ADAPTIVE AGENT COMMUNITIES FOR PROVIDING SERVICES IN DYNAMIC NETWORKS

by

NAYANTARA MALLESH

Presented to the Faculty of the Graduate School of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements
for the Degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

THE UNIVERSITY OF TEXAS AT ARLINGTON
December 2005

To my parents and brother who have supported me
and inspired me to reach my goals
at all odds

ACKNOWLEDGEMENTS

I would like to thank my supervising professor Dr. Mohan Kumar for guiding me throughout the course of my thesis work. He has been a constant source of motivation for me to give my best. I would also like to thank Dr. Ramesh Yerraballi and Dr. Hao Che for taking time to be on my thesis committee. Their constructive criticism of my work served to give my thesis work overall improvement and clarity.

I would like to express my gratitude to the Dean of Graduate Studies and Department of Computer Science and Engineering for the funding provided to me. Working with the department as a teaching assistant has given me valuable experience. I would like to thank the University of Texas at Arlington for the academically stimulating environment.

Finally, I would like to thank my parents, Wg. Cdr. G. R. Mallesh and Mrs. Gopika Mallesh for their love, guidance and support and for instilling in me a belief in my own abilities. I would like to thank my brother Rishikesh Mallesh for his motivation, inspiration and words of support during the times I needed them most. I would like to express my thanks to Jeevan and Usha for their encouragement and for always being there for me. I would like to thank all my friends for their friendship, advise, cooperation and help in my work. Most of all I would like to thank Swapna for her friendship, support and the late night study sessions throughout our time at UTA.

November 16, 2005

ABSTRACT

ADAPTIVE AGENT COMMUNITIES FOR PROVIDING SERVICES IN DYNAMIC NETWORKS

Publication No. ______

NAYANTARA MALLESH, M.S.

The University of Texas at Arlington, 2005

Supervising Professor: Mohan Kumar

New network applications are being created everyday to accommodate diverse user needs. Delivering services to the user in a timely manner taking into account network conditions, resources allocated and network load is a challenge. Multiprotocol Label Switching attempts to overcome best-effort service by providing a method for routing traffic around network congestion, resource reservation and quality of service (QoS) capabilities. IntServ and DiffServ are two other QoS models in use today. IntServ provides per-flow guarantee of quality while DiffServ is based on aggregate service classes. Adaptive Network Service (ANS) is a community of adaptive, collaborating agents residing in the network that aims to provide enhanced performance, better quality of service and improved efficiency of network resources. ANS agents are distributed across the network and are strategically located to provide services and monitor network conditions. ANS agents gather network information in these locations and exchange this information with other agents in the community. Awareness of current network status enables the agent community to efficiently allocate resources and provide services in response to incoming user requests. Sharing information allows agents in one part of the network to be aware of conditions in other parts of the network. ANS uses this information to route user data flows away from congested network areas. The agents also share resource utilization information in different ANS nodes. Routing user connections to different service points based on current resource utilization leads to efficient use of resources. In our implementation we provide a TCP based service for transferring bulk data from a source to a destination. When a user contacts ANS, ANS first determines the best available node to service the user request. The ANS node to service the request is determined before the start of data flow and is selected on the basis of i) least congestion around the ANS node and ii) maximum availability of resources in the ANS node. This thesis describes an architecture for ANS and derives a mathematical model for ANS's bulk data transfer service while presenting simulation results for a various experiments demonstrating the benefits of using ANS. Simulation results and model estimates show that ANS is able to achieve superior data transfer throughput compared to connections that do not use ANS scheme. Simulation results show that use of ANS improves data transfer performance by a factor of 2 in low traffic conditions and by a factor of up to 3.8 times in high traffic conditions. Future work in this direction includes introducing new services into the ANS framework and improving ANS agent intelligence to deliver these services in a user friendly way. We intend to enhance ANS for applications in pervasive computing environments.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Introduction

Traffic in the Internet has shown a consistent growth in past the few years. The number of devices that connect to the Internet is estimated to reach a billion by the year 2006 [1]. The variety of services available to the user on the Internet has grown vastly, creating a greater demand for instant availability, maximum reliability and minimum delay in accessing these services [1]. The growth and variety of services on the Internet has increased the demand for better quality of service and at the same time has led to an increase in network congestion and delay of service perceived by the user. The Internet has also seen the increasing deployment of multimedia applications like video-on-demand, IP telephony and video conferencing.

Best-effort delivery provided by the Internet is not best suited to requirements of multimedia traffic. The dynamic routing protocols used in the Internet adapt data routes to changes in network topology. These basic features of the Internet do not actively provide user performance and do not provide sufficient support for efficient network resource utilization. Many architectures and technologies have been proposed to enhance the service quality provided by IP networks [2]-[3]. Multiprotocol Label Switching (MPLS) attempts to overcome the limitations of best-effort Internet Protocol (IP) by providing a mechanism for routing traffic around link failures and network congestion [4],[5]. Integrated Services (IntServ) is a protocol architecture designed to handle traffic using Reservation Protocol (RSVP) to specify service priority and resource requirements along the path of data transfer[6]. Differentiated Services (DiffServ) is an alternative mechanism that allocates different service level to different classes of user traffic. DiffServ is scalable and works in the network core by categorizing traffic based on quality of service requirements[7].

MPLS attempts to provide a connection-oriented mechanism over the connectionless service provided by IP. This reduces the basic flexibility of the IP protocol. MPLS is also likely to be applicable within well managed networks where every node is able to provide support for MPLS and its related protocols [8]. DiffServ is not adaptive and works with static service agreements. In the real world, network topologies are subject to change. Real time flows like video conferencing require service guarantees on flow-by-flow basis which DiffServ is unable to provide since it operates on aggregates of data flows. The IntServ reservation model, on the other hand, provides traffic control on a flow by flow basis. IntServ's granularity causes serious concerns with its capacity to scale to the ever increasing concentration of user flows in the network.

1.2 Motivation

Efficient allocation of resources in the Internet is an issue of prime importance. Service providers want efficient usage of the deployed resources to better profits. Efficient resource allocation also balances load on the network and helps to evenly spread traffic load. Load balancing in the network is handled by routers. Load balancing occurs when a router has multiple equal cost paths to a destination.

Routing data flows away from congestion is an advantage to the network as well as to the user. It helps to channelize incoming traffic to network areas experiencing less traffic which in turn have greater residual capacity. This allows the network to provide better service quality to the user flows.

Dynamic IP routing adapts to changing network conditions. If a path breaks down and becomes unavailable for data to pass through, the IP routing mechanism detects this failure and over time begins to route data around the problematic location. But, dynamic IP routing has several shortcomings. 1) Too late to respond 2) Short sightedness and not knowing the overall picture.

Congestion control in the internet is implemented by the TCP congestion avoidance and control mechanism. TCP controls the flow of packets into the network by regulating the number of packets the source sends into the network. A TCP source increases or decreases its sending rate based on the rate it receives acknowledgements from the destination. Acknowledgements from the destination indicate the level of congestion the packets are experiencing on the path between source and destination in either direction. TCP modulates packet flow based on its perception of congestion which relies on TCP's ability to detect increased delay and segment loss. Congestion control by TCP alone is not enough to ensure good congestion control in the Internet which carries a sizeable percentage of non-TCP data.

1.3 Contributions of this Thesis

Adaptive Network Service (ANS) is a community of adaptive, collaborating agents residing in the network that aims to provide enhanced performance, better quality of service and improved efficiency of network resources. In this work we show the latency of user connections is dramatically reduced when the ANS scheme is used. ANS routes incoming user requests to service nodes that are experiencing relatively less traffic. ANS considers available resources and traffic conditions around the service node before routing user requests to the node.

ANS agents monitor congestion indicators in the network to gather information on the state of congestion in the network. This information is then distributed to other agents in the ANS community. By distributing network information, ANS agents become aware of traffic conditions in different parts of the network. This allows agents to make decisions on the best route for incoming user requests. Suppose an agent is aware that the network around a given ANS node is congested then it routes the user connection to a different ANS node thus moving the data flow away from existing congestion. Agents in the ANS community also monitor usage of resources in the community. User requests are routed to available ANS nodes, thus routing new connections away from busy nodes. This supports load balancing of user data flows in the network. By exchanging information pertaining to current state of network congestion and current resource usage levels, ANS is able to provide a traffic aware service provisioning mechanism in dynamic networks. ANS agents and services are present on a subset of nodes in the network and do not need to be deployed on every node in order to make this an effective service. This is a major advantage over architectures like MPLS and DiffServ.

Network Simulator 2 (NS2) results show that the ANS scheme improves data transfer performance by a factor of 2 in low traffic conditions and by a factor of up to 3.8 times in high traffic conditions. Simulation results for bulk data transfer using TCP and multimedia data transfer using Realtime Transport Protocol (RTP) testify that the ANS scheme can be used very effectively to improve data transfer performance in the network. ANS is a scheme that introduces a traffic and resource aware system aimed at adapting incoming user traffic to network conditions. ANS at the same time provides improved user performance and efficient usage of available network resources.

1.4 Organization of Thesis

The thesis is organized as follows. Chapter 2 gives a detailed overview of of current literature in network congestion control schemes and and quality of service architectures. Chapter 3 explains ANS architecture in detail and describes the functionality of ANS service. Chapter 4 details the mathematical model derived to predict the latency of a TCP flow using ANS service. The simulation environment, details of the simulated network and the experimental results and analysis are presented in Chapter 5. Conclusions from this work are drawn and the scope for future work is discussed in Chapter 6.

CHAPTER 2

BACKGROUND AND LITERATURE REVIEW

2.1 Growth of the Internet

Since its inception, the Internet has seen a continous growth in the number of devices connected to it as well as in its capacity to handle growing traffic demands. The number of users accessing the Internet has grown with an increasing number of people connecting to the Internet every year [1]. Users connect to the Internet using a vast array of devices. These devices include mobile devices that aim to provide network connectivity to the user anytime and anywhere. Hand held mobile computing devices are incorporating processing and data intensive applications to suit user needs [9],[10]. Applications on such devices frequently need access to data and support from applications residing in the core of the network. Users also access a variety of applications like email, text information on web sites, multimedia applications like streaming audio and video, Voice over Internet Protocol (VoIP) and distributed applications. Many users require Virtual Private Networks (VPN) functionality to securely connect geographically remote establishments. Applications that satisfy these user requirements have become increasingly complex and data-intensive.

Multimedia applications transfer text, graphics, voice, data and video traffic between users. Multimedia data transfers are usually large and consume high amounts of bandwidth. Multimedia applications have specific quality of service requirements to ensure basic service quality [11]. They require real time data transfer and place bounds on throughput, delay, variation in delay and error rate among other network factors. VoIP applications are gaining popularity due to their low cost communications abilities. VoIP or Telephony offers new supplementary features and multicast services over the Internet at no extra cost [12]. Bandwidth guarantees are generally used to provide service to such applications to ensure better performance than best effort traffic. Distributed applications run on a number of computers that may be geographically separated from each other and are connected using some kind of transmission media. Distributed computing applications are decomposed so that data and processing requirements are distributed across the network. Popular applications like distributed multimedia demand large bandwidth and network resources for their operation. The network must accommodate demands made by these applications in order to satisfy user requirements. The Internet best-effort service is not equipped to handle large bandwidth data transfers with stringent bounds on delays, loss and throughput.

