanoseconds. The identifier is 0100 and the second 32-bit value corresponds to this.

The 0000 identifier ends this list.

Examples

Interactive applications like Videoconferencing/Audio Conferencing or other real time applications.

A-2.2-A.2 Controlled Load Service Example

The example of a video application cited in section 5, which is Controlled Load Service, can be defined in the following way.

0	8	16	Type	24	Length
Next header	Hdr Ext Len		1 0 0	1 0 0 1 1	
QoS Identifier	0 0 1 0 0 1 1 0	0 0 0	0		
32-bit value – average bandwidth in kbps					
32 bit value – buffer req. in bytes					

Explanation

The first 3 bits numbered 16,17,18 being 1,0,0 say that if the router is not able to recognize the option type, it should discard the packet and, regardless of whether or not the packet's Destination Address was a multicast address, send an ICMP Parameter Problem, Code 2, message to the packet's Source Address, pointing to the unrecognized Option Type and the value of the option data field should not be changed en route by any routers [RFC 2460].

The value of 19 in the 5 bits numbered 19,20,21,22,23 defines this QoS type of IntServ and Controlled Load Service. The numeric decimal value specifying this type is 147.

The Resource Required List and its Specification.

- a. Average Bandwidth Requirement: The bit value of 0010 after the QoS identifier is the identifier for this and the first 32-bit value gives the required value in kbps.
- b. Buffer Requirement: The bit value of 0110 following the Average Bandwidth Resource type is the identifier for this and the second 32 bit value gives the number of bytes to be reserved.

This list is ended by the 0000 identifier.

Examples

Video/Audio applications that require buffering involving video/audio.

Acknowledgements

Authors acknowledge technical inputs and support from the members of the "Project IPv6@BITS" especially Sumeshwar Paul Malhotra and Mahaveer M. at the Birla Institute of Technology Science, Pilani, India, Dr. Latif Ladid of Ericsson Telebit, (Luxembourg); Dr. Torstern Braun of University of Bern (Switzerland); Dr. Pascal Lorenz of I.U.T. at the University of Haute Alsace, Colmar (France); Dr. Sathya Rao of Telscom A.G. (Switzerland); Dr. Bernardo Martinez of Versaware Inc. (Spain); Dr. Juan Quemada of UPM, Madrid (Spain); Dr. Merce G-Fisa and Dr. Paulo D'Sousa at the EC, Dr. Glenn Morrow of Nortel, Dr. Pekka Savola of CSC/FUNET and Prof. Zoubir Mammeri of IRIT (France).

References

- [RFC 2460] RFC 2460, Internet Protocol version 6 Specification.
[RFC 2711] RFC 2711, IPv6 Router Alert Option.
[Paul et al] QoS in Data Networks, Protocols and Standards by Arindam Paul.
[RFC 2676] RFC 2676, QoS Routing Mechanisms and OSPF Extensions.
[NGNI-MMI-QoS: D2] Rahul Banerjee (BITS), Juan Quemada (UPM), P Lorenz (UHA), Torsten Braun (UoB), Bernardo Martinez (Versaware): "Use of Various Parameters for Attaining QoS in IPv6-based Multimedia Internetworks", Jan. 15, 2002 available at <http://ipv6.bits-pilani.ac.in/ngni/> and <http://www.ngni.org/>.

Disclaimer

The views and specification here are those of the authors and are not necessarily those of their employers. The authors and their employers specifically disclaim responsibility for any problems arising from correct or incorrect implementation or use of this specification.

Author Information

Rahul Banerjee / Preethi N. / M. Sethuraman
3256, Centre for Software Development
BITS, Pilani - 333031
Rajasthan, India.

Phone: +91-159-7645073 Ext. 335
Email: rahul@bits-pilani.ac.in

Full Copyright Statement

Copyright (C) The Internet Society (2002). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are

included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an

"AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Appendix-3
A Quick-View Chart of Select Internetworking Research & Development Initiatives Around the World

