DISTRIBUTED CONTROL ARCHITECTURE FOR MULTISERVICE NETWORKS

Thesis submitted in accordance with the requirements of
University College London for the degree of Engineering Doctorate

by

RADHAKRISHNAN KADENGAL

November 2011

Department of Electronic & Electrical Engineering

Declaration

I, Radhakrishnan Kadengal, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

ABSTRACT

The research focuses in devising decentralised and distributed control system architecture for the management of internetworking systems to provide improved service delivery and network control. The theoretical basis, results of simulation and implementation in a real-network are presented. It is demonstrated that better performance, utilisation and fairness can be achieved for network customers as well as network/service operators with a value based control system.

A decentralised control system framework for analysing networked and shared resources is developed and demonstrated. This fits in with the fundamental principles of the Internet. It is demonstrated that distributed, multiple control loops can be run on shared resources and achieve proportional fairness in their allocation, without a central control. Some of the specific characteristic behaviours of the service and network layers are identified. The network and service layers are isolated such that each layer can evolve independently to fulfil their functions better. A common architecture pattern is devised to serve the different layers independently. The decision processes require no co-ordination between peers and hence improves scalability of the solution. The proposed architecture can readily fit into a clearinghouse mechanism for integration with business logic. This architecture can provide improved QoS and better revenue from both reservation-less and reservation-based networks. The limits on resource usage for different types of flows are analysed. A method that can sense and modify user utilities and support dynamic price offers is devised. An optimal control system (within the given conditions), automated provisioning, a packet scheduler to enforce the control and a measurement system etc are developed. The model can be extended to enhance the autonomicity of the computer communication networks in both client-server and P2P networks and can be introduced on the Internet in an incremental fashion.

The ideas presented in the model built with the model-view-controller and electronic enterprise architecture frameworks are now independently developed elsewhere into common service delivery platforms for converged networks.

Four US/EU patents were granted based on the work carried out for this thesis, for the cross-layer architecture, multi-layer scheme, measurement system and scheduler. Four conference papers were published and presented.

TABLE OF CONTENTS

TABLE OF FIGURES

LIST OF ABBREVIATIONS & SYMBOLS

AF	Assured Forwarding
ANSI	American National Standards Institute
API	Application Program Interface
AS	Autonomous System
ATM	Asynchronous Transfer Mode
BE	Best Effort
BGP	Border Gateway Protocol
BRM	Benefit/Revenue Meter
BSS	Billing Support System
BW	Bandwidth
CBQ	Class Based Queuing
CLI	Command Line Interface
CMIP	common management information protocol
COPS	Common Object Broker System
CORBA	Common Object Broker Architecture
CoS	Class of Service
CP	Charged Price
CPU	Central Processing Unit
CR-LDP	Constrained Routing - Label Distribution Protocol
D/S	Destination/Source
DiffServ	Differentiated Services
DRC	Distributed Resource Control
DSP	Digital Signal Processing
DWDM	Differential Wavelength Division Multiplexing
ECMP	Equal Cost Multipath
ECN	Explicit Congestion Notification
EF	Expedited Forwarding
EFSFQ	Expedited Forwarding Start time Fair Queuing
eth	ethernet
EU	European Union
FCFS	First Come First Served
ftp	ftp protocol
I/O	Input/Output
ID	Identifier
IEC	International Electrotechnical Commission
IEE	Institution of Electrical Engineers
IEEE	Institution of Electrical and Electronic Engineers
IGP	Interior Gateway Protocol
IntServ	Integrated Services
IP	Internet Protocol
IPD	Inter Packet Delay

IPTV	IP Television
IR	Ingress Router
ISO	International Standards Organisation
ISP	Internet Service Provider
IXP	Internet Exchange Point
J2EE	Java 2 Enterprise Edition
JAIN	Java APIs for Integrated Networks
LAN	Local Area Network
M/D/1	Markovian input/Deterministic service/1 server
M/M/1	Markovian input/Markovian service/1 server
M/M/s	Markovian input/Markovian service/Several servers
MEMS	Micro Electro Mechanical Switch
MPLS	Multi Protocol Label Switching
NAP	Network Access Point
NE	Network Element
NIB	Network Information Base
NoC	Network on Chip
NOC	Network Operations Centre
OEM	Original Equipment Manufacturer
OS	Operating System
OSI	Open Systems Interconnection
OSPF	Open Shortest Path First
OSS	Operation Support System
OSS/J	OSS/Java
O-UNI	Optical User Network Interface
P2P	Peer to Peer
PDP	Policy Decision Point
PEP	Policy Enforcement Point
PIB	Policy Information Base
PID	Proportional Integral Derivative
POMDP	Partially Observable Markov Decision Processes
POTS	Plain Old Telephony System
PP	Provisioning Potential
PSTN	Public Switched Telephony Network
QoS	Quality of Service
RC	Router control
REM	Random Early Marking
RF	Radio Frequency
RFC	Request for comments
RMI	Remote Method Invocation
RS232	serial link
RSVP	Resource Reservation Protocol
S/D	Source/Destination
SFQ	Start time Fair Queuing
SLA	Service Level Agreement
SNMP	Simple Network Management Protocol

SoC	System on Chip
SONET	Synchronous optical networking
SRLG	shared risk link group
ssh	secured shell
tc	traffic control
TCP	Transmission Control Protocol
TCS	Traffic Control System
TDM	Time Division Multiplexing
telnet	telnet protocol
Tier 1 n/w	National ISPs (in general)
Tier 2 n/w	Regional ISPs (in general)
Tier 3 n/w	Local ISPs (in general)
Tkined	Tcl/tK based Interactive Network EDitor
TOS	Type of Service
UDP	User Datagram Protocol
UML	Unified Modelling Language
UNI	User Network Interface
US	United States
VoIP	Voice over IP
WAN	Wide Area Network
WDM	Wavelength Division Multiplexing
WFQ	Weighted Fair Queuing
WRR	Weighted Round Robin
XML	Extensible Markup Language

List of symbols

z	z-transform operator
ZOH	Zero Order Hold
:	extends
Λ	and
$\forall$	for all
$\Re$	set of real numbers
Δ	difference
$\in$	is an element of
O	the output space and a region surrounding this solution of the system
O	Order of

List of variables

A_k	Potential willingness to Pay by the user, in 'Provisioning Potential unit' (also denoted PPNk)

C_k	Set point for control, in ‘resource unit’, set by the n/w operator
C_{Rk}	Set point for control for resource R, k is the iteration number
E_k	Difference variable, in ‘resource unit’. The difference between the set point and controlled output is given by $E_k = C_k – Y_k$
E_{Rk}	Difference variable for resource R
k	Sample step number, iteration steps for computation
K	Proportionality multiplicand for the ‘availability figure’ of the resource, its value controls the stability with time. This parameter is scheduled into the controller based on simulation results
K_R	Proportionality multiplicand for availability figure of the resource R, this is scheduled into the controller based on simulation results
n	Number of users making N traffic aggregates
N_R	Traffic aggregate using resource subset *R*
N_T	Total number of traffic aggregates (active paths)
P_k	Resource price function in ‘price unit for resource unit’
P_{Nk}	Total resource price function for traffic aggregate N
Pnk	Price function for individual user making up total aggregate resource price P_{Nk}
PP_{Nk}	Provisioning Potential for the traffic aggregate N (also denoted A_k). This is the potential willingness to pay by the traffic aggregate N
PPnk	Potential willingness to Pay by user making up aggregate PP_{Nk}
PP_{Rk}	Total share of Provisioning Potentials for resource R
P_{Rk}	Resource price function for resource R
R_N	Resource subset used by traffic aggregate N
R_T	Total number of resources
U_N	Resource consumption by traffic aggregate N (e.g. Flow rate of active path)
un	User resource consumption making up aggregate consumption U_N
U_R	Total resource consumed in a given resource R
v	Number of individual paths making V aggregate paths
V_T	Total number of paths (or combinations)
Y_k	Output level, in ‘resource unit’
Y_{Rk}	Output level of resource R

LIST OF PUBLICATIONS

1. Patents granted:

1.1 Kadengal, R., et al. Scheduling and reservation for dynamic resource control systems, US Patent 6888842, April 2000.

The invention deals with a packet scheduling scheme that maintains the inter-packet delay for the expedited traffic flow within the given limits while operating in a work-conserving mode. The scheduler sends packets from a set of queues with differing priorities onto an outgoing link. The scheme associates a weight and a virtual start time with each of the queues. Queues are selected, in order of the virtual start time, until a non-empty queue is selected. One or more packets are sent from the selected queue, then the virtual start time is updated, based on the length of the transmitted packet, and the weight associated with the selected queue. This patent relates to the scheduler devised for this thesis (section 5.1.1).

1.2 Kadengal, R. Determining traffic information in a communications network, US Patent 6804196, November 2000.

The invention deals with the provisioning of a network information database for each region in a communication network and related data acquisition. The link occupancy data from each of the network element within the region is acquired and stored in the network information database. An edge router wishing to dispatch packets into the core network acquires the link occupancy data from the network information databases in those network regions through which packets are to be routed so as to schedule dispatch of those packets into the core network at a rate commensurate with resource availability within the core network. The link occupancy data is classified into quality of service classes. This patent relates to the link occupancy measurement system devised for this thesis (section 6.2.5).

1.3 Kadengal, R. Dynamic resource control in telecommunications networks, US Patent 6928053, December 2000.

The invention deals with a decentralised network control system where each edge-router allocates bandwidth to a traffic aggregate based on a provisioning potential to the traffic aggregate and the total resource-price, an intermediate variable, of the resources used by that traffic aggregate. The allocation of network resources settles to a proportional fair solution across all the edge-routers, without the need for any communication between the edge-routers. The allocation is based on the provisioning potentials and the set point capacity of the resources. The system assigns set point values for a network performance parameter for the routers as well as an initial value for the intermediate variable called resource-price.

The system operates a control loop that takes the differences between the actual performance and the set point for the router and brings the difference between them to a minimum. A second control loop operates between the hosts and the edge-routers where the negotiation for service allocation takes place. This patent relates to the distributed control architecture devised for this thesis (section 3.1.2).

1.4 Kadengal, R., et al. Management and control of multi-layer networks, US Patent 7269185, May 2001.

The invention deals with the network resource allocation across the different layers of a multi-layer communication network. The different layers include IP router layer, MPLS layer, Optical/Photonic layer, Wireless link layer etc. The system incorporates a distributed system and protocols to allocate resources requested by one layer from another layer. At a first layer, the management structure provides an indication to a second layer of the required resources that are to be allocated from the second layer. The second layer automatically offers the required resource together with a condition for use of those resources. This condition includes a notional resource-price factor which is dependent on current utilisation. The first layer determines if the condition for use of the offered resources is acceptable and, if so, automatically accepts the offered resources from the second layer. This patent relates to the cross-layer architecture devised for this thesis (section 3.7).

2. Conferences:

2.1 Kirkby, P., Kadengal, R. Traffic management and control using a single 'congestion price' like variable across multiple layers of network hierarchy, Proceedings of the colloquium on Control of Next Generation Networks, IEE, 1999

2.2 Kirkby, P., Kadengal, R, Carrol, J, Sabesan, S, Biddiscombe, M, et al. The use of economic and control theory analogies in the design of policy based dynamic resource controlled (DRC) network architectures, Proceedings of the International Teletraffic Congress ITC-16, Elsevier, 1999

2.3 Kadengal, R. Dynamic Resource Control and Management middleware for Carrier networks, UKTS16, IEE, May 2000.

2.4 Kadengal, R. Advanced router and switch designs for Quality of Service in distributed routing fabric, UK Teletraffic Symposium, IEE, May 2001.

3. Presentations:

3.1 Kadengal, R. Adaptive Bandwidth Management in IP/Photonic networks, Presentation UCL April 2001.

3.2 Kadengal, R. Bringing bandwidth to access in the improved Internet architecture, Presentation UCL, March 2002.

4. Technical Reports:

4.1 Kadengal, R. Distributed Digital Control System for Dynamic Resource Management of Carrier Networks, Nortel Networks Technical Report, September 2000.

4.2 Kadengal, R. Proposal for IP over Photonic switch control interface design, Nortel Networks Technical Report, Feb 2001.

ACKNOWLEDGEMENTS

I thank my academic supervisors Prof Jon Crowcroft (Cambridge), Prof George Pavlou, Dr Lionel Sacks and Dr Miguel Rio (UCL) and my industrial supervisor Dr Paul Kirkby (Nortel Networks). I also thank Prof DG Smith (Strathclyde), Prof John O'Reilly (UCL), the colleagues at the Harlow Labs of Nortel Networks and the EngD staff of UCL.

1. INTRODUCTION

1.1 General

This thesis contributes to the de-centralised and distributed resource allocation and traffic management areas of the Internet engineering. The de-centralised and distributed control architecture supports autonomic network systems that are visualised to allow dynamic self-organisation of the network according to the professional, economical and social needs of the users [64, 93]. With the exponential growth in the number of systems and subsystems that forms part of the Internet, without autonomic networking the management of the network to deliver expected performance will become impractical, as there will be far too many disparate systems to administer. While the individual devices are getting cheaper, major portion (>70%) of the total cost of ownership of the networked systems is spent in the management of the network [172].

The system proposed controls traffic admission to the network at the edge routers, on an edge-to-edge basis such that the network is not overloaded, at the same time maintains higher utilisation of the network elements. The edge routers treat the network as a common resource. However, semaphores like token passing are not used for ingress control. A centralised bandwidth broker is not used either. It is demonstrated that a system that provides direct feedback to its users (in the given case the users are edge routers) and the users acting on that feedback to control the usage of the system will achieve proportional fairness in a de-centralised way without the need for inter-user communication.

The idea is to build on the de-centralised and distributed principles of IP systems and developing a higher-level control system that compliments the existing IP systems. The Diffserv protocol does not maintain any per-flow state in the core elements of the network; similarly, the proposed system also

does not require any per-flow state to be maintained in the core elements of the network. However, Diffserv alone cannot guarantee inter-packet delays for premium traffic and perform appropriate traffic management. Moreover, a system that is protocol-agnostic is required to provide a stable management environment, as the transport protocols change in course of time.

The resources are allocated in parallel. Each resource is associated with an intermediate variable called ‘price’, which is in effect a ‘price-like variable’ and is not connected to any currency. When there is imminent congestion, the price function of that resource is increased and the ingress flow is limited to admit only the higher value flows.

Concept diagrams of the overall system presented in this thesis is given in Figure 1 and Figure 2. Internet can be visualised as a decentralised network of different types of networks. It may be noted that Figure 1 is only a very top-level abstract diagram, depicting the connectivity architecture of the Internet[1]. The different types of networks include Tier 1 networks (generally the national ISPs), Tier 2 networks (generally the regional ISPs) and Tier 3 networks (generally the local ISPs). These different networks themselves operate as autonomous systems (AS). The idea is to manage the traffic input to the network by de-centralised decision processes in the edge routers as indicated. The only feedback required is that of the cost functions of the network resources for traffic/route aggregates. This information is available in databases for every autonomous network (AS) that constitutes the network (shown as NIB). In a full-mesh autonomous network consisting of n nodes, the number of unique connections possible is $n.\ (n-1)/2$ i.e. $O(n^2)$. Collecting this information will be an enormous problem. However organising the cost function information per autonomous network helps to reduce the complexity of retrieving it to $O(n)$.

[1] The end-to-end transit of datagrams through the different types of networks could be traced using programs like *traceroute*

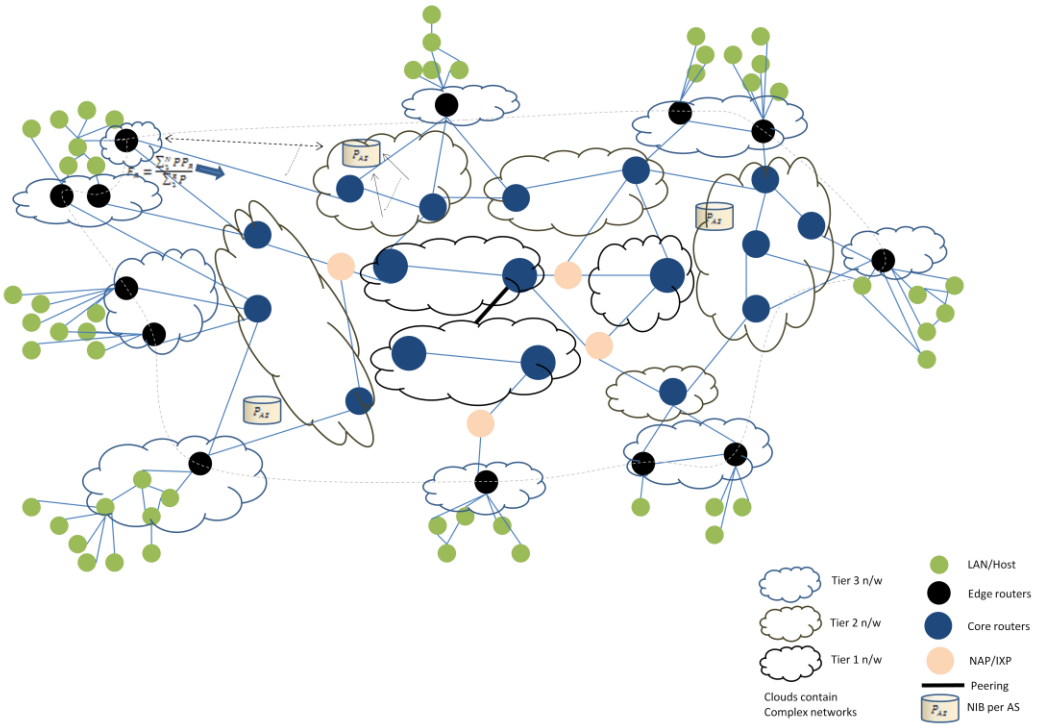


Figure 1 Concept diagram-I of the system presented in this thesis – ingress traffic and feedback in the network

1.2 Problem statement

The Internet can be considered as a multi-service delivery platform with three distinct types of entity: the user, the network provider and the service provider. The services are delivered by different types of transport mechanisms (e.g. TCP, UDP) in different type of system organisations (e.g. client-server, P2P).

In general, the flow rate/QoS achieved by user traffic is determined by various transport & network issues; not user/operator demand. At the moment, the resources used for the service delivery are provisioned manually as per the estimated requirements of the users, as decided by the business contracts and associated policies. Below this management level, while there are some control mechanisms like TCP for microflows, there are no integrated

control mechanisms available for macroflows i.e. traffic aggregates. The reservation based flow control like the one used in Intserv did not succeed.

The UDP traffic and aggregates of TCP flows does not have a control system to manage them in the client-server or P2P architectures. This gives rise to inefficiencies and unfairness in service delivery.

Therefore, a new control architecture is necessary to put user/operator demand in control of the service achieved.

Using the system architecture methodology given in section 1.6.2, the different viewpoints on how the Internet evolved has been looked at. Earlier works from the view points of resilience, queuing and resource sharing by P Baran, L Kleinrock and D Davies respectively resulted in the basic designs of the current architecture [167]. The disparate packet networks were then connected via gateways with traffic engineering and routing techniques, which resulted in the current Internet. The next step would then be adding further attributes like quality of service and fairness. This coincides with the needs identified above.

It may be observed that the application part of the OSI stack is moving from a monolithic structure to that of distributed structure, with the advent of distributed file sharing as well as computing applications. Similarly, the network part of the stack is also moving from a monolithic structure to that of distributed structure, with the advent of the multitude of network substrates that are seamlessly integrated to provide connectivity. The proposed top-level control system provides the necessary framework to maintain the isolation between these two parts. This helps to control all types of flows with a single framework and help in monetising the usage more effectively.

It has been proven by Ashby and Conant [50] that any control system that is effective and simple must be isomorphic with the system being regulated i.e. should be a model of the system itself. The connection-less, de-

centralised and distributed nature of the control system proposed in this thesis fulfils this theory.

Considering the enormity of the distribution of Internet resources and the different architectures used for service delivery, the control framework required to harmonise the resource management has to be agnostic of the transport mechanisms, in order to be future proof. This is because innovation in transport mechanisms is still happening. Such a framework has to be essentially distributed in nature, not only to allow continued growth (scalability) but also to facilitate incremental deployment.

This requires devising several components- a control system that provides optimal control within the given conditions, automated provisioning (predictive and reactive), a control enforcement mechanism and a measurement system.

The control system operates over traffic trunks/route aggregates at the edge-routers of the network. It makes decisions regarding the provision of resources to the traffic trunks/aggregates. These decision commands are enforced by schedulers that control the bandwidth allocated at the links. While the schedulers are local to the edge-routers, the decision commands are based on the packet transport and resource availability issues across the network between the edge routers. Therefore, it is fair to say that the decision command system operates above the transport layer. It may be noted that there is no resource reservation involved.

The following Figure 2 shows a depiction of the level at which the proposed control system operates. The figure shows a depiction of end hosts communicating with each other using various end-to-end protocols. The hosts communicate via their edge routers. The edge routers use core routers to carry the traffic across. The edge routers aggregate the traffic depending on their service requirements and routes. These aggregates are transported between the edge routers as traffic trunks. The edge routers operate control decisions at this is the level.

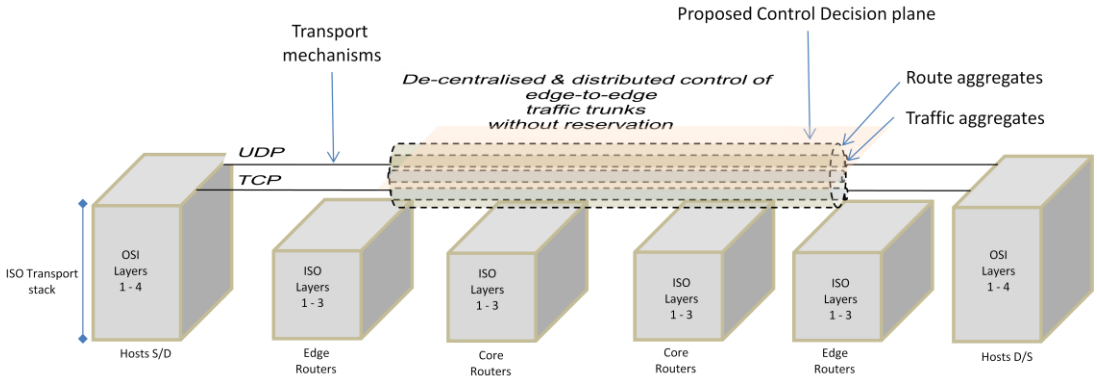


Figure 2 Concept diagram-II of the system presented in this thesis – traffic aggregates & control plane

The control layer also serves as an aggregation layer to bundle the various flows using the same set of resources and therefore providing higher scalability when compared to methods that use admission control per flow.

1.3 Thesis work

This thesis work started in 1999 and the main body of the work was complete by 2001. The original title was 'Optimal Infrastructure for a Universal Electronic Enterprise', to research into the design, development and implementation of a system architecture and components to efficiently operate, control and manage a Universal Electronic Enterprise based on Computer Communications Control. The electronic enterprise is constituted by networked communication (telecom/datacom) and IT (email, web servers, storage etc) software and hardware systems. These are in fact systems of systems and are not monolithic in nature. They run on heterogeneous distributed platforms and offer a variety of flexible, programmable, and multimedia services to the users. The project was envisaged to make original contributions in inter-disciplinary systems science, distributed engineering technology and real-time protocols. The main thrust was to find a confluence of communications, control, signal processing and economics/ management in the emerging Internet. The idea was that a value added and interactive service provisioning, intra-domain and inter-domain traffic engineering and

resource optimisation for network and computing services and resources, operating with appropriate price function models would provide optimum quality of service and balanced network growth for multi-service networks. This thesis presents the results of the work carried out within the constraints of the industry.

The Internet resources are provisioned manually as per the estimated requirements of the users, as decided by the business contracts and associated policies. While there is some basic self-configuration available, there is no framework available to suit a dynamic system like the Internet to work as an autonomic system[158]. As such, best effort is made to carry the traffic generated by the users within the constraints of the service level agreements and the resources. This scenario gives rise to various issues, e.g. fairness of the resource allocation, quality of service (delay, jitter, loss) of the packet flows etc. While there are some control mechanisms like TCP for microflows, there are no integrated mechanisms available for macroflows i.e. traffic aggregates. Hence, the issues remain largely unresolved. This thesis is an investigation into such issues using the distributed control systems approach and provides some theoretical ideas, tools and techniques and various elements for building autonomic networks. It also provides some realised application examples in IP and Photonic networks.

The thesis is structured as follows: chapter 1 provides the introduction, motivation, business framework and engineering process used, chapter 2 gives background and related work, chapter 3 provides the system analysis and modelling, chapter 4 gives architecture simulation, results and interpretation of results, chapter 5 describes the scheduler, chapter 6 describes real network implementation and results, chapter 7 gives conclusions and future work.

This thesis contributes the following: A decentralised control system framework enable easy analysis of networked and shared resources is developed. It is demonstrated that distributed, multiple control loops can be run on shared resources and achieve proportional fairness in their allocation,

without a central control. Some of the specific characteristic behaviours of the service and network layers are identified. These layers are isolated such that each layer can evolve independently to fulfil their functions better. A common architecture pattern is devised to serve the different layers independently. The decision processes require no co-ordination between peers and hence improving the scalability. The architecture can fit into a clearinghouse topology to readily integrate with business logic. This architecture can provide improved QoS and better revenue from both reservation-less and reservation-based networks. The limits on resource usage for different types of flows are analysed. A method that supports dynamic price offers and modify/sense user utilities is devised. A control system that is optimal in given conditions, automated provisioning, a packet scheduler to enforce the control and a measurement system etc are developed. The model can be extended to enhance the autonomicity of the computer communication networks in both client-server and P2P networks and can be introduced on the Internet in an incremental fashion.

1.4 Relevance of the contributions in today's Internet & Systems

The idea of a universally connected network that this thesis started from is gaining more and more strength. In fact, Licklider suggested *Intergalactic Computer networks* far back in 1963 [167].

The idea of control system architecture for de-centralised and distributed networks, macro-scheduling with defined attributes of fairness, separation of service and network provision, absence of peer to peer co-ordination etc are gaining popularity in areas like autonomic systems, provision of on-demand real-time services, ensuring QoS in cloud computing, multi-processor job scheduling etc. The idea of IP traffic control above the transport layer, described in this thesis has found recent interest in 'Quality of Transport' (QoT) studies. The framework is applicable in different areas of the Internet Traffic Control and Management business viz. IP Routers, MPLS switches, Optical switches, Photonic switches, Wireless resource management, Server

farm load balancers, Content provisioning etc. On the micro-level, the ideas are applicable in multiprocessing, multiprocessor, multicore, System-On-Chip and Network-On-Chip communication systems.

The idea of scheduler, with reduced inter-packet delay and dynamic scheduling, described in this thesis has found recent use in wireless networks where a session being transmitted suddenly breaks due to RF channel quality issues.

The framework and principles presented in this thesis can be generalised to find applications in any system of systems- from multi-agent software systems to robotic systems.

1.5 Motivation

The Internet as a physical substrate is a shared resource. A variety of applications interacts with each other on this substrate using different protocols. Naturally, there are a number of issues that crops up when using a shared resource that affects the Quality of Service delivery.

A prime example is the congestion caused by applications competing for resources. Protocols like Ethernet, and later TCP etc had some amount of congestion control built into them. However, the effectiveness of such measures is defeated by applications that spawn multiple TCP instances (like the P2P torrents) as well as non-TCP applications (like VoIP, IPTV etc). Without QoS class tiering, those who generate the most traffic will dominate the Internet bandwidth. Providing QoS in IP has been a major thrust area in packet switching networks and the applications that demand QoS include human communication, scientific computing, online financial trading, distributed games etc. It is known that measurement based admission control techniques usually tend to favour small and short lived flows [32] as the short flows have higher probability of not getting sampled (for rejection) than flows that traverse a larger number of hops. Non-TCP protocols, torrents etc are

generally considered unfair. Thus a mechanism to enforce fairness in the resource allocation, regardless of the type of protocol used, is needed.

Perhaps a better architectural decision would be not to burden the data transport protocols (like TCP) further with the issues like congestion control that is due to the shared resource. The rationale is as follows: TCP has several functions like packet re-ordering/re-transmission/error correction, flow control based on buffer fill, congestion avoidance based on timers etc. Due to the wait-times and re-transmissions involved, TCP already incurs longer delays than, say UDP. The functions of TCP are indeed going to stay, as they are important for the micro-flow. However, to improve the performance, it is required to move from ‘congestion avoidance’ at the micro-flow level to ‘congestion control’ at macro-flow level, by resource provisioning and usage control. By doing this, the obstacles encountered by protocols like TCP is reduced.

Therefore, issues due to the shared resource will be better solved by an automatic provisioning protocol that operates above the transport layer between the edge routers, handling the aggregate, macro-flows. This protocol will ensure fair sharing of the physical substrate i.e. the network resources among the many users and leave the lower layer transport protocols to do the jobs they are designed for.

In a highly distributed network, the effect of multiple control loops is another issue. In this thesis, it is demonstrated that multiple control loops can be allowed in a highly distributed automated network, yet their isolation and tractability can be maintained.

The attempt in this thesis is first to analyse such a sharing mechanism and then to synthesis and implement a solution, and provide a value based admission control to deliver specific quality of service and ultimately improve the average revenue per user.

1.6 Business framework and Engineering process

1.6.1 Business context framework used

An overview of the electronic business enterprise entities in the network communication space, their relationships and importance can be gained from the Zachman frame work [229], shown in Figure 3. The figure is applicable to enterprises including Internet Service and Infrastructure Providers. The rows represent various viewpoints from which the aspects can be described. Each cell, formed by the intersection of a column and a row represents an aspect of the enterprise modelled from a particular viewpoint. This framework is introduced here only to show the wider context of this thesis work. The thesis considers only the system and technology models for the network aspect.