Quality of service (QoS) is an indirect measure of the quantity and quality of network resources that can be provided to data flows traversing the network. QoS mechanisms reserve resources and provide priority processing for select data transfers. The ability to discriminate data flows is an important aspect of providing quality network service. Differentiating between data flows allows the network to allocate resources according to the resource requirements of each data flow. The network is able to guarantee specific performance levels for data intensive application flows like interactive multimedia and VoIP well before beginning the data transfer.

2.2 Quality of Service

Quality of Service is the ability of the network to provide a quantifiable level of service to selective data flows traversing the network. QoS ensures sufficient resources for specific applications and at the same time allow network resources to be used efficiently. QoS mechanisms satisfy the need to quantify the quality of service that applications expect from the network and ensure that values of parameters like available bandwidth, delay, loss and variation in delay lie within specified bounds. Bandwidth is the rate at which data is carried by the network. Delay or latency specifies the time for data transmission from source to destination. Jitter is variation in delay. Loss or reliability is the ratio of the number of packets lost by the network to the number of packets sent from source to destination [13].

Quality of service support allows service providers to have better control over their network resources. Bandwidth can be pre allocated to service high priority data flows without over-provisioning resources. Service providers can use QoS mechanisms to implement differentiated pricing schemes. Resources can be exclusively reserved for business critical transaction processing and interactive applications. Such applications can be given priority treatment on backbone links by limiting bandwidth used by low priority data flows like email and bulk data transfers.

Quality of service concepts also allow for preferential treatment of data flows in the network. This allows the network to recognize that some applications are more critical than others and thus require higher performance from the network. Best-effort delivery treats all data flows equally, which may result in low priority data-intensive applications choking the network and denying precious bandwidth to critical data flows. Applications can be categorized based on the quality of service they expect from the network. Applications like file transfer and email are not sensitive to delay. These applications are sensitive to loss. On the other hand video-on-demand, telephony and video conferencing are less affected by loss but are extremely sensitive to delay [14]. Different traffic in the network have different priorities. Traffic prioritization may be based on a number of things. Customers who pay for greater bandwidth require data flows that emanate or end in their networks to enjoy a higher priority over other traffic. In times of network congestion, resolving congestion is of primary importance and network management traffic must have greater priority over other data flows. Therefore, it is important for the network to have the ability to differentiate between different traffic flows and provide required service quality to different flows [15].

2.3 Congestion in the Internet

As networks move towards higher speeds, handling of greater amounts of data supporting interactions between a wider variety of protocols and end systems results in under utilization of many resources in the network. Congestion occurs when the number of packets present in the network begins to exceed the capacity of the network [15]. Many times, congestion is a result of traffic being concentrated in certain parts of the network while other areas of the network are relatively under utilized.

2.4 Congestion Avoidance and Control Techniques

Congestion control involves controlling the number of packets that enter the network and aiming to keep this number well below the level at which network performance begins to degrade.

2.4.1 Issues related to congestion control

2.4.1.1 Resource Reservation

Resources for an application can be reserved before the first byte of data is sent across the network. The application must be able to specify its requirements to the network before it can start using network resources. This allows the network to reserve resources in favor of the application. In the event that enough resources are not currently available to support the user flow, the network can refuse the connection or negotiate with the application to either connect at a later time or downgrade the quality requirements. Negotiating resource requirements prior to sending data also allows the network to monitor network usage of the particular application. Bandwidth is guaranteed to an application within the agreed bounds. If an application exceeds the service agreement then packets are marked by the network and are candidates for discarding if congestion develops or if network resource are limited along the way.

2.4.1.2 Equal treatment of flows

While the network differentiates flows to ensure quality of service to critical applications it must ensure that in the absence of these requirements all flows are treated equally.

2.4.2 Backpressure

Backpressure can be exerted on the basis of links or logical connections. In the figure Fig. 1, if the buffers in node z become filled and the node is congested, node z can slow or stop the flow of packets from node y. Due to this restriction from node z, node y may also become congested since the incoming rate continues but the outgoing packet rate has reduced. Node y in turn can then signal node x to reduce or stop its packet flow. Thus backpressure, much in the same way as it works in fluids, propagates in the direction opposite the direction of packet flow. The backpressure propagates backwards and eventually makes its way to the data sources. The sources are then expected to restrict data flow into the network which aids in easing congestion. Backpressure can be applied on specific data links or to specific logical connections. The idea is to restrict data on those flows that are burdening the network.

This technique however cannot be used in networks where there is no facility to implement flow control between one router to the next. IP networks traditionally do not have any facility for regulating packet flow between routers along the path from source to destination.

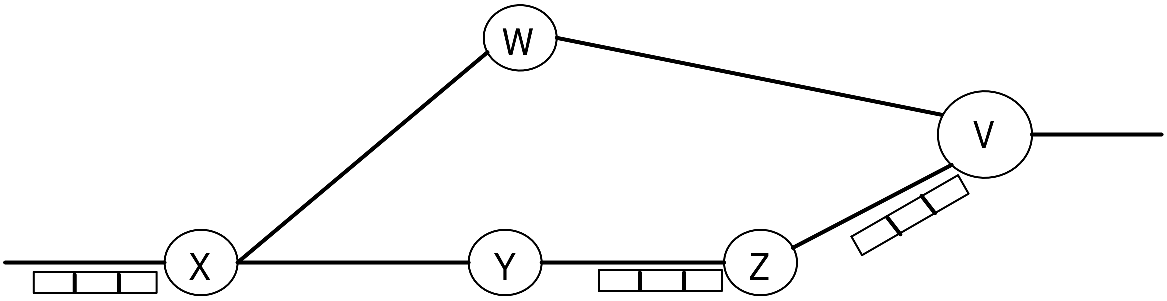


Figure 2.1: Backpressure

2.4.3 Source Quench

Internet Control Message Protocol (ICMP) source quench packet is generated at a congested node to signal a source node to restrict traffic flow. The source quench packet maybe used by an intermediate node or a destination node to request the sender to reduce the rate at which it is sending traffic. The source of packets is expected to immediately reduce the rate at which it is sending traffic to the destination node. The advantage of using this method is that it is relatively quicker to perceive congestion. The downside to this method is that more traffic gets generated in the form of source quench packets during congestion. It is also found that allowing the congested node to just drop a packet has the same effect as sending the source quench packet in a congested node which is running on low available resources.

2.4.4 Implicit Congestion Signalling

Congestion is implicitly perceived by a source sending data into a network when the sources observes that data packets are being discarded or when the transmission delay of the packet is significantly higher than the time for the packet to propagate from source to destination. Implicit congestion signalling is an effective technique in situations where a majority of the sources are able to detect congestion. Congestion is relieved when such sources control the rate of input of packets into the network. The advantage of implicit signaling is that it does not require support from routers and is usually implemented by the transport protocol in the end systems. TCP congestion avoidance and control mechanism, for example, uses implicit congestion signaling to control the flow of packets from the source into the network. Another advantage of this approach is that there are no explicit packets generated to signal congestion.

2.4.5 Explicit Congestion Signalling

In this method of congestion control, the network explicitly informs end systems of the presence of congestion inside the network. It is then the responsibility of end systems to react to congestion by reducing the number of packets sent into the network. Explicit notifications can be sent in either the in the direction of data flow (Forward direction) or in the in the direction of the source (Backward direction). Forward notification notifies the destination system that the packets received have experienced congestion and the higher layers using the data flow must exercise some form of flow control. Backward notification informs the source of congestion in the network. The source is then expected to control the flow of packets into the network. Explicit congestion works by either setting a bit in packets headed to the source or destination system. The end system checks these bits to gather information about congestion in the network. Explicit congestion method may also use independent control packets to signal congestion to end systems.

2.5 Traffic Engineering

Traffic Engineering is the ability to map traffic routes onto the existing physical network topology while adhering to traffic performance and resource utilization objectives. As defined in [16], it involves the measurement, modeling and characterization of traffic patterns in the network. The information generated out of these techniques is used to tune the network to achieve specific performance targets. As user expectations from the network become more demanding, complex provisioning of resources is required to satisfy user demands and achieve optimal resource allocation. The expense of network resources, the critical nature of business needs and growing competition only serve to enhance the need for Traffic Engineering.

Traffic Engineering concentrates on providing traffic based performance while providing resource based performance. Providing good performance to traffic flows involves keeping predetermined network characteristics within acceptable bounds. Examples of these network parameters were discussed in Section 2.2. In networks where there is no differentiation between traffic flows, keeping packet loss to a minimum holds highest priority. On the other hand, it is important to ensure that resources in the network are optimally utilized while providing good quality of service to user data flows. Resources distributed across the network may not always be utilized equally. Based on traffic patterns, most times, traffic load is concentrated on certain parts of the network while other parts go under utilized. Such situations lead to congestion in the heavy traffic areas and may result in denial of service to users while resources to handle requests exist. Traffic Engineering ensures that traffic is routed through alternate routes in the network in times of high demand.

Minimizing congestion is a goal that is advantageous to optimal resource management as well as to traffic performance objectives. Congestion usually occurs when the network is unable to handle the offered load. Networks become congested either as a result of resource shortage or as a result of improper routing of traffic among existing resources. When network capacity shortage exists, the problem can be dealt with using the techniques like admission control which regulate inflow of data into the network and other flow control techniques [16],[17]. Traffic Engineering uses a load balancing approach to minimizing network congestion. This approach argues that congestion can be avoided by keeping resource utilization below maximum resource capacity. This is achieved by routing traffic using efficient resource allocation techniques. As a result user flows experience less packet loss and transit delay and better throughput.

The functionality of current Internet interior gateway protocols do not accommodate Traffic Engineering goals [16]. These routing protocols are based on a single routing parameter. Routing decisions are based on network topology. Current resource availability and traffic characteristics are not considered while making routing decisions. Interior gateway protocols often suggest the same path for different data streams, thus resulting in concentration of traffic load on certain areas of the network. Even when other parts of the network are relatively under utilized, network congestion results from convergence of traffic in parts where maximum utilization is already reached. Overlay models like IP over Asynchronous Transfer Mode (ATM) create virtual topologies over the physical network topology [18]. Such approaches take significant steps towards achieving Traffic Engineering goals. ATM allows administrators to provision virtual data paths across the network. Resources for these paths can be configured before data transfer begins.

2.6 Overlay Networks

Overlay networks are built over existing netowkrs and add a layer of abstraction over the existing network setup. Overlay networks are built to provide services and functionalities that maynot be available in the existing infrastructure [19].

2.7 Multi Protocol Label Switching

Multi Protocol Label Switching is a QoS mechanism that has been the focus of recent standardization efforts related to tag switching. MPLS has a distinct advantage over other QoS mechanism in that it provides a broader spectrum of traffic management and QoS support than the DiffServ and IntServ architecture discussed earlier. Most importantly, MPLS provides five basic capabilities that have ensured its emergence as a technology generating tremendous interest in the past few years.

2.7.1 MPLS Capabilities

2.7.1.1 Enhanced Network Resiliency

A pure IP based network falls short in providing strictly defined Service Level Agreements to its customers. In such a scenario, MPLS can positively bring about a change in terms of increasing the network resiliency by implementing link and node protection. This is achieved by using MPLS for faster re-routing of packets. New LSPs thus created help to keep the traffic QoS on target. This type of MPLS functionality is not limited to a network having a uniform transport layer. Since the mechanism is transport independent it can be applied to heterogeneous networks and network infrastructures.