Project	Organization	Areas of Research & Development
Project WIDE	(Japan)	IPv6 deployment and testing
Mobile IPv6, VoD, LANDMARC (LU-Microsoft collaboration)	Lancaster University, (UK)	IPv6-based distributed multimedia over fixed and mobile networks
Project KAME	Kame.org	Reference implementation of IPv6 and IPSec
IPv6 Test bed	Telecom Lab. Italia (Italy)	IPv6, MPEG-x
IPv6 Test bed	IRISA (France)	IPv6 Testing
Next Generation Networks Initiative	EC's NGNI (International)	IPv6, Optical and Mobile Technologies
Next Generation Networks	NGN (USA)	Next generation multi-technology areas
ISABEL	UPM, Madrid (Spain)	Video-on-Demand with
Digital Library Initiative	NSF's DLI (USA)	Digital Libraries
6-Bone	6bone.org	IPv6 Backbone and Test-bed
Internet-2	Internet.edu (basically USA with collaborators from around the globe)	High-speed next generation Internetworking
M-Bone	Mbone (International)	Multicast Backbone and Test-bed
Project BITS-MOS	BITS, Pilani (India)	Distributed Multimedia O.S
Project BITS-WearComp	BITS, Pilani (India)	Wireless IPv6-Bluetooth-UMTS combination-based Wearable Computing System
Project d-Lib	BITS, Pilani (India)	Distributed Digital Library
Project IPv6@BITS	BITS, Pilani (India)	IPv6, Video-on-Demand, Desktop Videoconferencing, Internetwork Management and Performance Analysis, Architecture Neutral Protocol Stack for Heterogeneous Systems
Numerous	Cisco Systems (USA)	IPv6 Routers, Embedded OS with IPv6 support, QoS research, MPLS

A large number of research and development initiatives are going on in Asia, Europe and North America in the area of next generation internetworking based distributed media delivery.

Appendix-4 Bibliography

A. Amandi and A. Price: **Object Agent Programming through Brainstorm System**, Proceedings of the PAAM '97 Conference, London, April 1997.

A. S. Tanenbaum: **Computer Networks**, Fourth Edition, Prentice-Hall, Upper Saddle River, 2002.

B. A. Forouzan & C. H. Fegan: **TCP/IP Protocol Suite**, Second Edition, Tata-McGraw-Hill Publishing Co. Ltd., New Delhi, 2002.

B. Grosz: **Building Commercial Agents: An IBM Perspective**, Invited Paper, PAAM '97 Conference, 1997.

B. Hayes-Roth: **Architecture for Adaptive Intelligent Agents**, Artificial Intelligence, Vol. 72, No., 1995, pp. 329-365.

B. O. Szuprowicz: **Multimedia Networking**, McGraw-Hill, New York, 1995.

B. Pell: **Agent Architectures for Autonomous Control Systems**, Tutorial at the PAAM '97 Conference, London, 1997.

B. Quendt: **An Agent-Based Resource Control of the Signaling System for the Open Telecommunication Market**, Proceedings of the PAAM '97 Conference, April 1997.

Bruce Schatz & H. Chen: **Building Large-Scale Digital Libraries**, IEEE Computer, May 1996, available at the URL: <http://computer.org/computer/dli/index.html>.

Bruce Schatz et al: **Building the Interspace**, 1996, available at the URL: <http://csl.ncsa.uiuc.edu/interspace.html>.

Bruce Schatz et al: **Federating Diverse Collections of Scientific Literature**, IEEE *Computer*, May 1996, pp. 28-36.

C. Huitema: **IPv6**, Second Edition, Prentice-Hall PTR, Englewood Cliffs, NJ, 1998.

C. Leckie et al: **A Multi-Agent System for Distributed Fault Diagnosis**, Proceedings of the PAAM '97 Conference, London, April 1997.

Cisco staff: **Internetwork Design Guide**, 1999 available at: <http://www.Cisco.com/univercd/cc/td/doc/cisintwk/idg4/nd2002.htm#xtocid229276>

Cisco staff: **Internetwork Design Guide**, Cisco Press / Techmedia, New Delhi, 1999.

Cisco staff: **Internetworking Case Studies**, Cisco Press / Techmedia, New Delhi, 1996.

Communications of the ACM, Special Issue on Digital Libraries, April 1995.

Cormac Long: **IP Network Design**, Osborne-McGraw-Hill, Berkeley, 2001.

D. Comer & D. L. Stevens: **Internetworking with TCP /IP**, Vols. 2-3, Prentice-Hall of India, New Delhi, 2000.

D. Comer: **Internetworking with TCP / IP**, Vol. -1, Fourth Edition, Pearson Education, New Delhi, 2001.

D. Martin et al: **Information Brokering in an Agent Architecture**, Proceedings of the PAAM '97 Conference, London, April 1997.

D. Pinnard, M. Wiess and T. Gray: **Issues in Using an Agent Framework for Converged Voice / Data Applications**, Proceedings of the PAAM '97 Conference, London, April 1997.