Abstractions / Perspectives	DATA what	PROCESS how	NETWORK where	PEOPLE who	TIMING when	MOTIVATION why
BUSINESS SCOPE (contextual) Planner	List of Things important to the business Entity: Class of Business Thing	List of Processes the business performs Process: Class of Business Processes	List of Locations in which the business operates Node: Business Location	List of Organisations important to the business People: Organisational Unit	List of Events/Cycles significant to the business Time: Business Event/Cycle	List of business Goals/Strategies Ends/Means: Business Goal/Strategy
BUSINESS MODEL (conceptual) Owner	e.g., Semantic model Entity: Business Entity Relationship = Business Relationship	e.g., Business Process Model Process: Business Process I/O: Business Resources	e.g., Business Logistics System Node: Business Location Link: Business Linkage	e.g., Work Flow Model People: Organisation Unit Work: Work Product	e.g., Master Schedule Time: Business Event Cycle: Business Cycle	e.g. Business Plan End: Business Objective Means: Business Strategy
SYSTEM MODEL (logical) Designer	e.g., Logical Data Model Entity: Data Entity Relationship: Data Relationship	e.g., Application Architecture Process: Application Function I/O: User Views	e.g., Distributed System Architecture Node: I/S Function (processor, storage..) Link: Line Characteristics	e.g., Human Interface Architecture People: Role Work: Deliverable	e.g., Processing Structure Time: System Event Cycle: Processing Cycle	e.g., Business Rule Model End: Structural Assertion Means: Action Assertion
TECHNOLOGY MODEL (Physical) Builder	e.g., Physical Data Model Entity: Segment/Table Relationship: Pointer/ Key etc	e.g., System Design Process: Computer Function I/O: Data Elements/Sets	e.g., Technology Architecture Node: Hardware/System Software Link: Line Specifications	e.g., Presentation Architecture People: User Work: Screen formats	e.g., Control Structure Time: Execute Cycle: Component Style	e.g., Rule Design End: Condition Means: Action
DETAILED REPRESENTATIONS (out-of-context) Subcontractor	e.g., Data Definition Entity: Field Relationship: Address	e.g., Program Process: Language Statement I/O: Control Block	e.g., Network Architecture Node: Address Link: Protocol	e.g., Security Architecture People: Identify Work: Screen Formats	e.g. Timing Definition Time: Interrupt Cycle: Machine Cycle	e.g., Rule Specification End: Sub-condition Means: Step
FUNCTIONING ENTERPRISE	e.g.: DATA	e.g.: FUNCTION	e.g.: NETWORK	e.g.: ORGANISATION	e.g.: SCHEDULE	e.g.: STRATEGY

Figure 3 Enterprise architecture framework (courtesy: Zachman)

1.6.2 System architecture methodology used

Developments in the architectural specifications by ANSI/IEEE/ISO/IEC [94, 95] as well as the UML [178] have made it possible to develop standard models that can be applied across the whole spectrum of engineering architecture.

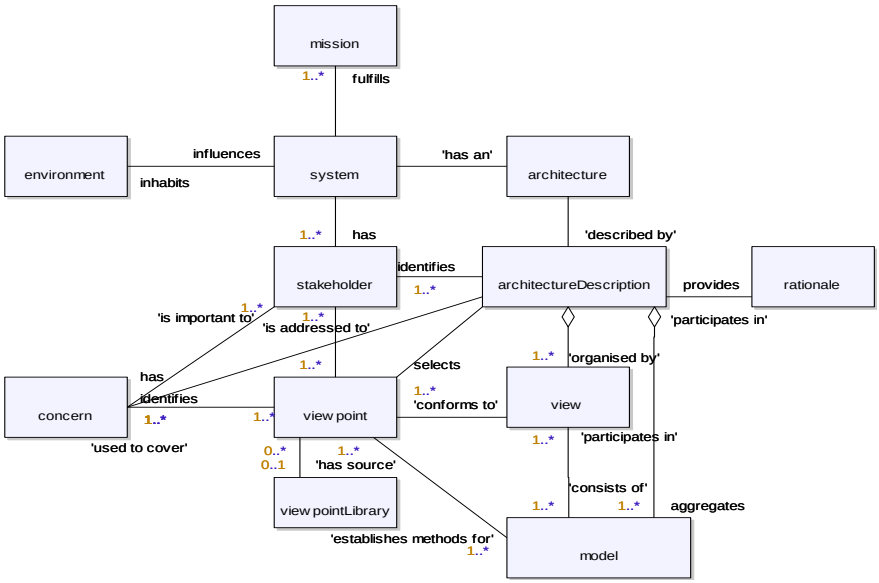


Figure 4 Conceptual model for architectural description (ref ANSI/IEEE 1471/ISO/IEC 42010)

Figure 4 shows the conceptual model for the system architecture followed in this work. A system is realised to fulfil one or more missions (use or operation) in an environment. The environment or context determines the usecases. The different stakeholders of the system have different concerns about the system. The concerns are interests pertaining to the development, operation etc as well as performance, reliability, evolvability etc. These concerns are considered and addressed from different viewpoints. The expression of a system's architecture with respect to a particular viewpoint is referred as view. Views are groups of models that conform to exactly one viewpoint by using its language and rules. The viewpoint taken here is that of distributed resource control.

The way in which the system architecture methodology helped in formulating the problem and solution in a systematic way is mentioned in section 1.2

1.6.3 System engineering process used

The incremental process for system engineering i.e. architecting, designing and testing etc is characterized as in Figure 5 below, described by the IEEE [94].

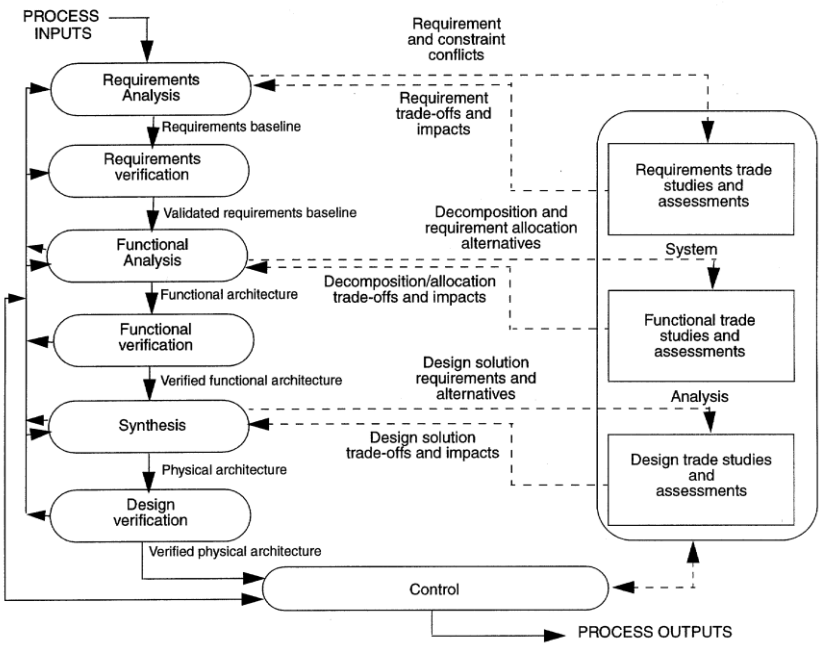


Figure 5 System engineering process (courtesy: IEEE 1220)

The process inputs are the requirements including the value judgments of needs, risk tolerance, cost, utility, quality, delivery, development platforms etc. All the inputs to the engineering process have very high variance in the initial development phase.

The system engineering process described above was used in the systematic development of the system requirements and architecture, functional block design and implementation as well as validation and verification of the system.

2. BACKGROUND

2.1 Related work

2.1.1 De-centralised and distributed networks

The connection-less communication system brought about by the Internet Protocol suite shares the distributed network resources in a very effective way. This is achieved by filling the network resources with fine granularity packets. Awduche et al [16] provides an overview of Internet traffic engineering. Prior research has been conducted on how these resources are shared and utilised in order to deliver value added services. Methods for dynamic routing, traffic management, capacity management etc have been studied earlier for telephony and IP networks [11, 12, 91, 157]. However, sophisticated, on-line, QoS routing and traffic engineering methods are not commonly deployed in real world IP networks [12]. The work described in this thesis is an effort to contribute further in this area of study.

On feedback control, Jacobson et al [97] introduce feedback from packet-receiver to packet-sender for congestion control. Keshav [121] has done further investigations into the feedback control methods and the usage of estimators and fuzzy logic etc. Further work by Breslau et al found that measurement based admission control techniques usually tend to favour small and short lived flows [32]. All these studies deal at the TCP flow level and assume round robin schedulers. The Internet however consists of different types of flows e.g. TCP, UDP etc. The work in this thesis is to build a system above this transport level, using schedulers that are more flexible.

On overall system philosophy, Nagle [163] introduces a game theoretic approach and proposed market based techniques to avoid 'tragedy of the commons' with network resources. In other words, users driven only by their self-interest (not driven by market forces) will over-exploit the resources. This will degrade rendering of services by those resources, in effect reducing the

utility of the resources to all the users. This thesis shows that by using the proposed system, the provision of resources will be proportionally fair to all the users and that over-exploitation and consequent degradation of quality does not happen due to the aggregate level admission control. MacKie-Mason and Varian [152] describes a per-packet charge based congestion control. However, Junko et al states that 'market theory' based studies are not warranted [165] at this layer. Kelly [117] describes the proportional fairness in resource sharing and uses ECN (Explicit Congestion Notification) and shadow prices. Low [151] describes an optimisation approach to flow control, later implemented by random early marking [14]; further development is given in Paganini, Doyle and Low [179].

Fischer et al shows that in a distributed and decentralised network like the Internet, achieving consensus to provide perfect fairness will be impossible as there will at least be some faulty nodes [66]. Interest in self-maintaining networks, a prime objective of autonomic systems for enterprises as well as networks, had suddenly caught on around 2003 [48, 93]. A good survey of autonomic networks is available in [57].

2.1.2 Schedulers

On schedulers, Nagle [163] proposed a fair queuing algorithm and Demers, Keshav, Shenker [55] provide an analysis of why fair queuing is better than first-come-first-served (FCFS) queues. Goyal proposed improved fair queuing algorithms [81, 82] to meet the demands of fairness, delay bounds, computational efficiency, heterogeneity etc. More details are given in section 5. Not much effort has gone into the aggregate level real-time feedback based scheduling in the communications networks [185, 209]. In this thesis work, further improvement is made to offer bounded delay for premium traffic in a work-conserving scenario.

The resources of the network (communication bandwidth, channels, CPU, memory etc) are shared between the user processes by appropriately scheduling their access to those resources. Schedulers operate at different

levels in the system, at the micro levels as well as the macro levels. A scheduler must provide assurances on how the resources are shared and what type of guarantees are given. Usually the scheduler ensures that every flow is eventually served, and in doing so ensures some sort of fairness as well.

The service level assurances provided to the user or the network operator requires the given assurance policies are enforced. Schedulers act as the enforcement points of output control policies as given in Figure 6

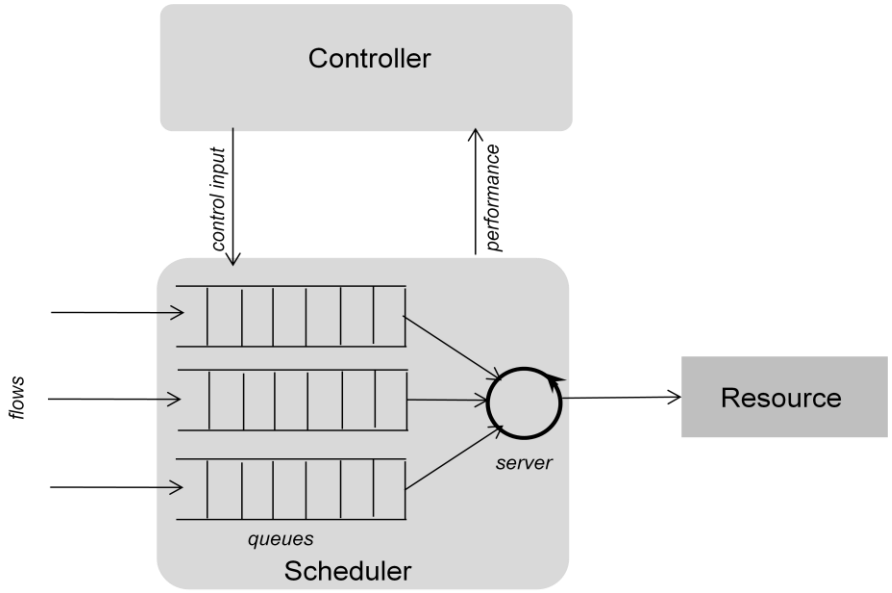


Figure 6 Scheduler as enforcement points for control policies

In order to control the multiservice network resources at the required service levels, the following criteria were identified by the author to satisfy the requirements of policy enforcement points/schedulers. The scheduler shall:

1. partition the output bandwidth according to bids
2. provide QoS guarantee and fairness even with variable capacity
3. does not require prior knowledge of the length of packets
4. flexible to incorporate forwarding strategies based on policies
5. use the bandwidth efficiently when a scheduled flow is blocked
6. handle the instantaneous nature of congestion
7. operate in hierarchical systems

In the domain of flow control in IP networks, the studies so far deal at the TCP flow level and assume round robin schedulers.

The work in this thesis is to build a system above the TCP level using schedulers that are more flexible.

2.1.2.1 Operation of Schedulers

A Queuing and scheduling mechanism is necessarily used whenever there are multiple user processes competing for limited resources. In this section, scheduling mechanisms for packet networks with no reservation, handling datagram packets of variable length is dealt with.

Often the queuing that is seen in human queuing systems is First-come First-serve. A similar system was used in the packet networks, termed FIFO (First-in First-Out). FIFO suffers from several fundamental issues. In the presence of congestion i.e. when the arrival rate is higher than the departure rate, FIFO starts to lose packets and no guarantee can be given for the flows involved. The reason for this is that in general data packets are transmitted with a ‘time-to-live’ information. If the size of the packet queue increases (this happens in congestion), packets that have zero time-to-live will be discarded from being queued for transmission. Further, flows that are higher in rate compared to the rest of the flows take up more share of the network. Therefore, the FIFO scheduler is also unfair. In order to make the flows behave better, the flows need to be rewarded for behaving better.

Nagle [163] proposed in RFC970 (1985) a round robin scheduling algorithm to resolve this issue of unfairness. The single first in, first out queue associated with each outgoing link was replaced with multiple queues, one for each source host. These queues were then serviced in a round robin fashion. It may be noted that the optimal strategy for a given user class is no more sending as many packets as possible, but sending at an appropriate rate that makes one packet available for service when its turn comes in the round robin

cycle. This way the host will be serviced each time the round-robin algorithm cycles, and the host's packets will experience less transit delay.

Nagle [163] does point out that although the round robin queue makes the packet switches fair, the whole of the network is not made fair. To achieve this, rules of the game has to be changed further, so that the optimal strategy for players results in a situation that is optimal for all.

Demers, Keshav, Shenker [55] provides an analysis of why round robin fair queuing is better than first-come-first-served (FCFS) queues. They proposed what was later called as Weighted Fair Queuing. In WFQ, the different queues can have different service shares. In here, the time at which the packet would finish being serviced is calculated and then the packets are serviced in order of their finish time. The priority assignment for the weighted schedulers could be either partially-preemptive or non-preemptive [81]. In partially-preemptive scheduling, the packet in service is always the packet with the highest priority, possibly by pre-empting the transmission of a packet with lower priority, either by discarding or fragmenting. WFQ scheduler however is unfair for the variable rate flows.

Goyal proposed Start-time Fair Queuing (SFQ) algorithm [81, 82] to meet the demands of variations in packet arrival rate as well as server rate. In SFQ the start and finish time are calculated similar to that in WFQ although the packets are serviced in the increasing order of start times. SFQ does provide fairness as well as delay guarantees as follows.

2.1.2.2 Fairness guarantee

In a general case, the allocation of link bandwidth is fair if equal share is given to all the flows. In a weighted queue scheduling scenario, the bandwidth is allocated proportional to the weight associated with the flow. If w_f is the weight associated with the flow f and S_f is the service received by the flow f in the time interval (t_1, t_2), then the allocation is fair if, for all intervals (t_1, t_2) in which the flows f and m are backlogged,

$$\frac{S_f(t_1,t_2)}{w_f} - \frac{S_m(t_1,t_2)}{w_m} = 0$$

Clearly, the assumption here is that the flows can be serviced in infinitesimally divisible quantities. As this is impractical, the objective of a fair allocation algorithm is to have $\left|\frac{S_f(t_1,t_2)}{w_f} - \frac{S_m(t_1,t_2)}{w_m}\right|$ as close to zero as possible. Golestani [78] has shown that, if a packet scheduling algorithm guarantees

$\left|\frac{S_f(t_1,t_2)}{w_f} - \frac{S_m(t_1,t_2)}{w_m}\right| \leq H(f,m)$ for all intervals (t_1, t_2) then

$H(f,m) \geq \frac{1}{2}\left(\frac{l_f^{max}}{w_f} + \frac{l_m^{max}}{w_m}\right)$ where $H(f,m)$ is called the *fairness measure* and l_f^{max} and l_m^{max} are the maximum packet lengths of the flows f and m respectively.

It has been shown in Goyal [81] that $H(f,m)$ value of SFQ is

$$\left(\frac{l_f^{max}}{w_f} + \frac{l_m^{max}}{w_m}\right)$$

2.1.2.3 Delay guarantee

Given the derivation of Finish time, the general deadline guarantee is derived based on its expected arrival time. The departure time *Td* is guaranteed as

$$T_d\left(p_f^j\right) \leq EAT\left(p_f^j, r_f^j\right) + \beta_f^j$$

where $T_d(p_f^j)$ is the departure time of packet p_f^j, $EAT\ (p_f^j, r_f^j)$ is the expected arrival time of packet p_f^j that has been assigned a rate r_f^j and β_f^j depends on l_f^j and the properties of the server as well as the other flows at the server.

It has been shown in Goyal [82] that for SFQ, the term β_f^j is

$$\beta_f^j = \sum_{n \in Q \Lambda n \neq f} \frac{l_n^{max}}{C} + \frac{l_f^j}{C} + \frac{\delta(C)}{C}$$

where C is the server capacity, $\delta(C)$ is the burstiness of the server, Q is the set of flows served

However, in the said SFQ scheduler, the Instantaneous nature of congestion was not accounted for and the inter-packet delay can grow very large.

Further detailed readings of schedulers available for packet networks are available in Zhang [231] and Parekh [181].

2.1.3 Discussion

This thesis work builds on the work carried out in different areas of network control and management summarised above. It can be seen that although feedback control has long been used in computer communication networks at physical, data and transport layers, automatic feedback control is not used above the transport layer. This thesis introduces a distributed control system framework, developed in order to facilitate tractable analysis, simulation and implementation.

In general, a control system can be open loop or closed loop. Open loop control systems use admission control and job scheduling etc while close loop control systems use one or many feedback signals in a centralised or distributed fashion. In a control system that is stabilised, the feedback signal of the resource indicates the condition of the resource. This feedback about the state of the network is used for pre-emptive, preventive or corrective actions. This feedback signal can also be used as a unitary value similar to the cost, which can be used by the appropriate price function models and trend analysis to price the service. A dynamic network resource management and control system that operates above the transport layer is analysed, devised and demonstrated in this thesis. This operates for a multiplicity of flows in both reservationless and reservation enabled networks, using local information and computation to achieve global optimisation within the given conditions. In order for such an implementation to be globally accepted, a form of guarantee of fairness to the multitude of users is considered an important characteristic feature. Fairness means every user gets an equitable service according to certain criteria that is applied uniformly to all the users.

On a wider appreciation, the system demonstrated in this thesis proves, in the given conditions, that a distributed set of resources/elements, with distributed feedback (with messages); can achieve a fair solution without the involvement of a central control.

2.2 Internet, System of Systems, Control theory

Although a flat network is possible to obtain (as in LANs), the Internet is organised in a multi-level hierarchy. The rule of thumb number of the acceptable number of levels if hierarchy is $ln(N)$ where N is the number of routers [127]. The number of levels, in practical terms, is decided by several factors including the cost/capacity of the routers, degree of administrative and security desired etc. The inter-networks are generally classified into three: the Internet, which is the global public network; the intranet that is private network

and the extranets that is hybrid- private networks/internetworks connected through the Internet. The Internet is, alternatively, a system of systems having three major layers 1) the access layer that provides local and remote access and services 2) the distribution layer that provides the edge functions and 3) the core layer that provides high-speed switching backbone [47]. In this work, the control theory framework is introduced to global scheduling functions in distributed computing and communication networks. A closed loop control overlay is used for regulating the sharing of the resources in this system of systems. This system eliminates the need for centralised control. This means an incremental introduction of the scheme into the inter-network is possible.

In the engineering of the network, the management system that overlays such a control system across the network will stabilise the resource management of the computing and communication network elements in a proportional fair manner. The theory holds well regardless of the tiering of the network and can be generalised to any system of systems.

2.3 Study on advantages of resource sharing

One of the advantages of IP packet based network is that the granularity of sharing the network resources is much finer than a connection based network. Therefore, the network resources are utilised much more efficiently. In order to demonstrate advantages of using a shared system, a brief comparative study of three candidate systems is given below. The three system models are a) connection oriented circuit-switched model b) reservation based datacom model and c) dynamically shared network model. In fact, the three models described represent three phases of progress in resource management technology. The study and simulations by the author uses high level abstract models of the system from the perspective of queuing theory. The service quality obtained from each of the systems is looked at, along with the demand on resource to provide that quality. The comparison between the models are given in section 2.3.3.1

2.3.1 Connection oriented circuit-switched model

In the circuit-switched model, the quality of service for the applications is at its best, for the given parameters like guaranteed bandwidth, due to the end-to-end connection with no store-and-forward. However, the availability or grade of service is characterised by the probability of blocking (proportion of calls that are rejected in the long term). In order to provide reasonable grade of service, the circuit-switched networks generally provide over-trunking. Over-trunking ratio is given by the ratio of required trunks to the average offered load s/ρ where s is the number of trunks and ρ is the average offered load [157]. For large values of offered load with a given blocking probability of 0.1%, the over-trunking ratio is about 120%, as can be seen from the following Figure 7. The standard Erlang-B formula $B(s,\rho) = \frac{\rho^s/s!}{\sum_{k=0}^{s}\rho^k/k!}$ is used in this simulation, where B is the probability of lost calls cleared.

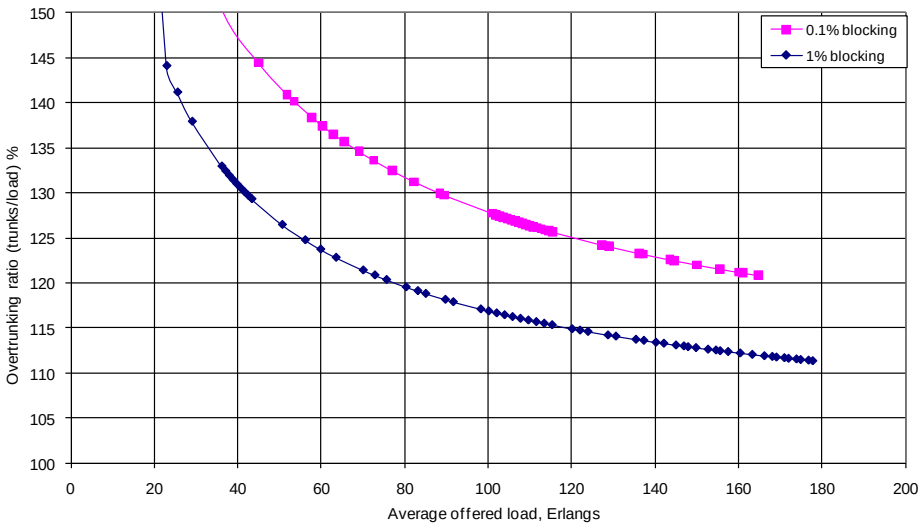


Figure 7 Over-trunking ratio in the 'circuit-switched model

Although this graph can be approximated to $1 + {}^{1}/_{\sqrt{\rho}}$ where ρ is the offered load, which shows an economy of scale for large systems, practical systems require an over-trunking ratio between 110 and 120%.

2.3.2 Reservation based datacom model

The reservation based (e.g. Intserv type) packet network can be modelled as separate router trunks as given in Figure 8:

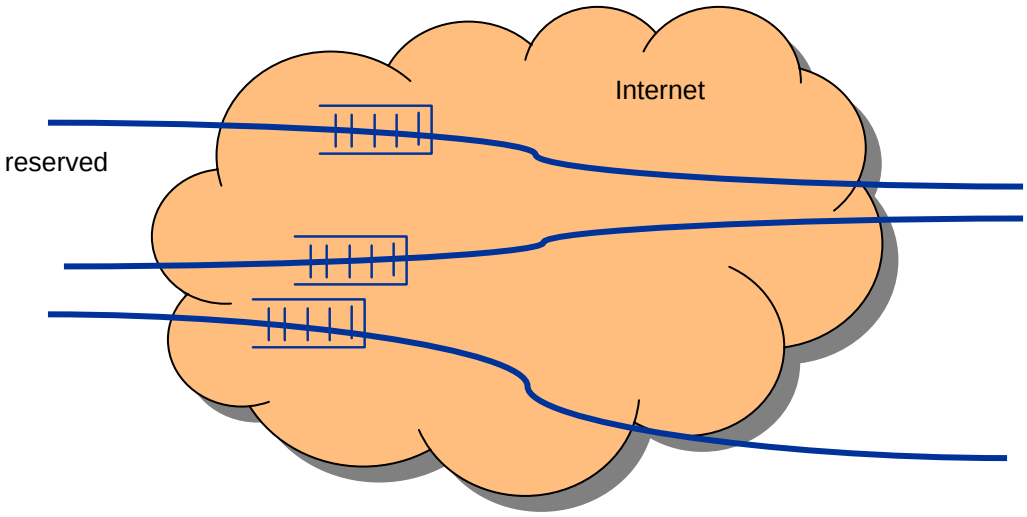


Figure 8 Reservation based packet trunks in the Internet

The packets go through a series of store-and-forward intermediate systems between the source and the destination for every channel. The simplest approximation of such a communication channel of tandem queues will be a queue with associated waiting, for a single server. Such a system could be approximated as a set of multiple M/M/1 queues. The performance is thus defined by the characteristic of an M/M/1 queue, E[k] = $\frac{u^2}{1-u}$ given in Figure 9 where E[k] is the average number of elements in the queue and *u* is the utilisation. This is a simplification of the Erlang-C[2] formula for the single

2 For an M/M/s queue, the probability of queuing

$$C(s,\rho) = \frac{\frac{\rho^s}{s!}\frac{s}{s-\rho}}{\sum_{k=0}^{s-1}\frac{\rho^k}{k!} + \frac{\rho^s}{s!}\frac{s}{s-\rho}}$$

where s is the number of servers and ρ is the offered load.

server case. At about 60% average load the average number of waiting connections is unity and from there on it increases approximately in an exponential fashion.

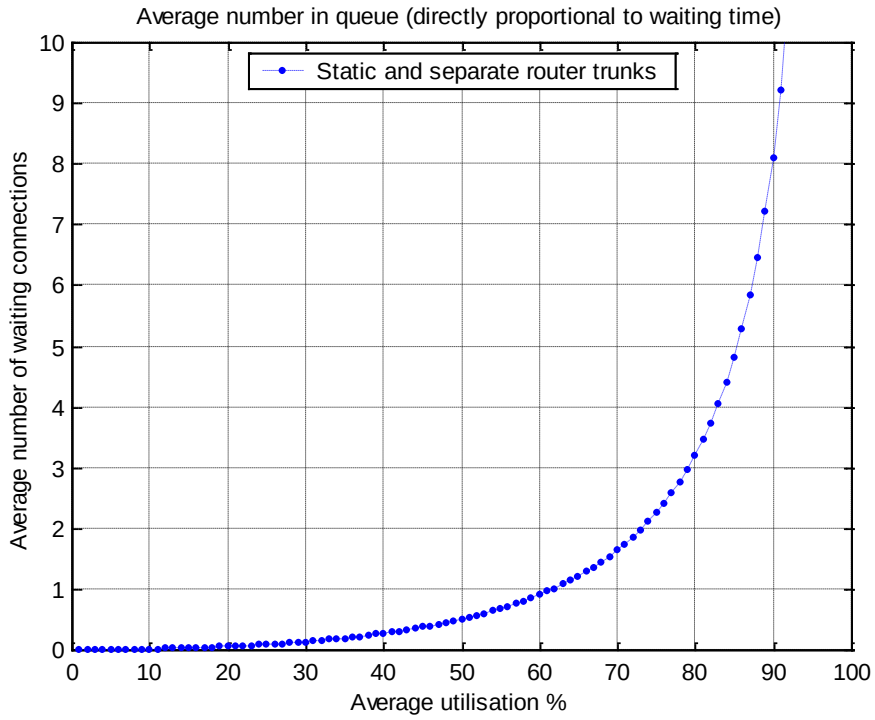


Figure 9 Waiting time in the ‘net’ model

2.3.3 Dynamically shared resource datacom model

In the distributed resource controlled (DRC) network proposed in this thesis, the closed loop feedback brings the system to dynamic load balance under controlled load. This is because the traffic is distributed within the network in accordance with the load on the resources. There are no reservations and all users share the system without static allocations. This is similar to the current Internet substrate but without the greedy flows. Greedy flows are contained within the edge-to-edge ‘traffic trunks’ and therefore such flows will be limited depending on their provisioning potential. Such a system can be approximated as a multiserver M/M/s queue as given in Figure 10:

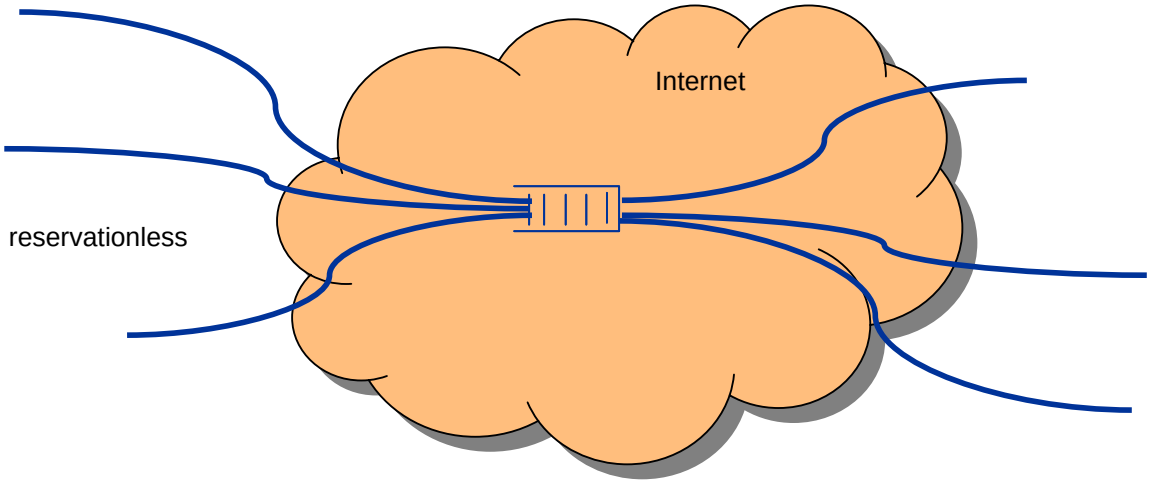


Figure 10 Limit case approximation: a dynamically load balanced system.