2.7.1.2 Connection Oriented QoS Support

Applications on the internet are now demanding an increasingly reliable and sophisticated QoS support mechanism to fulfill their data communication needs. Some of the requirements include guarantee of fixed capacity for specific applications, controlled latency and jitter characteristics, ability to provide stringent service level agreements and the ability to measure and abide by the contract and capacity to configure varying degrees of service as demanded by the specific user and specific application.

2.7.1.3 Virtual Private Networks

From the beginning of the Internet era, service providers have looked for ways to provided suitably customized services to keep everyone of their users satisfied. A service that epitomizes customization for the end user is the Virtual Private Network concept.

Introduction of VPNs crops up a number of issues - Security being the least unimportant of these. Issues such as overlapping IP addresses and operational scalability further complicate the implementation of VPNs. MPLS uses the concept of stacked labels on each packet, to provide a transparent tunnel through which traffic of a given enterprize or group passes. This is a way of effectively segregating the traffic from other user traffic. This also allows it to obtain a class of service much different in quality from other traffic multiplexed on the same network.

2.7.1.4 Traffic Engineering

The MPLS Traffic Engineering approach encompasses the functionality provided by the overlay approaches like IP over ATM [18]. MPLS uses traffic trunks to aggregate data and route flows across the network. A traffic trunk is defined as an aggregation of traffic flows belonging to the same traffic class [20]. Traffic trunks are routed through the MPLS network using Label Switched Paths (LSP). LSPs specify the actual physical path the data takes to traverse the network from source to destination. Traffic Engineering in MPLS is a three stage process. the first stage involves assigning packets from different flows to appropriate traffic classes, second is attaching traffic classes to appropriate traffic trunks and the third step is mapping traffic trunks to LSPs that carry data across the network [16]. MPLS uses Reservation Protocol (RSVP) to signal the path and resources used by LSPs. RSVP is used to reserve resources on routers along the path of the LSP.

2.7.1.5 Multi protocol Support

Multi protocol support is the ability of MPLS enabled routers to inter operate with ordinary IP routers. MPLS can be deployed in ATM and Frame relay networks thus creating MPLS enabled ATM/FrameRelay switches. MPLS can be used in a purely homogenous network or in a mixture of different networks. It is useful to provide a unified control plane across these networks, thus greatly simplifying management by a single provider owning a number of different types of networks.

2.7.2 MPLS Operation

MPLS transfers data from source to destinations using Label Switched Paths. LSPs are established before the transfer of data begins and are defined by a series of MPLS labels at each node along the path from source to destination. MPLS labels are identifiers that are distributed using Label Distribution Protocol (LDP) [21]. Labels may also be piggy-backed on IP routing protocols like Open Shortest Path First (OSPF). MPLS enabled routers are classified into Label Switching Routers (LSR) and Label Edge Routers (LER). LSRs reside in the core of the MPLS network. They participate in label distribution and switching functions of MPLS. Labels are distributed and stored in the label information base (LIB) of LSRs. Every MPLS packet is prefixed with one or more MPLS labels that are used to make forwarding decisions for the packet at each LSRs along the path from source to destination. LERs are present at the edge of the MPLS network. LERs act as gateways between MPLS and non MPLS networks. When traffic enters the MPLS network, LERs determine the route that will be taken to traverse the MPLS network. LERs prefix incoming packets with MPLS labels and are responsible for removing the label when packets exit the MPLS network. The Forwarding Equivalence Class (FEC) groups traffic flows based on their quality of service requirements. Packets that belong to the same FEC are provided the same level of service by the network. Packets are assigned to FECs as they enter the network at the LERs.

An MPLS label identifies the path that a packet should take. Labels are distributed to MPLS enabled routers along the LSP from source to destination. Each MPLS packet carries a label that is examined by a receiving router on the LSP. The receiving router determines the next hop based on the value of the label. At every node the MPLS

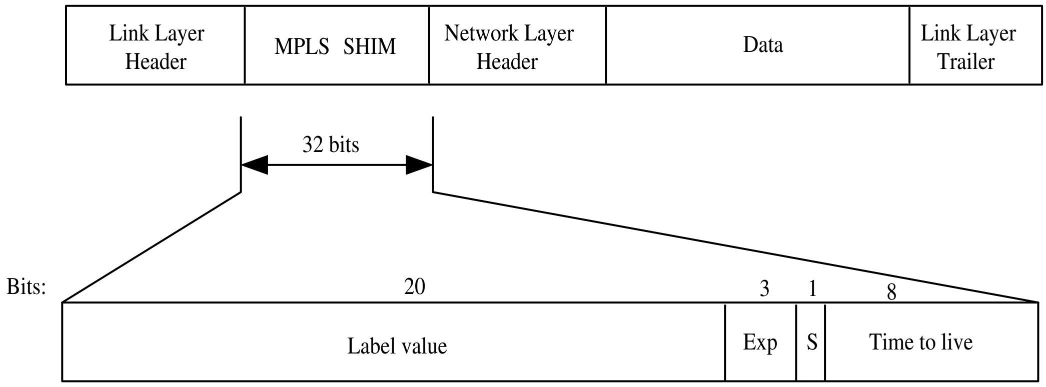


Figure 2.2: MPLS Label Format

incoming label is swapped with an outgoing label that will be used at the next router. Thus the label value has significance only between two routers. At the destination the incoming label is removed and the packet is handed to the network layer protocol for further processing. The format of the MPLS label can be seen in Figure 2.7.2.

2.8 Differentiated Services Architecture

The Differentiated Services Architecture (DiffServ) is designed to provide differential quality of service based on the performance requirements of different applications. DiffServ offers quality of service capability to IP networks using low over head and uncomplicated implementation functionality. There are a number of key characteristics of the DiffServ architecture. Traffic is divided into a number of classes based on different performance requirements. Available resources are allocated based on the requirements of each Class of Service (CoS). Differential treatment of classes is achieved using provisioning, priority processing and admission control. DiffServ contrasts with the end-to-end resource reservation typically offered by the Integrated Services (IntServ) architecture. DiffServ eliminates the need for a per flow allocation of resources. It also requires no signaling before packets are sent on the network. DiffServ basically aims to keep the forwarding functionality simple within the core of the network and move any complexity associated with traffic profiling and classification to the edge.

The following characteristics of DiffServ enable the advantages discussed earlier. 1)There is no extra space overhead associated with DiffServ enabled IP traffic. The IP TOS field is used to hold the DiffServ Code Point (DSCP). 2)Existing applications need not undergo any change to accommodate DiffServ. The SLA is established between the user and the service provider in advance of any traffic flow. This enables the applications to be shielded from any DiffServ related processing. 3)Aggregation of traffic is a part of DiffServ by the virtue of its functionality. All traffic with the same DSCP is treated in the same way and experiences the same per hop forwarding behavior at each router along the way. This may be disadvantageous when service providers need to fine tune their service offerings to customers with vastly different and wide ranging QoS requirements. 4)Routers in a DiffServ enabled network do not have to maintain state associated with any of the forwarding classes.

DiffServ (DS) enables a variety of services to be built from a small number of functionalities that are deployed in IP nodes across the network. These functionalities implemented in edge nodes as well as core network nodes. DiffServ functionalities include marking the type of service (ToS) byte in the IP header of each packet as it enters the network, using the ToS bits to decide how to forward the packet inside the network and enforcing traffic conditioning rules that include metering, marking, shaping and dropping. Traffic entering the DiffServ enabled network is classified and may be conditioned to enforce rules of the Traffic Conditioning Agreement (TCA). Traffic is assigned to different aggregate classes. Each aggregate classes is identified by a single DiffServ codepoint and is associated with a per hop behavior (PHB) specification. In the core of the network each packet is forwarded based on the per hop behavior associated with its DS codepoint.

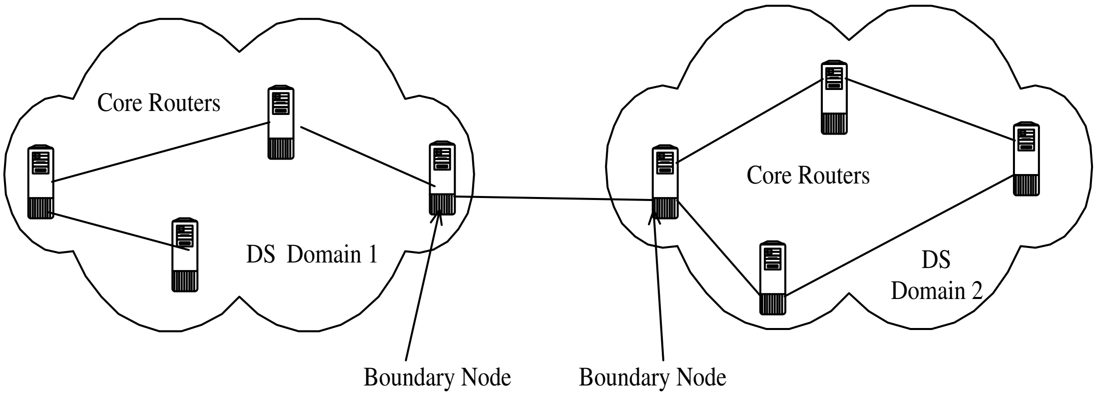


Figure 2.3: DiffServ Domains

A DiffServ domain is a contiguous set of DiffServ enabled nodes which operate on a common service provisioning policy [7]. The DS domain is bounded by a set of DiffServ nodes that classify and may condition incoming traffic into the available PHB classes. Nodes inside the DS domain select the forwarding behavior for packets based on their DS codepoint. A DS domain is generally administered by a single organization. Nodes at the edge of the DS domain are called boundary nodes and connect to other DS domains or DS incapable domains.

The advantage that DiffServ enjoys over other QoS architectures is that DiffServ concentrates most of its functionality at the edge of the network data enters the network. Once packets pass the boundary nodes and enter the DiffServ cloud,

2.9 IntServ

The services traditionally offered by established Internet protocols are unsuitable for real-time applications because variable queuing delays and packet loss because of congestion can occur at any time. Integrating real-time QoS mechanisms, which can control end-to-end packet delay, requires admission control and network management facilities that give network operators the ability to control the sharing of bandwidth and other QoS resources among different traffic classes. This form of network operation is known as controlled link sharing.

To meet this need a number of network operators have implemented integrated service architecture in routers and end system software. The main function of ISA is to manage congestion and provide QoS for both elastic and inelastic traffic by classifying the traffic into different traffic classes, selecting optimal route and scheduling packets on the queues for one or more output ports at each router, which are the main tasks of the router. To do this it uses the following functions: Admission control, Routing algorithm, Queuing discipline, Discard policy.

The main components of ISA are the reservation protocol, the admission control, Management agent and routing protocol, these are executed for each packet and are therefore highly optimized. The categories of service defined for ISA are guaranteed service, controlled load service and best effort service, the controlled load service tightly approximates the behavior visible to applications receiving best-effort service under unloaded conditions. The traffic is characterized using the token bucket scheme.

Various queuing discipline have been proposed like FIFO, Fair Queuing, Processor Sharing, Bit Round Fair Queuing, Generalized Processor Sharing and Weighted Fair Queuing, all these queuing techniques have there own tradeoffs between efficiency and fairness. The random early detection (RED) queuing technique is one of the most widely used as it helps in congestion avoidance, global synchronization avoidance, avoidance of bias against bursty traffic and provides a bound on average queue length.

The ISA and RSVP are intended to support QoS offering in the internets and private internets. But they are very complex to deploy and they don't scale well enough to handle large volume of traffic because of the control signaling required to coordinate integrated QoS offerings and the fact that the state information has to be maintained in the routers.

