This is to certify that the
dissertation entitled

SUPPORTING REMOTE SENSING AND CONTROL OVER
IP NETWORKS

presented by

ZHIWEI CEN

has been accepted towards fulfillment
of the requirements for the

Ph.D. degree in Computer Science and Engineering

Major Professor's Signature

January 10, 2007

Date

SUPPORTING REMOTE SENSING AND CONTROL OVER IP NETWORKS

By

Zhiwei Cen

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Department of Computer Science and Engineering

2007

ABSTRACT

SUPPORTING REMOTE SENSING AND CONTROL OVER IP NETWORKS

By

Zhiwei Cen

In IP network based remote sensing and control systems, the operator controls a slave manipulator with sensing capabilities located at a remote site. Control commands and sensory feedback are transmitted through computer networks. The control and sensing data involved in such systems differentiate themselves from other media types in that they require both reliable and smooth delivery. Reliable delivery requires that the transport service have TCP style semantics. By being smooth, the transport service should be able to deliver the control and sensing data with both the average latency and the standard deviation of the latency bounded and reduced. Traditional transport services have great difficulty meeting the latency requirements of delivering sensing and control data over the communication networks.

Remote sensing and control systems could be deployed in distant geographical locations, or places where network infrastructures are not available. In the former case, the Internet is an ideal communication channel due to its availability. In infrastructureless environments, Mobile Ad-hoc Networks (MANETs) can be a good choice. However, the transport protocols in both the Internet and MANETs have difficulty meeting the needs of this type of applications. We proposed a Supermedia TRansport over Overlay Networks (STRON) framework to improve the QoS of teleoperation systems over wide area networks. A QoS management frame work is also proposed to support STRON based teleoperation systems. A Transport service for Remote Sensing and Control (TRSC) over Wireless Networks is designed for MANET based applications. The proposed approaches were verified using experiments, net-

work simulators and theoretical models.

The popularity of third-generation wireless communication, handheld devices and local area wireless technologies makes pervasive computing environment a good incubator for remote sensing and control applications. Supporting remote sensing and control applications in pervasive computing environments poses new challenges to the transport protocols. In this part of the research we proposed a collaborative approach to support remote sensing and control in heterogeneous environments. We explore the spatial diversity provided by dual-mode terminals that are equipped with both WLAN and cellular interfaces. Multiple terminals serve as proxies to provide better QoS for the targeted application. We built a dynamic mechanism to determine the appropriate number of proxies to use. The mechanism utilizes the MAC layer retransmission rate as an indicator of the contention level in the MANET.

We also examine the organization structures of the networks that support remote sensing applications. In order to adapt to the changing environments and provide resilient QoS, it is often desirable that the sensor network reorganize itself. We explore the opportunities and challenges of deploying low cost hopping sensors and utilizing their mobility to ensure coverage and maintain energy efficiency within a sensing field. Hopping sensors are able to maintain mobility in harsh terrain but may lack the movement accuracy of those sensors that are powered by wheels. We focus on the problem of transporting a number of hopping sensors from multiple sources to a destination. Probabilistic methods are used to contain the movement inaccuracies along the hopping course. We also consider the impact of wind under an aerodynamic setting. Two transport schemes are designed to minimize the number of hops needed while considering other constraints, such as sustaining the capability of relocating sensors within the whole network. The performance of the two schemes are evaluated and compared through simulations.

ACKNOWLEDGMENTS

Foremost, I am deeply grateful to my adviser, Professor Matt Mutka. I want to express my sincere gratitude to Professor Mutka for his years of encouragement, his scientific influence on me, his infinite patience, his insights in our numerous discussions, his financial support and his careful review of paper manuscripts. Without these generous assistance, this dissertation could not have come into light. To be one of his students is my great honor.

I also wish to express my gratitude to my dissertation committee, Professor Abdol-Hossein Esfahanian, Professor Charles Steinfield, and Professor Li Xiao for sparing their precious time to serve on my committee and giving valuable comments and suggestions. Special thanks also go to Professor Ning Xi for his guidance in the supermedia project.