The characteristic of the M/M/s queue E[k] = $\frac{uC(s,su)}{1-u}$ compared with the case of the M/M/1 queue mentioned before is given in Figure 11 below:

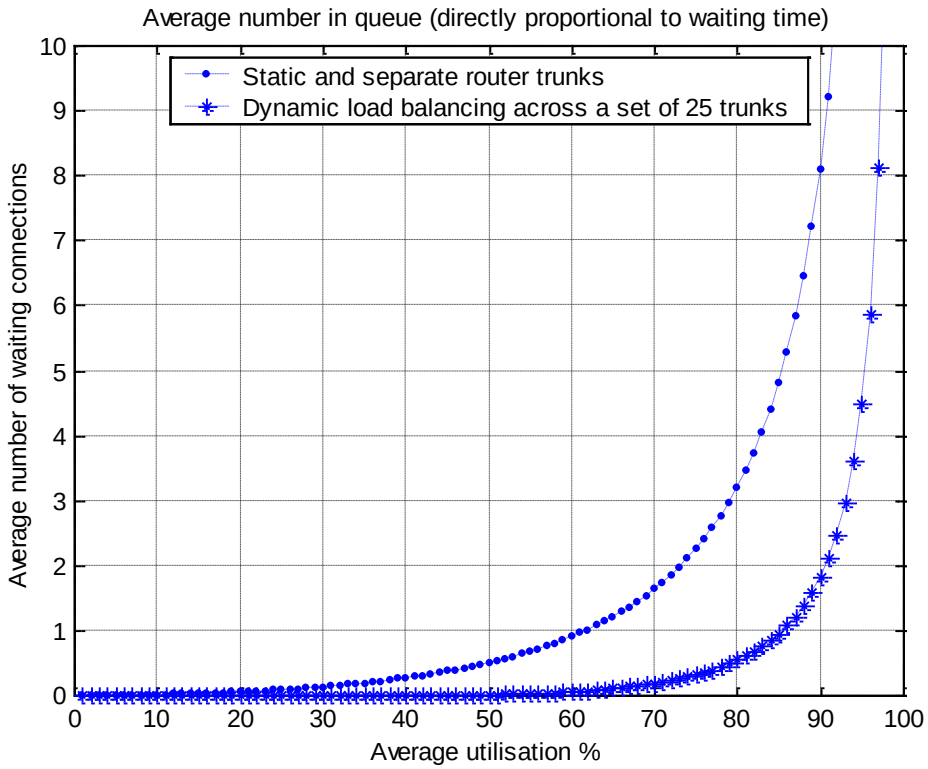


Figure 11 Improvement in the waiting time in the ‘net’ model

In the given case, at 85% utilisation, the waiting time is reduced to one fifth. As the number of servers increase, the utilisation can be further increased for a given delay parameter.

Hence, the probability that delay P[*d*] in the proposed closed-loop, reservation-less system exceeds a given value *x*, at a given loading *l* is less than the probability of delay P[*d*] in an open-loop, reservation-based system i.e. $\{P[d]_{\text{reservation-less}(l)>x}\} << \{P[d]_{\text{reservation-based}(l)>x}\}$. Although the maximum utilisation is approximated to be proportional to the inverse of the number of hops [40], the improvement seen above will hold good for any number of hops.

2.3.3.1 Dynamic sharing with multiple paths

Comparing the three models, it can be seen that

a) connection oriented circuit-switched model: at 60% load there is 0.1% blocking, 125% overtrunk

b) reservation based datacom model: at 60% load there is only 1 waiting connection in the system. In comparison with the connection oriented circuit-switched style trunks, characterised by blocking probability and over-trunking, the connection less packet data system offers savings obtained from reduced network provisioning for the same class of service.

c) dynamically shared network model: at 60% load, with sharing across 25 trunks, there is hardly any waiting connection in the system. In comparison with the ‘reservation based Intserv’/Diffserv style, characterised by queued calls through the network, the dynamically resource shared packet data system offers savings obtained from less waiting and its associated drop.

The reduction in waiting time improves the QoS as well as reduces the number of abandoned connections. This improves profit for the operator, depending on the traffic and value mix. Therefore, a dynamically shared network model is superior to the previous models.

The idea can be directly extended to sharing the multiple routes available between a given source-destination pair. The following graph in Figure 12 shows the advantage that can be gained from adaptive multipath load balancing over 25 trunks with shared queue.

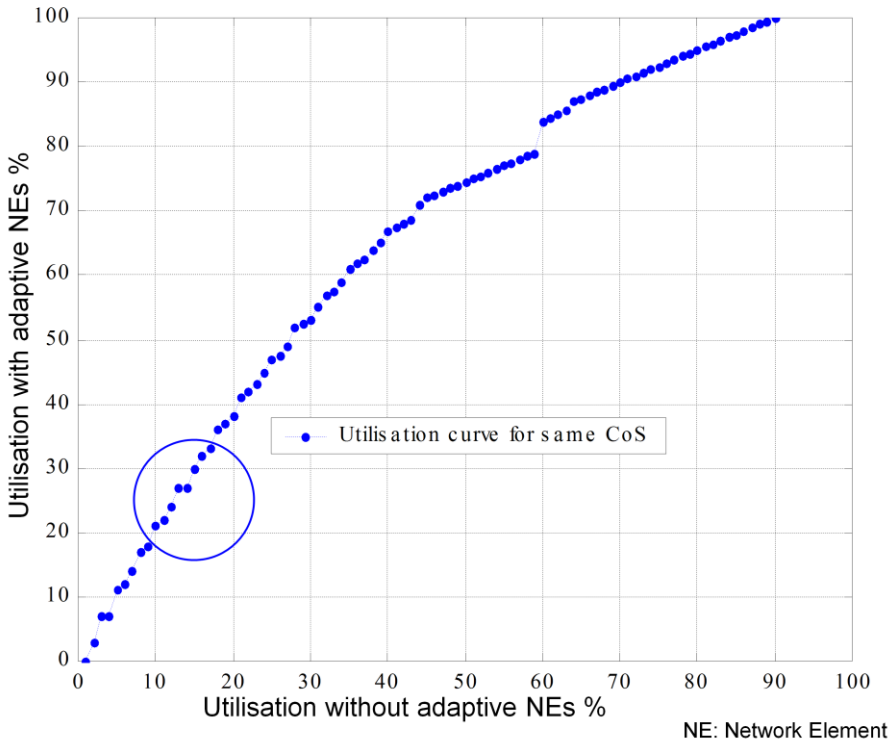


Figure 12 Route sharing and network utilisation

The curve shows the figure of merit of such a network with a given QoS parameter (in this case the number of elements in the queue), the utilisation with adaptive network elements plotted against the utilisation without adaptive network elements, for the same QoS parameter.

It is known that the current utilisation figure for the Internet is between 10 to 20% [174]. It can be seen that the network utilisation can be doubled in this regime while maintaining the same QoS.

A network overlay offering such dynamical, shared resource features can further increase the utilisation of the network by prompting the user to use the network when the network is lightly loaded. Further, with the advertisement of associated price, people can buy the bandwidth/QoS in an interactive fashion.

Further, since such a system is above the transport layer, fairness can be enforced on any type of transport protocol including torrents.

Hence, the feedback-controlled system is advantageous in many ways:

- Due to the dynamic load sharing and balancing behaviour, the proposed system provides better performance in terms of the number of satisfied customers at given CoS, with given provisioning
- Due to network usage information feedback and knowledge of customer value, the proposed dynamic system can be directly used for network state dependent charging and value based admission control
- Provides a common framework to manage user and business policies/utilities as well as network elements

3. SYSTEM ANALYSIS AND MODELLING

3.1 Control system reference model

In a computer communications network, the service/network operator and the user are the actors. These actors aim to maximize their benefit from the usage of the network. These actors would have, at any particular instant, a defined set of resources allotted to them, which is a subset of the global network substrate. The user uses a subset of the resources owned by a single or several network operators. The target requirement of the actors is to increase their respective utility of the system. For example, the user would want to improve their satisfaction (from their use of the network) at minimal expense, the network operator would want to maximise the resource usage and maximise the profit they derive from their business of network ownership. In this thesis, this primal – dual problem [117] is re-cast in the control system model. This makes the system analysis more tractable and implementable using appropriate mechanisms.

In a feedback control system, the potential target requirement (of the actor) is first captured. The resources are then controlled using an intermediate feedback variable to achieve the targets. This is mechanised by passing messages that incorporate a function of the intermediate variable between the controller and the resource plant.

A control system regulates the system resources to maintain the targets requirements.

Mapping this concept of regulation to the network operator/ traffic-aggregate scenario, this is equivalent to:

- The operator wants to fulfil the requirement of the traffic-aggregate for the different classes of traffic
    - The operator has to maintain the isolation between those classes

    - For each resource, there will be a set control level for each class of traffic
    - The operator would want the total traffic level through each resource to be maintained as close as possible to the said control levels. In other words the given ‘supply’ has to be fully consumed to maximise resource usage efficiency
- The intermediate variable is manipulated according to the difference between the resource supply and its consumption (in other words the demand) so as to reach equilibrium
    - This variable is a forcing function and is usually termed as the resource price function as is analogous to price that drives an economic system[3]. This is a low bandwidth mechanism to convey aggregate information about a set of cost functions
- In the steady state, the difference between the available resources and the consumed resources is zero; the multiple traffic-aggregates sharing the resource receive an even treatment across all of them[4].
    - These properties ensure a particular form of fairness known as proportional fairness [117].

In such a scenario, the variable set of resource requirements set by the individual actors in order to receive an allocation is akin to a potential set to receive a flow. Thus this requirement is termed the provisioning potential for the individual actor. In an economic model [132], this potential would be akin to how much those actors are willing to pay for the given resources. The provisioning potential is set by the operator. In this case, the feedback variable would be equivalent of price for that resource. These terms are useful in using the model at different layers of the system hierarchy spanning the different aspects of technology and business.

[3] Kolmogorov complexity of such a number is much less than that of having to specify all the parameters of the resource.

[4] in a time multiplexed scenario this is achieved by a weighted fair scheduler

3.1.1 Single resource, single actor, single path model with price function

A concrete simulation model[5] of such a system for a single actor with a single resource is shown below in Figure 13. The single resource/ single actor scenario in fact does not require any sharing method; however this example is given to introduce the modelling approach.

Driving on the similarity from the economic system where the price of an item is determined by its demand/supply, the controller output in this context is termed ‘price’, which is in effect a ‘price-like variable’ and is not connected to any currency. Generally, this parameter provides an indication of the demand the system is experiencing and can be mapped on to a suitable parameter in any given domain. This allows the users (in this case edge routers) of the system to decide on their responses as to how to bid, expend their potential and share the common resources.

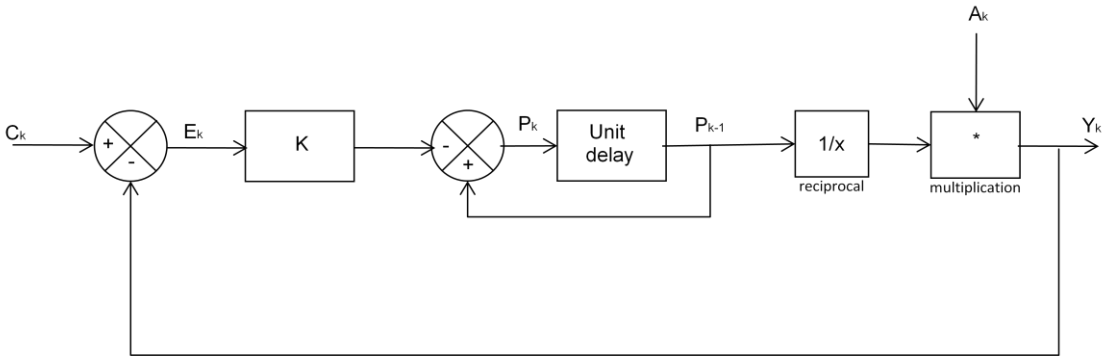


Figure 13 Generic model of the proposed resource control system

For the model above, the following parameters can be defined:

C_k Set point for control, in ‘resource unit’, set by the n/w operator

E_k Difference variable, in ‘resource unit’. The difference between the set

[5] Implemented in matlab/simulink tool

point and controlled output is given by $E_k = C_k – Y_k$

$_k$ Sample step number, iteration steps for computation

K Proportionality multiplicand for the ‘availability figure’ of the resource, its value controls the stability with time. This parameter is scheduled into the controller based on simulation results

P_k Resource price function in ‘price unit for resource unit’

A_k Potential willingness to Pay by the user (in this case edge router), in ‘Provisioning Potential unit’ (later termed as PP_k)

Y_k Output level, in ‘resource unit’

3.1.1.1 Control system equations and transfer function:

The difference equations for the control system model given in Figure 13 could be written as

$$P(k) = P(k-1) - K.E(k) \quad (1)$$

$$Y(k) = \frac{A(k)}{P(k-1)} \quad (2)$$

Taking the z-transform of (1)

$$P(z) = P(z).z^{-1} - K.E(z)$$

$$P(z) = -\frac{K.E(z)}{(1-z^{-1})} \quad (3)$$

Taking the z-transform of (2)

$$Y(z) = \frac{A(z)}{P(z).z^{-1}} \quad (4)$$

Substituting (3) in (4)

$$Y(z) = -\frac{A(z)(1-z^{-1})}{K.E(z).z^{-1}}$$

$$K.E(z).z^{-1}.Y(z) = -A(z).\ (1-z^{-1})$$

$\big(C(z) - Y(z)\big).K.z^{-1}.Y(z) = -A(z).\ (1-z^{-1})$, as $E(z) = \big(C(z) - Y(z)\big)$

$$Y^2(z) - C(z).Y(z) - \frac{A(z)}{K}.(z-1) = 0 \quad (5)$$

The transfer function $Y(z)$ can be obtained by taking the impulse response of (5). Input $C(k) = \delta(n)$, the z transform of $C(k)$ is $C(z) = 1$

$$Y^2(z) - Y(z) - \frac{A(z)}{K}.(z-1) = 0$$

$$Y(z) = \frac{1 \pm \sqrt{1 + 4\frac{A(z)}{K}(z-1)}}{2} \quad (6)$$

3.1.1.2 Stability of the resource control system

Lemma 1:

The given resource control system model shown in Figure 13 is always stable so long as the provisioning input $A(z)$ has no poles i.e. $A(z) = a + b(z) + C.z^2 + ..$

Proof:

It can be seen that the transfer function (6) has no poles as K is a constant, if $A(z)$ has none. Therefore, the control system is always stable if $A(z)$ is bounded.

When there are poles in a system, stability is guaranteed when $|z| < 1$

It may be observed that if $A(z)$ is a step function, $(z) = \frac{1}{(1-z^{-1})}$; the pole is cancelled by the numerator's $(z-1)$. Therefore, the system is stable for step change in $A(z)$.

3.1.1.3 Resource price function block

The inner control loop is a 'deccumulator', a concept introduced in this thesis for price function updates.

This update algorithm introduced in the controller provides a tractable model in the given context and is called the price function block. What this

does is to introduce the concept of a price-like variable in the controller where the price function parameter tracks the resource consumption.

It may be observed that the resource price function block receives a proportion of the resource availability (difference between the supplied resource and their usage) as negative feedback for price function stabilisation. This is done so that as the availability increases, the price must go down and vice versa.

3.1.1.4 Resource sharing block

In the following resource-sharing block, the resource price function later denominates the resource consumption[6] to each user (in this case edge router). The numerator is the provisioning potential that controls the transfer function. Sharing of the resources in this way is akin to a commercial transaction. The allotted share of resource usage is regulated by the set point.

3.1.1.5 Initial and boundary conditions

The initial and boundary conditions are set as follows. The initial operating condition for the resource price function (P_0) is set by the network operator and forced on the system. In this case, P_0 is set to unity. The output of the resource price function is bounded to upper and lower limits to define a sustainable operating regime. Likewise, the resource consumption is also bounded. The feedback ensures sustained operation within the boundary conditions.

[6] in this case ingress traffic flow allocation

3.1.2 Multiple resources, actors and paths model with price function

In the real scenario, there are multiplicity of resources, actors, aggregates and paths. There are a number of ways in which such a system can be modelled.

3.1.2.1 Model for simulation and development

However, it is important that such a model should be tractable and can demonstrate the concepts developed in this thesis. Further, this model should yield itself to further synthesis. With this in mind, a three-flow – two-resource model shown in Figure 14 is developed, where the two resources are the core routers. It is further shown that this model is sufficient to demonstrate the principles developed in this thesis as well as the interactions between the component flows.

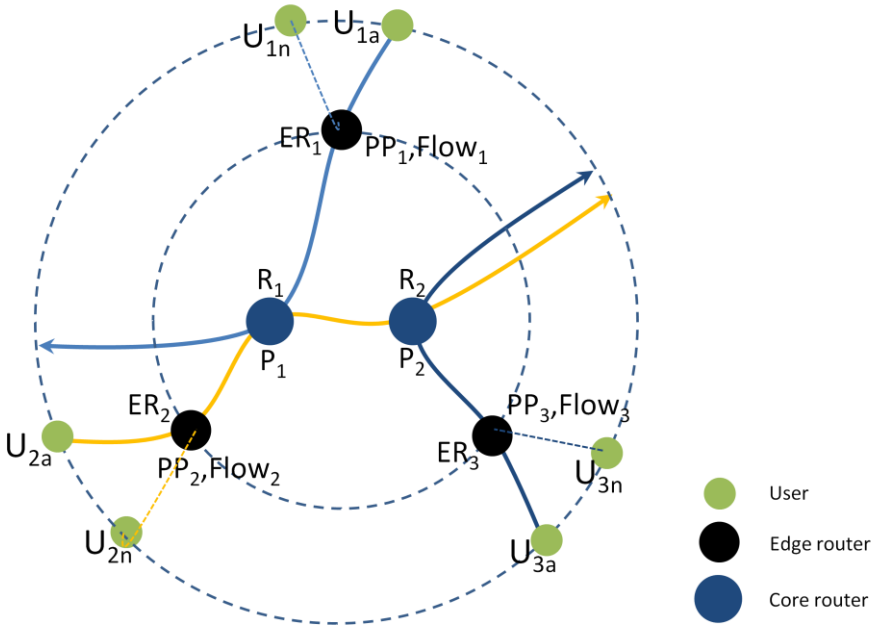


Figure 14 Simulation reference model using multiple resources and actors

The edge routers serve a set of individual users from its captive area. The edge routers can utilise any of the core routers they have business agreement with. In the given model, the edge router ER_1 uses core router R_1, the edge

router ER_2 uses core routers R_1& R_2, the edge router ER_3 uses core router R_2. This arrangement is used such that the interactions between the constituent flows could be readily seen during the analysis and synthesis. In addition to the flows identified in Figure 14 above, couple of 'disturbance flows', D_1 using both resources R_1& R_2 and D_2 using only resource R_1 will also be used in the analysis and synthesis. It may be noted that the edge routers aggregate the user traffic per traffic path.

A concrete simulation model of such a system is shown in Figure 14 below. The model is built within the matlab/simulink system so that various simulations can be run and stability, sensitivity etc assessed. The two core routers R_1& R_2 are shown to the right of the diagram. The core routers receive traffic from the edge routers ER_1, ER_2 , ER_3. The core router is also shown to be receiving traffic from a 'disturbing source' D. This is to simulate scenarios where a non-confirming flow loads the core routers without following the edge-router mechanism presented in this thesis. The 'monitors' shown at the various output blocks are from observing the data generated. The data generated are then used in various graphs provided in this thesis. It can be seen that the aggregate traffic sent to the core routers by each of the edge routers depends on the ratio of the provisioning potential PP and price P. The price function is calculated at the deccumulator. The deccumulator takes its input from the controller K, which is fed with the difference of the core router capacity and core router utilisation. The dynamic state of the core routers is available from the network information base. Analytical proof of this model is given in the following section.

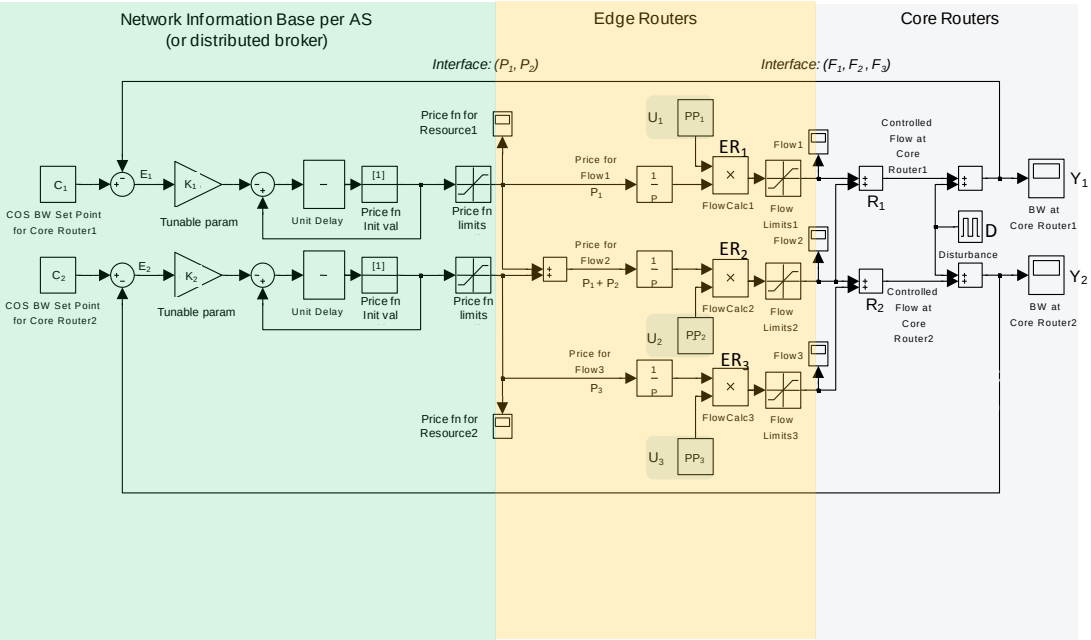


Figure 15 Control system reference model for multiple resources, actors and paths

The results of the simulation are given in section 4.2 in this document.

3.1.2.2 Analytical proof with multiple elements

Let the total number of resources be R_T and the total number of traffic aggregates be N_T (active paths). The parameters of interest in section 3.1.1 are then modified to incorporate the different sets of resources, traffic aggregates and paths as follows:

C_{Rk}	Set point for control for resource R, k is the iteration number
E_{Rk}	Difference variable for resource R
K_R	Proportionality multiplicand for availability figure of the resource R, this is scheduled into the controller based on simulation results
N_R	Traffic aggregate using resource subset *R*
N_T	Total number of traffic aggregates (active paths)
P_{Nk}	Total resource price function for traffic aggregate N
PP_{Nk}	Provisioning Potential for the traffic aggregate N (formerly A_k)
PP_{Rk}	Total share of Provisioning Potentials for resource R

P_{Rk} Resource price function for resource R

R_N Resource subset used by traffic aggregate N

R_T Total number of resources

U_N Resource consumption by traffic aggregate N (e.g. Flow rate of active path)

U_R Total resource consumed by users (in this case edge router) in a given resource R

V_T Number of paths (or combinations)

Y_{Rk} Output level of resource R

The equations in section 3.1.1 get modified as follows:

$$E_{Rk} = C_{Rk} – Y_{Rk} \quad (7)$$

$$P_{Rk} = P_{Rk-1} – K_R E_{Rk} \quad (8)$$

With N_T traffic aggregates using the resources in V_T paths, the resource price function for each traffic aggregate is calculated as:

$$P_{Nk} = \Sigma P_{Rk},\ R \in R_N \quad (9)$$

The resource consumed by traffic aggregate N is given by:

$$U_{Nk} = PP_{Nk} / P_{Nk} \quad (10)$$

The output level is now given as:

$$Y_{Rk} = U_{Rk} = \Sigma U_{Nk},\ N \in N_R \quad (11)$$

Lemma 2:

The multi-loop feedback control system drives the system to fully use up the available resources.

Proof:

When the multi-loop feedback control system is in equilibrium, $E_{Rk} = 0$.

From (7)

$$Y_{Rk} = C_{Rk} \text{ when } E_{Rk} = 0 \quad (12)$$

From (11) & (12),

$$\Sigma U_{Nk} = C_{Rk} \quad (13)$$

This means that the resource consumption by different users (in this case edge router), summed up at the resource R is equal to the capacity of the resource. It follows that the actors fully use-up the resources.

3.1.2.2.1 Proportional allocation and fairness[7]

In other words, no resource is left un-allocated; all the resources are used up and shared according to the provisioning potential of the users (edge router). It may also be observed that all the users pay the same price, as opposed to the pricing schemes in a bid scenario where the tender offers differ. These are properties of a proportional fair system [117].

The proportional allocation of resources for individual traffic aggregates is proved as follows:

The provisioning potential *PP* for any given traffic aggregate is divided across the resources that the traffic aggregate uses. The utilisation of any given resource is given by

$$U_{Rk} = PP_{Rk} / P_{Rk} \quad (14)$$

When the resource is fully used, U_R equals the capacity of resources C_R. In this case $P_{Rk} = PP_{Rk} / C_{Rk}$ (15)

[7] As far as the author is aware, results (16) and (18) shows a new way of presenting proportional fairness, and more tractable, when contrasted with the available literature

As the resource price is equal for all the traffic aggregates using the given resource, substituting (15) in (10), for the case of full utilisation,

$$U_{Nk} = (PP_{Nk} / PP_{Rk}). C_{Rk} \quad (16)$$

The result (16) shows that the service obtained by the traffic aggregates is proportional to their provisioning potential[8].

The fairness across individual traffic aggregates is proved as follows:

From (16), considering the ratio of resource allocation to provisioning potential,

$$U_{Nk} / PP_{Nk} = C_{Rk} / PP_{Rk} \quad (17)$$

Since C_{Rk} / PP_{Rk} is a resource related quantity and applies to all the traffic aggregates,

$$U_{1k} / PP_{1Rk} = C_{Rk} / PP_{Rk} = U_{2k} / PP_{2Rk} \quad (18)$$

for infinitesimally divisible fluid flows.

The result (18) shows that the service obtained by the traffic aggregates is fair across the traffic aggregates.

3.1.2.2.2 Distributed multi-node macro-scheduling

It may be observed that result (16) is similar to the WRR algorithms present in the literature [78]. WRR algorithms operate on a single resource, e.g. a link. However, it may be observed that result (16) extends the WRR theory to work across multiple resources. The provisioning potential is distributed and shared across the multiple nodes and the scheduling decision takes care of global schedulability across the multiple nodes. Therefore, the system described is a top-level distributed multi-node scheduling system, working above the individual node schedulers.

[8] This shows that the price-function is only an intermediate variable and that the resource allocation is entirely determined by the provisioning potentials and the capacity of the resource

It may be noted that the intermediate variable called resource-price disappears from the resource allocation equation (16)

3.1.2.2.3 Isolation and scalability

The above analysis shows that the traffic aggregates compete for the resource share within the distributed network independently and receive their fair share depending on their provisioning potential. This means each control loop operates independently and the system maintains isolation between the traffic aggregates.

Scalability is the ability of the system to deliver the required performance as the network grows. The growth could be either vertical or horizontal.

Vertical scalability means the size and number of flows using given set of resources increase. As the flows are aggregated in the edge router, increasing the capacity of the edge router e.g. the CPU, I/O etc will resolve this.

The horizontal scalability means the number of routes and resources in the network increase. It can be seen that no co-ordination between the routers are required for making resource allocation decisions by individual edge routers. The only information required is the network load information from the network information bases. As the number of administrative domains is limited, this does not pose a problem either. Therefore, the system is scalable to multiple layers of hierarchies.

3.1.2.3 Traffic aggregation

The following is a note for completeness: In cases where multiple individual users contribute to traffic aggregates, the aggregation happens as follows:

N	Number of users making N traffic aggregates
N_T	Total number of traffic aggregates
P_{Nk}	Total resource price function for traffic aggregate N
P_{nk}	Price function for individual user making up total aggregate resource price P_{Nk}
PP_{Nk}	Potential willingness to Pay by the traffic aggregate N
PP_{nk}	Potential willingness to Pay by user making up aggregate PP_{Nk}
U_N	Resource consumption by traffic aggregate N
u_n	User resource consumption making up aggregate consumption U_N
V	Number of individual paths making V aggregate paths
V_T	Total number of paths

$N \in n$

$P_{Nk} = \Sigma P_{nk}$

$U_N \in u_n$

$V \in v$

$PP_{Nk} = \Sigma PP_{nk}$

3.1.3 Features of importance

The following features can be noted from the analysis model.

3.1.3.1 Resource price function

When the system senses that the utilisation of the resources is less than its capacity supply (say in state 1), the output of the price function is reduced (in state 2) so as to get the users (in this case edge router) to consume more resources.

Comparing the two states 1 and 2, when $Y_{R1[1]} < C_{R1[1]}$,

$P_{R1[1]} < P_{R1[0]}$

3.1.3.2 Resource utilisation gain

When the output of the price function is reduced (say in state 1) so as to get the users consume more resources, the feedback control ensures better resource utilisation (in state 2). This improvement in resource usage is called Resource utilisation Gain

Comparing the two states 1 and 2, when $P_{R1[1]} < P_{R1[0]}$,

$Y_{R1[2]} > Y_{R1[1]}$

3.1.3.3 Resource revenue

It may be observed that for any resource R, for a given sum of PPs, the revenue for that resource (product of total resource usage Y and resource price P) remains same regardless of the state as long as there is demand for the excess capacity. This product equals the sum of PPs for that resource.

Comparing the two states (1) and (2),

$$Y_{R1[1]} * P_{R1[0]} = PP_{N1[1]} + PP_{N2[1]} /2 = Y_{R1[2]} * P_{R1[1]} = PP_{N1[2]} + PP_{N2[2]} /2$$

Thus the revenue from the resource is guaranteed despite the variation (in above case, reduction) in price; so long as there is demand.

3.1.3.4 Operator revenue gain

The following shows how the self-adjustment mechanism maximises revenue across the multiplicity of resources in the given conditions.

Due to the resource price function, which has an adjustment mechanism that invites more traffic, the total resource price of the network reduces, and the operator is able to input more traffic into the network, increasing the utilisation of the resources. This extra traffic is permitted to be input at no extra cost to the operator. In this case,

$[Y_{R1[2]} \ Y_{R2[2]}] > [Y_{R1[1]} \ Y_{R2[1]}]$ and

$[U_{N1[2]}\ U_{N2[2]}\ U_{N3[2]}\ ] > [U_{N1[1]}\ U_{N2[1]}\ U_{N3[1]}]$

If the user is paying a real price per unit share of resource (unit bandwidth in this case), the extra revenue to the operator from the user can be calculated as follows. *CP* here denotes the real charged price. (For convenience of calculation, *CP* is taken as the PP of individual users and remains same at both step1 and step2).