2.10 I-TCP

I-TCP is an indirect transport layer protocol that allows a mobile host to communicate with the fixed network via its Mobile Support Router [22]. The TCP connection between the mobile host and the fixed host is established by the mobile support router (MSR) on behalf of the mobile host. This model suggests that the TCP connection between the fixed host and the mobile host be split into two separate connections. One connection is between the fixed host and the MSR and the other connection is between the MSR and the mobile host. Splitting the TCP connection achieves a number of important objectives in the transport layer. Firstly, the flow and congestion control schemes on the two connections are separated. This separation is advantageous because of the vastly different communication characteristics between wireless and wired media. Secondly, a different transport protocol on the wireless link can be used to support notification of events to mobile applications. Thirdly, I-TCP allows the base station to manage communication on behalf of the mobile host. In most cases, mobile hosts are devices with limited memory and processing capacity. The mobile device can access many fixed network services like FTP or streaming media through the MSR even though it may not carry a full TCP/IP stack which is a requirement for such accesses. Lastly, I-TCP supports mobility as it accommodates faster reaction to mobility and wireless related events [22].

CHAPTER 3

ADAPTIVE NETWORK SERVICE

3.1 ANS Architecture

3.1.1 Functionality

ANS is a logical community of collaborating software agents that reside and execute on a subset of network nodes. ANS community comprises ANS agents and ANS nodes. ANS agents monitor, collect and exchange information about network conditions among agents in the ANS community and ANS nodes provide services to incoming user requests. The information collected by each ANS agent is communicated with other ANS agents in the community. Exchanging information between agents allows ANS to get an overall view of network conditions. ANS uses this information to route user data flows across the network. Before start of service, ANS decides the best available ANS node to service the user request. ANS considers traffic conditions in different parts of the network while deciding the best available ANS node to service the user request. ANS also monitors resources utilization in different ANS nodes. ANS uses this information to route incoming user requests around congestion and towards resource high areas of the network. ANS tries to engineer traffic flow in the network to increase user performance at the same time maximize network resource utilization. ANS nodes provide requested service to user flows and provision resources to user flows on a request by request bases. ANS nodes offer functionality to alter, buffer and route each packet that flows through the node.

ANS nodes register with the ANS agent in its vicinity. Each ANS agent monitors a number of nodes. When a user requests service from ANS, the contacted agent com-

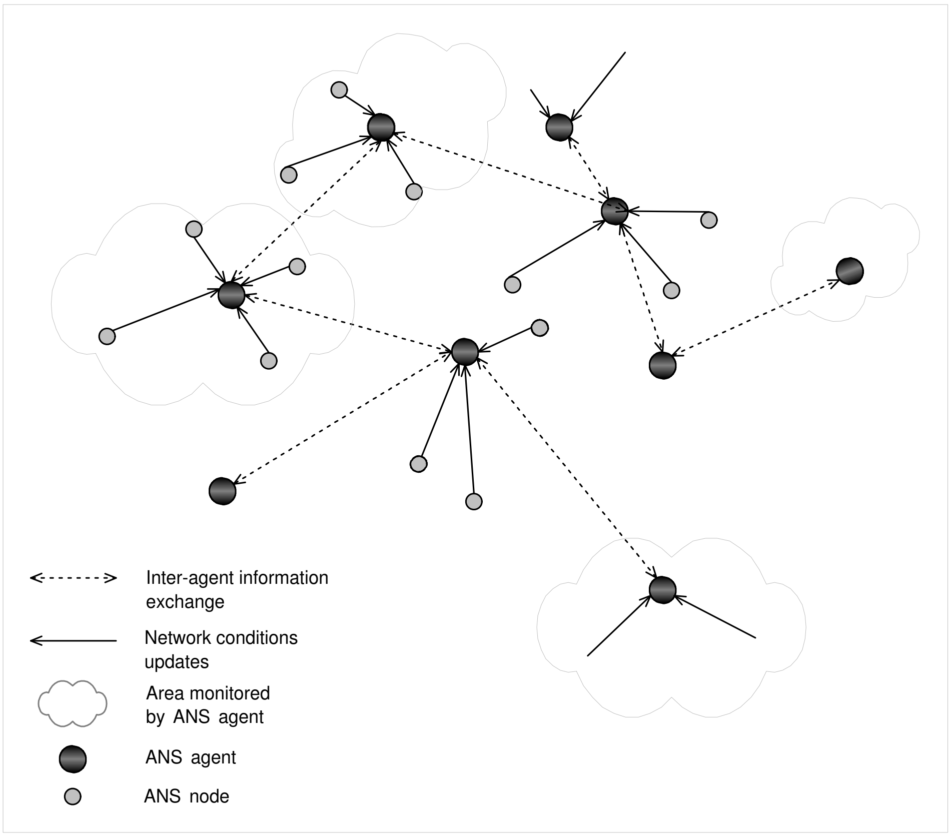

Inter-agent information exchange
Network conditions updates
Area monitored by ANS agent
ANS agent
ANS node

Figure 3.1: Adaptive Network Service

municates with the agent monitoring the selected ANS node. The agent monitoring the ANS node communicates this request to the selected ANS node to notify the ANS node of the new user request.

The information collected by an ANS agent is communicated with other ANS agents in the community. The information received from other ANS agents as well as information obtained from monitoring the local area is stored in a lookup table. Congestion indicators like mean TCP congestion window size allow ANS to know if the user flow encountered congestion in the network. By collecting such data from different locations, ANS can identify the least congested network area. ANS agents monitor congestion indicators while the data flow is in progress. Statistics like RTP receiver reports are collected from the receiver at the termination of RTP flows using ANS.

3.1.2 Location

ANS agents are distributed across the network and are strategically located to provide services and monitor network conditions. Agents are deployed on nodes on the network monitor network resources and current traffic conditions. ANS agents are present on a subset of network nodes. This is an advantage over other schemes like MPLS, DiffServ and IntServ wherein support from every router along the path from source to destination is a necessity. Moreover ANS does not require changes to existing router functionality.

3.1.3 Service Requests

A user wishing to use ANS must contact an ANS agent in its vicinity. The ANS agent consults the local lookup table and depending on the type of service requested and current conditions in the network, chooses the best ANS node to service the request. The decision is based on resource availability and network traffic conditions. The ANS agent communicates the address of the ANS node to the user. The user proceeds to use the ANS service by directly contacting the ANS node.

3.1.4 Resource Provisioning

Many network services require resources to be reserved within the network before the start of data transfer. ANS provides resource reservation for user data flows using the ANS service. Resources are reserved before the first byte of data is sent and are freed once the data flow is complete. ANS nodes communicate resource utilization data to the ANS agent in its vicinity.

3.1.5 Statistics

ANS agents collect data of previous flows and compile information about performance and network conditions. This information can be used to route new incoming user requests.

3.2 ANS Implementation

In our implementation we provide a TCP based service for transferring bulk data from a source to a destination. ANS needs inputs from the service in order to make a decision on the best ANS node to service a user request. The best ANS node is determined on the basis of 1) least congestion around the ANS node 2) maximum availability of resources in the ANS node.

An ANS agent obtains TCP congestion window size of current ANS connections from the nodes it is monitoring. The mean value of the congestion window is calculated and is periodically exchanged with neighboring agents. Each agent accepts updates from surrounding agents on the congestion window values in the surrounding nodes and is able to maintain a table of ANS node address against mean congestion window values. When a client connects to an ANS agent, the agent consults its local table to make a decision on the ANS node for the client to use for data transfer. The agent must take into account the availability of an ANS node prior to assigning a data buffer for data transfer. In the current implementation, an ANS node that is not busy and has the highest mean congestion window value is chosen. By considering these factors, ANS ensures that the service point is not a bottleneck. ANS locates an alternate service point if the first choice is busy. This allows the traffic to be spread across the network and service resources and ensures that traffic is not concentrated in a single part of the network. By choosing an ANS node that has the highest mean congestion window, ANS implicitly routes the new data flows away from congestion to a location that has experienced less congestion. This traffic aware behavior helps ANS ensure better quality service to clients while helping to balance traffic in the network. Having multiple service points ensures that failure of a single ANS node does not affect service on other ANS nodes.

In the course of data transfer, congestion may develop in the path of the transfer such that the service performance is affected. ANS continuously monitors the network and will suggest a new node to continue the service based on the current traffic conditions and resource allocation status. ANS informs the client of the new service point to continue service. This adaptive nature of ANS allows the service to react to congestion and avoid further exacerbating congestion. If the client moves to a new location, ANS chooses a new ANS node in the vicinity of the client to service the client request. ANS finds the best service point for the client in its current location and inform the client of the service point.

ANS bulk data transfer service transfers data from a source to a destination using two TCP connections. The two TCP connections, TCP-1 and TCP-2, are such that TCP-1 connects the source node and the ANS node and TCP-2 connects the ANS node and the destination node. Data is sent from the source to the ANS node on TCP-1 and from the ANS node to the destination on TCP-2. The ANS node receives incoming data on TCP-1 and stores the data in the ANS node buffer. ANS then sends data present in the buffer to the destination node on the TCP-2 connection. The mechanism is shown in Figure 3.2. In this figure the network of agents is shown by the filled circles inside the upper box. In reality the agents are implemented within the network of nodes shown in the lower box. They form a logical layer above this data flow layer. The agents monitor the nodes in the network and periodically send this information to other agents. This process is detailed in Figure 3.3

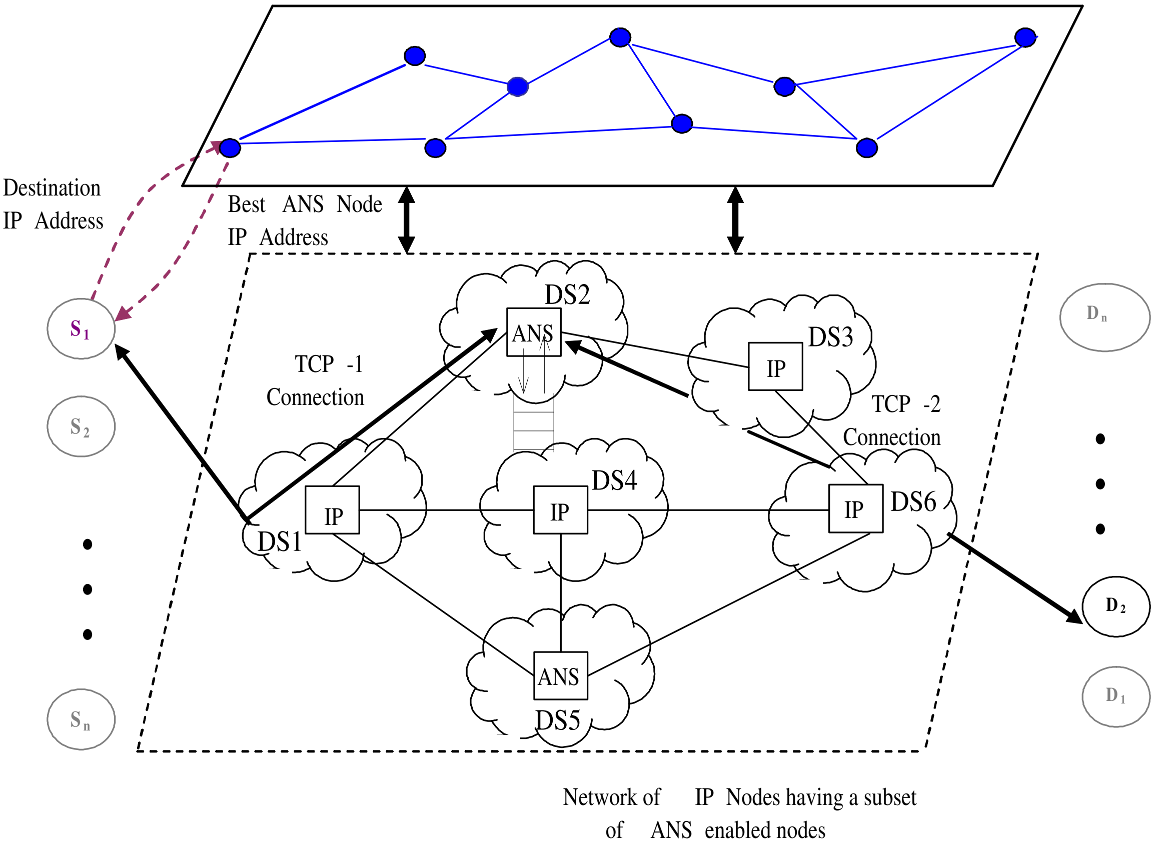
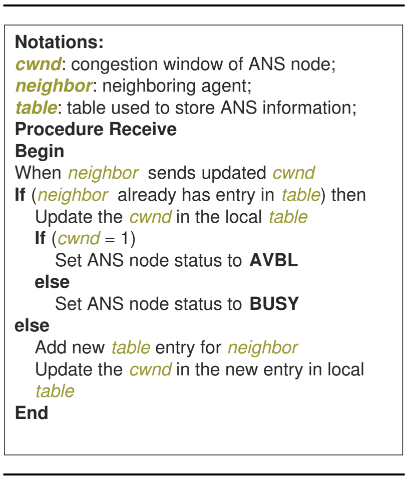