D. Santa Cruz, T. Ebrahimi, J. Askelof, M. Larsson and C. Christopoulos: **Coding of Still Pictures: JBIG and JPEG**, ISO/IEC JTC1/SC29/WG1 (ITU-T SG-8), N1816, July 2000. (Status: Informational document) This is based on two papers:

D. Santa Cruz, T. Ebrahimi, J. Askelof, M. Larsson and C. Christopoulos: **JPEG 2000 Still Image Coding versus Other Standards**, Proceedings of the SPIE. Vol. 4115.

D. Santa Cruz, T. Ebrahimi: **An Analytical Study of JPEG 2000 Functionalities**.

Dave Koiur: **IP Multicasting: The Complete Guide to Interactive Corporate Networks**, John Wiley & Sons, New York, 1998.

DLI Staff: **Home page of the DLI National Synchronization Effort**, available at the URL: <http://www.grainger.uiuc.edu/dli/national.htm>.

DLI Staff: **The Digital Library Forum home page**, accessible at the URL: <http://www.dlib.org/>.

Donald Steiner: **Issues in Agent Interaction**, Invited Paper, Proceedings of the PAAM '97 Conference, London, April 1997.

Dr. Tim Berners Lee: **The World Wide Web: A very short personal history**, available at the URL: <http://www.w3.org/People/Berners-Lee/ShortHistory.html>

G. Taubes, "Indexing the Internet," *Science*, Sept. 8, 1995, pp. 1,354-1,356.

Garry R. McClain (Ed.): **Handbook of Networking and Connectivity**, AP Professional, 1994.

Gilbert Held: **Data and Image Compression**, 4th Edition, John Wiley and Sons, Inc. 1996.

Grady N. Drew: **Using SET for Electronic Commerce**, Prentice-Hall PTR, 1998.

H. Baumgartel, S. Bussmann and M. Kolsterberg: **Multi-Agent Coordination of Material Flow in a Car Plant**₁, Proceedings of the PAAM '97 Conference, London, April 1997.

H. Chen, **Collaborative Systems: Solving the Vocabulary Problem**, IEEE *Computer*, May 1994, pp. 58-66.

H. Ghosh and S. Chaudhury: **An Abductive Framework for Retrieval of Multimedia Documents in a Distributed Environment**, Proceedings of the KBCS '98 International Conference, NCST, Bombay, Dec. 1998, pp. 153-165.

IEEE Internet Computing, Special Issue on Digital Libraries, April 1998.

IITA Staff: **Interoperability, Scaling, and the Digital Library Research Agenda**, IITA report, 1995, available at the URL: <http://www-diglib.stanford.edu/diglib/pub/reports/iita-dlw/main.html>.

ISO/IEC: **Information Technology – Coded Representation of Picture and Audio Information – Progressive Bi-Level Image Compression**, May 1993.

ISO/IEC: **Information Technology – Coding of Audio-Visual Objects, Part-2: Visual**, Dec. 1999.

ISO/IEC: **Information Technology – Lossless and Near-Lossless Compression of Continuous-Tone Still Images: Baseline**, Dec. 1999.

ISO/IEC: **JPEG 2000 Image Coding System: Core Coding System**, WG 1 N 1646, March 2000, available at the URL: <http://www.jpeg.org/FCD15444-1.htm>.

J. Arthursson et al: **A Platform for Secure Mobile Agents**₁, Proceedings of the PAAM '97 Conference, London, April 1997.

J. Bradshaw: **Software Agents: The Next Generation**₁, Tutorial at the PAAM '97 Conference, London, 1997.

J. Dospisil, E. Kendall and T. Polgar: **Multimedia Presentation Scheduling with ILOG**₁, PAP '97 Conference, London, April 1997.

J. E. Whatley and P. J. A. Scown: **Simultaneous Multiple Agents Working in Real-Time: From Interaction Framework to Prolog**, PAP '97 Conference, London, April 1997.

J. F. Koegel (Ed.): **Multimedia Systems**, ACM Press, Addison-Wesley, New York, 1994.

James Kurose & Keith W. Ross: **Computer Networking**, Second Edition, Pearson Education, New Delhi, 2002.

John Fox: **Intelligent Agents Which Reason About Beliefs, Decisions and Plans: Logical Foundations and Practical Applications**, Invited Paper, PAP '97 Conference, London, April 1997.

K. Bajaj & D. Nag: E-Commerce: **The Cutting Edge of Business**, Tata McGraw-Hill, New Delhi, 1999.

K. Downes, Marilee Ford, H. K. Liu, Steve Spanier & T. Stevenson : **Internetworking Technologies Handbook**, Second Edition, Cisco Press / Techmedia, 1999.