Difference in revenue from step1 to step2 = revenue in step2 – revenue in step1

$$= \left[CP_{N1[2]}\ \mathrm{CP}_{N2[2]}\ \mathrm{CP}_{N3[2]}\right]\begin{bmatrix}U_{N1[2]}\\U_{N2[2]}\\U_{N3[2]}\end{bmatrix} - \left[CP_{N1[1]}\ \mathrm{CP}_{N2[1]}\ \mathrm{CP}_{N3[1]}\right]\begin{bmatrix}U_{N1[1]}\\U_{N2[1]}\\U_{N3[1]}\end{bmatrix}$$

Since, $[U_{N1[2]}\ U_{N2[2]}\ U_{N3[2]}\ ] > [U_{N1[1]}\ U_{N2[1]}\ U_{N3[1]}]$ and *CP* remains the same, revenue in step2 > revenue in step1

This improvement in operator revenue is called operator revenue gain. This revenue can then be utilised by the operator for his benefit/ passing on to individual users according to business strategy.

3.1.3.5 Fairness to user traffic flow

Lemma 3 :

The allocation of bandwidth to the set of users (in this case edge routers) in the given network is proportionally fair.

Proof:

As evident, each traffic aggregate traffic flow pays the same resource price per resource unit consumed. In equilibrium, reached at step 2:

$PP_{N1[2]} / U_{N1[2]} = (PP_{N2[2]} /2)/ U_{N2[2]} = PP_{N3[2]} / U_{N3[2]}$

This property observed at step2 shows that the distribution is proportionally fair [117]. Users of each resource pay equal resource price per resource used. The feedback control provides proportional fairness and eliminates the possibility of cheating.

3.1.3.6 Value added service protection

Now suppose that user1 is a more valuable user. The service manager increases the provisioning potential for user1 when an unexpected reduction in throughput is sensed (e.g. when the price for the resources goes high). In this case, the allocation for other user2 is further reduced to maintain the provisioning for user1.

It may be noted that the method of increasing the provisioning potential for a given user should be compatible with lemma 1.

3.2 Stability of the multi-loop system

It has been demonstrated in section 3.1.1 that the system is always stable so long as the function of provisioning potential input has no poles.

For the multiple resources, actors and paths model, the very nature of the system like the distributedness, adaptiveness, uncertainty in the time delays, inaccuracy of available instantaneous data etc makes it hard to analyse the stability of the system in the traditional control systems model e.g. using the z transforms. Although the resource provisioning decisions on one router affects flows on another, it may be noted that this type of interaction is akin to that in a WRR system where the allocation to one flow affects the other. The system described in this thesis is a distributed multi-node macro-scheduling

similar to WRR (see section 3.1.2.2.2). Therefore, a mechanism for further modelling and analysis is available although it is left for future work.

However, these types of issues were understood earlier by Lyapunov [72]. Lyapunov's Second method provides a tool to analyse such systems. Rather than depending on the unknown system parameters, the Lyapunov's Second method uses a Lyapunov function to assess the stability of the system. Lyapunov function uses the finite nature of the physical system to provide conclusions about the stability, for e.g. in physical systems, a Lyapunov function is the system energy (as a physical system can only store a finite energy[9]); in economic systems, it is the cost; and for computational systems, it is the 'error'.

3.2.1 Lyapunov Stability

If there exists a Lyapunov function, $V : \Re^O \rightarrow \Re$, defined in a region of state space near a solution of a dynamical system such that

1. $V(0) = 0$
2. $V(x) > 0 : \forall x \in O ; x \neq 0$
3. $V(x(t_{i+1})) - V(x(t_i)) = \Delta V(x) \leq 0 : \forall x \in O,$

then the solution of the system is said to stable in the sense of Lyapunov.

$x = 0$ represents a solution of the dynamical systems and $\Re^O$, O represent the output space and a region surrounding this solution of the system.

3.2.2 Asymptotic Stability

If in addition to conditions (1) and (2) of the definition in section 3.2.1, the system has a negative-definite Lyapunov function

$\Delta V(x) < 0 : \forall x \in O$

then the system is Asymptotically Stable.

---

[9] For example, in a mechanical system if it can be shown that the energy is always being dissipated except at the equilibrium point, then the system finally reaches equilibrium when the energy is gone.

Asymptotic stability adds the property that in a region surrounding a solution of the dynamical system, trajectories are approaching this given solution asymptotically.

3.2.3 Global Asymptotic Stability

If in addition to conditions (1) and (2) of the definition in section 3.2.1, the Lyapunov function is constructed such that,

$\lim_{t\to\infty} V(x) = 0$,

over the entire state space, then the system is said to be Globally Asymptotically Stable.

The difference between asymptotic stability and global asymptotic stability is the fact that the latter implies any trajectory beginning at *any* initial point will converge asymptotically to the given solution, as opposed to the former where only those trajectories beginning in the neighbourhood of the solution approach the solution asymptotically.

3.2.4 Stability of the given system

The error function E_k in the system under discussion satisfies the conditions of the Lyapunov function, as it can be seen that

$Y_{R1[2]} > Y_{R1[1]}$ and $E_{R[2]} < E_{R[1]}$

This function achieves stability in one or more cycles. This demonstrates that the system described is stable in the sense of Lyapunov.

3.3 Sensitivity

The variation in the proportionality multiplicand K has its effect on the resource price function, as $P_{R[1]} = P_{R[0]} – K_R E_{R[1].}$ When the resources are fully utilised (resources to be shared have hard physical limits to their availability),

the output of the price function stops changing as the second term in the RHS goes to zero.

A lower value for the proportionality multiplicand will gradually settle to the optimum value within the given conditions. However, this will take additional cycles to settle[10].

A larger value for the proportionality multiplicand would settle the price function at a sub-optimal level, due to the coarse gradient jump in the price function update. Note that although the settled price function level is lower, the users do not get more of the resources than their proportional share. This is taken care of by the fair scheduler of the resource, explained in section 5 in this document.

3.4 Robustness

As can be seen from the preceding sections, the system under discussion is robust to perturbations. It can also be seen that the system is robust to expansions in the structure as well as to the inclusion of multiple layers.

The sensitivity can be decreased by robust design techniques as well as by having a ‘robustness monitor’ to monitor whether the price function had settled to a level that is different from the optimal value.

3.5 Theorem of de-centralised/ distributed feedback and network optimisation

The theoretical understanding developed so far could be summarised as follows.

[10] Techniques from the neural networks learning process like the momentum factor (a factor of the previous price change) could be used here to speed up.

A networked system will optimize its resource usage within the given conditions and achieve proportional fair solution for the whole of the network without the need for any other additional co-ordination by the decision-making unit so long as information regarding the state of the set of networked devices is used when making decisions that affects that set.

Proof:

Lemma **1**, Lemma 3, Lemma 3 and the given description proves the theorem by induction

3.6 Value added products and services

From the outset, the purpose of a communication network system is to provide services that users want, in order to satisfy their utilities. The services are built on top of the networked systems that enable those services. The differentiation between the service delivery part and network operation part is described in section 3.7. It can be seen that these two parts are two distinct and separate sections of the communications system, which lend themselves to layering.

3.6.1 Isolation

Traditionally the features of telecommunications services were embedded into the equipment at the user end as well as the operator end e.g. the customer handset, infrastructure switch, provisioning platforms etc. This made it difficult to introduce new services and features. The control system framework helps to model the different layers of the system of communicating systems separately and provides tractable solutions to each closed loop while maintaining their isolation, without interference between them. Thus, operators are free to adopt independent policies (subject to provisioning potential being bounded) to maximise the overall value of the service without interaction or interference with network resources.

3.7 System architecture of a multi-layer communication network

Firstly, a control layer for the fair sharing of the underlying network resources (CPU time slots, frequency, links, storage memory etc) is developed using a pareto-optimal[11] control system. Secondly, to enforce the control on how the real-time requirements of the different classes of services[12] are satisfied, a job scheduler is developed. Thirdly, the service layer that provides the ability to sense user utilities[13] and provide charging interfaces where different product bundles can be formulated based on the economic policies (including game theories) is developed. The system provides a framework for negotiating between the constituent elements according to the policies, and provides fair solutions to its users (in this case edge routers) across the different layers of the network hierarchy. The tractability of the control system approach helps to monitor this easily.

Figure 16 shows a layered architectural view of the communication network. The service layer consists of different hosts running various applications. It uses the IP service provided by the service provider (service provider uses the network provided by the network provider).

The service provider delivers IP services to an aggregate of hosts with a quality level decided by the value it places on the user SLA.

Inside the network ingress router, packets arriving from the users are classified according to their route and buffered at the output port. To a first approximation, the quality of service delivered for services using a particular route can be sensed from the buffer backlog at the output ports. This backlog

---

[11] i.e. no further resource allocation possible in the system that can make one party/criteria better off without making another party/criteria worse off [173]

[12] different classes of service for DiffServ are EF (expedited forwarding, low loss/latency traffic), AF (assured forwarding, assured delivery under conditions), BE (best-effort forwarding)

[13] Customers are only required rank one bundle of service over the other in order to reach equilibrium, no need to measure 'utility' [140]

is measured by setting a small virtual threshold, for faster measurement and remedial purposes.

The QoS controller uses a provisioning potential to control this backlog. The more the backlog, the less the quality of service delivered. The provisioning potential of the queue is now raised as an attempt to increase the service (bandwidth) provisioning. The provisioning potential is raised (e.g. a step change) so long as the queue contains valuable traffic, until it reaches absolute limits specified. It may be noted that the method of increasing the provisioning potential for a given user should be compatible with Lemma 1. Valuable traffic contains individual users whose provisioning potential is raised according to certain policies decided by the management.

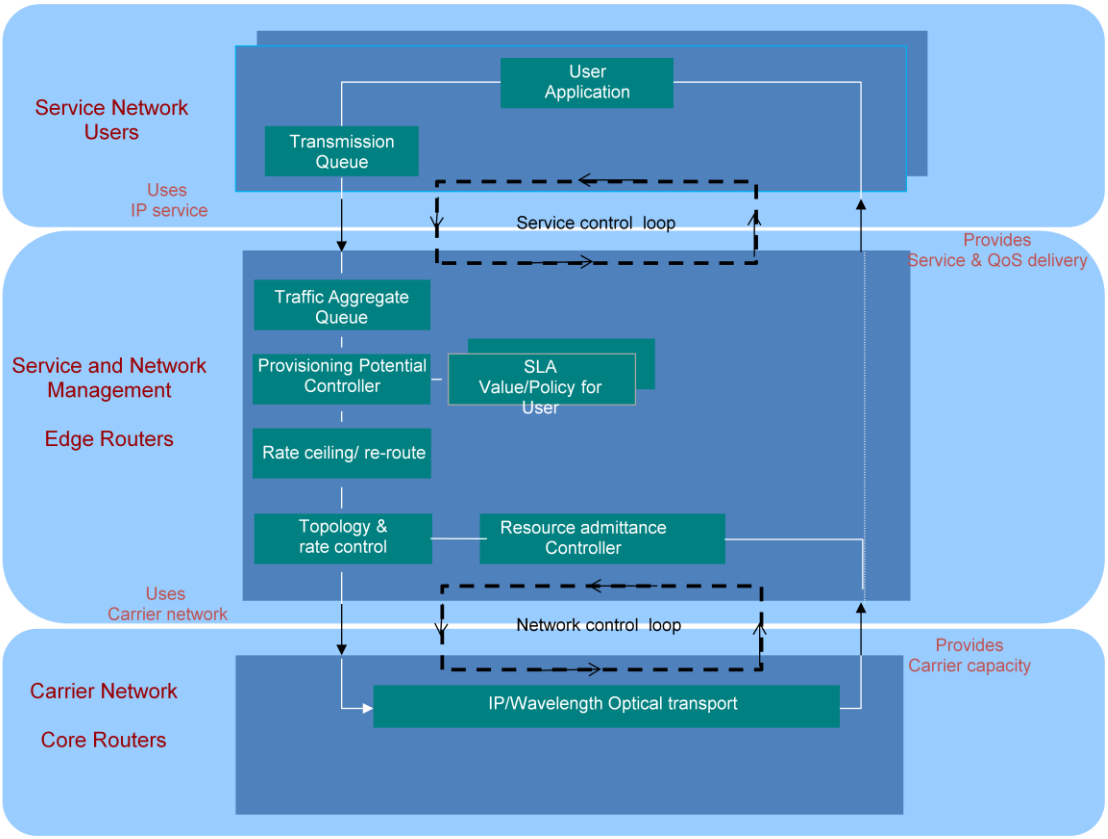


Figure 16 A three-layer model of the communications network

If the route receives more bandwidth (in terms of lines, wavelengths or radio channels), the buffer backlog reduces immediately. If the system is unable to provide more bandwidth, it attempts to use additional routes.

The network management part of the service provider uses the carrier network for transport. The loading per route is controlled by the availability of carrier capacity. The carrier network provides availability information to the service and network manager.

As the usage of the carrier capacity is increased, the admittance controller reduces the loading, such that congestion is avoided. If the usage is less, more traffic is input to the network. This way the resources are always used efficiently. A proportional fair sharing of the resources is achieved using a distributed control system that is employed to control the network load.

3.7.1 Service and Network layers

As shown, the service layer (Customer-Operator loop) deals with the provisioning of user services. The network layer (Operator-Network loop) deals with the provisioning of the network equipment.

The service layer uses the network layer. Each of these layers could have further layers within them. For example, the network layer could have an IP layer, an (optional) MPLS layer, a wireline/wireless/optical layer etc. The user/provider relationship in multi-layer systems holds good for multiple levels of system hierarchies as given in Figure 17. The relationships generally map to sequential levels of abstraction. The transfer functions f, g and h of each layer maps to inter-layer adaptation functions f(x,t), g[f(),y,t], h{g[],z,t} as the level goes higher.

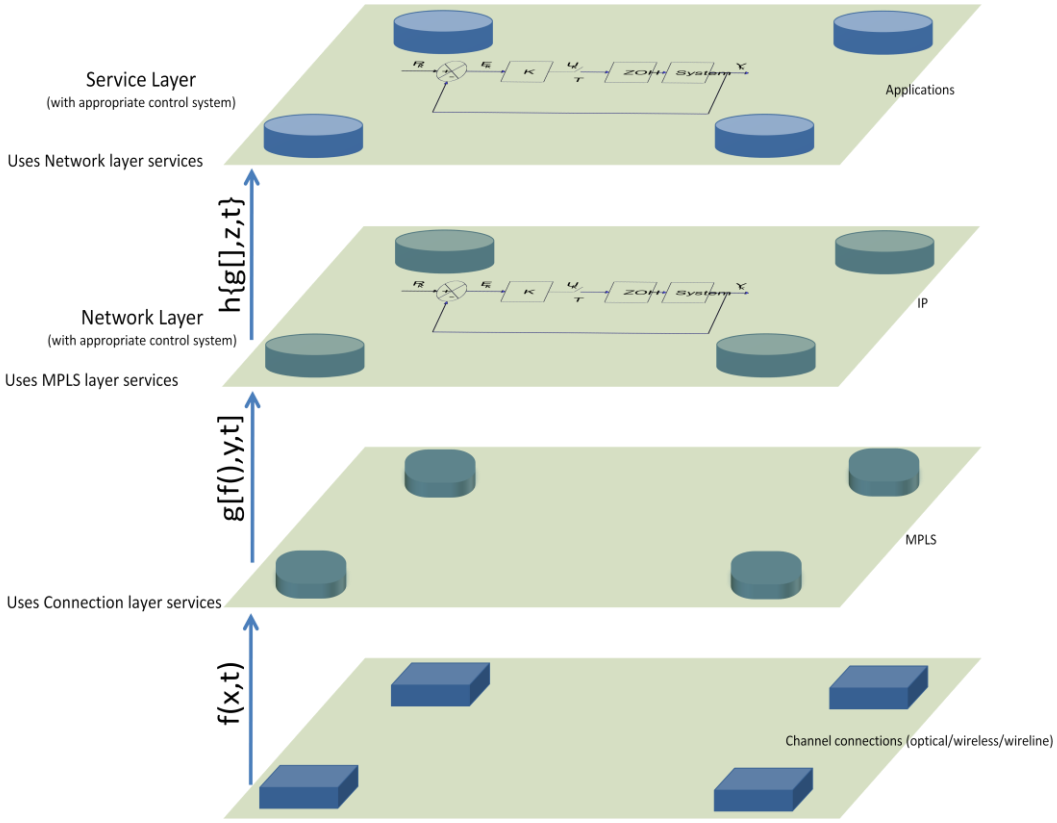


Figure 17 Cross-layer optimisation & in-layer control system

Each of these layers can have their own control system for in-layer resource optimisation within the given conditions. In such a system, a user layer operating with the resource availability levels in the provider layer can provide cross-layer optimisation within the given conditions. The multiple resource allocation layers and their inter-layer adaptation sub-layers provide a dynamic resource allocation system and cross-layer optimisation within the given conditions. Such a system satisfies the traffic demands of end user at faster pace due to automated negotiations.

These service and network layers have different types of economics associated with them. The service layer is like a product market where the price offering depends on the business proposition to the particular customer. The network layer is similar to a facility market where each user pays the same charge for using the service.

P_{Nk} and PP_{Nk} are parameters in the carrier network level for traffic aggregates and are decoupled from each other. These are appropriately decided, negotiated, distributed and mapped to the real world customer offers based on the economics of service management. It could be a linear or non-linear relationship as decided by various policies defined by the operator. In a measurement-based system, the operator can dynamically modify PP_{Nk} to satisfy user requirements. A value based provisioning could use the similar type of controllers used in the control loops.

3.7.1.1 Service layer

The operation at this point of presence level is mainly decided by the business policy. It is defined by how individual SLAs are made, how the utilities are assessed and distributed and how the individual actors are charged. This is like a customer ‘product market’ where the strategy is to enhance the revenue from the individual customers. In here, the individual users could be paying different prices depending on the value proposition.

Although this thesis does not deal with the service layer activities mentioned above, it may be noted that the system proposed can be used to facilitate a clearinghouse for service and network owners. For example, the end customers receive money from their own sources and give it to the service operators in return for the services and the service operators in turn give it to the network operators in return for the network resources. This closed-loop has its own regulations and it will not be possible for any party to act arbitrarily. For example, an ingress router cannot raise the provisioning potential for a given set of traffic arbitrarily as a) it has to receive money from the end customer and b) it has to pay to the network operator. Such arrangements also help to contain the so-called ‘Slashdot effects’ that tend to create huge amount of flash-traffic for websites.

At a static level, there will be hard limits on the resource usage anyway defined in the SLAs.

3.7.1.2 Network layer

It may be noted that in the case of the operation dealing with the network resources the strategy is to maximise the resource utilisation, which is akin to ‘facility market’. In here, every user pays the same price to use the facility.

4. SYSTEM ARCHITECTURE SIMULATION AND RESULTS

The system architecture for the de-centralised and distributed network management system using distributed feedback is simulated in this chapter. The results of the simulation of the architecture are also given, demonstrating the value based QoS as well as network optimisation within the given conditions.

4.1 Model used for simulation and software development

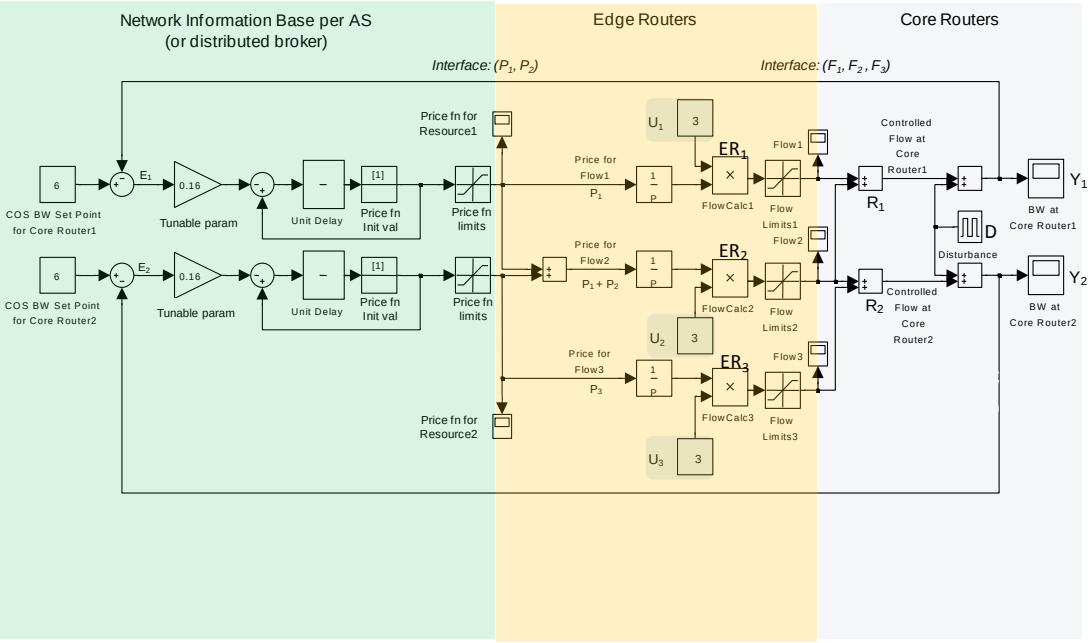


Figure 18 Simulink diagram of the three flows/two resources system reference model with parameter values

A simulation model of the system described above is given in Figure 18 together with the parameter values. The various constituent blocks are explained earlier in section 3.1.1 and 3.1.2. A concrete model, demonstrated on a real network, is explained in chapter 6.

There are two core router resources with their respective, settable control levels[14], three actors using three ingress routers and three flows taking three routes. Flow 1 uses core router 1, flow 2 uses both core routers 1 and 2 while flow 3 uses core router 2 only. It can be seen that the distributed bandwidth broker elements in the ingress controller works out the resource price function per traffic -aggregate path.

4.1.1 Controller

The controller determines the speed of response and robustness of the system. In the simulation model above, the controller used is proportional type.

4.1.1.1 Estimation of the proportionality multiplicand

In this example, a value for the proportionality multiplicand is chosen that would settle the resource price function to its equilibrium value in a single step. The equilibrium value is calculated using an offline algorithm given below. This value, for the given availability figure, is estimated as 0.16. The estimation uses the parameter values given in Figure 18. From equation (7) & (8),

$$K_R = (P_{Rk-1} – P_{Rk})/ (C_{Rk} - Y_{Rk})$$

In order to find the value of K_R, P_{Rk-1} is to be the initial open-loop value of the price-function and P_{Rk} is its expected closed-loop value at steady state. C_{Rk} is the capacity[15] of the resource (also the final steady state utilisation) and Y_{Rk} is the initial open-loop value of total resource utilisation.

a) $P_{Rk-1} = 1$

---

[14] In general the ‘usable limit’ of a resource is set at a lower level than the maximum capability of the resource, to allow for some operational margin

[15] In general the ‘usable limit’ of a resource is set at a lower level than the maximum capability of the resource, to allow for some operational margin.

b) From equation (15), $P_{Rk} = PP_{Rk} / C_{Rk.}$

PP_{Rk} is the share of provisioning potential for the given resource R. In the given case, $PP_{Rk} = \sum_{i=1}^{N} \frac{PP_i}{H_i}$ where *N* is the number of traffic aggregate flows through the given resource, PP_i is the *PP* for traffic aggregate *i,* H_i is the number of hops transited by traffic aggregate flow *i*

$PP_{Rk} = PP1/1 + PP2/2 = 4.5$

$C_{Rk} = 6$

$\therefore P_{Rk} = 4.5 / 6 = 0.75$

c) From equation (15), $Y_{Rk} = PP_{Rk} / P_{Rk}$, given that the open-loop utilisation requires to be found, the open-loop value of P_{Rk} given in (a) is used $\therefore Y_{Rk} = 4.5/1 = 4.5$

$\therefore K_R = (P_{Rk-1} – P_{Rk})/ (C_{Rk} - Y_{Rk}) = (1 – 0.75) / (6 - 4.5) = 0.1666$

To avoid the system settling to a non-optimal price function (see the discussion on sensitivity in section 3.3), it is always safe to start with a low proportionality multiplicand although it may take longer to settle.

4.1.2 Measurement feedback

In the given design model, unity feedback function is used and the measured value is polled every second. Due to the unreliability of the packet switched network, one cannot fully rely on the instantaneous feedback. Hence, the previous feedback value is required to be stored in the system so that this value can be used if the instantaneous value is not available.

4.1.3 Initial values for the price function and provisioning potential

The initial operating condition for the price function is set to unity by the operator. The provisioning potential (PP) for all users (in this case edge router) is chosen and set at 3 units per unit bandwidth in the given reference topology.

4.2 Simulation results of the multi-user, multi-resource reference model

Result of the Matlab simulation of the multi-user, multi-resource model is given below in Figure 19:

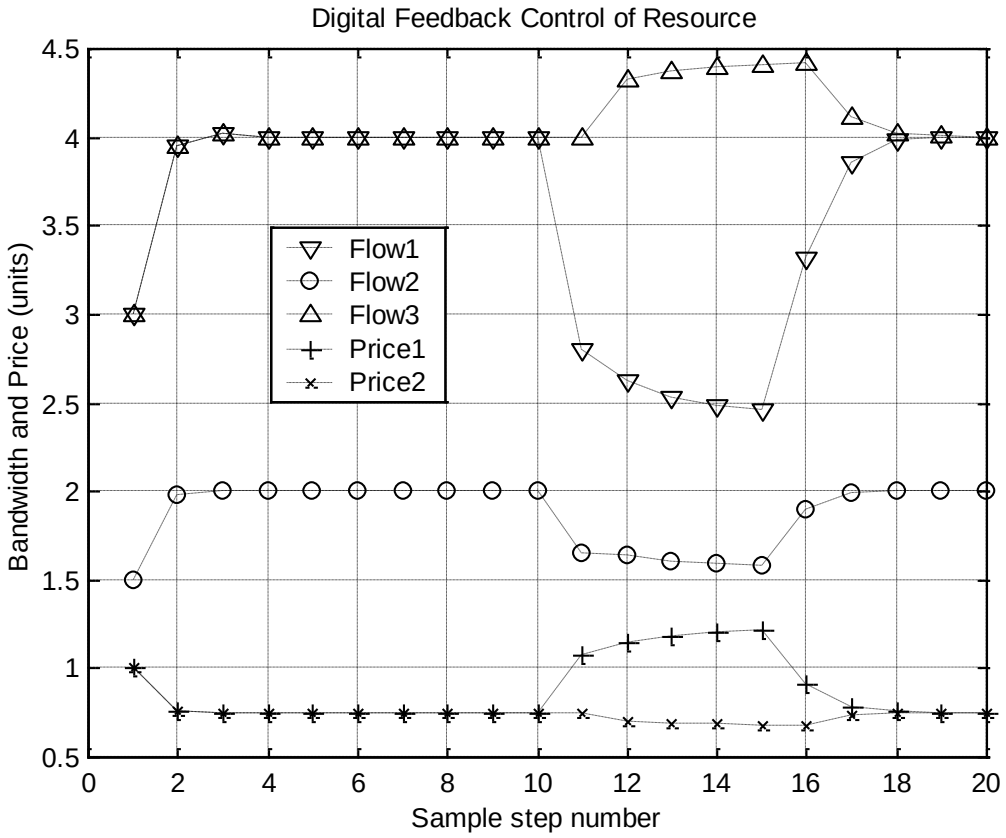


Figure 19 Matlab simulation results of the reference model

4.2.1 Explanation of the results graphs

Step1: At step1, the program starts up.

As this is a feedback control system, calculations take effect only in the (n+1) iteration. This is the reason for lack of instant response. Further, using sub-optimal value of K (0.16 instead of 0.1666 as calculated in section 4.1.1.1) meant that the calculations take 2 iterations before reaching the final values. This is why there is a slight undershoot/overshoot (0.25%) in steps 2 & 3 for the flow allocations. This is an artefact of numerical calculations, an

error band, and not indicative of control loop instability per se. In practical systems, the overshoots are usually capped by hard-limits.

$C_{R1} = 6$

$C_{R2} = 6$

From given equations in section 3.1.1 and 3.1.2:

The resource price for each user is

$P_{N1[1]} = P_{R1[0]} = 1$

$P_{N2[1]} = P_{R1[0]} + P_{R2[0]} = 2$

$P_{N3[1]} = P_{R2[0]} = 1$

The flow bandwidth allocated to each user is

$U_{N1[1]} = PP_{N1[1]} / P_{N1[1]} = 3$

$U_{N2[1]} = PP_{N2[1]} / P_{N2[1]} = 1.5$

$U_{N3[1]} = PP_{N3[1]} / P_{N3[1]} = 3$

The total output at the resources (resource usage) is

$Y_{R1[1]} = U_{N1[1]} + U_{N2[1]} = 4.5$

$Y_{R2[1]} = U_{N2[1]} + U_{N3[1]} = 4.5$

Step2: In this first iteration, the above usage information Y_R is now fed back to the controller.

The difference signals (availability information) are

$E_{R1[1]} = C_{R1[1]} – Y_{R1[1]} = 1.5$

$E_{R2[1]} = C_{R2[1]} – Y_{R2[1]} = 1.5$

The system now updates the resource prices as

$P_{R1[1]} = P_{R1[0]} – K_{R1}E_{R1[1]} = 0.76, K_{R1} = 0.16$

$P_{R2[1]} = P_{R2[0]} – K_{R2}E_{R2[1]} = 0.76, K_{R2} = 0.16$

The resource prices for individual users is

$P_{N1[2]} = P_{R1[1]} = 0.76$

$P_{N2[2]} = P_{R1[1]} + P_{R2[1]} = 1.52$

$P_{N3[2]} = P_{R2[1]} = 0.76$

The flow allocated to each user is

$U_{N1[2]} = PP_{N1[2]} / P_{N1[2]} = 4$

$U_{N2[2]} = PP_{N2[2]} / P_{N2[2]} = 2$

$U_{N3[2]} = PP_{N3[2]} / P_{N3[2]} = 4$

The total output at the resources is

$Y_{R1[2]} = U_{N1[2]} + U_{N2[2]} = 6$

$Y_{R2[2]} = U_{N2[2]} + U_{N3[2]} = 6$

The capacity of the resources is thus fully utilised.

An explanation of the notable features is given below. An explanation of later steps (from Step10) is given in section 4.2.2.8.

4.2.2 Features of importance

4.2.2.1 Resource utilisation gain

Comparing the utilisation of the network between step1 and step2

$Y_{R1[1]} = 4.5$ units

$Y_{R2[1]} = 4.5$ units

$Y_{R1[2]} = 6$ units

$Y_{R2[2]} = 6$ units

There is 33% gain (increase) in the usage (in this case flow) with the feedback control.

4.2.2.2 Resource revenue

$Y_{R1[1]} * P_{R1[0]} = 4.5 = PP_{N1[1]} + PP_{N2[1]} /2$
$Y_{R1[2]} * P_{R1[1]} = 4.5 = PP_{N1[2]} + PP_{N2[2]} /2$

Thus, the revenue from the resource is guaranteed despite the reduction in price so long as there is demand.