Figure 3.2: ANS Implementation Architecture



```
Notations:
cwnd: congestion window of ANS node;
neighbor: neighboring agent;
table: table used to store ANS information;
Procedure Receive
Begin
When neighbor sends updated cwnd
If (neighbor already has entry in table) then
    Update the cwnd in the local table
    If (cwnd = 1)
        Set ANS node status to AVBL
    else
        Set ANS node status to BUSY
else
    Add new table entry for neighbor
    Update the cwnd in the new entry in local
    table
End
```

Figure 3.3: Monitor and collect cwnd values from surrounding ANS nodes

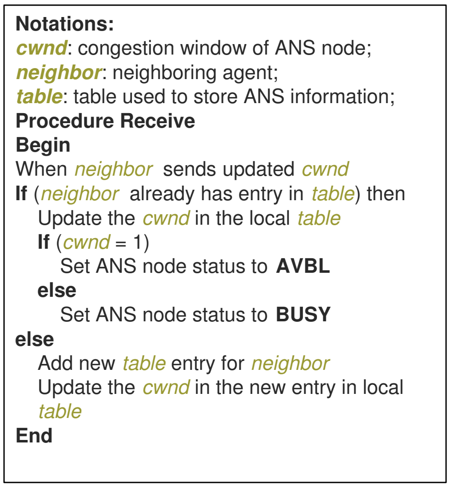
Each agent maintains a local ANS table that contains current information about that status of ANS nodes across the network. An example of this table is shown in Figure 3.1. Every agent accepts update messages from its neighboring agents and updates its lookup table.

Table 3.1: ANS lookup table

ANS Node Address	Congestion Window value	ANS Node Status	Client Address	Destination Address
129.10.29.21	1	AVBL	-	-
123.29.32.12	69	BUSY	129.10.12.12	192.4.13.2
192.4.16.2	35	AVBL	-	-

The processing steps that occur when an update message is received are described in Figure 3.4. The client S_1 requests service by contacting an ANS agent in its vicinity. The client sends the address of the destination node to which the data needs to be transferred along with the service request message. The ANS agent does a look up of its ANS table to locate a suitable ANS node. The agent searches for ANS node with status available (AVBL) and having the highest CWND value. The algorithm for the lookup process is described in Figure 3.5.

If the agent finds a suitable ANS node to service the client request, the agent contacts the ANS node with the destination address and requests the ANS node to setup the TCP-2 connection. TCP-2 connects the ANS node with the destination node. The agent awaits a reply from the ANS node which may respond with 'Service Ready' if all goes well, or 'Service Failed' if ANS node cannot connect to the destination or 'Service Busy' if the ANS node is busy with another connection. When the agent hears from the ANS node it replies to the client with the corresponding message. If ANS node sends 'Service Ready' the agent will send the ANS node address to the client along with the



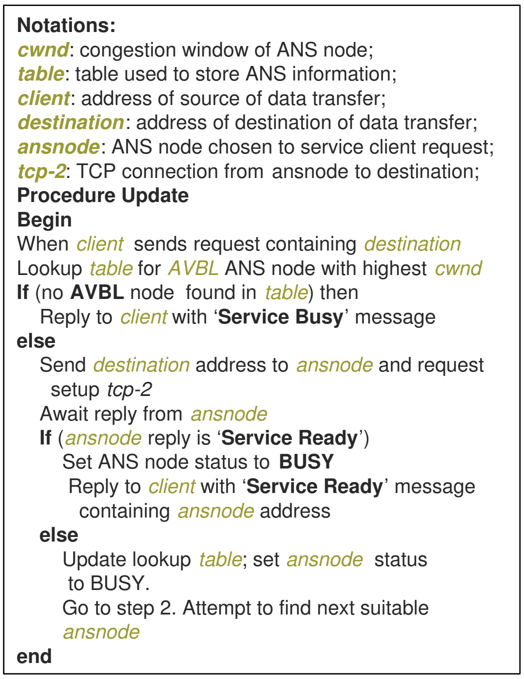
```
Notations:
cwnd: congestion window of ANS node;
neighbor: neighboring agent;
table: table used to store ANS information;
Procedure Receive
Begin
When neighbor sends updated cwnd
If (neighbor already has entry in table) then
    Update the cwnd in the local table
    If (cwnd = 1)
        Set ANS node status to AVBL
    else
        Set ANS node status to BUSY
else
    Add new table entry for neighbor
    Update the cwnd in the new entry in local
    table
End
```

Figure 3.4: Receive agent update

'Service Ready' message. The client can then connect to the ANS node and to establish TCP-1 connection and start data transfer.

3.3 Localizing Congestion

In a best effort service network such as the Internet it is difficult to guarantee a minimum level of service. Loss probabilities may be very high in some parts of the network and low in other parts. For a regular TCP connection flowing from source to destination, the sender's congestion window growth is subject to losses/congestion along the length of the link from sender to destination. No part of the network can be isolated out and pointed to as the cause of the congestion and re-routed around. Any rerouting is done by the IP dynamic routing mechanism. Thus in absence of or ignoring IP dynamic routing TCP source is at the mercy of the network from source to destination as far as the growth of its congestion window is concerned. ANS solves this problem by breaking the path into two distinct parts. Thus for any congestion or losses in the path of TCP-2, the source TCP entity is never affected. Of course if there are losses due to congestion



```
Notations:
cwnd: congestion window of ANS node;
table: table used to store ANS information;
client: address of source of data transfer;
destination: address of destination of data transfer;
ansnode: ANS node chosen to service client request;
tcp-2: TCP connection from ansnode to destination;
Procedure Update
Begin
When client sends request containing destination
Lookup table for AVBL ANS node with highest cwnd
If (no AVBL node found in table) then
    Reply to client with 'Service Busy' message
else
    Send destination address to ansnode and request
      setup tcp-2
    Await reply from ansnode
    If (ansnode reply is 'Service Ready')
        Set ANS node status to BUSY
         Reply to client with 'Service Ready' message
           containing ansnode address
    else
        Update lookup table; set ansnode status
         to BUSY.
        Go to step 2. Attempt to find next suitable
        ansnode
end
```

Figure 3.5: Receive client request

in the path of TCP-1, then the performance of TCP will be the same as the performance of ANS. Thus ANS localizes the effects of congestion to the ANS node, if the congestion is in the part of the network between ANS and destination.

3.4 ANS Location

Due to the ability of ANS explained above, we can make traffic aware decisions. If we detect that there is congestion in the path of TCP-2, then we can actively route the TCP-2 connection away from the current path. This can be done by choosing another ANS node in a different part of the network such that the path from that ANS node to the destination is not congested. Of course the new ANS node should be chosen such that there is also no congestion between the source and new ANS node.

3.5 ANS in Heterogenous Networks

The model does not place restrictions on the ANS location. Simulation experiments have shown that the ANS bulk transfer mechanism gives the best performance when the intermediate ANS node is located midway between the source and destination. That is, round trip time of connection 1 is equal to round trip time of connection 2.

3.6 Mathematical Model and Performance Analysis

In this section we present a mathematical model of the bulk transfer service. The model is used to predict the time taken to transfer a given amount of data from a source to a destination. The main inputs to the model are the probability of loss on TCP-1 and TCP-2 and the round trip times (RTT) RTT-1 and RTT-2 of the two connections respectively, refer Figure 3.2. With these inputs the model is able to predict the time to transfer the data from source to destination. The basis for our model is the well known analytical model described in [23]. An improvement to this model is the model presented in [24] that includes connection establishment, slow start, fast retransmit and delayed acknowledgements, factors that are not considered in the model proposed in [23].

3.6.1 TCP-1 Connection

The time T_1 to transfer D bytes from sender to the ANS node on TCP-1 is given by the sum of the expected values the TCP sender spends in slow start, timeouts, congestion avoidance and delayed acknowledgement. The expected value of t is denoted by $E[t]$ in the rest of this paper. T_1 is given by

$$T_1(D) = E[T_1 ss] + E[T_1 loss] + E[T_1 ca] + E[T_1 delack] \tag{3.1}$$

3.6.1.1 Slow Start

The time spent in slow start E[T1ss] is given in equation (3.2) below. This equation is derived in [24].

$$E[T_{1ss}] = \begin{cases} RTT_1.log_{\gamma 1}(\frac{E[d_1ss](\gamma 1-1)}{W_{1i}} + 1), \\ \qquad\qquad\qquad\qquad when \quad E[W_{1ss}] \leq W_{max} \\ \\ RTT_1.[log_{\gamma 1}(\frac{W_{1max}}{W_{1i}}) + 1 + \frac{1}{W_{1max}}(E[d_{1ss}] - \frac{\gamma 1 W_{1max} - W_{1i}}{\gamma 1 - 1})], \\ \qquad\qquad\qquad\qquad when \quad E[W_{1ss}] > W_{max} \end{cases} \tag{3.2}$$

$E[d_{1ss}]$ is the expected value of the number of segments of data that will be sent during the slow start period. $E[d_{1ss}]$ is a function of the probability of loss that a packet faces on the path from sender to the ANS node. $E[W_{1ss}]$ is the expected value of the congestion window at the sender at the end of the slow start period. $E[W_{1ss}]$ is a function of the expected number of segments $E[d_{1ss}]$ sent during the slow start period, the initial value W_{1i} of the congestion window at time $t = 0$, and the growth rate of the congestion window denoted by $\gamma 1$.