Marvin Minsky: **Society of Mind**, Simon & Shuster, New York, 1980.

Michael Afergan et al: **Web Programming Desktop Reference**, Prentice Hall of India, New Delhi, 1998.

N. Bensaid and Ph. Mathieu: **A Hybrid and Hierarchical Multi-Agent Architecture Model**, Proceedings of the PAAM '97 Conference, London, April 1997.

N. R. Jennings: **Agent Software**, Preprint, QMWC, Univ. of London, 1995. (Subsequently published.)

Nalin K. Sharada: **Multimedia Networking**, Prentice-Hall of India, New Delhi, 2002.

P. Charlton et al: **An Open Architecture Supporting Multimedia Services on Public Information Kiosks**, Proceedings of the PAAM '97 Conference, London, April 1997.

P. Ciancarini, D. Rossi, F. Vitali, A. Knoche and R. Tolksdorf: **Coordinating Java Agents for Financial Applications on the WWW**, Proceedings of the PAAM '97 Conference, London, April 1997.

Pattie Maes: **User-facing Software Agents**, Tutorial at the PAAM '97 Conference, London, 1997. <http://www.demon.co.uk/ar/PAAM99/>

Paul T. Ammann: **Managing Dynamic IP Networks**, Tata-McGraw-Hill Publishing Co. Ltd., New Delhi, 2001.

R. K. Arora et al (Ed.): **Multimedia 98 --- Shaping the Future**, Tata McGraw-Hill, 1998.

R. Pool, **Turning an Info-Glut into a Library**, *Science*, Oct. 7, 1994, pp. 20-22.

Rahul Banerjee: **An Intelligent System for Behavioral Analysis**, Ph. D. Thesis, AU, Amt., 2001.

Rahul Banerjee: **Lecture Notes on Computer Networks**, Nov. 2002, available on-line at: <http://www.bits-pilani.ac.in/~rahul/CN/index.html/>

Rahul Banerjee: **Lecture Notes on Internetworking Technologies**, Oct. 2002, BITS, Pilani, available on-line at: <http://www.bits-pilani.ac.in/~rahul/eac451/index.html/>

Rahul Banerjee: **Lecture Notes on Internetworking Technologies**, Oct. 2002, BITS, Pilani, available on-line at: <http://www.bits-pilani.ac.in/~rahul/eac451/index.html/>

Raj R. Reddy: **Challenges in the AI**, ACM Computing Surveys, Vol. 27, No. 3, 1995, pp. 301-303.

Ravi Kalakota & Andrew B. Whinston: **Frontiers of Electronic Commerce**, Addison-Wesley Longman, Inc., Reading, 1996 (IE: 1999).

RFC 1009 (Requirements for Internet Gateways)

RFC 1009 (Requirements for Internet Gateways)

RFC 1011 (Official IP)

RFC 1042 (IP over IEEE 802.3)

RFC 1124 (Policy Issues in Interconnecting Networks)

RFC 1125 (Policy Requirements for Inter-Administrative Domain Routing)

RFC 1147 (FYI: A list of Network Management Tools)

RFC 1175 (FYI: A very useful reference-list on Internetworking related information)

RFC 1208 (Glossary of Networking Terms)

RFC 1209 (IP over SMDS)

RFC 1254 (Gateway Congestion Control)

RFC 1360 (Official Protocol Standards of the Internet Architecture Board)

RFC 1630 (Universal Resource Identifiers in the WWW)

RFC 1738 (Uniform Resource Locators)

RFC 1809 (IPv6 Flow Labels: An Informational RFC)

RFC 1825 (IP Security Architecture)

RFC 1826 (IP Authentication Header)

RFC 1827 (IP Encapsulation Security Payload)

RFC 1828 (IP Authentication using MD5)

RFC 1883 (Older IPv6 Specification)

RFC 1884 (IPv6 Addressing)

RFC 1886 (IPv6 DNS Extensions)

RFC 1887 (IPv6 Unicast Addressing)

RFC 1971 (IPv6 Address Autoconfiguration)

RFC 1972 (IPv6 over Ethernet)

RFC 2019 (IPv6 over FDDI)

RFC 2023 (IPv6 over PPP)

RFC 781 (IP Timestamp)

RFC 791 (IP version 4)

RFC 815 (IP Datagram Reassembly)

Richard Murch & Tony Johnson: **Intelligent Software Agents**, Prentice-Hall PTR, New Jersey, 1999.