4.2.2.3 Resource price function adjustment

The total resource price of the network is reduced between step1 and step2

Total Resource $Price_{R1R2[1]} = P_{N1[1]} + P_{N2[1]} + P_{N3[1]} = 4$ units
Total Resource $Price_{R1R2[2]} = P_{N1[2]} + P_{N2[2]} + P_{N3[2]} = 3$ units

The total resource price is reduced by one unit because there was spare capacity in supply. This is an incentive for other customers to use the resource.

4.2.2.4 Operator revenue gain

Extra traffic allowed at resourceR1 = $Y_{R1[2]} - Y_{R1[1]} = 1.5$ units
Extra traffic allowed at resourceR2 = $Y_{R2[2]} - Y_{R2[1]} = 1.5$ units

Extra revenue from user =

$$= \left[CP_{\mathrm{N1[2]}}\ \mathrm{CP_{N2[2]}}\ \mathrm{CP_{N3[2]}}\right]\begin{bmatrix}\mathrm{U_{N1[2]}}\\ \mathrm{U_{N2[2]}}\\ \mathrm{U_{N3[2]}}\end{bmatrix} - \left[CP_{\mathrm{N1[1]}}\ \mathrm{CP_{N2[1]}}\ \mathrm{CP_{N3[1]}}\right]\begin{bmatrix}\mathrm{U_{N1[1]}}\\ \mathrm{U_{N2[1]}}\\ \mathrm{U_{N3[1]}}\end{bmatrix}$$

= 30 units - 22.5 units = 7.5 units

There is 33% gain (increase) in revenue in a feedback-controlled network.

4.2.2.5 Proportional fairness to user traffic flow

The pricing function drives flow levels to a proportionally fair distribution. It may be noted that each user traffic flow pays the same resource price per resource unit consumed:

Resource price/ resource for $user_{N1[2]} = PP_{N1[2]} / U_{N1[2]} = 0.75$
Resource price/ resource for $user_{N2[2]} = (PP_{N2[2]} /2)/ U_{N2[2]} = 0.75$
Resource price/ resource for $user_{N3[2]} = PP_{N3[2]} / U_{N3[2]} = 0.75$

Lemma 3 is therefore demonstrated in actual results.

4.2.2.6 Scalability

It can be seen that

a) the resource allocation at the core routers (the shared resources) maintains the isolation for the constituent flows as the allocation is determined only by their respective provisioning potentials and
b) no co-ordination is required between the edge routers for the calculation of ingress rates.

Thus, the system exhibits the qualities required for scalability and distributability.

4.2.2.7 Sensitivity

The proportionality multiplicand for availability figure of the resource determines the convergence. For small values of proportionality multiplicand, as calculated in section 4.1.1.1 (in this case 0.16) the system settles to the optimum value of the price function (in this case 0.76). For a higher value of proportionality multiplicand e.g. 0.25, the corresponding output of the price function is 0.625. Although this is sub-optimal, the system is found to be stable, as observed from the matlab/simulink simulations.

4.2.2.8 Interactions

The following interactions can be clearly seen in the graph in Figure 19.

At step10, a perturbation of 2 resource units increase is introduced (for 5 time units) to resource R1 that carries flows 1 & 2 (perturbing signal is not plotted). The resource price function of R1 can be seen going high immediately. Flow1 and Flow2 reduce in a proportional manner to maintain the total load at the set point. In other words, when the demand is more than supply, the price is increased to force decreased usage.

As a consequence of reduction in Flow2, the resource price function of R2 goes down and lets in more traffic from Flow3. In other words, when the supply is more than the demand, the price is reduced to force increased usage.

After a while, the perturbation is removed. The resource price functions and hence the flows then regain their respective equilibrium values.

4.2.2.9 Pareto-optimal solution

As can be seen from the results, in the equilibrium, it is not possible to allocate additional resources to one user without making another user worse off. This property of the system is called pareto-optimality.

4.2.2.10 Value added service protection

Now suppose that U1 in Figure 18 is a more valuable user than other users. The service manager increases the provisioning potential for U1 when a reduction in throughput associated with U1 is sensed, e.g. when the price allocated with the given set of resources has increased. In this case, the allocation for U2 is further reduced to maintain the provisioning for U1.

4.3 Simulation of the reference model in channel switching scenario

The discussion so far considered IP traffic aggregates using the solid physical media and homogenous network equipment to route the traffic. It may be noted that changing the physical media to RF and optical links will not make any difference in the principles described, as the resource, e.g. bandwidth can be controlled in a similar fashion.

Of further interest is the discrete capacity switching scenario in order to add/remove capacity in the existing network. In this case one can consider a) solid links b) RF channels c) optical wavelengths.

The model using optical wavelengths is of particular interest at this time, particularly due to the emergence of MEMS photonic technology. The optical switches made using this technology are called optical cross-connects. They use small mirrors to switch wavelengths. The wavelengths can be added and removed at will. The control technology developed in this thesis has been demonstrated in a real photonic cross-connect as described in section 6.5.

An illustrative reference topology of the system for space/channel switching (wired and wireless (optical or radio)) network control is given in Figure 20

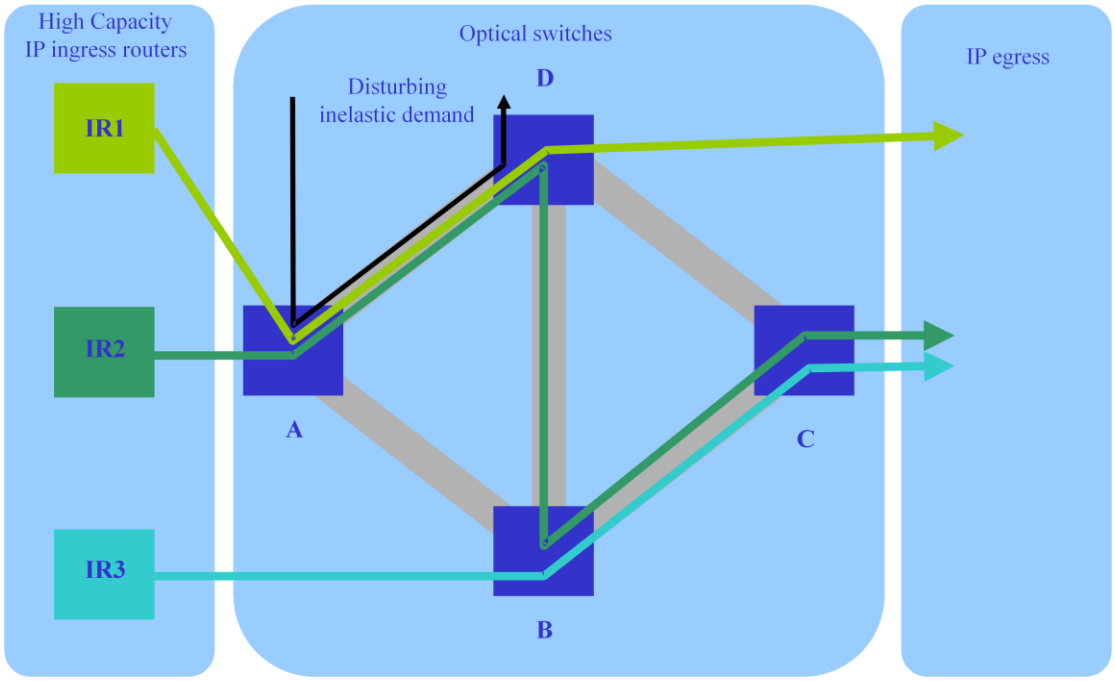


Figure 20 Model of the IP ingress routers using optical/radio wavelength/channels on demand

Consider a heavily loaded network configured as shown in Figure 20. There are three high capacity IP ingress routers that use the optical network to transport packets. IR1 uses fiber link AD, IR2 uses three fiber links AD, DB and BC and IR3 uses link BC.

For easy illustration, consider that each fiber consists of twelve wavelengths and that each ingress router has equal provisioning potential, say six units.

The wavelengths are allocated as a function of the ratio of the provisioning potential and a function of the resource demand for the given route.

4.3.1 Results of the simulation in channel switching scenario

The resultant wavelength allocations as well as the function of demand for wavelength are plotted on the vertical axis against unit sample steps along the horizontal axis as shown in Figure 21. Please note that due to the contention

in fiber links AD and BC (due to allocation to both IR1 and IR2), the demand for the route ADBC is twice that for routes ADC and ABC.

Initially, at step 1, the number of wavelengths allocated to IR1 is six as the demand for this route is unity; IR3 is similar. IR2 receives three wavelengths as it uses route ADBC. Note that the total number of wavelengths allocated is only nine, in link AD, which is sub-optimal. When the usage information is fed back to the controller, this is effectively signalling the lack of allocation for full utilisation i.e. network optimisation. This is immediately reflected in the reduced demand function at step 2. The allocation immediately increases to fill the set capacity of the fiber. The new allocations can be seen to be proportionally fair.

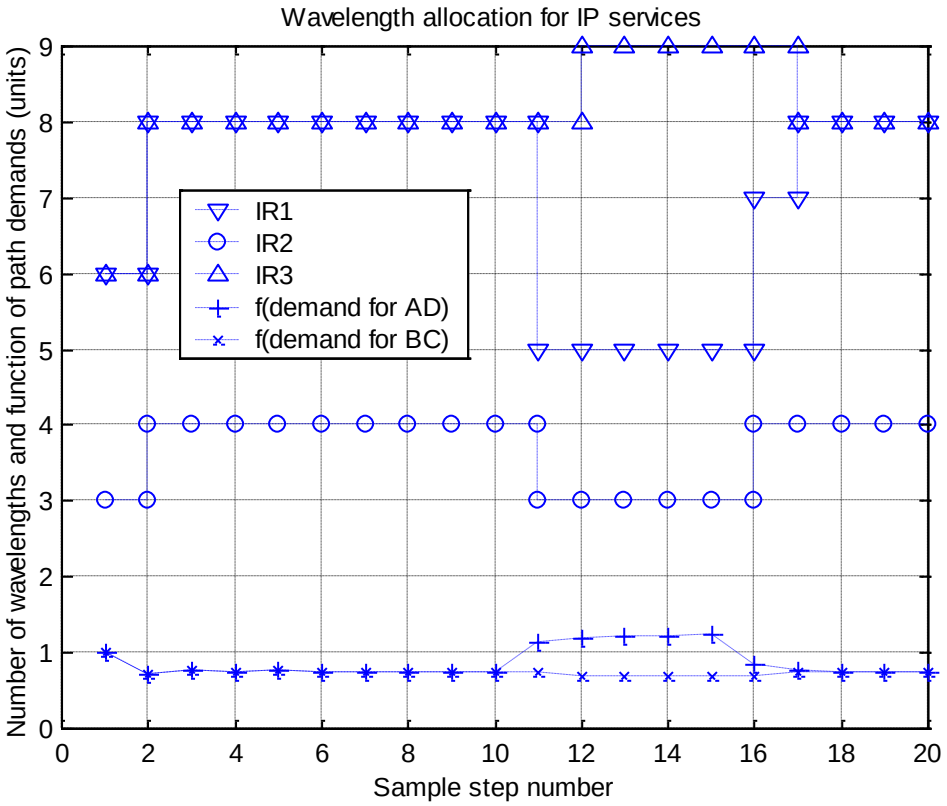


Figure 21 An illustrative reference topology behaviour of the optical layer optimisation

4.3.2 Explanation of the results graphs

Step1: At step1, the program starts up.

The initial operating condition for the resource price function is unity and the PP for all users is set at 6 units per wavelength. The proportionality multiplicand K_R is set to 0.08

From given equations in section 3.1.1 and 3.1.2:

The resource price for each user is

$P_{N1[1]} = P_{R1[0]} = 1$

$P_{N2[1]} = P_{R1[0]} + P_{R2[0]} = 2$

$P_{N3[1]} = P_{R2[0]} = 1$

The wavelengths allocated to each user is

$U_{N1[1]} = PP_{N1[1]} / P_{N1[1]} = 6$

$U_{N2[1]} = PP_{N2[1]} / P_{N2[1]} = 3$

$U_{N3[1]} = PP_{N3[1]} / P_{N3[1]} = 6$

The total output at the resources is

$Y_{R1[1]} = U_{N1[1]} + U_{N2[1]} = 9$

$Y_{R2[1]} = U_{N2[1]} + U_{N3[1]} = 9$

Step2: The above usage information is now fed back to the controller.

The error signals are given by

$E_{R1[1]} = C_{R1[1]} – Y_{R1[1]} = 3$

$E_{R2[1]} = C_{R2[1]} – Y_{R2[1]} = 3$

The system now updates the resource prices as

$P_{R1[1]} = P_{R1[0]} – K_{R1}E_{R1[1]} = 0.76$, $K_{R1} = 0.08$

$P_{R2[1]} = P_{R2[0]} – K_{R2}E_{R2[1]} = 0.76$, $K_{R2} = 0.08$

The resource prices for individual users is

$P_{N1[2]} = P_{R1[1]} = 0.76$

$P_{N2[2]} = P_{R1[1]} + P_{R2[1]} = 1.52$

$P_{N3[2]} = P_{R2[1]} = 0.76$

The flow allocated to each user is

$U_{N1[2]} = PP_{N1[2]} / P_{N1[2]} = 8$

$U_{N2[2]} = PP_{N2[2]} / P_{N2[2]} = 4$

$U_{N3[2]} = PP_{N3[2]} / P_{N3[2]} = 8$

The total output at the resources is

$Y_{R1[2]} = U_{N1[2]} + U_{N2[2]} = 12$

$Y_{R2[2]} = U_{N2[2]} + U_{N3[2]} = 12$

It may be noted that introduction of the feedback results in full utilisation of the resources.

Since the channels are in discrete numbers and cannot be switched in fractional form (as opposed to packets that has a lower level of granularity and hence could be scheduled in a continuous fashion for fractional bandwidths) additional software is required to be run on the background to make the system practical so that discrete channels are switched at suitable thresholds.

4.3.2.1 Interactions between the individual loops

At step10, a non-co-operative demand for 2 wavelengths is introduced to fiber link AD (this perturbing demand is not plotted). The resource demand for AD can be seen going high immediately. Wavelength allocations for IR1 and IR2 reduce in a proportional manner to maintain the total load at the set point.

As a consequence of reduction in allocation to IR2, the demand for BC goes down and allocates more traffic to IR3.

These interactions can be clearly seen in the graph in Figure 21.

4.4 QoS and service Protection

This thesis deals with the QoS offered by the packet transport. As the loading of the Internet increases, the ability of the network to cope with the traffic reduces. As a result, the quality of service offered by the network to its users diminishes drastically.

To a first level approximation, the QoS can be considered as inversely proportional to the number of packets in the system. In general, the type of service is considered deterministic.

To protect a service provision for a particular operator or user, the service rate and the QoS requires to be maintained. However, this is only possible with a) higher transmission rate and b) reducing the queue sizes of the schedulers. Since the resources are always limited, introducing a feedback control that maintains the service rate according to the dominant parameter of the service assurance will be necessary for scheduling the resource usage. With an inner optimisation loop, the system assures fairness to operators and optimal resource use in the given conditions. The service assurance for bandwidth, with feedback from ‘near’ and ‘far’ dynamically varying congestion points was demonstrated (see section 6.3).

4.5 QoS and the traffic mix

In the public network, the QoS obtained for the users of the premium traffic depends on the mix of the traffic in the network. The traffic generally contains packets with differing QoS requirements. For example, the different classes of service for DiffServ are EF (expedited forwarding, low loss/latency traffic), AF (assured forwarding, assured delivery under conditions), BE (best-effort forwarding). In the short study below, The EF load is the premium traffic and the BE load is the best effort traffic. The following graph in Figure 22 shows the simulation results of the effect of traffic mix on the premium flow. The plot shows percentage of the EF load and EF drop against the total load.

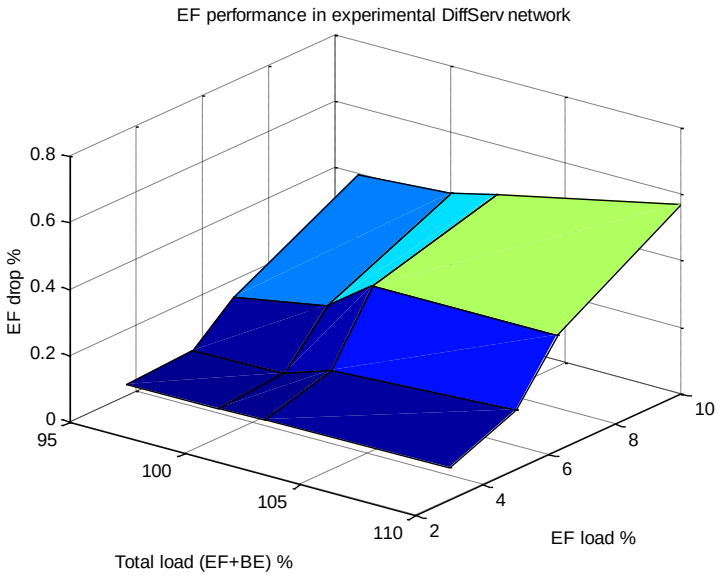


Figure 22 Matlab simulation of the diffserv traffic mix

The EF load is the premium traffic and the BE load is the best effort traffic. The plot shows percentage of the EF load and EF drop against the total load.

As can be inferred from the graph, the priority premium traffic has to be capped to a small percentage in all scenarios, in order to provide assured QoS. The traffic management policies and link schedulers are implemented such that the bandwidth partitioning for different classes of traffic are not broken.

4.6 Scalability of the Distributed Architecture

Previous sections demonstrated how the de-centralised and distributed controller decisions are enforced by the edge-router schedulers within an autonomous network. The architecture presented in this thesis is shown to be scalable within the autonomous domain, as well as across the multiple layers. Further, this architecture can be demonstrated to be scalable across multiple autonomous domains, as described below.

The Internet is constituted by a collection of autonomous systems connected together at network access points (private peering points or IXPs). Data flows use single or multiple autonomous systems to reach from the source end to the destination end. An abstract sample of such a system is shown in Figure 23.

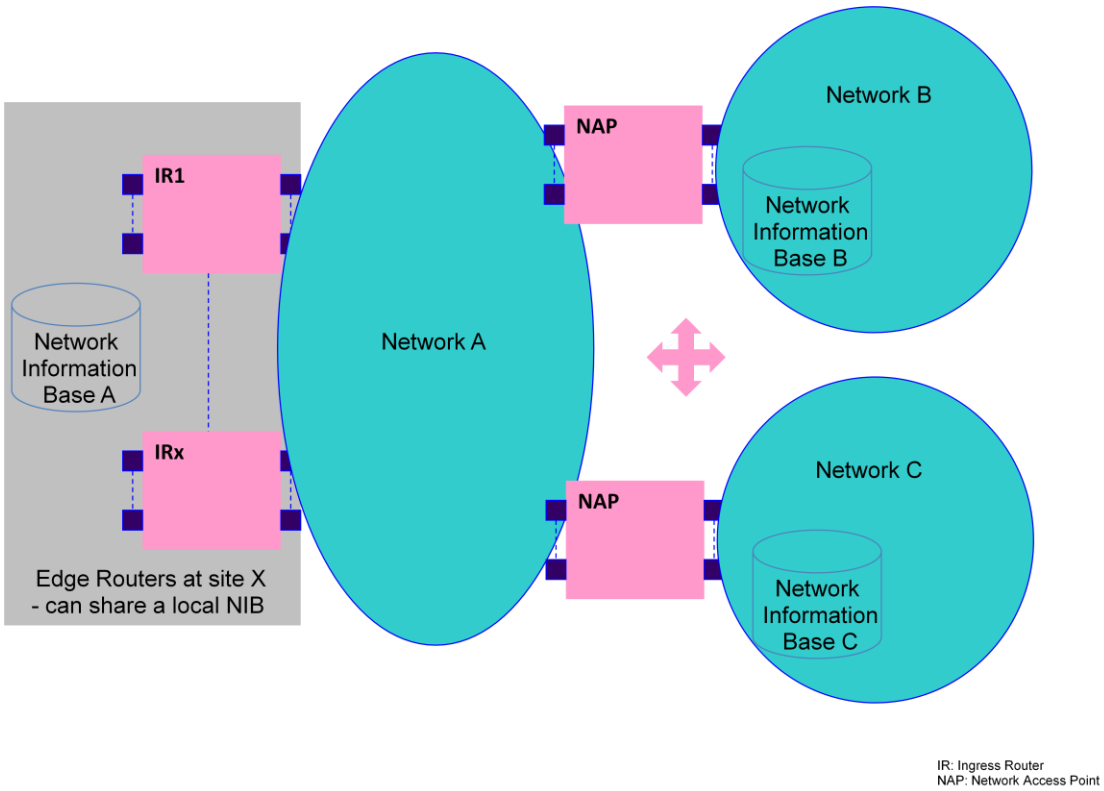


Figure 23 Illustration of the system across the multiple autonomous networks

In the distributed architecture proposed, the complexity of signalling load from individual routers is tackled by storing the measurement data in a network information database within each autonomous network, which can be queried to retrieve the required information. This also helps in maintaining the integrity of the individual autonomous systems.

There is an issue with the extent to which ASs will exchange pricing information. In this case, techniques to work with sub-optimal information has to be developed. Further work is suggested in this area (see section 7.2).

The number of AS addresses are defined as 16-bit integers (changed to 32 bits in 2008). There was over 5000 unique autonomous systems in 2000,

estimated to grow 10 fold over the next decade. Any given AS consists of a few routers to over 100 routers. While the Internet has more than 100,000 routers, the given organisation reduces overhead of passing resource information by way of aggregation.

The measurement data acquisition is done either by a the client asking for information or the client being told of the information. In the first model, also called the 'pull' model, the client (edge-router or NIB) requests the server (NIB or core router) and the server replies with the information. The second model, also called the 'push' model, the server send out the information either periodically or triggered by an event. The client can either subscribe to the server or receive the broadcast and decide what to do about it. The second model might be more suitable for larger networks. The scalability of the system depends on the number of ‘states’ to be stored and the additional load due to the messages.

In the experiment network given in section 6, the pull model was used in order to acquire information from the core routers. The measurement requests were send every second.

4.6.1 Simulation across multiple autonomous networks

A simulation model, built for the multiple networks scenario, is given in Figure 25. The inter-network model uses two networks and three flows, so that the interaction characteristics can be studied. The analysis is similar to the reference model used earlier in section 3.1.2 for the intra-network case. The results can be seen to be following similar pattern i.e. the intra-network case and the inter-network case have the same pattern of behaviour, which demonstrates that the system operation is scalable.

The autonomous network systems have their own intra-network flows as well as inter-network flows that use multiple autonomous systems. The resource information about individual routers can be stored in network

information bases for later collection. Flows c1, c2 & c3 works with the output of the resource price function available at network access points. This eliminates the need to poll individual routers within an autonomous network.

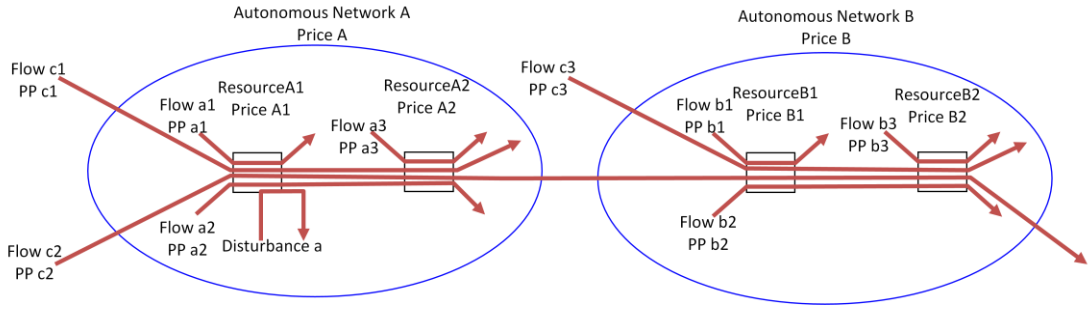
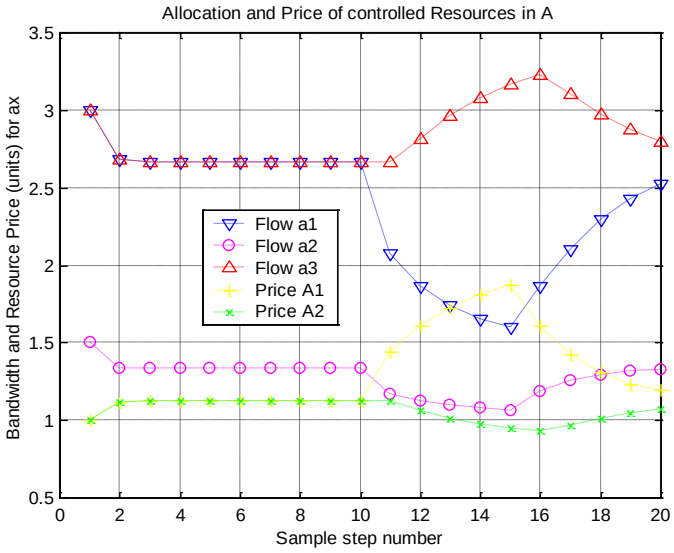
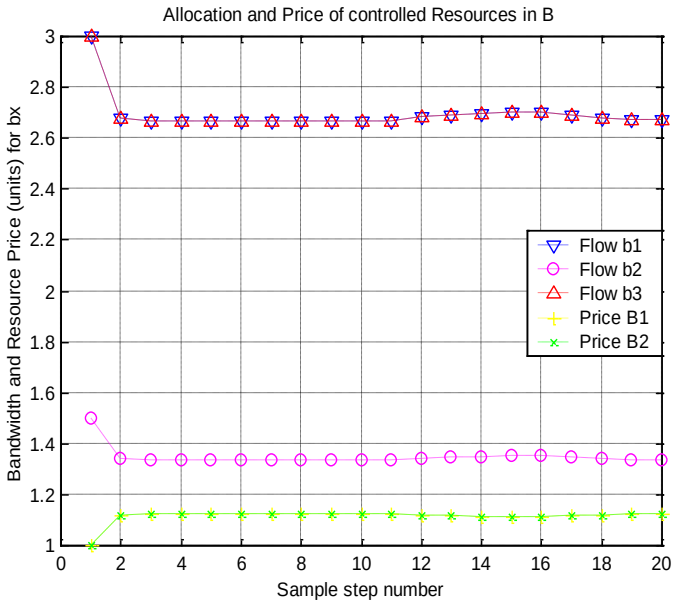


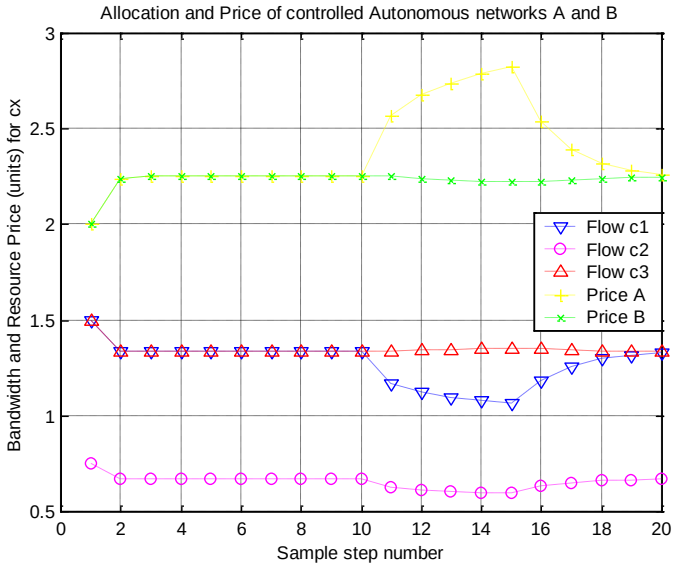
Figure 24 Simulation model for the multi-AS scenario



a) Allocation and price of resources in autonomous network A for intra-AS flows



b) Allocation and price of resources in autonomous network B for intra-AS flows



c) Allocation and price of autonomous networks A & B for inter-AS flows

Figure 25 Matlab simulation results for inter-AS (flows a and b) and intra-AS flow (flow c)

The autonomous system AS1 and AS2 each has three internal flows and two resources. These two autonomous systems are used as the resources for three other flows that pass through them. The provisioning potentials for all the flows are given as 3 units and the initial operating point for the price function is set as unity. When the simulation starts up at step1, due to the

demands from both internal and external flows, the price function for all the resources goes high. At step2, the allocations are settled to their proportionally fair solutions. At step10, a flow that is deliberately admitted at the resource A1 creates congestion at A1 and hence the price function for this resource goes high. This results in the reduction of all the flows that use this resource, in varying degrees, as the provisioning potential is kept constant in this reference topology. The reduction in the transit flow has the knock on effect on autonomous system B, in that the price function for both resources decreases and hence all the flows that use only this autonomous system benefits with increased flow.

When the perturbation flow is removed, the system settles back to the equilibrium values. The system characteristics remain same across multiple domains. This shows that the performance pattern in similar to the single network case given in section 3.1.2, and is stable and scalable.

5. SCHEDULERS

5.1 Scheduler for Diffserv Expedited Forwarding

As the Diffserv Expedited Forwarding for premium traffic is being standardized in RFC2598 (1999) it is required that the EF traffic cannot be pre-empted for more than a packet time at the configured rate. It has been proposed in the RFC2598 to use either a) use input throttling of the packets or b) have one queue in a weighted round robin scheduler where the share of the output bandwidth assigned to this EF queue is equal to the configured rate, in order to meet this delay requirement.

With the overall design and implementation concept being driven by the need for ever faster network processing capabilities, the option (a) to use input throttling is perhaps the last option.

When considering the option (b), use of WRR type queue (WFQ, SFQ etc) will not meet the packet time requirement in a work-conserving scheme. A work conserving scheduler idles only when there are no packets queued. The non-work conserving scheduler idles during the period between the last packet served and the next packet to be served but has not become eligible for service yet. The latter, although it wastes bandwidth, helps reducing the jitter and makes the downstream traffic more predictable. Making the scheduler non-working conserving, however, is not an efficient option, as it wastes bandwidth. The issue in meeting the packet time requirement can be demonstrated as follows.

Consider the state diagram of a packet scheduling system is given in Figure 26 below. The state transitions are self-explanatory.