3.6.1.2 Timeouts

The time that the TCP sender spends in retransmission timer timeouts and TD acknowledgement followed by fast retransmit is given by the probability that a retransmission timeout or triple duplicate acknowledgement will occur multiplied by the time taken for the sender to recover from this state and enter congestion avoidance state. The time taken for the fast recovery period is taken to be a single round trip time (RTT), also assumed in [23]. The expected time for retransmission timeouts and fast recovery after the initial slow start phase is given by T_{1loss}

$$T_{1loss} = l_{ss}[Q(p_1, E[W_{1ss}]).E[{Z_1}^{T_{1o}}] + (1 - Q(p_1, E[W_{1ss}])).RTT_1] \qquad (3.3)$$

3.6.1.3 Congestion Avoidance

From [24] we approximate the time spent in sending the remaining data, if any, after slow start and loss recovery phases. The expected amount of data that will be remaining is denoted by $E[D_{ca}]$

$$E[D_{ca}] = D - E[D_{ss}] \qquad (3.4)$$

3.6.2 Delayed Acknowledgment

The most common reason for delayed acknowledgement is when the source TCP entity sets its *cwnd* to an initial value of 1. In this scenario the receiver waits for the second segment which never arrives. The receiver waits until its delayed ACK timer expires and the receiver then sends an ACK [24]. In most implementations the delayed ACK ranges between 0ms and 200ms.

3.6.3 Data Transfer Time

We use the steady-state model given by [23] for this purpose. This model gives the throughput R_1 as a function of loss rate p_1, round trip time RTT_1, average RTO value T_{1o} and the maximum window size advertised by the receiver W_{max}. R_1 is given by the below equation from [23].

$$R_1 = \begin{cases} \dfrac{\frac{1-p_1}{p_1}+\frac{W(p_1)}{2}+Q(p_1,W(p_1))}{RTT_1(\frac{b}{2}W(p_1)+1)+\frac{Q(p_1,W(p_1))G(p_1)T_{1o}}{1-p_1}} & if \quad W(p_1) < W_{max} \\ \\ \dfrac{\frac{1-p_1}{p_1}+\frac{W_{1max}}{2}+Q(p_1,W_{1max})}{RTT_1(\frac{b}{8}W_{1max}+\frac{1-p_1}{p_1 W_{1max}}+2)+\frac{Q(p_1,W(p_1))G(p_1)T_{1o}}{1-p_1}} & otherwise \end{cases} \tag{3.5}$$

where,

$$W(p_1) = \frac{2+b}{3b} + \sqrt{\frac{8(1-p_1)}{3bp_1} + \left(\frac{2+b}{3bp_1}\right)^2}$$

Considering the results from (3.4) and (3.5) we get the expected time to complete the data transfer in the congestion avoidance phase,

$$E[T_{ca}] = \frac{E[D_{ca}]}{R_1}$$

The expected value of the delayed acknowledgement is $100ms$ for BSD-derived stacks and $150ms$ for Windows [24].

3.6.4 TCP-2 Connection

Similarly we can depict the time taken by TCP-2 to transfer D bytes of data to the destination by the equation below

$$T_2(D) = E[T_2 ss] + E[T_2 loss] + v[T_2 ca] + E[T_{delack}] \tag{3.6}$$

where $E[T_{2ss}]$, $E[T_{2loss}]$ and $E[T_{2ca}]$ can be derived similar to the corresponding equations for TCP-1. In the case of TCP-2, the loss rate is p_2, round trip time is RTT_2, the growth rate of the congestion window during the slow start phase is , initial congestion window size is W_{2i}, maximum congestion window size advertised by receiver is W_{2max}.

However, we must remember that TCP-2 starts the data transfer with a lag of $RTT1/2$ seconds. This is the time taken by the first byte from the sender to reach the ANS node.

Therefore,

$$T_2(D) = \frac{RTT_1}{2} + E[T_{2ss}] + E[T_{2loss}] + E[T_{2ca}] + E[T_{delack}] \tag{3.7}$$

Once the first byte reaches the ANS node it begins to send data to the destination. While data is transferred from ANS to the destination, one of two situations can occur. There is always data in the ANS buffer to be sent on the TCP-2 connection or there is at least one interval of time where the ANS buffer is empty. In the first case, TCP-2 is never starved of data and the ANS behaves like an infinite source of data and hence the time to send D bytes of data from source to destination is the time taken to send the data from ANS to the destination, $T_2(D)$.

$$T(D) = \frac{RTT_1}{2} + E[T_{2ss}] + E[T_{2loss}] + E[T_{2ca}] + E[T_{delack}] \tag{3.8}$$

In the second case, the time to send D bytes of data from source to destination via the ANS node is sum of the time taken to send D bytes on TCP-1 from source to ANS and half the round trip time of the second connection TCP-2. The round trip time is considered because after the last byte of data arrives at the ANS buffer, it will take $RTT_2/2$ seconds to transfer the data from ANS to the destination. Hence, the time to transfer D bytes from source to destination via ANS is

$$T(D) = \frac{RTT_2}{2} + E[T_{1ss}] + E[T_{1loss}] + E[T_{1ca}] + E[T_{delack}] \tag{3.9}$$

The table below summarizes equations for the time to transfer D bytes of data from source to a destination.

Table 3.2: Summary of equations for bulk data transfer

Using TCP	Using ANS when TCP-1 lags TCP-2	Using ANS when TCP-2 lags TCP-1
$T(D) =$ $[T_{ss}] + E[T_{loss}] + E[T_{ca}]+$ $E[T_{1ca}] + E[T_{delack}]$	$T(D) =$ $\frac{RTT_2}{2} + E[T_{1ss}] + E[T_{1loss}]+$ $E[T_{2ca}] + E[T_{delack}]$	$T(D) =$ $\frac{RTT_1}{2} + E[T_{2ss}] + E[T_{2loss}]+$ $E[T_{delack}]$

CHAPTER 4

SIMULATION AND RESULTS

4.1 Simulation Setup

We employed Network Simulator 2 (NS2) for carrying out the experimental studies. In our simulations the network consists of about 8 nodes of which 5% of nodes were ANS enabled. The links connecting the nodes have delays ranging from 5-20 milliseconds and bandwidth of 5 Mbps each. Traffic generated by the source for the data transfer follows Poisson distribution. The hop count from source to destination is varied from 13-18 hops for each data transfer from source to destination. The simulations are carried out on different data sizes ranging from 1 Mb to 100 Mb.

4.2 Simulation Results for Bulk Data Transfer

The results presented in this sub section present simulations that were conducted on two networks of different sizes. The first network consists of 80 nodes and the second network comprises 25 nodes. The results for the larger network are presented in Table 4.1. Four data flows following Poisson distribution were observed with one flow using the ANS bulk transfer service and other flows using TCP connection between source and destination. Results for the smaller network are presented in Table 4.2. This table shows results for ANS scheme for data sizes of 1 Mb, 10 Mb and 100 Mb in the presence and absence of other traffic flows in the network.

Table 4.1: Table showing timing results of data transfer between source and destination with and without the ANS scheme. These results are for four Poisson flows with average hop count 13, 5 ms delay and 5 Mbps per link bandwidth. Results are shown for 1 Mb and 10 Mb data transfers in a network of 80 nodes.

Flow Identifier	Time to transfer data with No ANS Node (sec)	Time to transfer data using ANS NODE for Poisson 2 only (sec)	Time to transfer data with ANS for Poisson 1,2,3,4 (sec)	Number of packets transferred (1000 bytes each)
Poisson 1	15.42	15.42	**7.2**	1000 Total = 1MB
Poisson 2	20.80	**11.48**	**11.51**	
Poisson 3	17.70	17.72	**8.78**	
Poisson 4	20.38	20.36	**9.79**	
Poisson 1	144.76	144.69	**67.92**	10000 Total = 10 MB
Poisson 2	195.63	**107.60**	**107.78**	
Poisson 3	166.13	166.14	**82.82**	
Poisson 4	191.09	191.98	**91.87**	

4.3 Comparison of Model Estimates and Simulation Results

The graphs presented in this section compare the performance results of simulations with the predictions of the proposed mathematical model. The time taken to transfer 1 Mb data is measured using Network Simulator 2. In each simulation a TCP Reno sender is used to transfer 1 Mb of data over a 5 Mbps link to a TCP receiver. The figures also show the results of using TCP from source to destination not using ANS service. The simulations results and model estimates shown here were obtained with the following values of parameters. Initial window size $w_1 = 1$, growth rate of TCP congestion window, $\gamma_1 = 1.5$, number of segments to be transferred, $d = 1000$, maximum

Table 4.2: Table showing timing results for data transfer between source and destination with and without the ANS scheme. These results are for a data transfer with hop count 6, 10 ms delay and 10 Mbps per link bandwidth. Results are shown for transferring 1 Mb, 10 Mb and 100 Mb data in a network of 25 nodes. Two cases are simulated 1) no other traffic in network and 2) CBR and FTP flows in the network that affect the data transfer under study.

Traffic	Time to transfer data with NO ANS NODE (sec)	Time to transfer data using ANS (sec)	Number of packets transferred (1000 bytes each)
No Traffic	14.09	**7.26**	1000 Total = 1 MB
CBR & FTP traffic	101.33	**26.08**	
No Traffic	134.44	**67.44**	10000 Total = 10 MB
CBR & FTP traffic	1147.26	**325.07**	
No Traffic	1337.95	**669.19**	100000 Total = 100 MB
CBR & FTP traffic	11510.18	**3310.17**	

segment size, $MSS = 1000$, maximum window advertised by receiver $w_{max} = 19$, round trip time from source to destination (no ANS service), $RTT = 0.384$, round trip time from source to ANS node, $RTT_1 = 0.204$, round trip time from ANS node to destination, $RTT_2 = 0.140$, retransmission timeout, $RTO = 100ms$. p is defined in [23] as probability that a packet is lost given that it is the first packet in its round or the preceding packet in its round is not lost. The simulations were carried out using loss probability values of $p = 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}$ and 10^{-1}. The results of simulation are depicted by the scatter plot overlaid upon the model results. The graphs clearly show that the mathematical model agrees closely with the simulation results. Further, the model also predicts that

ANS scheme will give a definitive performance improvement as compared to use of direct transport connection between source and destination.

Figure 4.1 shows the performance of ANS and TCP for bulk data transfers. Simulation results are presented as scatter plots and the mathematical model estimates are depicted by continuous line plots. In this plot, p is the loss probability in the network region closer to the TCP source and for ANS scheme p is the loss probability in the TCP-1 connection. In our simulations a Bernoulli loss model was applied to the links that were shared by TCP and ANS TCP-1 connection. We see for a data size of 1 Mb at a loss probability of 0.001 for example, ANS has a throughput of $81Kbps$ as compared to $45Kbps$ of transport connection from source to destination without using ANS. The simulation and mathematical plots clearly show that ANS scheme outperforms a direct transport link from source to destination.

We attribute this performance increase to the fact that the round trip time of ANS TCP-1 connection is half round trip time of the TCP connection. Smaller RTT allows acknowledgements to reach the sender in half the time enabling the sender to increase the growth rate of its congestion window. As the congestion window opens the sender is able to send more data into the network without awaiting acknowledgements. In case of congestion, the sender receives triple duplicate acknowledgements quicker and is able to retransmit and recover from a packet loss earlier than a connection with longer RTT. Retransmission timeout is smaller for connections with smaller RTT and the sender is able to respond faster to packet losses.