I am thankful to my colleagues in ELANS lab and CSE department. Their friendship makes my years at MSU unforgettable. Special thanks go to Danyu Zhu for helping me understand the cellular networks area and Alexander Chobanyan for many discussions regarding network bandwidth measurements. I am also thankful to my colleagues in RAlab, especially Amit Goradia for his help in robotics and Yang Liu, Clayton Haffner for helping me conducting the supermedia experiments.

I am grateful to my parents, my wife Lijuan Wang and our son Evan for being there for me all the time.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

- APSM: Active Path Selection Module
- DQFM: Decoding and QoS Feedback Module
- ETM: Encoding and Transport Module
- MANET: Mobile Ad Hoc Network
- MRM: Multi-path Routing Module
- NMM: Network Measurement Module
- ONA: Overlay Network
- OPSM: Optimal Path Selection Module
- QMM: QoS Management Module
- QoS: Quality of Service
- RONA: Receiver-side Overlay Network Agent
- RTT: Round Trip Time
- SONA: Sender-side Overlay Network Agent
- STRON: Supermedia TRonspart over Overlay Networks
- TDI: Task Dexterity Index
- WWAN: Wireless Wide Area Network
- WLAN: Wireless Local Area Network

CHAPTER 1

Introduction

1.1 Motivation

The development of the Internet sees its impact on many fields. The mingling of the Internet with these fields is changing the way industries operate and people live their lives. The Internet connects not only traditional computers, but also mobile phones, personal digital assistants, as well as sensors of various sizes and capabilities, and robots that can perform different manipulations. The data transferred over the Internet are not only text, image, audio and video, but also sensing feedback and control commands. In remote sensing and control applications, automatic manipulating and sensing devices are deployed at a remote site. The user or operator control the systems or accept feedback through network connections.

A typical application that uses remote sensing and control is a bilateral teleoperation system [178, 133, 45, 67, 53, 56, 54, 55, 57, 58, 191, 16, 128, 190]. In such a system, a human operator controls a mobile manipulator (robot) to perform certain tasks and receives force and other feedback from the remote environment. In areas of tele-medicine, a physician can feel the patient at a remote location for abnormalities, or in tele-commerce, a customer can touch and manipulate an item before its purchase. In the scope of the military, the ability of a soldier to control the fighting devices remotely and sense the remote environment is one of the important objectives of Future Combat Systems (FCS). In other areas, such as intelligent surveillance or navigation, hazardous materials handling, remote control and sensing through wireless network is also a critical component of the system. Another example of remote sensing and control systems is Mobile Surveillance Networks (MSNs) [72, 73]. In a mobile surveillance network, mobile computer nodes equipped with multiple functional sensors may recognize, characterize and track certain objects. Information of the target objects in the form of image, binary data and even audio and video need to be exchanged. The mobile nodes also need to coordinate the tracking movements in order for the target to be covered by at least one of the nodes. The information exchange is also time sensitive depending on the movement features of the target.

1.2 Challenges and Difficulties

The biggest challenge of Internet based remote sensing and control system is making the unpredictability caused by the Internet to be transparent to upper layer of the system. Random time delay, packet loss, network buffering effects and disconnects in the Internet's best-effort service model present major difficulties. Among all the uncertainties, time delay is one of the biggest obstacles to build a stable remote control system. High latency may cause the robot to stall in the middle of a task or the operator to lose control of the remote robot. The uncertain variations of the network latency makes the control system unstable, which in turn affects of the performance of the remote sensing and control application. This problem may be approached through two directions:

1. Try to modify the control mechanism of the system to accommodate the unpredictable nature of the Internet. The traditional control system assumes a dedicated and reliable communication channel between the operator and the robot. In order for the system to work over the Internet, some of its assump-

tions must be changed.

2. Try to improve the quality of service of the communication channel to make it close to the dedicated communication channels used by the traditional system. One of the possible ways to help build a stable control system is to reduce the end-to-end latency of the communication channels.