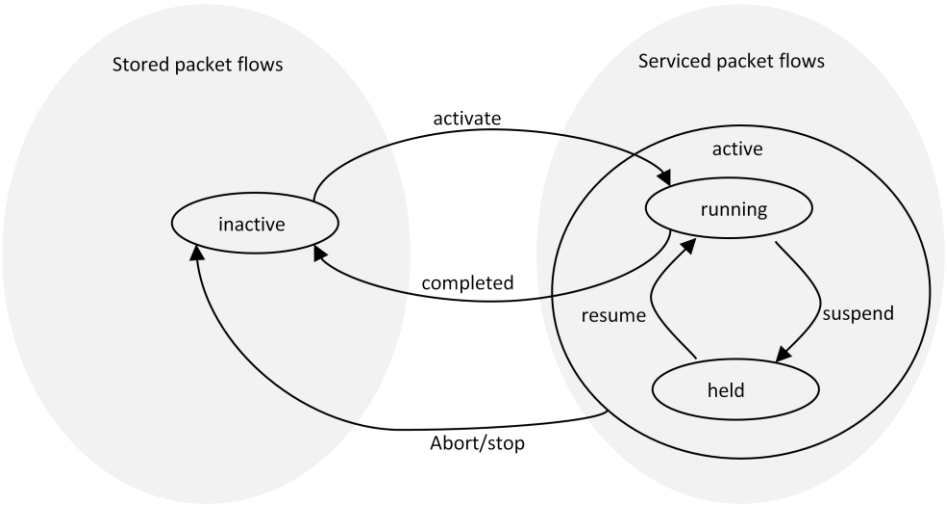


Figure 26 State diagram of a packet scheduling system

Assume there are two flows *B* and *P* (B stands for BestEffort and P stands for Premium) being serviced from time t_0. At time t_{4+} flow *B* is being served and let us take that flow *P* is scheduled for a time in future, t_f (t_8 in this case). Even in a well-designed system, bursts and malfunctions of different kinds can occur and suspend a running flow. Now, assume that flow *B* got blocked at t_7. In order to increase the efficiency of resource utilisation, the work-conserving scheduler services the flow that is not suffering the blocking. In a work conserving scheduler, the bandwidth will now be taken over by flow *P at* t_7. Due to this, flow *P* gets expedited service when flow *B* is blocked. When flow *B* becomes runnable subsequently, it captures the service as given in Figure 27. Flow P relinquishes control on demand from flow B.

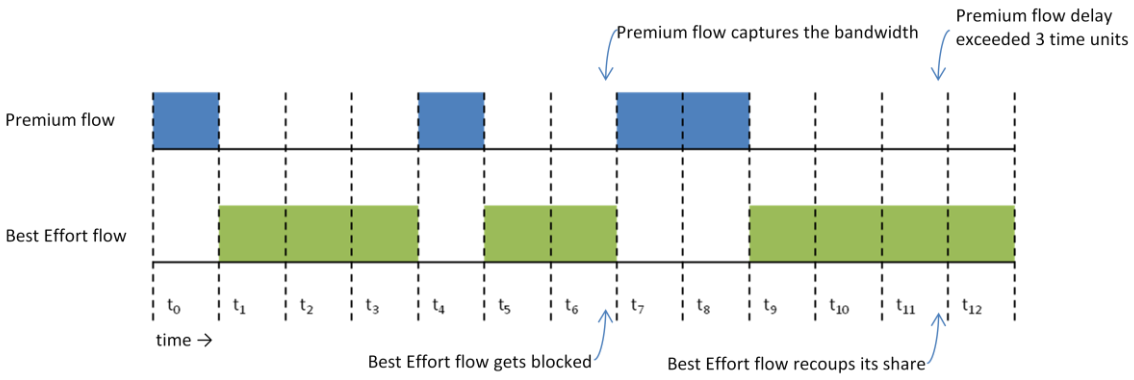


Figure 27 Effect of bandwidth conservation in round robin scheduler

However, the time that flow B did not get serviced (because it was not available) is not counted as a loss in effective bandwidth to flow B. Therefore, the effective bandwidth share that was assigned to the flow B remains to be served. Due to this, flow P will now be scheduled for a time later than t_f (later than t_{12}). Flow P, however, cannot be penalised for using the extra bandwidth that was available due to the suspension of the blocked flow.

In this case, the delay bound for flow *P* could exceed the service agreement.

Hence, in work conserving mode (the mode in which the schedulers are preferred to be operated in order to maximize link utilisation), the delay experienced by a premium flow in Diffserv Expedited Forwarding could exceed the service level agreement as given in the RFC2598. Devising a scheduler to achieve this and all the prior requirements is an open problem.

Further, the flows that are really required to be scheduled are the flows whose exogenous rates exceed the rate at which they are being serviced. The exogenous flow rate is defined as the rate the flow would have, if the links were of infinite capacity. Thus, the weights are required to be modified on-the-fly that takes into account the instantaneous nature of congestion as well. This is applicable in various scenarios including P2P.

5.1.1 Expedited Forwarding SFQ (EFSFQ)[16]

An improved scheduling algorithm that eliminates the delay experienced by the premium flows in Diffserv Expedited Forwarding is developed as part of this work. The efficiency of the scheduler is increased, at the same time expediting the premium flow. The flow is expedited by using the work-conservation principle, at the same time preventing further delays due to it.

16 This scheduler was previously called Offset Adjusted Fair Scheduler in other documents and US patent US06888842 by the author. It is renamed in this thesis to directly convey the purpose.

The general framework of EFSFQ is similar to that of WFQ and SFQ. However, the calculation of *virtual start time* is modified for any flow that is blocked. The flows are serviced in the increasing order of *virtual start time*; therefore, EFSFQ is more similar to SFQ than WFQ. When there is a tie in the *virtual start time*, the ties are broken either arbitrarily or according to a policy. The weight can be changed, when required, by the ingress controller to take care of the instantaneous nature of congestion. The algorithm is invoked once per packet transmitted. The algorithm is given below.

EFSFQ can dynamically change the bandwidth allocation according to the policies required to be enforced. Each flow is transported packet by packet. The future time of service for a flow will be proportional to the current length of packet / its bid, the unit being [bits/ (bits/sec)] i.e. in seconds. This future time of service is called *virtual start time*.

Let p_f^j, l_f^j and r_f^j denote the j^{th} packet of flow f, its length and its weight respectively. Let $A(p_f^j)$ denote the time at which the j^{th} packet is requested i.e. comes to the head of the queue. If the flow remains runnable, it is the time at which its previous packet finishes. $S(p_f^j)$ and $F(p_f^j)$ denote start time and finish time respectively. For the analysis, the following virtual time assignments are made:

1. Virtual time, *v(t)* = $S\left(p_{f_{IN-SERVICE}}^{j}\right)$ when CPU is busy

$\max\left\{F\left(p_f^j\right)\right\}$ when CPU is idle

2.a. Virtual start time (normal flows), $S\left(p_f^j\right) = \max\left\{v\left(A\left(p_f^j\right)\right), F\left(p_f^{j-1}\right)\right\}$

In EFSFQ, if a flow is blocked, and hence not available for service, the *virtual start time* of the blocked flow (*B*) is updated in the background and is carried along as if it were serviced. This way flow *P* is assured of being

serviced at the previously scheduled time t_f. In this way, the scheduler can provide reduced inter-packet delay in a work-conserving[17] mode.

2.b. Virtual start time (blocked flow),

$S\left(p_B^i\right) = S\left(p_B^{i-1}\right) + \left(F\left(p_P^j\right) - F\left(P_P^{j-1}\right)\right) + \frac{l_P^j}{r_B^i}$ where *i* is the virtual packet count for flow *B* for each packet of flow *P*.

3. Virtual finish time, $F\left(p_f^j\right) = S\left(p_f^j\right) + \frac{l_f^j}{r_f^j}$

The share of resource capacity received by each flow would be:

$C_f = C_R \,.\, r_f / \Sigma\, r$

where C_f is the capacity received by the flow *f*, C_R is the total capacity of the resource, r_f is the weight of flow r and the denominator is the sum of the weights of all the flows.

5.1.1.1 Delay minimization for the Expedited Forwarding traffic

In the new scheduler devised, it is argued that penalising the flow *B* for getting blocked (no show) is a justifiable option. In addition, from a game theoretic point of view, at the network level this forces the individual flows to behave well.

Lemma 4: Given a queuing system and assume that at least one of the queues is blocked during the period (T1, T2), then during (T1, T2), the following property holds

$\sum_{\mathrm{EFSFQ}}(\mathrm{IPD}|\mathrm{IPD} > 0) \leq \sum_{\mathrm{SFQ}}(\mathrm{IPD}|\mathrm{IPD} > 0)$

where IPD is the inter-packet delay for Expedited Forwarding traffic.

[17] This scheduler works in the non-work conserving mode as well. However, a non-work conserving scheduler is less efficient in general

Proof:

The lemma is intuitively depicted in Figure 28. As can be seen, flow *B* had to give away the service time allotted to it when it is blocked. This lost share of time will not be given back to flow *B*.

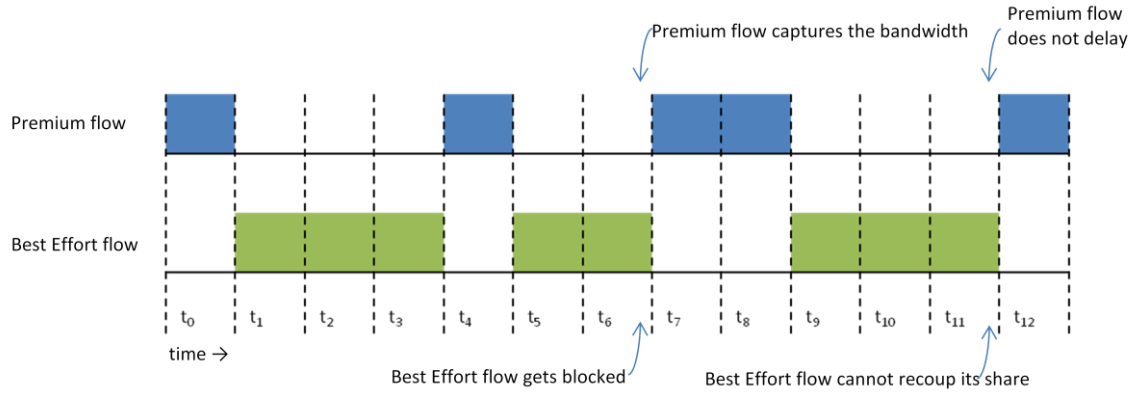


Figure 28 Effect of virtual time offset adjustment

Here the flow *P* gets the service at the assured future times and hence does not suffer more delay than what is specified. In other words, the adjacent flow (in this case *P*) receives more bandwidth when other flows are blocked or do not show. More importantly, the real-time requirement for the flow *P* is ensured in all situations. The inter-packet gap of the Premium flow is thus not allowed to grow beyond given limits.

5.1.1.2 Use of free bandwidth

Further, as can be seen, in this case the adjacent flow gets the bandwidth share absolutely free. Instead of giving this free bandwidth wholly to the one adjacent flow, it could be fairly distributed among all the flows by incrementing the virtual start time of all eligible flows by an amount that would spread the

time slot evenly. A simple case would be to increment it by an amount equal to $\frac{l_f^j}{w_f^j} \Big/ N_f$ where N_f is the number of flows.

.

Although it may appear that this would cost additional computation, it is not necessarily so. The clock base can be offset by the given amount, and all the flows get the extra share instantaneously without additional computational load. This is may be done to achieve near ideal fairness.

5.1.2 Complexity and Scalability of the scheduler

As the calculation of virtual time involves only one parameter (start time), the computation is inexpensive. Consequently, the computation of start time and finish time are also inexpensive. The scheduling order is decided by a FIFO queue (containing packets that are at the head of their respective queues) prioritised in the order of the start time. The complexity of the operations of this queue is O(log N) where N is the number of flows in the active list [51]. Since the complexity of the scheduler lies in the said mechanism, the complexity of ESFQ is O(log N).

This complexity is sustainable as the number of active flows in the Internet is found to be about 1% of the total flows [131].

6. SYSTEM IMPLEMENTATION

This chapter describes a concrete implementation of the de-centralised and distributed dynamic resource management system described in previous chapters. Results from the real network experiments are captured and shown. The system provided quality of service for different types of flows.

The system was implemented in a network of Linux routers using various hardware, software and technologies from the IP suite. The implementation follows the same design as the simulation model, in order to compare and contrast the results.

6.1 Mapping the DRC system to generic architecture

The laboratory reference model for implementation of the Distributed Resource Control system (DRC) was devised to be compliant with the COPS (Common Open Policy Server) Resource Broker model[18] as shown in Figure 29

[18] The COPS model was selected because of the similarities in top-level blocks that were independently visualised for the DRC

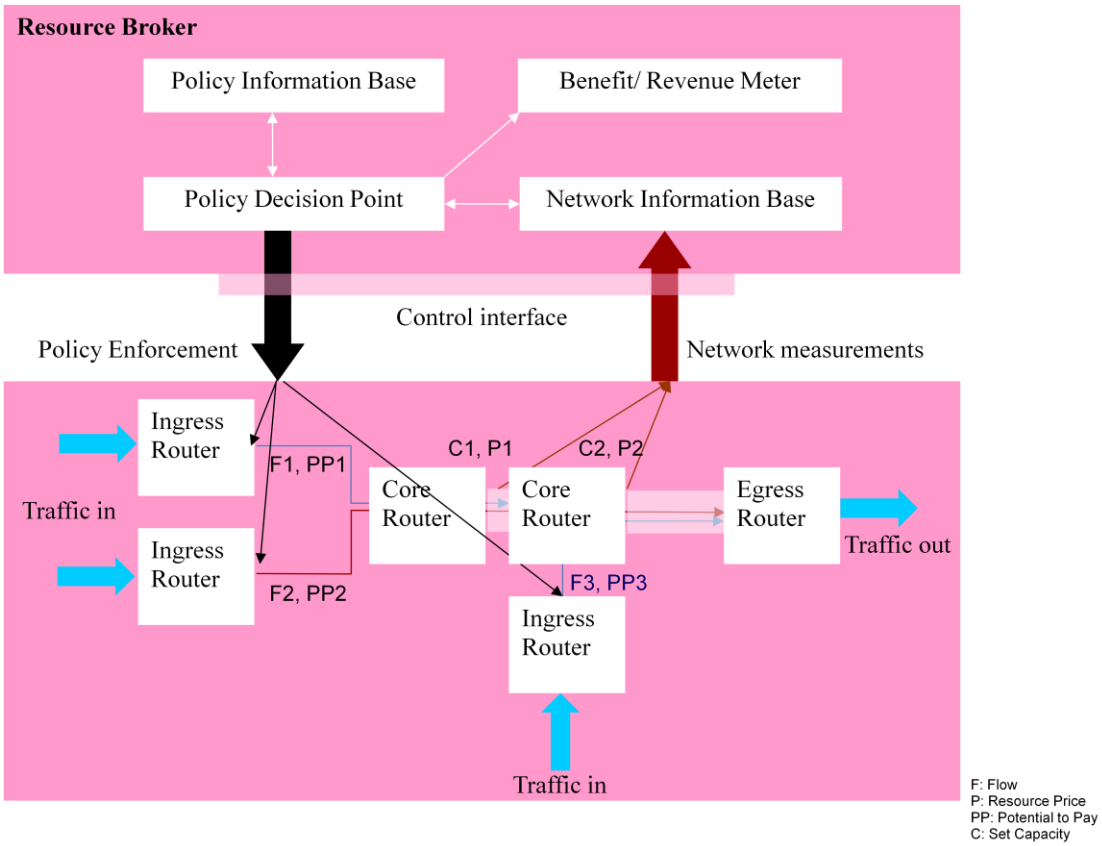


Figure 29 Simplified DRC System - COPS compliant architecture

The user policies to be implemented are stored in the Policy Information Base (PIB). The implementation decisions are made at the Policy Decision Point (PDP). These policies are then enforced at the Policy Enforcement Points (PEP). The network topology and measurement information is stored and collected at the Network Information Base (NIB). The benefit to the user and revenue to the network operator are displayed by the Benefit/Revenue Meter (BRM). The NIB and BRM are added to the COPS architecture by the author. The various functional modules of the system are described below.

6.2 Detailed functional blocks

6.2.1 Network Information Base

The network information base collects the resource utilisation/ price function information. This is done by polling or feedback from the resources

that are actively used. Polling is done only if the data is stale i.e. not current. In the case of multiple networks, output of resource price functions that are propagated to traffic merge points reduces the measurement load- this works as a hierarchical system. A schematic is given in Figure 30

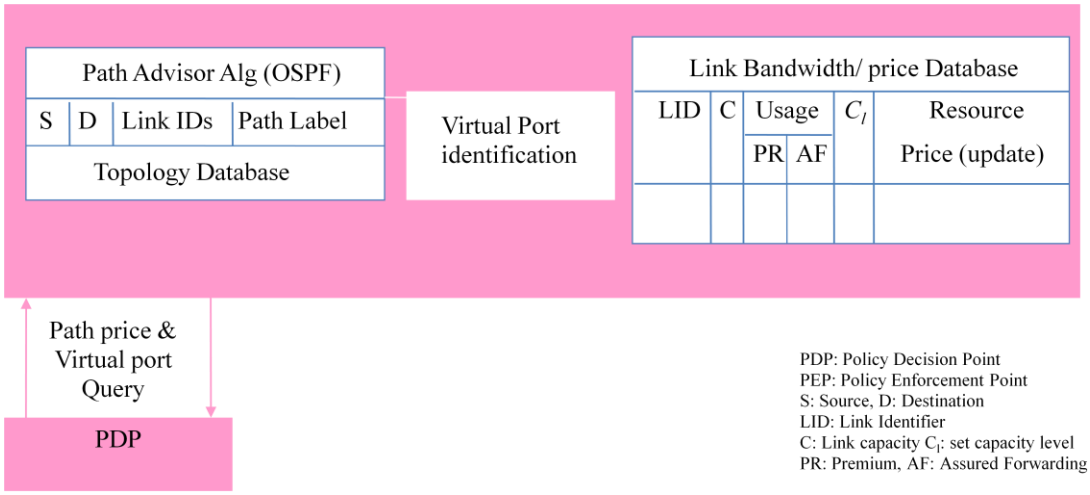


Figure 30 Network Information Base

The capacity partitioning (C*l*) could be preloaded in the core routers (this could be changed by the operator), and only the difference between this given and used bandwidth need be sent out. If the price function calculation block is incorporated within the router, then only the output of the price function calculation need to be sent out.

6.2.2 Ingress Router

The flow control policy enforcement is performed at the ingress router. Admission control policy enforcement is done at the traffic input part of the edge router and network ingress policy enforcement is done at the output of the forwarding path. The edge routers are designed to have their egress port scheduler controlled by the Provisioning Potential and the resource price function- in this case the scheduler becomes the Policy Enforcement Point.

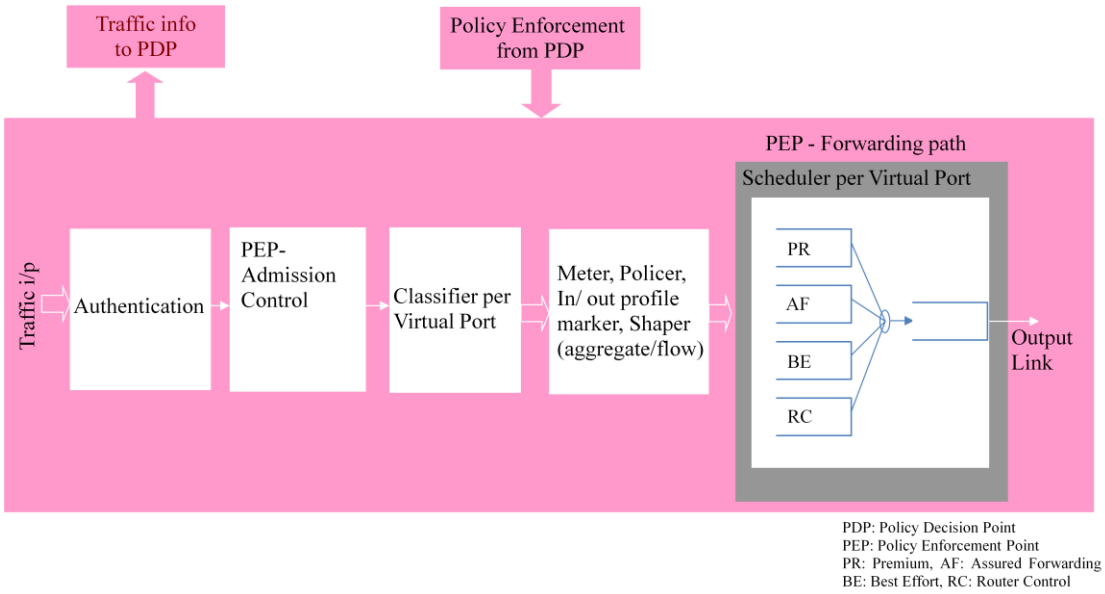


Figure 31 Ingress Router

To calculate the resource price function, the ingress controller issues a 'resource price function query' to the local database (see Figure 33) for a given source-destination pair, per traffic class. The program uses the routing table, finds the participating core routers, finds matches in the 'resource price function database' and responds with the output value of the resource price function.

6.2.2.1 Policy enforcement using Linux traffic control

The linux ports have two methods for controlling- either via netlink interface or via the system call interface as given in Figure 32:

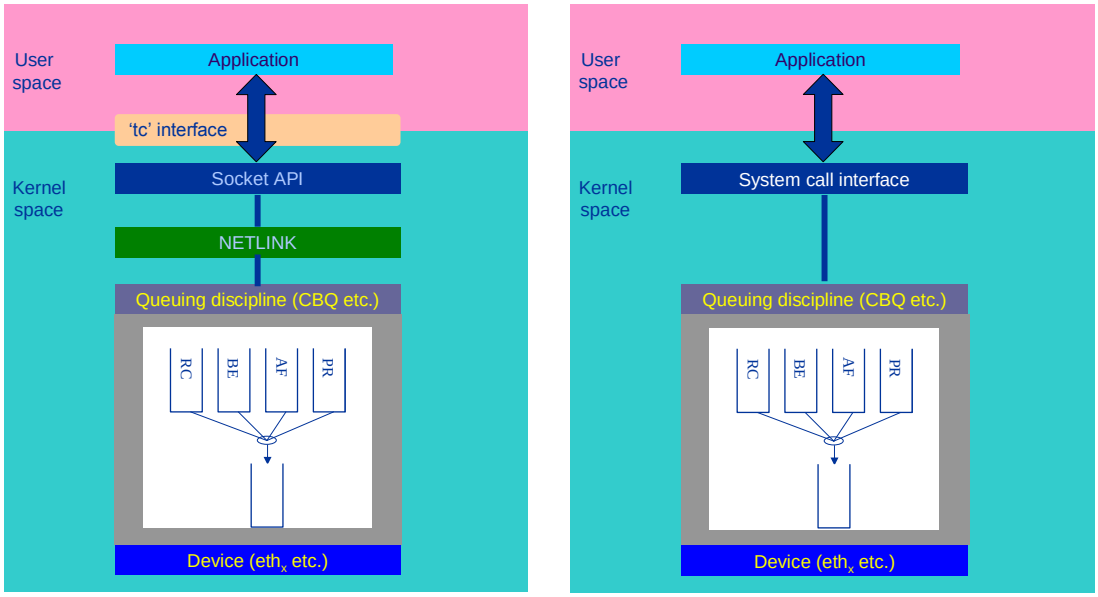


Figure 32 Different traffic control interfaces to Linux kernel

The 'tc' interface was used for the current implementation.

6.2.3 Policy decision point

The policy decision point makes the admission decision based on the policies set apriori. This decision could be to block, modify priority marking or admit the flow to the network. A schematic of the reference PDP is shown below in Figure 33:

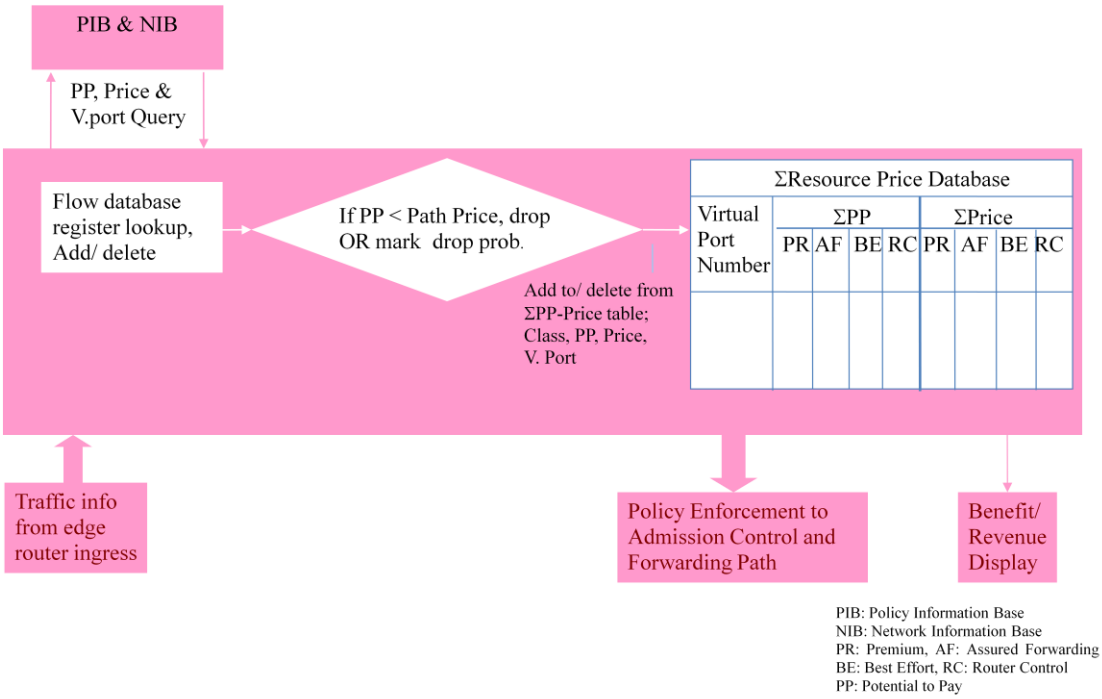


Figure 33 Policy Decision Point

6.2.4 Policy Information Base

The policy information is stored in a lightweight database; a general schematic is shown below in Figure 34:

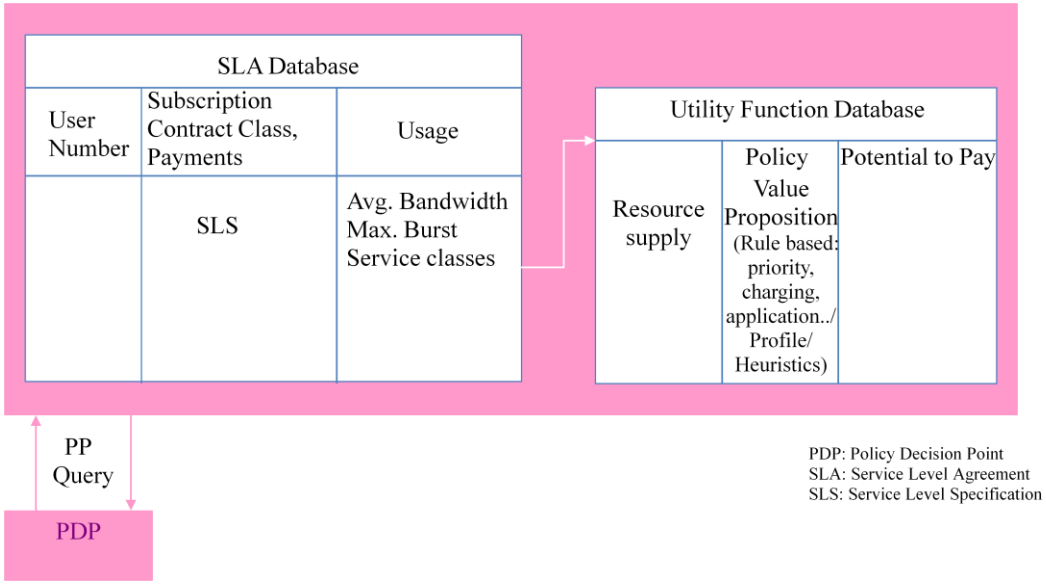


Figure 34 Policy Information Base

6.2.5 Link occupancy measurement sub-system

The implementation of the distributed resource control system uses its own sub-system for monitoring, measuring and feeding back the data from the network nodes. Such a sub-system could be used for carrying all types of data that the controller requires. In the current implementation, the traffic load on the link was measured using this sub-system.

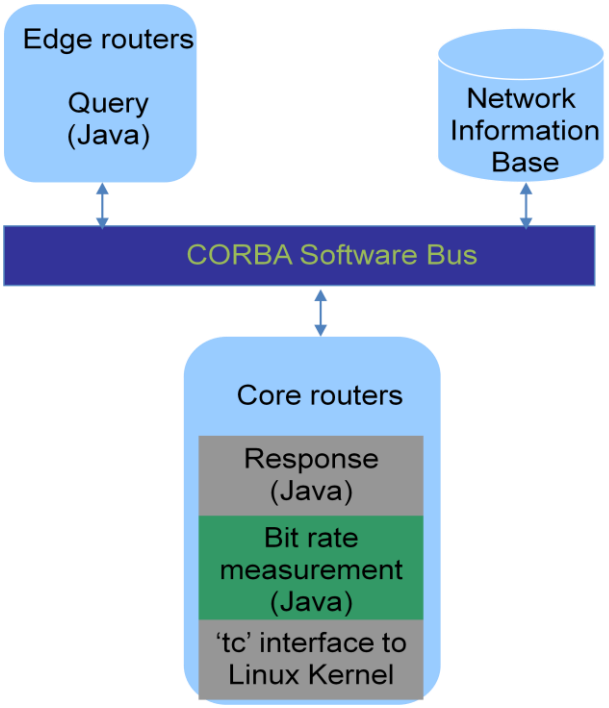


Figure 35 Link Occupancy Measurement sub-system

At the core router, (Linux based network element in this case), a java application issues 'tc' commands to the kernel every second. The kernel queuing discipline responds with the traffic statistics for the given interface device. This response is parsed to get the data (e.g. 'sent bytes' per class of traffic). This value is compared with the previous data. The bit rate per class of traffic at the given device is then deduced as follows

Bit rate = $data_{[new]}$ - $data_{[previous]}$ / (sample time * scale adjustment).

The scale adjustment figure is to take care of any delays in the process.

At the edge router (Linux based network element in this case), a java application issues RMI or CORBA calls to the core router program to fetch the

bit rate information. This data is stored in stored in the NIB database and updated as and when required.

6.3 Network Implementation of the architecture

The architecture described above was implemented in the router research laboratory of Nortel Networks Harlow Labs. The network configuration used is that of the reference model used for the simulation, described in previous chapters.