The plot also shows that the mathematical model agrees closely with the simulation results. The model graphs are generated using mathematical software using parameter values detailed earlier. The graphs show the change of throughput of the data transfer with and without using ANS scheme with loss probability varying from 1 to 10^{-5}. Figure 4.2 shows the performance of ANS and TCP. Simulation results are presented as scatter

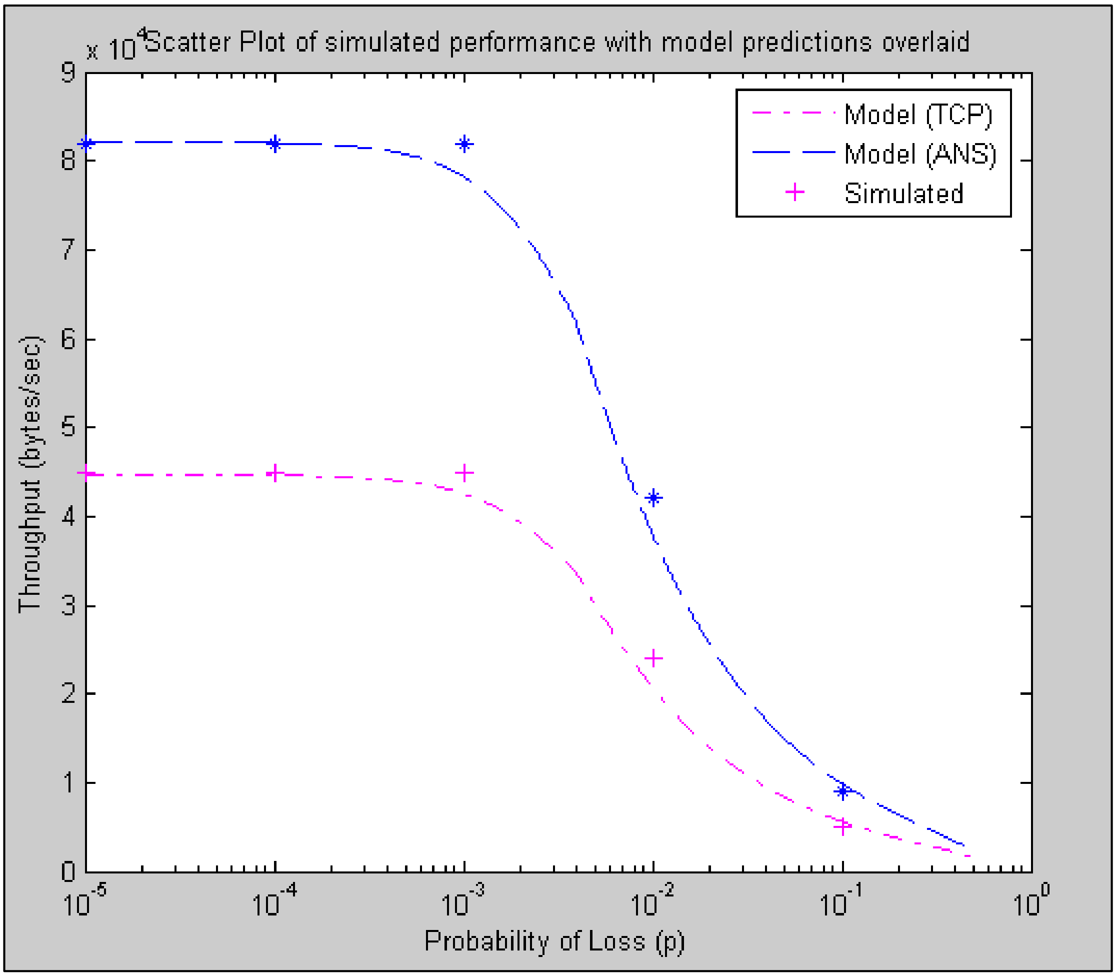


Figure 4.1: **Comparison of ANS and TCP: Equal loss probability in TCP and TCP-1.**

1 MB transfers with $w_1 = 1, \gamma_1 = 1.5, d = 1000, MSS = 1000, w_{max} = 19$, round trip time from source to destination (no ANS service), $RTT = 0.384$, round trip time from source to ANS node, $RTT_1 = 0.204$, round trip time from ANS node to destination, $RTT_2 = 0.140$, $RTO = 100ms$. Probability of loss p varies from 1 to 10^{-5}. In this figure both ANS and TCP face equal congestion at the same point in the network. The losses occur in TCP-1 connection of ANS. For TCP the losses occur in the network region nearer to the source. The plot shows ANS and TCP simulation performance as scatter plots and model estimates by continuous plots.

plots and the mathematical model estimates are depicted by continuous line plots. In this plot, the loss probability p, depicts probability of packet loss in the in the TCP-2 connection. The TCP connection is not affected by packet losses and has a constant throughput of $45Kbps$. A Bernoulli loss model was applied to the TCP-2 connection. We see for a data size of $1Mb$ at a loss probability of 0.1 ANS has a throughput of $14Kbps$ as compared to $45Kbps$ of transport connection from source to destination without using ANS. Thus in the case when there is a high probability of packet loss in the path of ANS connection, use of a different ANS node is recommended or else use of direct TCP connection is recommended if ANS is not available. We see that as the probability of loss decreases, ANS throughput quickly overtakes direct TCP and at loss probabilities of < 0.01 ANS is a better choice even if TCP faces 0 loss probability. A good example is when $p = 0.001$, ANS has throughput of $81Kbps$ as compared to $45Kbps$ without using ANS on a 0 loss probability connection.

Figure 4.3 shows the same results as described above when the loss probability p is applied in the network region towards the destination of the TCP connection and on TCP-2 connection of ANS. This plot is similar to Figure 4.1 showing that ANS gives better performance. The graph confirms that the mathematical model agrees closely with simulation results.

4.4 Simulation Results for RTP Data Transfer

This section presents the results of RTP simulations conducted in a network consisting of 80 nodes. Table 4.3 shows timing results for RTP data transfer between source and destination with and without the ANS scheme. These results are for RTP data transfer with hop count 16, 10 ms delay and 5 Mbps per link bandwidth.

The table shows the transfer time for a 1 Mb Poisson data flow transferred using RTP. Two cases are simulated 1) no other traffic in network and 2) Traffic in the form of

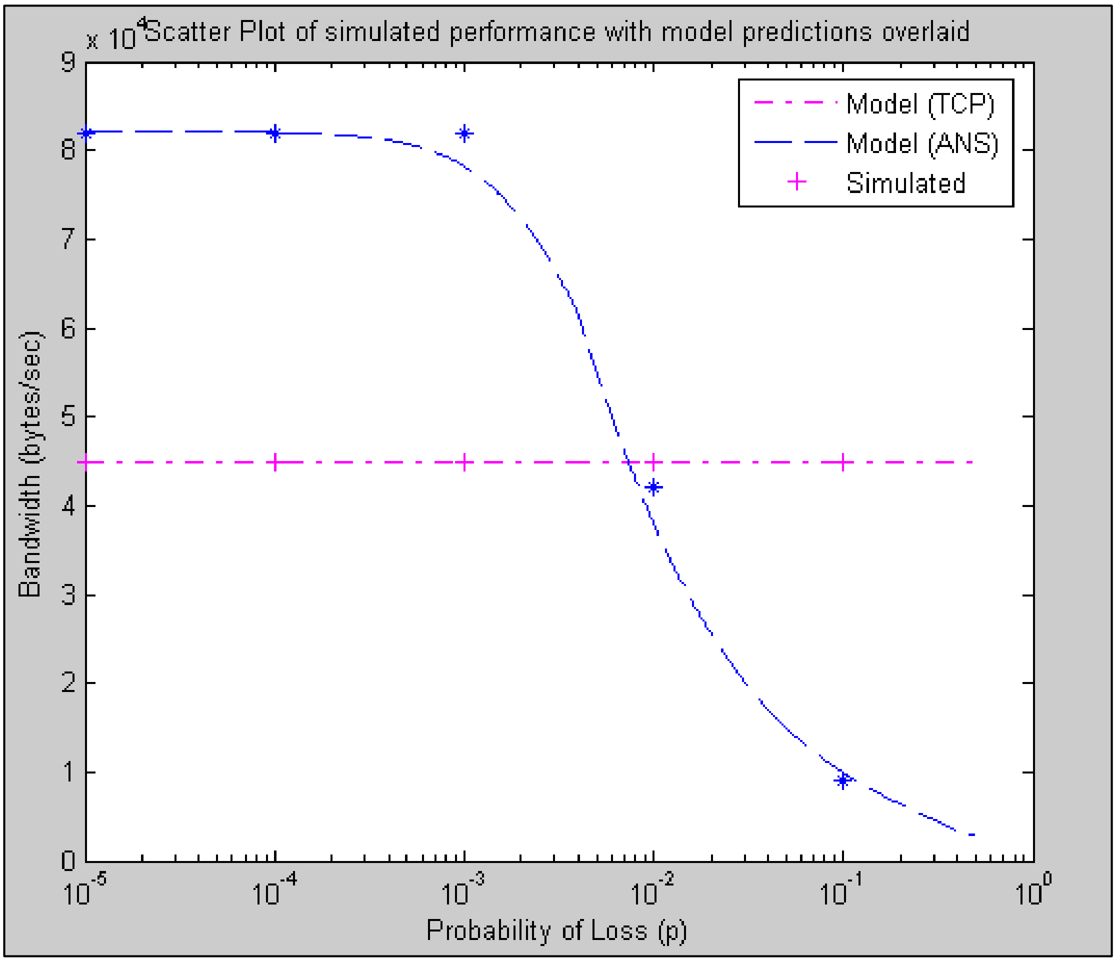


Figure 4.2: **Comparison of ANS and TCP: TCP has 0 loss probability, p varies in TCP-1.**

1 MB transfers with $w_1 = 1, \gamma_1 = 1.5, d = 1000, MSS = 1000, w_{max} = 19$, round trip time from source to destination (no ANS service), $RTT = 0.384$, round trip time from source to ANS node, $RTT_1 = 0.204$, round trip time from ANS node to destination, $RTT_2 = 0.140$, $RTO = 100ms$. Probability of loss p varies from 1 to 10^{-5}. In this figure TCP flow does not encounter loss. ANS flow encounters loss with probability p. The plot shows ANS and TCP simulation performance as scatter plots and model estimates by continuous plots.