S. Keshav: **An Engineering Approach to Computer Networking**, Addison-Wesley, Reading, 1997.

S. Mitaim and Bart Kosko: **Profile Learning with Neural Fuzzy Agents**, Proceedings of the PAAM '97 Conference, London, April 1997.

Smoot Carl-Mitchell & John S. Quarterman: **Practical Internetworking with TCP / IP and UNIX**, Addison-Wesley, Reading, 1993. (This book does not really discuss the IPv6. This however, helps the reader to take a look at the pre-IPv6 days and realize the wisdom of evolution of the IP.)

T. S. Dahl, S. Pearson and C. W. Priest: **An Agent Communication Platform in Object Oriented Prolog**, PAP '97 Conference, London, April 1997.

The US Digital Library Initiative: **Agency perspectives**, available at the URL: <http://computer.org/computer/dli/r50022/agencies.htm>.

Tim Finin: **Agent Communication Languages: KQML, KIF and the Knowledge Sharing Approach**, Tutorial at the PAAM '97 Conference, London, 1997.

Uyless D. Black: **TCP / IP & Related Protocols**, Second Edition, McGraw-Hill, N. Y., 1995.

V. Braun, B. Steffen and H. Wendler: **Service Definition of Intelligent Networks: Experience in a Leading-edge Technological Project Based on Constraint Techniques**, PAP '97 Conference, London, April 1997.

V. V. S. Sarma: **Intelligent Agents**, Journal of IETE, Vol. 42, No. 3, 1996, pp. 105-109.

W3C: **PNG (Portable Network Graphics) Specification**, Oct. 1996, available at the URL: <http://www.w3c.org/TR/REC-png>.

William Buchanan: **Advanced Communications and Networks**, Chapman & Hall, London, 1997.

William Stallings: **Cryptography and Network Security**, Second Edition, Prentice-Hall, Upper Saddle River, 1999.

Y. Han et al: **Agents for Citation Finding on the World Wide Web**, Proceedings of the PAAM '97 Conference, London, April 1997.

Y. Zheng & S. Akhtar: **Networks for Computer Scientists and Engineers**, Oxford University Press, New York, 2001.

About the Book

This work is the first of its kind in terms of simple introduction to the key concepts related to frontier areas of Internetwork-specific research and development. The book has been written as a simple text on internetworking technologies that should also cater to the needs of the working engineers who wish to update themselves about various associated frontier technologies or those who wish to have a brief survey of the state-of-the art so as to decide the exact direction they may wish to take for their research and development initiatives. However, this small volume can very well serve as the secondary reading material for an advanced course in Internetworking.

About the Author

Rahul Banerjee earned his first degree (**B.E.**) from the Government Engineering College, A. P. S. University, Rewa (India). Subsequently, he did his further studies at the Maulana Azad College of Technology (an REC), Bhopal (India) and the Birla Institute of Technology & Science, Pilani (India) in the areas of Computer Science and Software Systems respectively. His **Master's** research focused on the kernel-specific issues of Unix and Unix-like operating systems and Memory Management of CISC and RISC Processors. At the Amravati University, Amravati (India) his **Ph.D.** work (in Computer Science and Engineering) focused on Intelligent Software System for Behavioural Analysis.

He is the author of three books and a research monograph (PHI, TMH, KP) in the area of Computer Science and Engineering apart from a number of research publications in refereed journals and a co-author of a few IETF documents including the IDs on IPv6. He has also held editorial / reviewing responsibilities for established journals and magazines including the IEEE Internet Computing.

He has led several international projects including those funded by the European Commission. He was instrumental in initiating the IPv6-specific research, development and deployment activities in India through the well-known "Project IPv6@BITS" (pronounced as the 'Project IP v 6 at BITS'). He has lectured in several countries including UK, France and Belgium.

His research interests include internetworking, operating systems, wearable computing architecture and machine intelligence. He leads a core group of researchers at BITS working on projects entitled "The Project BITS-MOS" (a project on a distributed multimedia operating system), "The Project BITS-WearComp" (a project on special purpose wireless wearable computing devices), "The JournalServer Project" (a collaborative Virtual Digital Library project led by Oxford University and BITS Pilani) and the "The NGNI-MMI-QoS Project" (a multinational European Commission funded project being lead by BITS Pilani).

Currently, he is with the Computer Science and Information Systems Group at the Birla institute of Technology and Science (BITS), Pilani (India) where he also holds the additional responsibilities as the Coordinator of the Centre for Software Development and an Assistant Dean of the Distance Learning Programmes Division.