For ease of demonstration as well as measurement, test and diagnostic purposes, the model was drawn up in the TkIned and appropriate ports were monitored using SNMP monitors. A screen capture (from the management computer) of the working system is shown below in Figure 36:

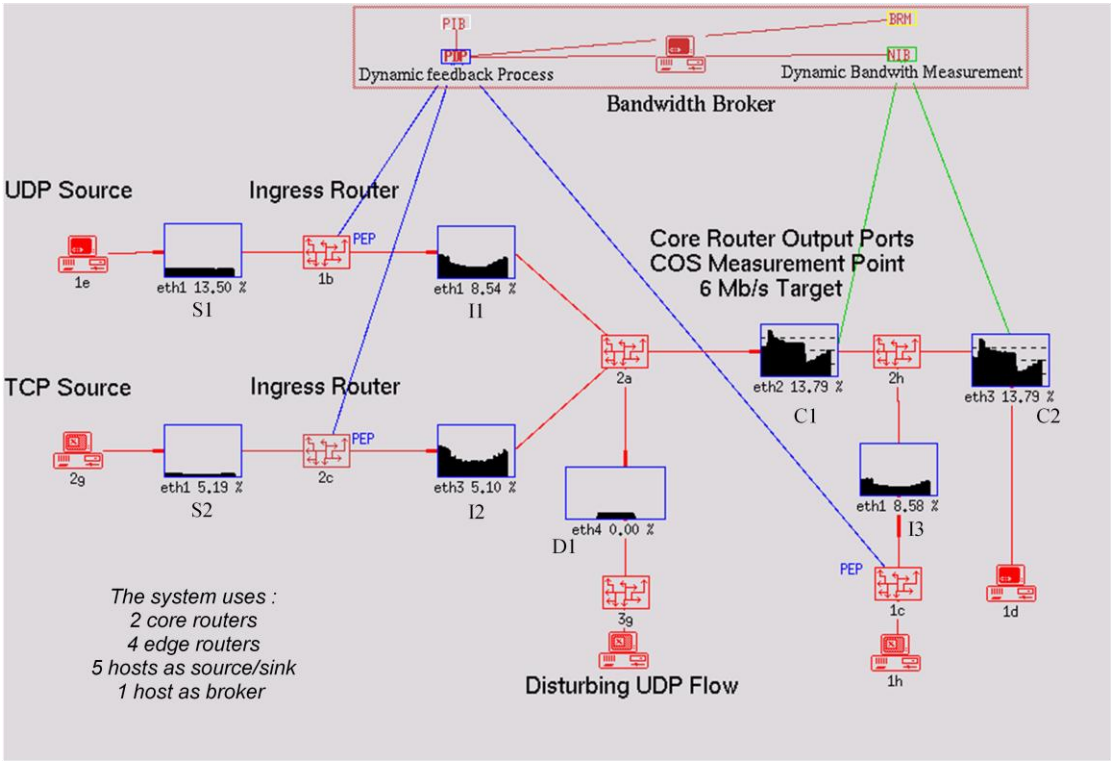


Figure 36 Screen capture of the live demonstration of the reference network

The software is implemented in a Python platform, Python being selected primarily for rapid prototyping. The communication interface was, at the time, done through ssh (secured shell) access. This can be done by other remote invocations methods as well.

Both TCP and UDP type flows were used in order to demonstrate their characteristic responses, in particular the packet trunking as explained in section 6.4.1. Only one class of traffic is used in the present demonstration. However, the system could be used for other classes of traffic.

6.3.1 Description of the live Network Implementation

As mentioned before, the network is designed around a set of Linux routers, host traffic sources and sinks, as in Figure 36, configured in the same connectivity pattern that was used for the simulation reference model. This pattern of connectivity is used in order to compare and contrast the performance of the real network and the simulation model.

6.3.2 Ingress routers

Routers designated 1b, 2c and 1c serve as the ingress routers. The output bandwidths of these routers are to be controlled to facilitate network ingress control per class of traffic. For this purpose, a differentiated services class based queuing (CBQ) discipline is attached to the output ports of these ingress routers. The command to control the bandwidth is given through the traffic control (tc) interface of Linux. In the demo, this control command is issued every second that updates the diffserv CBQ.

6.3.3 Core routers

Routers 2a and 2h form the core network. These are the system resources used by the ingress routers, the usage of which is to be controlled. The output bandwidth (resource utilisation limit) of these routers is set at a specified level (in this case 6Mbps) and is measured. To control the limits per class of traffic,

the diffserv CBQ is attached to their output ports. The measurement is done through the tc command interface. The statistics sub-command provides the measurement.

6.3.4 Traffic sources

Flow1: Host 1e provides a UDP traffic source that uses core router 2a only. The output is sunk at the input port of 2h. The ingress controller for this flow is 1b.

Flow2: Host 2g provides a TCP traffic that uses both resources 2a and 2h. The sink is 1d. The ingress controller for this flow is 2c.

Flow3: Host 1h provides a UDP source that uses resource 2h only. The sink is 1d again. The ingress controller for this flow is 1c.

6.3.5 Perturbation

Host 3g is used to generate a traffic pulse to create an overflow at the core routers, to study the performance and demonstrate the various effects. Two flows are generated: D1, that uses both resources and terminates at 1d and D2 that uses only one resource (2a) and terminates at 2h.

6.4 Operation of the live network implementation

As described earlier, there are three traffic aggregate flows taking three routes. User1 shares resource1, user2 shares both resource1 and resource2 and user3 shares resource2.

At start up the system allocates the bandwidth for these traffic aggregates according to the pre-determined initial operating point for the resource price function. This allocation decision is made at the PDP using the algorithm given in section 3.1.2.2. The bandwidth usage at the core resource outputs

are now measured and fed back. This information is collected at the Network Information Base. The algorithm then modifies the allocated bandwidth to force optimum utilisation in the given conditions. This is achieved by varying the intermediate variable that is the resource price function. The steady state is reached in two sample steps. In the current implementation, the time step is one second. The graphs in Figure 36 show the changes in the traffic flow.

6.4.1 Explanation of the results graphs

The graphs on the screen capture shown in Figure 36 shows the network load measured via snmp daemons. Please note that the vertical axes are not exact to scale (they were modified to give better visibility). The horizontal axis represent about 50 seconds duration.

6.4.1.1 Transport of UDP flows

The graph on the left-hand-top (S1) shows the UDP traffic that is generated. The second graph on the same line (I1) shows the controlled output. The dip in the flow is due to the effect of the perturbation that increases the resource usage suddenly. The perturbing flow can be seen in the lower middle (D1). It may be observed that the excess UDP flow gets dropped at the ingress router (I1).

The lower graph on the right-hand side (I3) shows another controlled UDP flow that is input to the core network system.

6.4.1.2 On-demand QoS

A mechanism had been proposed to signal packet drops back to the user for the user to take corrective measures in case they are necessary- like increasing user Potential willingness to pay that would provide on-demand QoS.

6.4.1.3 Improving the transport of TCP flows

The lower left-hand graph (S2) shows the TCP flow that is input to the system. The second graph on the line (I2) shows the controlled flow. It is interesting to note that the TCP flow changes its rate by itself at the source, since the TCP works underneath the control provided by this resource control system. We refer to this as *Packet trunking* and may find very useful commercial applications because the network operator can provide preferential treatment to specific edge-differentiated TCP user flow aggregates [133]. In addition, in a heterogeneous network, TCP delays in the core are not uncommon. As the control is done at the ingress, these delays and their effect on other core traffic are avoided. The system thus enables the network operator to treat both high priority TCP traffic and high priority real time (UDP) traffic as one elastic trunk whose resource usage is carefully optimised to avoid core buffer overload in any traffic demand loading pattern. This basically suggests that TCP performance can greatly be improved if the control system proposed is overlayed. The overlay control provided by the proposed system eliminates congestion in the network for premium traffic and hence eliminates the need to over-stretch the transport protocols and stray them from what they are supposed to do best.

6.4.1.4 Core routers and Ingress routers

The two graphs on the top right-hand side (C1 & C2) show the outputs of the core routers. The sudden rise is due to the perturbation traffic pulse. The bandwidth can be seen to be controlled back to the set point very fast. The time delays are due to the various time lags in the network and system implementation.

It can be seen that the perturbation traffic pulse affects all the ingress controllers. This is because the perturbation flow passes through both resources. If only one resource is perturbed, only the respective resources will be affected. This emphasizes the fact that the calculation is not dependent on global knowledge. This scenario is better understood in the analysis part

where one can see interactions between various components. This is explained in the simulation sections 3.1.2 and 4.2.

6.5 IP/Photonic network controller

As stated earlier, the distributed resource controller framework can be extended to multiple layers. A controller for wavelength switching is designed and developed for the dynamic wavelength provisioning and routing system for MEMS optical switches. This system is also demonstrated in a real network at the Router Research Laboratory, Harlow Labs, Nortel Networks.

6.5.1 Scope of the IP/Photonic network controller

The model must be capable of demonstrating basic functionalities of auto-discovery, wavelength provisioning, routing, protection, management, re-configuration and fast restoration. Static (e.g. SRLG) as well as dynamic (e.g. degradation in bit error rate) information and fiber break information are to be used for control and test purposes. A distributed routing protocol is to be used for the photonic layer i.e. OSPF.

6.5.2 Network elements

The proposed experimental optical network consists of a total number of six MEMS (micro electro mechanical systems) based photonic switches in a given topology shown in

Figure **37**. Each photonic switch has a control node associated with it. These control nodes can initiate, modify, teardown and monitor the connectivity and performance of the photonic cross-connects.

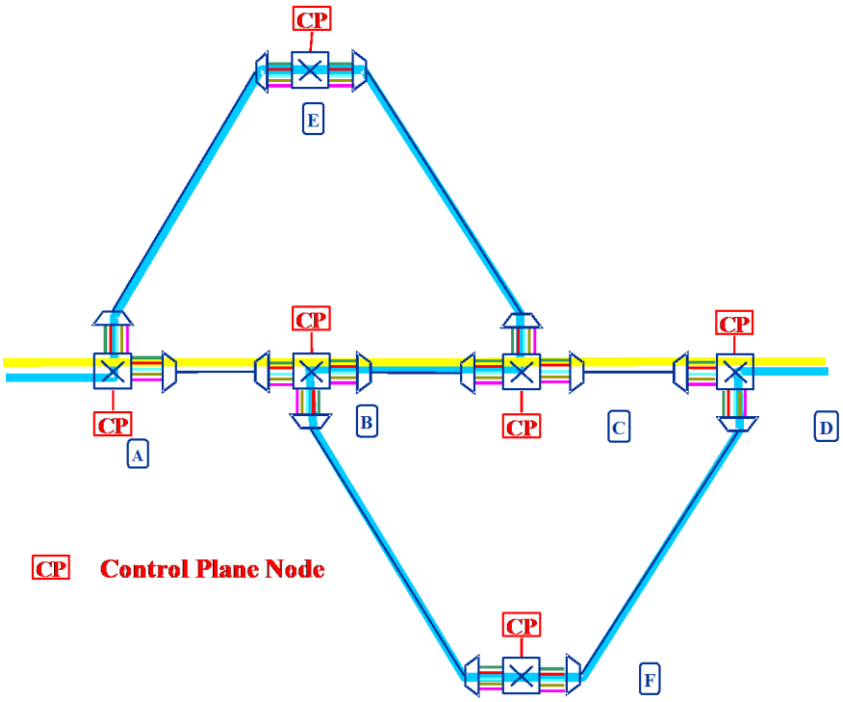


Figure 37 IP/Optical reference network topology

6.5.3 Overlay model

In the overlay model shown in Figure 38, both IP and Optical layers use MPLS technology i.e. label switching, but independently. Both layers will have their own IGP for routing. Because of this, the overlay model is expected to command more trust from the operators as this will allow multiple clients under weak trust boundaries. In the overlay model, the internals of the photonic cloud is not visible to the clients of wavelength services, i.e. IP routers etc. This is in contrast with the peer-to-peer model in which both layers share the same MPLS space, the scalability of which is not fully understood. In the overlay model, control nodes associated with the photonic cross-connects exchange the connectivity information. The overlay model proposed is more similar to the Multi-Protocol over ATM scheme.

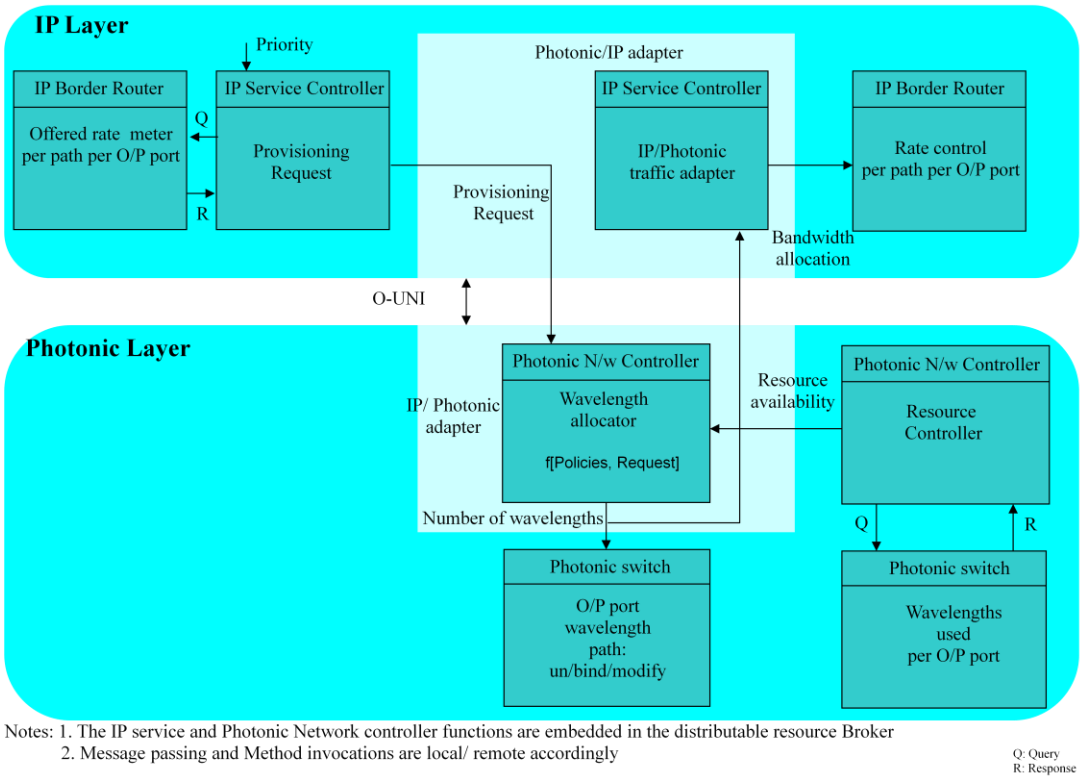


Figure 38 IP/Photonic network: abstract system architecture

6.5.4 Signalling

In-band signalling is difficult in photonic networks. At the same time, dedicating a complete channel for out-of-band signalling is less efficient. This problem area is under study.

Meanwhile, in the initial phases of the present work, out-of-band signalling through Ethernet ports is proposed to be used. This will emulate the point-to-point photonic network and current routing techniques can be used straight away.

6.5.5 IP/Optical Interface

The boundary routers and boundary Photonic cross-connects communicate via a control channel called the optical user-network interface (O-UNI).

6.5.6 Service discovery

The boundary router reachability across the optical cloud is via a service discovery mechanism across the O-UNI in conjunction with optical IGP (OSPF).

6.5.7 Provisioning

The boundary router issues light path requests across the O-UNI through extended RSVP or CR-LDP.

In the static overlay model, the path endpoints are specified by a network management system. The paths may be laid out either by the management system or by the network elements.

In the data-driven shortcut approach, the boundary routers use traffic measurements to autonomously control the number of light paths. Static provisioning information, SRLG information etc will also be used.

6.5.8 Protection

The protection and fast restoration may be designed in the local repair fashion as is done in the MPLS based fast re-route scheme.

6.5.9 Performance monitoring

Performance monitoring will be used for control of allocation. Decision thresholds for bit error rate degradation are set apriori and are used for early warning. In such a case the protection switching is set to act before the SONET triggers to action, achieving faster protection switching than the 50ms SONET limit. In the experimental set-up, 25ms protection switching was achieved.

6.5.10 Other general requirements

It was decided that the timing accuracy is not important at the initial phase of development. However, a stable control is important.

6.5.11 System design

6.5.11.1 Photonic switches

A total of six switches are connected in the given topology. These are MEMS, not opto-electronic switches. The switches have 8x8 ports with a traffic rate of 10GHz. The number of wavelengths used is 12.

Communication with the photonic switch is via one TCS (Traffic Control System) card that controls the 8x8-switch module. This card resides in the switch module. Each TCS card is linked to a Control Node by Ethernet. The control nodes can be accessed via telnet or RS232. To access the switch, one has to login to the TCS card from the control node and send set/modify instructions. Status monitoring is also possible.

6.5.11.2 System controller

In the initial phase of the development, before venturing to a fully distributed system, one way to implement the system would be to run the separate controller processes in one host computer. This host computer functions as the network and service delivery platform.

The IP system and Photonic system controllers are separate processes run by Python scripts within the host computer. Each control node can be accessed from the script by telnet. This way, CLI commands can be issued to the switches. The responses are parsed to obtain the relevant data for further processing. The script would contain the programs for auto-discovery etc as well. A fully distributed system would have border routers directly talking to the photonic border cross connects.

6.5.12 Progress of IP/Photonic network controller development

This system was developed in the Optical system division of Nortel Networks Harlow Labs and had been demonstrated to OEMs and Service Providers.

7. CONCLUSIONS AND FUTURE WORK

7.1 Conclusions

This work was originally set out develop an architecture for the autonomic, de-centralised and distributed management and control of computing and communication platforms of the electronic enterprise using minimal overhead parameters.

A theoretical framework for the study of distributed network control, feedback, resource management, simulation mechanisms, control and measurement systems, packet scheduling algorithms, multilayer architecture and OSS/BSS sub-system for value added service delivery has been developed.

It is demonstrated that, in the given conditions, the proposed distributed architecture can manage the QoS priorities, provide on-demand real-time services and ensure fairness to the users and processes. The system described can provide de-centralised and distributed management and control for the Internet traffic and resources.

The system presented has several advantages that are complimentary to the overall architectural principles of the Internet. For example, there is no co-ordination required between the ingress systems in order to use the common resource that is the Internet. Many other systems proposed uses semaphores like token passing. There is no central bandwidth broker or even a multi-tier bandwidth broker system. The term bandwidth broker, used when referring to the work presented, is an abstract noun used for convenience that is realised in a de-centralised and distributed fashion. The system presented has no single point failure in its decision path. This further contributes to the scalability attribute of the system. The work presented is scalable both in terms of the number of flows and in terms of the number of resources in the network.

7.1.1 Practical deployment considerations and associated issues

While the theoretical work and demonstration in a live network gave promising results, there are several issues to be resolved before deploying the architecture presented in a real world network. These include delay issues control/feedback information, hierarchical schedulers, inter-domain interactions etc. These are further detailed in the next section on future work.

In order to assess the limitations of the work so far and present areas of future work, the overall system could be split into a set of major areas e.g. the monitoring of network state, decision processes, the control systems, routing in different layers and communication protocols; and implementation aspects.

7.1.1.1 Incremental introduction strategy

As demonstrated, this super-system does not require global knowledge of policies and information in order to achieve fair allocation of resources. In fact, the distributed control eliminates the need for the global knowledge i.e. communication between edge-routers is not required. The signalling requirements are affordable as the super-system works with the aggregate path resource information (like price functions) fed back from network access points rather than signalled from every intra-system resource.

The computation complexity is negligible as the system uses a simple algorithm.

This super system can be overlayed, which allows it to be transparent and allows easier modification. As the system can work with partial deployment, ubiquitous deployment of the protocol is not required. This makes an incremental introduction strategy viable.

7.2 Future work

7.2.1 Monitoring and measurement

Monitoring in itself does not pose any limitation for the current implementation. With appropriate interfaces to the network hardware and software systems adequate information can be collected for input to the control system. For larger networks, event notification from the core routers to the edge routers could be used.

7.2.1.1 Loss of data in transmission

The control data is prioritized data. However, when the network grows larger, loss of data could result in the decision process making use of stale information. The decision process algorithm has to be robust against this.

7.2.1.2 Information bottlenecks

However, it is difficult to take corrective measures to deliver the QoS as the performance bottlenecks are not readily known. An example case would be where the application requires more bandwidth, it asks for more bandwidth (over and above what it currently receives) however there are some highly shared/slow servers or links in its path that prevents the application from performing well. The feedback information is not always available from the Internet equipment that is owned by disparate entities. Akella et al [3] found that about 50% of the Internet paths contain non-access bottlenecks, equally split between intra-ISP and inter-ISP links.

Referring to Hu et al [89], it looks like the best tools for Internet tomography are only 80% successful in detecting the bottlenecks. Such tomography methods would require DSP techniques for system identification etc. It is important that more work be done in this area so that the bottlenecks

are detected and corrective measures taken in order to ensure the QoS required by the user.

7.2.2 Decision Processes

The information collected from the network information base for any set of resources could be stale or partial. This could be due to a number of issues like delays, reachability issues, protocol issues etc. The system however needs to operate in this sort of scenarios. Although different types of averaging of available data are used at the moment, approaches that are more sophisticated are necessary for a wider system. One of the methods to consider could be Markov decision processes, entropy based methods etc. The idea is that as the number of resources is very large and set to be growing exponentially, the information lost from the calculations could be considered 'rare events'. These rare events could be analysed using the techniques of Partially Observable Markov Decision Processes (POMDP). POMDP models contain the sources of uncertainty i.e. stochasticity of the controlled process, and imperfect and noisy observations of the state (Kaelbling et al [112])

7.2.3 Control system techniques

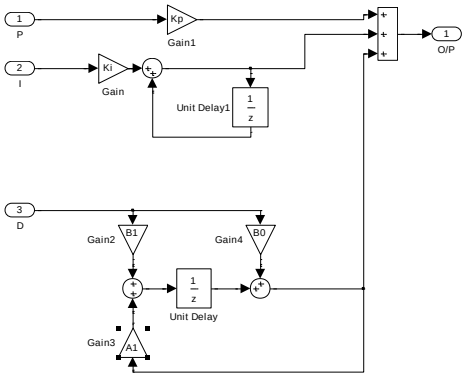
In the real networks, the control system properties will be affected by various attributes of the network e.g. the propagation delays in control as well as feedback data. The control system techniques to cope with such effects need to be studied in future work.

In the work presented, only a proportional controller was used as the tunable parameter, called proportionality multiplicand. In this, zero steady state error is achieved because of the integrator present in the controlled system. The derivative was not used to avoid amplification of the load variations. However, the use of PID controllers, Kalman filters etc need to be explored to achieve a flexible control system. The parameter tuning to achieve optimum performance is yet another area to work on.

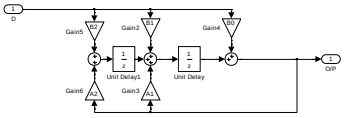
To improve the settling time, techniques such as PID (Proportional-Integral-Derivative) and/or adaptive controllers (fuzzy, neural networks etc), predictor-corrector (Kalman), feed-forward, parameter scheduling etc can be used.

7.2.3.1 Proportional-Integral-Derivative (PID) Controller

PID controllers provide full flexibility in the design and stabilisation of networks, the representation of which is given in Figure 39



Proportional+Integral+Derivative (PID) control for resource pricing block



Canonical representation of the direct digital PID controller with expandable structure

Figure 39 PID Controller model

The advantage is that while the proportional part provides better response to perturbations and increases speed of response, it still has a small steady state error and transient overshoot. The overshoot results in the price function settling to non-optimal value. By adding a term proportional to the integral of the error, the steady state error can be eliminated however the dynamic performance deteriorates. With another term proportional to the derivative of the error, the dynamic response can be damped. This helps in settling the price function to an optimal value, with a speed of response that is adequate. The PID controller includes all three terms (proportional, integral and derivative). These parameters are usually tuned to get the desired response.

7.2.3.2 Provisioning potential input

As can be seen from

Lemma **1**, the nature of the provisioning potential has an impact on the stability of the control system. This is an area for further work, to understand the nature as well as effect of the provisioning input.

7.2.3.3 Stability of multi-loop interactions

As given in section 3.2, the interaction between the flows in the distributed multi-node macro-scheduling could be modelled to study the performance under various scales and scenarios

7.2.4 Routing

In the present work, the routing information from OSPF was used in the demonstration network. This routing table is updated every 30 seconds. Similar type of protocols for photonic systems and wireless systems could be used. This has the advantage of re-using the route-changing techniques when the existing path does not provide the necessary QoS.

7.2.5 Inter-domain scalability

In the present work, a brief analysis of the inter-domain scalability is provided. Further work is necessary in inter-domain analysis, scalability in large systems, etc.

7.2.6 Communication protocols

The ad-hoc protocols used in the present work to communicate between the various players in the network- the end customers, service operators, network operators, network resources in different layers etc require to be improved in line with the protocol design principles

7.2.7 General architecture for DRC system

The communication networks has been embracing packet based technologies for a long time, due to its efficiency to share resources. The technologies available to both the individual users and enterprise users (including carrier networks) are based on digital technology using the Internet Protocol. However, the resource sharing brings in its own issues in service assurance. The proliferation of the number of devices makes centralised management and control too costly. The availability figure of the Internet based systems should be improved from the 99.6% to at least 99.999% that is typical of circuit-switched model [9]. Efficient management and control of the Internet requires distributed, autonomic, dynamic resource control system.

The analysis, simulation, prototype development and real network demonstrations carried out as part of this work gives the proof of concept required to develop a network management and control system for distributed network control and service delivery. The following is a sketch of distributed resource control system showing how the system can be architected.

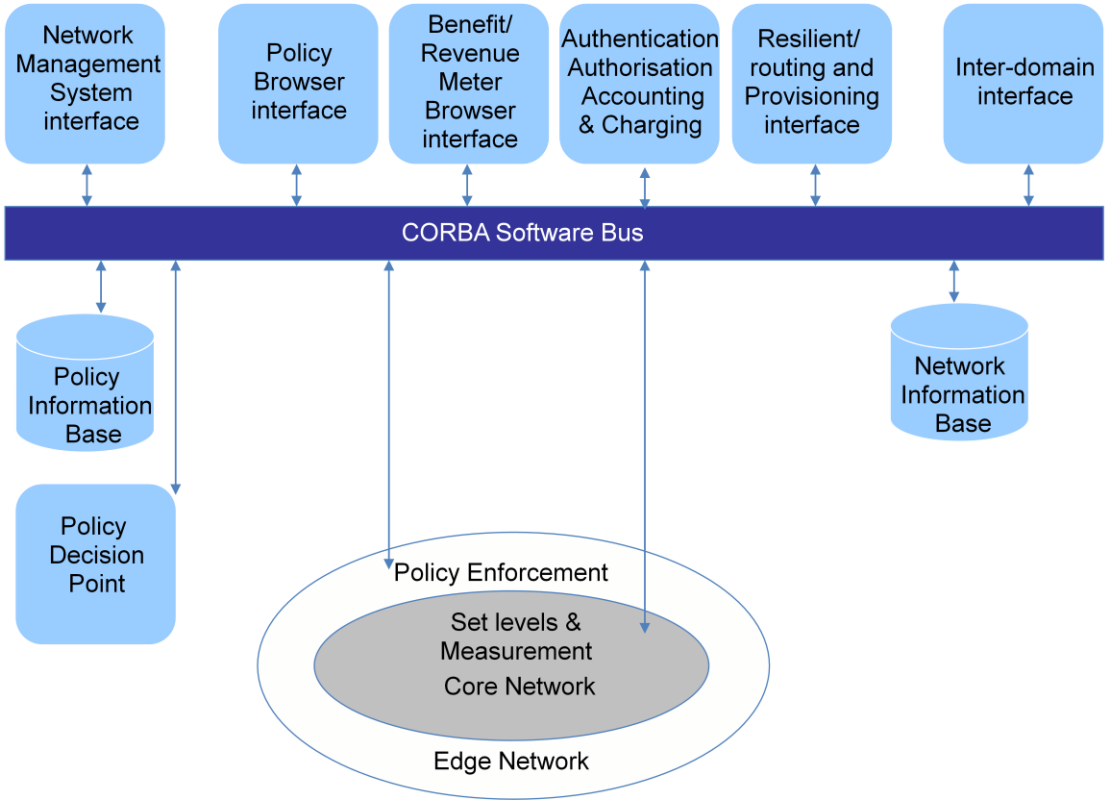


Figure 40 DRC system for the network and service delivery platform

The general architecture of the management system is shown in Figure 40. The system uses publish-subscribe architecture and hence there is no requirement for rigidly controlled servers running on network elements.

The DRC system designed as middleware provides appropriate development and run-time environment for distribution. Layered between the application and the network/OS, this is a compact and fast layer4+ solution.

As it has been explained, it is important to collect feedback from the system in order to ensure fairness to all the users and processes. It is relatively easy to sense when a given application is not performing well and to activate a ‘turbo switch’ to order more QoS.

7.2.7.1 Components and middleware for the DRC system

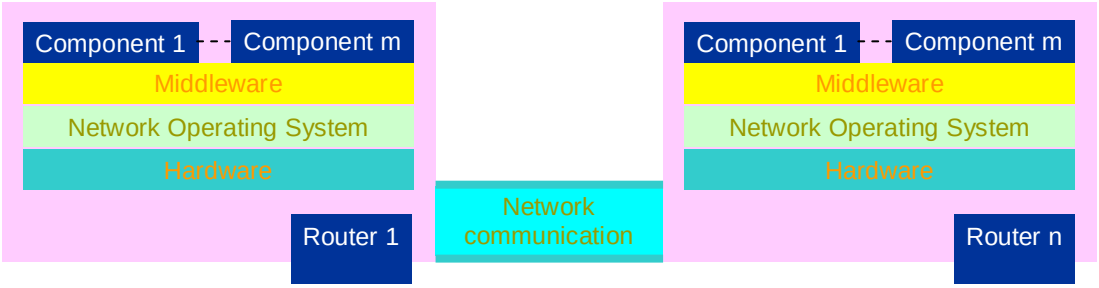


Figure 41 Middleware model

The choice of distribution architecture, language and software engineering tool for the development of a component-based model as given in Figure 41 is briefly mentioned below:

The Common Object Request Broker Architecture (CORBA) provides a flexible, scalable alternative to CMIP and SNMP. The transactions are between functional objects as object requests. This provides versatile coordination and policy enforcement.

The Java platform provides a universal interface to most of the devices and operating systems as well as easy run time installation of new functionality in other network elements. The multicore systems may require other suitable languages. Browser interfaces via XML is now very common. The Unified Modelling Language (UML) gives an effective notation for modular design and hierarchical generalisation for software engineering the system. New UML based tools deliver executable code for both hardware and software development from the architectural specification models [104].

7.2.7.2 Universal distributed network and service delivery system

The generic implementation of the de-centralised and distributed control system for the emerging Internet is based on the development of APIs for

independent functions and components orchestrated by service-oriented architecture. The concepts developed in this thesis will be realised by combining enterprise business with the intelligent network functions and OSS/BSS functions through web services.