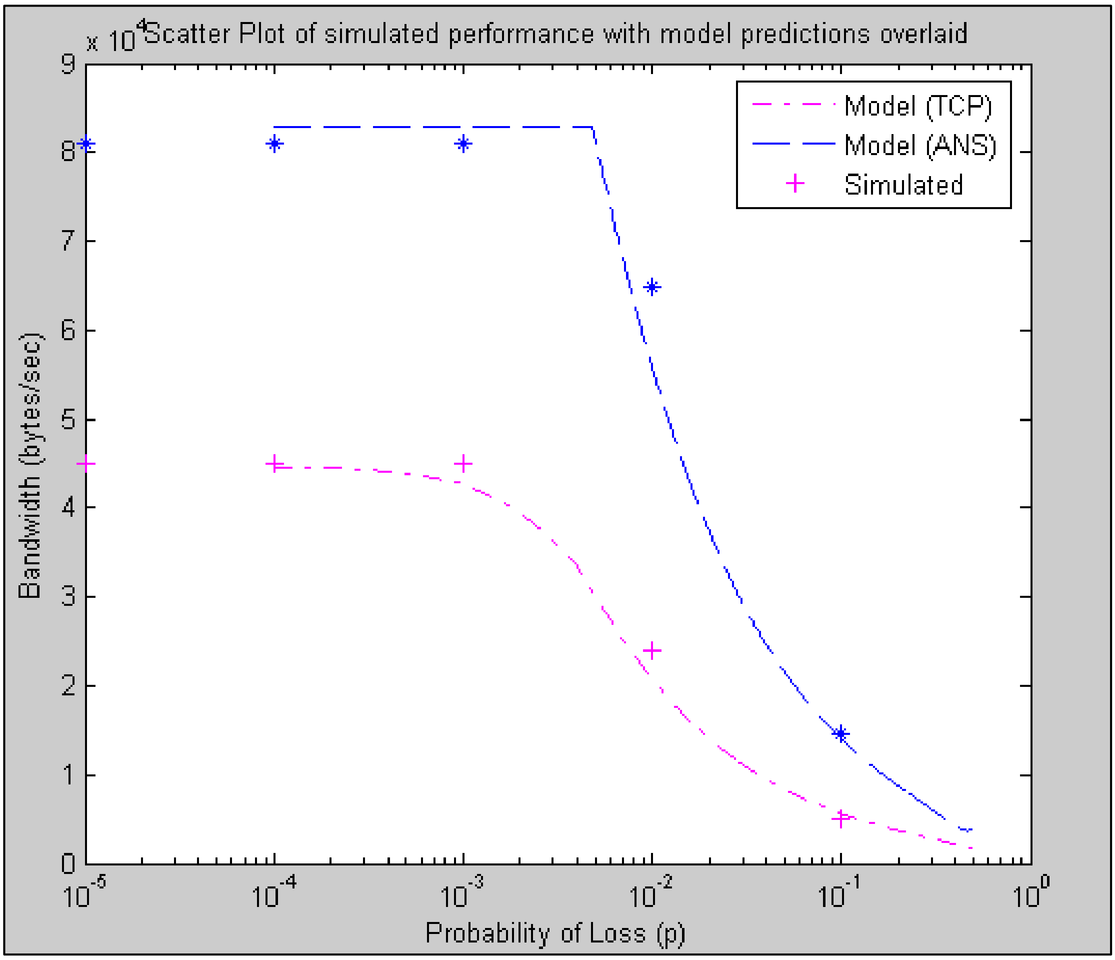


Figure 4.3: **Comparison of ANS and TCP: Equal loss probability in TCP and TCP-2.**

1 MB transfers with $w_1 = 1, \gamma_1 = 1.5, d = 1000, MSS = 1000, w_{max} = 19$, round trip time from source to destination (no ANS service), $RTT = 0.384$, round trip time from source to ANS node, $RTT_1 = 0.204$, round trip time from ANS node to destination, $RTT_2 = 0.140$, $RTO = 100ms$. Probability of loss p varies from 1 to 10^{-5}. In this figure both ANS and TCP face equal congestion at the same point in the network. The losses occur in TCP-2 connection of ANS. For TCP the losses occur in the network region nearer to the destination. The plot shows ANS and TCP simulation performance as scatter plots and model estimates by continuous plots.

Table 4.3: Comparison of data transfer time for RTP flow using ANS scheme and without using ANS scheme

		No ANS Node	Using ANS Node
No Traffic	Jitter	0.21×10^{-2}	0.07×10^{-2}
	End-to-End delay (sec)	16.6×10^{-2}	6.1×10^{-2}
	Throughput (bits/sec)	9.9×10^{5}	9.9×10^{5}
10 Poisson flows	Jitter	0.28×10^{-2}	0.07×10^{-2}
	End-to-End delay (sec)	18.2×10^{-2}	6.2×10^{-2}
	Throughput (bits/sec)	8.8×10^{5}	10.5×10^{5}

ten Poisson flows in the network, that share network links used by the RTP data transfer under study. In this simulation, the ANS node is located in a region of the network that is away from the path taken by RTP packets when ANS is not used. The path taken by RTP over User Datagram Protocol (UDP)packets when ANS is not used is determined by the IP routing protocol being used in the network. Table 4.3 shows that when the ANS node is used the end-to-end delay and jitter values are reduced almost three times. When traffic is present in the path of the RTP flow, we see an increase in end-to-end delay and jitter and a decrease in throughput when ANS is not used. When the RTP flow makes use of ANS scheme, the flow is routed away from traffic. This results in significantly lesser values of end-to-end delay and jitter when compared with the same value when ANS is not used. We also observe an improvement in throughput when ANS is used in the presence of traffic.

Table 4.4 shows timing results for RTP data transfer between source and destination with and without the ANS scheme. These results are for RTP data transfer with hop

Table 4.4: Comparison of data transfer time for RTP flow using ANS scheme and without using ANS scheme

		No ANS Node	Using ANS Node
No Traffic	Jitter	0.18×10^{-2}	0.08×10^{-2}
	End-to-End delay (sec)	13.5×10^{-2}	7.23×10^{-2}
	Throughput (bits/sec)	10.03×10^{5}	10.03×10^{5}
10 Poisson flows	Jitter	1.56×10^{-2}	1.1×10^{-2}
	End-to-End delay (sec)	14×10^{-2}	8.6×10^{-2}
	Throughput (bits/sec)	1.13×10^{5}	1.13×10^{5}

count 12, 10 ms delay and 5 Mbps per link bandwidth. The table shows the transfer time for a 1 Mb Poisson data flow transferred using RTP. Two cases are simulated 1) no other traffic in network and 2) Traffic in the form of ten Poisson flows in the network, that share network links used by the RTP data transfer under study. In this simulation, the ANS node is located on the path taken by RTP packets when ANS is not used. This path is determined by the IP routing protocol being used in the network. Table 4.4 shows that when the ANS node is used the end-to-end delay and jitter values are reduced by almost half. When the ANS node is not present and traffic is present in the path of the RTP flow, we see an increase in end-to-end delay and jitter and a decrease in throughput. When the ANS scheme is used we observe that the end-to-end delay and jitter are decreased even in the presence of traffic.

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

5.1 Discussion

This thesis presents the ANS scheme for providing network services in a way that ensures performance, efficiency and quality of service. The ANS architecture is presented and the use of ANS for bulk transfer of data is described in this thesis. ANS is a logical community of collaborating software agents that reside and execute on a subset of network nodes. ANS community comprises ANS nodes and ANS agents that monitor, collect and exchange network information among agents in the ANS community. ANS nodes provide services to incoming user requests. Simulations show that ANS significantly improves performance when it is used to transfer bulk data across the network. The performance of ANS bulk data transfer service is modeled using a mathematical model which is based on a well known model of TCP latency. The mathematical model is used to estimate data transfer throughput. Simulation results and model estimates show that ANS is able to achieve significantly better data transfer throughput compared to connections that do not use ANS scheme. ANS routes incoming user requests away from network areas experiencing congestion and towards areas where more resources are available. User data flows experience less congestion and are exposed to better resources, thus increasing the quality of service perceived by the end user. Simulations also show that ANS improves throughput and decreases end-to-end delay and jitter for RTP data flows.

5.2 Future Work

Future work in this direction includes introducing new services into the ANS framework and improving ANS agent intelligence to deliver these services in a user friendly way. Adaptive services like ANS make well-informed decisions to provide network services to promote network efficiency, quality of service and good performance. We intend to enhance ANS for applications in pervasive computing environments.

REFERENCES

[1] V. Cerf, “Beyond the post-pc internet,” *Commun. ACM*, vol. 44, no. 9, pp. 34–37, 2001.

[2] V. Firoiu, J. L. Boudec, D. Towsley, and Z. Zhang, “Theories and models for internet quality of service,” *Proceedings of the IEEE*, vol. 90, no. 9, pp. 1565–1591, 2002.

[3] D. Wetherall, U. Legedza, and J. Guttag, “Introducing new internet services: Why and how,” 1998.

[4] D. Awduche, “Mpls and traffic engineering in ip networks,” *IEEE Communications Magazine*, vol. 37, no. 12, pp. 42–47, Dec. 1999.

[5] E. Rosen, A. Viswanathan, and R. Callon, “Multiprotocol label switching architecture,” RFC 3031, Jan. 2001.

[6] R. Braden, D. Clark, and S. Shenker, “Integrated services in the internet architecture: an overview,” RFC 1633, June 1994.

[7] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, “An architecture for differentiated services,” RFC 2475, Dec. 1998.

[8] L. Mathy, C. Edwards, and D. Hutchison, “The internet: A global telecommunications solution?” *IEEE Network*, vol. 14, no. 4, pp. 46–57, 2000.

[9] R. Want, T. Pering, G. Danneels, M. Kumar, M. Sundar, and J. Light, “The personal server: Changing the way we think about ubiquitous computing,” in *Proceedings of the Ubiquitous Computing Conference*, Gteborg, Sweden, 2002.

[10] Apple, “ipod: Apple’s mp3 player.” [Online]. Available: www.apple.com/ipod

[11] A. Vogel, B. Kerherve, G. Bochmann, and J. Gecsei, "Distributed multimedia applications and quality of service: a survey," in *Proceedings of the Conference of the Centre for Advanced Studies on Collaborative research*, Toronto, Canada, 1994.

[12] M. Korpi and V. Kumar, "Supplementary services in the h.323 ip telephony network," *IEEE Communications*, vol. 37, no. 7, pp. 118–125, 2001.

[13] MicrosoftTechNet, "Qos technical reference." [Online]. Available: http://technet.microsoft.com/default.aspx

[14] R. Jain, "Myths about congestion management in high speed networks," *Internetworking: Research and Experience*, vol. 3, pp. 101–113, 1992. [Online]. Available: citeseer.ist.psu.edu/jain92myths.html

[15] W. Stallings, *High-Speed Networks and Internets: Performance and Quality of Service.* 2/E Prentice Hall, 2002.

[16] J. Malcolm, J. Agogbua, M. O'Dell, J. McManus, and UUNET (MCI Worldcom), "Requirements for traffic engineering over MPLS," RFC 2702, Sept. 1999.

[17] C. Yang and A. V. S. Reddy, "A taxonomy for congestion control algorithms in packet switching networks," *IEEE Networking*, vol. 9, no. 5, pp. 33–44, 1995.

[18] M. Laubach, "Classical ip and arp over atm," RFC 1577, Jan. 1994.

[19] S. Ratnasamy, M. Handley, R. Karp, and S. Shenker, "Topologicallyaware overlay construction and server selection," 2002. [Online]. Available: citeseer.ist.psu.edu/ratnasamy02topologicallyaware.html

[20] T. Li and Y. Rekhter, "A provider architecture for differentiated services and traffic engineering (paste)," RFC 2430, Oct. 1998.

[21] L. Andersson, P. Doolan, N. Feldman, A. Fredette, and B. Thomas, "Ldp specification," RFC 3036, Jan. 2001.

[22] A. Bakre and B. R. Badrinath, "I-TCP: Indirect TCP for mobile hosts," *15th International Conference on Distributed Computing Systems*, 1995.

[23] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, “Modelling TCP reno performance: A simple model and its empirical validation,” *ACM Computer Communications Review*, vol. 28, no. 4, 1998.

[24] S. S. Neal Cardwell and T. Anderson, “Modelling tcp latency,” in *Proc. IEEE INFOCOM*, Tel-Aviv, Israel, Mar. 2000.

BIOGRAPHICAL STATEMENT

Nayantara Mallesh completed her Masters in Computer Applications from Bangalore University, Bangalore in 1998. She worked in the Telecom and Networking division of Cognizant Technology Solutions, Chennai, India and later with Metro-Optix, headquartered at Allen, Texas, USA. After 5 years of work in the industry, she enrolled at the University of Texas at Arlington to pursue her MS in Computer Science. Her work at UTA involves adaptive agent communities for providing services in dynamic networks. Her research interests include pervasive computing, telecommunication and networking protocols and wireless networks. She plans to pursue a PhD in Computer Science.