The JAIN, OSS/J, J2EE initiatives are now being heralded as a standard method to achieve this solution over the IP Multimedia Subsystem.

BIBLIOGRAPHY

1. Adiseshu, H., Parulkar, G., Varghese, G. A reliable and scalable stripping protocol, SIGCOMM, ACM Press, 1996.
2. Aidarous, S., et al. Telecommunications Network Management Technologies and Implementations, IEEE Press, 1998
3. Akella, A., Seshan, S, Shaikh, A. An Empirical Evaluation of Wide-Area Internet Bottlenecks, Proc. Internet Measurement Conf. (IMC), ACM Press, 2003.
4. Alaettinoglu, C., et al. Towards millisecond IGP convergence, draft-alaettinoglu -ISIS-convergence-00, IETF, 2000.
5. Albers, S., et al. Delayed information and action in on-line algorithms, 30th Annual ACM Symposium on the Theory of Computing, ACM, 1998, 416-425.
6. Altman, E., et al. Congestion control as a stochastic control problem with action delays, Automatica, Elsevier, 1999.
7. Alur, R., et al. Finitary fairness, ACM Transactions on Programming Languages and Systems (TOPLAS) ACM, 1998, 1171-1194.
8. Ammann, P. Managing dynamic IP networks, McGraw Hill, 2000.
9. Andersen, D.G. Improving End-to-End Availability Using Overlay Networks, PhD Thesis, MIT 2004.
10. Apon, A., et al. The circulating processor model of parallel systems, IEEE transactions on computers, IEEE, May 1997.
11. Ash, G., et al. 20 Years of Dynamic Routing in Circuit-Switched Networks: Looking backward to the Future, Global Communications Newsletter, IEEE Communications Society, October 2004.
12. Ash, G. Dynamic routing in Telecommunications networks, McGraw-Hill, 1997.
13. Ash, G. Performance evaluation of QoS-routing methods for IP-based multiservice networks, Computer Communications Elsevier, 2003, 817-833.
14. Athuraliya, S., Low, Steven. Optimisation Flow Control II: Implementation, Technical report, Caltech, 2000.
15. Atsushi, I., et al. Pareto Set, Fairness, and Nash Equilibrium: A case study on Load Balancing, Eleventh International Symposium on Dynamic Games and Applications, December 2004.
16. Awduche, et al. Overview and Principles of Internet Traffic Engineering, RFC 3272, IETF, 2002.
17. Awerbuch, B., Azar, Yossi. Local Optimization of Global Objectives: Competitive Distributed Deadlock Resolution and Resource Allocation, Proceedings of the 35th Annual Symposium on Foundations of Computer Science, IEEE, 1994, 240–249.
18. Banerjee, R. Internetworking technologies: an engineering perspective, Prentice-Hall 2002.
19. Bartal, Y., et al. Global optimisation using local information with applications to flow control, IEEE Symposium on the Foundations of Computer Science (FOCS), 1997, 303-312.
20. Basar, T. Feedback and the Internet: Control Driving a Rapidly Advancing Technology, The S.S. Penner Distinguished Lecture in the Aerospace and Mechanical Engineering Sciences, University of California, San Diego, November 21, 2003.
21. Benjamin, D., et al. Optical services over the intelligent optical network, IEEE communications magazine, September 2001.
22. Bennet, J., Stephens, D, Zang, H. High speed, scalable and accurate implementation of packet fair queueing algorithms in ATM networks, IEEE ICNP, 1997.
23. Bennet, J., Zhang, H. WF2Q: Worst case fair Weighted Fair Queueing, INFOCOM, 1996.
24. Bennet, J., Zhang, H. Why WFQ is not good enough for the integrated services networks, Proceedings of NOSSDAV, 1996.
25. Bensaou, B., et al. Credit based fair queueing (CBFQ): a simple and feasible scheduling algorithm for packet networks, Proceedings of IEEE ATM Workshop, Spain, 1997.
26. Bertsekas, D., Gallager, R. Data Networks, Prentice Hall, 1992.

27. Bhatti, S., Knight, Graham Notes on a QoS information model for making adaptation decisions, Proc. HIPPARCH 98 - 4th International Workshop on High Performance Protocol Architectures, London, June 1998.
28. Biddiscombe, M.D., Midwinter, J.E., Sabesan, S. Application of free-market principles to telecoms resource allocation, Electronics Letters, 18 February 1999, 264-266.
29. Bierman, H.S., et al. Game theory with economic applications, Addison Wesley, 1995.
30. Bonomi, F., et al. Adaptive algorithms for feedback based flow control in High speed, wide area ATM networks, IEEE journal on selected areas in communications, September 1995
31. Breslau, L., Shenker, Scott. Best effort versus reservations: a simple comparative analysis, Sigcomm 1998.
32. Breslau, L., et al. Comments on the Performance of Measurement-Based Admission Control Algorithms, Infocom IEEE 2000.
33. Breslau, L., Jamin, S, Shenker, S. Measurement based admission control: What is the research agenda?, Proceedings of the 7th International Workshop on Quality of Service (IWQoS), IEEE, London, 1999.
34. Briscoe, B. The Direction of Value Flow in Open Multi-service Connectionless Networks, Proc. 1st International COST264 Workshop on Networked Group Communication (NGC'99), Springer LNCS, 1999.
35. Brodal, G. Fast meldable priority queues, Proceedings of the 4th International Workshop on Algorithms and Data Structures, 1995.
36. CapeClear. The Service Delivery Platform: A Collaborative Approach to Agile Business, Cape Clear, May 2005.
37. Chandok, N., et al. IP over Optical Networks: A summary of issues, draft-osu-ipo-mpls-issues-00.txt, IETF, July 2000.
38. Chandra, A., et al. Surplus Fair Scheduling: A ProportionalShare CPU Scheduling algorithm for Symmetric Multiprocessors, Proceedings of the 4th USENIX Symposium on Operating Systems Design and Implementation, 2000
39. Charny, A., Le Boudec, JY. Delay bounds in networks with aggregate scheduling, QoFIS 2000.
40. Chen, Y., et al. On the stability of network distance estimation, SIGMETRICS Perform. Eval. Rev., 2002.
41. Chiu, Jain,. Analysis of the Increase and Decrease Algorithms for Congestion Avoidance in Computer Networks, Journal of Computer Networks and ISDN, June 1989, 1-14.
42. Chiussi, F., et al. Implementing fair queueing in ATM switches- part 1: A practical methodology for the analysis of delay bounds, Globecom, 1997.
43. Chiussi, F., et al. Implementing fair queueing in ATM switches- part 2: the logarithmic calendar queue, Globecom, 1997.
44. Chiussi, F., et al. Implementing fair queueing in ATM switches: the discrete rate approach, Globecom, 1997.
45. Chiussi, F., et al. Minimum delay self clocked fair queueing algorithm for packet switched networks, INFOCOM, 1998.
46. Cho, S., et al. Dynamic weighted cell multiplexing methd for real-time traffic in ATM, Electronic letters, 15 october 1998.
47. Cisco. Internetworking design basics, Cisco, 2002.
48. Clark, D.D., Partridge, Craig, Ramming, Christopher, Wroclawski, John T. A Knowledge Plane for the Internet, SIGCOMM, 2003.
49. Cobb, J., et al. Time-shift scheduling- fair scheduling of flows in high speed networks, IEEE/ACM transactions on networking, June 1998.
50. Conant, R.C. Every Good Regulator of System Must be a Model of that System. Ross Ashby, W. ed., International Journal of Systems Science, Taylor & Francis, 1970, 89-97.
51. Cormen, T.H. Introduction to Algorithms. Leiserson, C.E. ed., MIT Press, Cambridge MA, 1990.
52. Crowcroft, J., Oechslin, Philippe. Differentiated End-to-End Internet Services using a Weighted Proportional Sharing TCP, ACM SIGCOMM Computer Communication Review, July 1998, 53-69.
53. Crowcroft, J., et al. TCP/IP and Linux protocol implementation, John Wiley & Sons, 2001.

54. Das, S., et al. Optimal and load balanced mapping of parallel priority queues in hypercubes, IEEE transactions on parallel and distributed systems, June 1996.
55. Demers, A., Keshav, S., Shenker, S. Analysis and Simulation of a Fair Queueing Algorithm, ACM, 1989.
56. Dijkstra, E.W. Self-stabilizing Systems in Spite of Distributed Control, Communications of the ACM, November 1974.
57. Dobson, S., et al. A Survey of Autonomic Communications, ACM Transactions on Autonomous and Adaptive Systems, December 2006.
58. Douglass, B.P. Doing Hard Time- developing real-time systems with UML objects and patterns, Addison Wesley, 1999.
59. Dunlop, J., Smith, DG. Telecommunication networks, Chapman & Hall, 1995.
60. Dutta, D., et al. Oblivious AQM and Nash Equilibria, Infocom, 2003.
61. Edell, R., Varaiya, Pravin. Providing Internet Access: What We Learn From INDEX, IEEE Network, Sept-Oct 1999, 18-25.
62. Emmerich, W. Engineering Distributed Objects, John Wiley & Sons, 2000.
63. Ericsson, N. Revenue maximisation in resource allocation: applications in wireless communication networks, PhD Thesis, Uppsala University, 2004.
64. EU-IST. http://www.ana-project.org, EU-IST.
65. EU-IST. http://www.autonomic-communication.org, EU-IST.
66. Fischer, et al. Impossibility of distributed consensus with one fault process, Journal of the ACM, April 1985, 374-382.
67. Fischer, M., Merritt, M. Appraising Two Decades of Distributed Computing Theory Research, Distributed Computing, 2003.
68. Fischer, M., et al. Fishspear: A priority queue algorithm, Journal of the association for computer machinery, January 1994.
69. Floyd, S. Link-sharing and Resource Management Models for Packet Networks, IEEE/ACM Transactions on Networking, September 1993.
70. Foster, I., et al. The GRID: blueprint for a new computing infrastructure, Morgan Kaufmann, 1999.
71. Fragouli, C., et al. Multimedia wireless link sharing via enhanced class based queueing with channel state dependent packet scheduling, INFOCOM 1998.
72. Franklin, G., et al. Digital control of dynamic systems, Addison Wesley, 1997.
73. Gabow, H.N., et al. An efficient approximation algorithm for the survivable network design problem, Mathematical programming, 1998 13-40.
74. Galli, D. Distributed operating systems concepts and practice, Prentice Hall, 2000.
75. Gibbens, R., Key, P. The use of games to assess user strategies for differential quality of service in the internet, Proc. Workshop on Internet Service Quality Economics (ISQE), 1999.
76. Gibbens, R.J., Kelly, F.P. Resource pricing and the evolution of congestion control, Automatica, 1999.
77. Golestani, J. A Class of End-to-End Congestion Control Algorithms for the Internet, Proceedings of ICNP, 1998.
78. Golestani, J. A self-clocked fair queueing scheme for high speed applications, Proc. INFOCOM, IEEE, April 1994, 636 - 646.
79. Goyal, A., et al. Distributed admission control scheduling and routing with stale information, Proc. of ACM-SIAM Symposium on Discrete Algorithms, October 2000
80. Goyal, P., et al. Determining end-to-end delay bounds in hetergenous networks, Multimedia systems, 1997, 157-163.
81. Goyal, P., Harrick, Vin. Generalized guaranteed rate scheduling algorithms: a framework, IEEE/ACM Transactions on networking, IEEE/ACM Transactions on networking, August 1997.
82. Goyal, P., Harrick, Vin. Start-Time Fair Queueing: A Scheduling Algorithm for Integrated Services Packet Switching Networks, IEEE/ACM Transactions on networking, October 1997.
83. Gremmelmaier, U., et al. Performance evaluation of the PNNI routing protocol using an emulation tool, XVI World telecom congress proceedings, 1997.
84. Grossglauser, M., et al. A Framework for Robust Measurement-Based Admission Control, IEEE/ACM Transactions on networking, June 1999.
85. Halsall, F. Data communications, computer networks and open systems, Addison Wesley, 1997.

86. Haverkort, B. Performance of Computer Communication systems: a model based approach, John Wiley & Son, 1998.
87. Hellerstein, J. Feedback Control of Computing Systems, IEEE Press, 2004.
88. Hong, S. Approximate analysis of timer controlled priority scheme in the single service token passing systems, IEEE/ACM trasactions on networking, April 1994.
89. Hu, N., et al. Locating Internet Bottlenecks:Algorithms, Measurements, and Implications, SIGCOMM, ACM 2004.
90. Huh, Y., Kim, C. New caching based location management scheme in personal communications systems, ICOIN, Beppu City, Japan, 2001.
91. Huitema, C. Routing in the Internet, Prentice Hall, 1995.
92. Hunt, G.C., et al. An efficient algorithm for concurrent priority queue heaps, Information processing letter, 1996, 151-157.
93. IBM. http://www.research.ibm.com/autonomic/overview/elements.html, IBM.
94. IEEE. IEEE Std 1220: IEEE Standard for Application and Management of the Systems Engineering Process, IEEE, 1998.
95. IEEE. IEEE Std 1471: IEEE Recommended Practice for Architectural Description of Software-Intensive Systems, IEEE, 2000.
96. ITU. Final report: IP-based networks: Pricing of telecommunication services, Market, Economics & Finance Unit, ITU, January 2003.
97. Jacobson, V., et al. Congestion Avoidance and Control, Proc. ACM SIGCOMM, 1988.
98. Jacobson, V. Towards differentiated services for the Internet, Technical Report, Lawrence Berkeley National Laboratory, 1998.
99. Jaffe, J. Bottleneck flow control, IEEE Transactions on communications, July 1981.
100. Jaffe, J. Flow Control Power is Nondecentralizable, IEEE Transactions on Communications, September 1981, 1301-1306.
101. Johansson, M. Resource allocation under uncertainty: Applications in mobile communications, PhD Thesis, Uppsala University, 2004.
102. Johari, R., et al. End-to-End Congestion Control for the Internet: Delays and Stability, IEEE/ACM Transactions on networking, December 2001.
103. Kadengal, R. Advanced router and switch designs for Quality of Service in distributed routing fabric, UK Teletraffic Symposium, IEE, May 2001.
104. Kadengal, R. Automated specifications for chip design using UML, UML for SoC Design Workshop, 44th DAC, San Diego, June, 2007.
105. Kadengal, R. Determining traffic information in a communications network, US Patent 6804196, November 2000.
106. Kadengal, R. Dynamic Resource Control and Management middleware for Carrier networks, UKTS16, IEE, May 2000.
107. Kadengal, R. Dynamic resource control in telecommunications networks, US Patent 6928053, December 2000.
108. Kadengal, R., et al. Management and control of multi-layer networks, US Patent 7269185, May 2001.
109. Kadengal, R. Measurement, Control and Charging strategies for ATM Networks, MSc Thesis, University of Strathclyde, 1997.
110. Kadengal, R. Proposal for IP over Photonic switch control interface design, Nortel Networks Technical Report, Feb 2001.
111. Kadengal, R., et al. Scheduling and reservation for dynamic resource control systems, US Patent 6888842, April 2000.
112. Kaelbling, L.P. Reinforcement Learning: A Survey. Littman, M.L. ed., Journal of Articial Intelligence Research, AAAI Press, 1996, 237-285.
113. Kakadia, D. Understanding Tuning TCP, Sun BluePrints, March 2004.
114. Kang, M., Wilbur, S. Handover compensation scheme for weighted fair queuing in cellular packet switched networks, Electronic letters, June 1997.
115. Kelly, C.T. Engineering flow controls for the Internet, PhD Thesis, Cambridge, 2004.
116. Kelly, F. Charging and rate control for elastic traffic, European Transactions on Telecommunications, 1997, 33-37.
117. Kelly, F., Key, Peter, Zachary, Stan. Distributed Admission Control, IEEE Journal on Selected Areas in Communications, December 2000.
118. Kelly, F., AK, Maulloo, DKH, Tan. Rate control for communication networks: shadow prices, proportional fairness and stability, Journal of the Operational Research Society, March 1998, 237-252.

119. Kephart, J.O., Chess, David M. The Vision of Autonomic Computing, Computer Magazine, IEEE Computer Society, January 2003.
120. Kershenbaum, A. Telecommunications network design algorithms, McGraw-Hill, 1993.
121. Keshav, S. A Control-Theoretic Approach to Flow Control, SigComm ACM, 1991.
122. Key, P., McAuley, Derek. Differential QoS and Pricing in Networks: where flow-control meets game theory, IEE Proceedings Software, March 1999.
123. King, C. JAIN and the Service Delivery Platform, BEA Systems April, 2004.
124. Kirkby, P., Kadengal, R. ARMAN (Advanced Resource Management) Work Package deliverable: Distributed Agreement and Policy Responsibility, DTI ref. yacc/08/02/1010, EPSRC/LINK/High Performance Interfaces and Protocols (HPIP) Project, 1998.
125. Kirkby, P., Kadengal, R. Traffic management and control using a single 'congestion price' like variable across multiple layers of network hierarchy, Proceedings of the colloquium on Control of Next Generation Networks, IEE, 1999, 1-6.
126. Kirkby, P., Kadengal, R, Carrol, J, Sabesan, S, Biddiscombe, M, et al. The use of economic and control theory analogies in the design of policy based dynamic resource controlled (DRC) network architectures, Proceedings of the International Teletraffic Congress ITC-16, Elsevier, 1999 447-456.
127. Kleinrock, L., Kamoun, F. Hierarchical Routing for Large Networks, Performance Evaluation and Optimization, Computer Networks, North-Holland Publishing Company, January 1977, 155-174.
128. Konstantinou, A.V. Towards Autonomic Networks, Columbia University, 2003
129. Korilis, Y.A., et al. Architecting noncooperative networks, Journal on Selected Areas in Communications, IEEE, September 1995.
130. Korilis, Y.A., et al. Why is flow control hard: optimality, fairness, partial and delayed information, Proc. 2nd ORSA Telecommunications Conference, March 1992.
131. Koterbi, A., et al. On the scalability of fair queuing, HotNets-III ACM, San Diego, November 2004.
132. Kumar, A., et al. Mobility modelling of rush hour traffic for location area design in cellular networks, WOWMOM, 2000.
133. Kung, H., Wang, SY. TCP trunking: design, implementation and performance, IEEE ICNP, 1999.
134. Kunniyur, S., Srikant, R. A Decentralized Adaptive ECN Marking Algorithm, Globecom, Nov. 2000.
135. Kunniyur, S., Srikant, R. End-to-end congestion control: utility functions, random losses and ECN marks, Infocom 2000.
136. Lago, P., Falcarin, Paolo, Andreetto, Alessandra, Licciardi, Carlo Alberto. Next Generation Networks: the service offering standpoint, Proceedings of the 7th IEEE-ITU International Conference on Intelligence in Networks (ICIN' 2001), Bordeaux, France, 2001, 256-261.
137. Lapsley, D., Low, Steven. An Optimiation Approach to ABR Control, ICC'98, IEEE, June 1998.
138. Law, K.E. The bandwidth guaranteed prioritized queuing and its implementations, GLOBECOM, 1997.
139. Le Boudec, J., Thiran, P. Network calculus: a theory of deterministic queueing systems for the internet, Springer, May 2004.
140. Lee, C., et al. Determination of the registration point for location update by dynamic programming in PCS, Wireless networks, 2001, 331-341.
141. Leeuwen, J.V., et al. Interval heaps, The computer journal, 1993.
142. Lerner, M., et al. Middleware networks: concepts, design and deployment of Internet infrastructure, Springer, 2000.
143. Leung, K., et al. Global mobility management by replicated databases in personal communication networks, IEEE journal on selected areas in communications, October 1997.
144. Leung, K. An update algorithm for replicated signalling databases in wireless and advanced intelligent networks, IEEE Transactions on Computers, March 1997.
145. Leung, K., et al. Use of centralized and replicated databases for global mobility management in personal communication networks, Universal Personal Communications, 1996.
146. Linux. Linux documentation, Linux Organization.
147. Lipsey, R., Chrystal, A. Economics, Oxford University Press, 2007.

148. Ljung, L. System identification: theory for the user, Prentice Hall, 1999.
149. Low, S., et al. Internet congestion control, IEEE control systems magazine, February 2002.
150. Low, S., Lapsley, David. Optimization Flow Control I: Basic Algorithm and Convergence, IEEE/ACM Transactions on Networking, December 1999.
151. Low, S. Optimization Flow Control with On-line Measurement, Proceedings of the 16th International Teletraffic Congress, June 1999
152. Mackie-Mason, J., Varian, H. Pricing Congestible Network Resources, IEEE Journal of Selected Areas in Communications, Sept. 1995.
153. Maheswaran, R., Bazar, T. Nash Equilibrium and Decentralized Negotiation in Auctioning Divisible Resources, Group Decision and Negotiation, 2003, 361-395.
154. Massoulie, L., Roberts, J. Bandwidth Sharing: Objectives and Algorithms, INFOCOM, 2002.
155. Mathworks. Matlab documentation, Mathworks.
156. Mc Dougall, R., et al. Resource Management, Sun Microsystems Press, 1999.
157. McDysan, D. QoS & Traffic Management in IP & ATM Networks, McGraw-Hill, 2000.
158. Melcher, B., Mitchell, B. Towards an Autonomic Framework: Self-Configuring Network Services and Developing Autonomic Applications, Intel Technology Journal, November 17, 2004.
159. Minh, H., et al. User profile replication with caching for distributed location management in mobile communication networks, SAC, 2001.
160. Mo, J., Walrand, Jean. Fair End-to-End Window-based Congestion Control, IEEE/ACM Transactions on Networking, September 1998.
161. Moore, A.W. Measurement-based management of network resources, Cambridge University, April 2002.
162. Mukherjee, B. Optical communication networks, McGraw-Hill, 1997.
163. Nagle, J. On Packet Switches With Infinite Storage, RFC 970, IETF, 1985.
164. Nakai, J., Rob, F, Van Der, Wijngaart. Applicability of Markets to Global Scheduling in Grids - Critical Examination of General Equilibrium Theory and Market Folklore, NAS Technical Report NAS-03-004, February 2003.
165. Nakai, J. Pricing in computer networks: reading between the lines and beyond, Technical report, Nasa January 2002.
166. Narvaez, P. Routing reconfiguration in IP Networks, Doctoral thesis, MIT, May 2000.
167. Naughton, J. A Brief History of the Future: the origins of the Internet, Phoenix, 2000.
168. Nekoogar, F., et al. Digital control using digital signal processing, Addison Wesley, 1999.
169. Newman, P., et al. Flow labeled IP: Connectionless ATM Under IP, Proceedings Networld +Interop, April 1996.
170. Newman, P., et al. Flow labeled IP: Connectionless ATM Under IP, Proceedings, Networld +Interop, April 1996.
171. Nieh, J., et al. Virtual-Time Round-Robin: An O(1) Proportional Share Scheduler, Proceedings of the 2001 USENIX Annual Technical Conference, June 2001.
172. Odlyzko, A. Data Networks are Lightly Utilized and will Stay that Way, Review of Network Economics, September 2003.
173. Odlyzko, A. The economics of the Internet: Utility, utilization, pricing, and Quality of Service, AT&T Labs Technical Report: 99-08, 1999.
174. Odlyzko, A. The Internet and other networks: Utilization rates and their implications, Telecommunications Policy Research Conference, 1998.
175. Odlyzko, A. Internet pricing and the history of communications, Computer Networks, 1999.
176. O'Donnell, D. TCO: don't let hidden IT expenses hurt your company, Software Magazine, August, 1998.
177. O'Mahony, M., et al. The application of optical packet switching in future communication networks, IEEE Communications magazine, March 2000.
178. OMG. Unified Modeling Language: Superstructure version 2.0, OMG, 1995.
179. Paganini, F., Doyle, J., Low, S.,. Scalable Laws for Stable Network Congestion Control, Proc of Conference on Decision and Control, December 2001.
180. Parberry, I. Load sharing with parallel priority queues, Journal of computer and systems sciences, 1995, 64-73.

181. Parekh, A.K. A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single-Node Case. Gallager, R.G. ed., IEEE/ACM Transactions on Networking, June 1993, 344-357.
182. Pareto, V. Manual of Political Economy, Augustus M. Kelley, 1906
183. Perlman, R. Interconnections: Bridges, routers, switches and internetworking protocols, Addison Wesley, 2000.
184. Perlmutter, B. Virtual private networking: a view from the trenches, Prentice Hall, 1999.
185. Ping, et al. A Market-Based Scheduler for JXTA-Based Peer-to-Peer Computing System, ICCSA 2004, 2004, 147-157.
186. Pinotti, M., et al. Parallel algorithms for priority queue operations, Theoretical computer science, 1995, 171-180.
187. Puliu, Y., et al. The studies of stability for dynamic routing in packet switching network, IEEE conference on computer, communications, control and power engineering, Beijing, Oct 19-21, 1993.
188. Putzolu, D., et al. The Pheonix Framework: A practical architecture for programmable networks, IEEE Communications magazine, March 2000 160.
189. Python. Python documentation, Python Organization.
190. Qui, J. Measurement-Based Admission Control with Aggregate Traffic Envelopes, IEEE/ACM Transactions on networking, April 2001.
191. Ramaswami, R., Segall, Adrian. Distributed Network Control for optical networks, IEEE/ACM Transactions on Networking, Dec 1997.
192. Rao, S., Petersen, ER Optimal pricing of priority services, Operations research, Jan-Feb 1998.
193. Reichl, P. Trends in Internet economics, Technical Report, Telecommunication research centre, Vienna, October 2003.
194. Rhissa, A.A., Hassnaoui, Adil. Global self-management of network and telecommunication information systems and services, IEEE/SITIS, 2005.
195. Sabry, M., Thronhill, NF, Midwinter, JE. An analytical approach to charging mechanisms using control theory, IEE Colloquium on Charging for ATM - The Reality Arrives, IEE, London, 20 Nov 1997.
196. Saha, D., etal. Multirate scheduling of VBR video traffic in ATM networks, IEEE Journal of Selected Areas in Communications, 1997.
197. Sakamoto, H., et al. Performance analysis method for system bus architectures in multimedia communication terminals, Electronics and communications in Japan Part 3, 1994.
198. Schmid, S., Eggert, Lars, Brunner, Marcus, Quittek, Jürgen. TurfNet: An Architecture for Dynamically Composable Networks, International Workshop on Autonomic Communication WAC'04, IFIP, Berlin, 2004
199. Schof, S. Efficient data structures for time warp simulation queues, Journal of systems architecture, 1998, 497-517.
200. Semret, N., et al. Pricing, Provisioning and Peering: Dynamic Markets for Differentiated Internet Services and Implications for Network Interconnections, IEEE Journal on selected areas in communications, December 2000.
201. Sharma, V., et al. Developing e-commerce sits: an integrated approach, Addison-Wesley, 2000.
202. Shenker, S. Fundamental design issues for the future Internet, IEEE Journal on Selected Areas in Communication, September 1995.
203. Shenker, S. Making greed work in networks: a game theoretic analysis of switch service disciplines, SIGCOMM ACM 1994.
204. Shenker, S. A Theoretical Analysis of Feedback Flow Control, SIGCOMM, 1990, 156-165.
205. Shreedhar, M., Varghese, G. Efficient fair queueing using deficit round robin, IEEE/ACM transactions on networking, June 1996
206. Smirnov, M. Autonomic Communication- Research Agenda for a New Communication Paradigm, Fraunhofer, November 2004.
207. Stallings, W. SNMP, SNMPv2, SNMPv3, and RMON 1 and 2, Addison-Wesley 1998.
208. Stiliadis, D., et al. Efficient fair queueing algorithms for packet switches networks, IEEE/ACM transactions on networking, April 1998.

209. Subramani, et al. Distributed Job Scheduling on Computational Grids using Multiple Simultaneous Requests, Proceedings of 11th IEEE Symposium on High Performance Distributed Computing (HPDC 2002), July 2002.
210. Sun. JAIN and Java in Communications, Sun Microsystems, March 2004.
211. Sun, J. Wireless Channel Allocation Using An Auction Algorithm, IEEE Journal on Selected Areas in Communications, to appear 2006.
212. Sun-microsystems. Java documentation, Sun Microsystems.
213. Suri, S., Varghese, G, Chandranmenon, G. Leap forward virtual clock: a new fair queueing scheme with guaranteed delays and throughput fairness, INFOCOM 1997.
214. Tan, D. Rate control and user behaviour in communications networks, 4th INFORMS Telecommunications conference, Florida, March 1998.
215. Tcl/Tk. Tcl/Tk documentation, TTDP.
216. TelelogicAB. Tau UML documentation, Telelogic, 2006.
217. Terplan, K., et al. Applications for distributed systems and network management, Wiley, 1995.
218. Van, E., Boas, P. Design and implementation of an efficient priority queue, Mathematical systems theory, 1977, 99-127.
219. Verdu, S., et al. On Minimax Robustness: A General Approach and Applications, IEEE Transactions on Information theory, March 1984.
220. Villamizar, C. Optimised Multipath, draft-ietf-ospf-omp-00, IETF, 2000.
221. Wolf, T., et al. Towards Autonomic Computing: Agent-Based Modelling, Dynamical Systems Analysis, and Decentralised Control. Tianfield, H., Unland, R ed., Proceedings of the First International Workshop on Autonomic Computing Principles and Architectures, 2003, 10.
222. Xiao, X., et al. Traffic engineering with MPLS, America's Network, November 15, 1999.
223. Yaïche, H., Mazumdar, Ravi R, Rosenberg, Catherine. A Game Theoretic Framework for Bandwidth Allocation and Pricing in Broadband Networks, IEEE/ACM Transactions on networking, October 2000.
224. Yeom, I., et al. Realizing throughput guarantees in a differentiated services network, Proc. of IEEE International Conference on Multimedia Computing and Systems- ICMCS, Florence, June 1999.
225. Yerramalla, S., Edgar, Fuller, Martin, Mladenovski, Bojan, Cukic. Lyapunov Analysis of Neural Network Stability in an Adaptive Flight Control System, Self-Stabilizing Systems, 2003, 77-91.
226. Youngmi, J. Nash equilibria of a generic networking game with applications to circuit-switched networks, Infocom 2003.
227. Yuming, J., et al. Measurement-Based Admission Control: A Revisit, Seventeenth Nordic Teletraffic Seminar, Fornebu, August 25-27, 2004.
228. Yusheng, J., et al. Virtual rate based queueing: a generalised queueing discipline for switches and high speed networks, IEICE Transations on communications, December 1994.
229. Zachman, J. Enterprise architecture: a framework, ZIFA, 1997.
230. Zappata, A., et al. Next-Generation 100-Gigabit Metro Ethernet (100 GbME) using Multiwavelength Optical Rings, Journal of light wave technology, November 2004.
231. Zhang, H. Service Disciplines For Guaranteed Performance Service in Packet-Switching Networks, Proceedings of the IEEE, October 1995.