Event based remote control systems [67, 57, 58, 191] were presented to address the problem from the first direction. The performance difficulty caused by the Internet based remote control system is a result of using time as a reference for different system entities. In the event-based control approach, a non-time based reference is used. An event reference is a monotonically increasing parameter to synchronize the operator and the robot. Through the event reference, low level sensing and control modules are integrated with high level task scheduling and action planning. The system is able to cope with the uncertainties caused by the Internet and provide event transparency to the system user. Fung, et al. introduced a Task Dexterity Index (TDI [67]) of the robotic task associated with each supermedia stream. The dexterity of a task represents the complexity of the controlled movement that the robot makes and the degree of attention the operator should give to tasks of the robot. A Task Dexterity Index (TDI) is generated for each data stream using a fuzzy logic system to describe the bandwidth requirement of the robotic task. Later, this TDI may be used in the bandwidth allocation algorithm to rank the data streams that need to be transmitted.

Extensive research efforts have been dedicated to provide QoS aware transport service over the Internet [169, 49, 14, 2, 160, 144, 181]. These efforts are closely related to our efforts to the second direction of approaching the remote control problem. Some of the most notable work, including Integrated Service (IntServ [144, 186]) and Differentiated Services (DiffServ [14]), aims to change the best-effort service model to a QoS aware infrastructure. However, IntServ and DiffServ are not deployed over major networks due to a variety of reasons. Many research efforts aim to improve QoS levels for multimedia applications over the current best effort Internet [79, 164]. However these transmission mechanisms do not solve issues in supermedia transmission due to the following reasons:

1. A remote sensing and control system involves several kinds of media types, such as video, audio, control commands, haptic information, and code uploads. Different types of supermedia stream have different QoS requirements. Video and audio streams may be unreliable but need to have an adaptive real time transmission service. Control commands and haptic information need a timely and reliable transmission service. Code uploads need reliable transmission service but are not very time sensitive.

2. Supermedia streams have dynamic QoS requirements associated with the task execution process. The priority of a supermedia stream will change dynamically when the remote sensing and control system switches from one task to another. For example, video streams may have higher priorities when the robot is approaching an object or trying to circumvent obstacles, while haptic feedback may become the highest priority when the robot starts to manipulate the object.

1.3 Approaches and Solutions

In order to provide the required QoS for remote sensing and control systems, we propose several approaches for different communication segments that can be possibly involved in such a system. We propose Supermedia TRansport for teleoperations over Overlay Networks (STRON) to improve the QoS for remote sensing and control systems based on wired wide area networks. An improved Transport service for Remote Sensing and Control (TRSC) is designed to solve the QoS issues for the class of applications over wireless MANETs. These efforts are combined and extended to provide QoS for remote sensing and control applications in heterogeneous environments, where both cellular and wireless LAN technologies are present and utilized to exploit the spatial channel diversities inherent in such a environment. Finally we propose a novel networking architecture that supports relocating sensor nodes through hopping mobility. This is of special significance for those remote sensing and control applications where the auto-organization and continuity of service are important.

The purpose of the proposed Supermedia TRansport for teleoperations over Overlay Networks (STRON) approach is to improve the reliability and efficiency of remote sensing and control systems from the second perspective. STRON aims to provide a fast transport service to transmit latency sensitive supermedia streams over current non-QoS capable wide area networks. STRON takes advantage of multiple disjoint overlay paths and forward error correction encodings to improve the QoS performance. The networking routes and encoding redundancy may be adjusted dynamically to meet the QoS requirements of the supermedia streams in face of networking performance degradation. TCP Friendly Rate Control (TFRC [79]) is used as the congestion control mechanism for each overlay connection, which ensures that the supermedia traffic remains friendly to other Internet traffic.

In the improved Transport service for Remote Sensing and Control (TRSC) over wireless networks, multiple disjoint paths between the sender and receiver are used to improve reliability and reduce end-to-end latency. Forward error correction encodings are applied to certain media streams to further increase the resilience over packet loss. TFRC is used as the transport protocol, which provides a smoother congestion control and at the same time remains friendly to other TCP traffic. The target applications of TRSC are remote sensing and control applications based on MANETs. Wireless networks have peculiarities that wired networks do not have, which include link performance, node movements, and unstable topology. These wireless specific characteristics pose significant challenges for a reliable and efficient transport service.

TRSC addresses these problems by introducing a resilient multi-path routing protocol and forward error correction encodings.

When the communication channel is a heterogeneous system, we provide acceptable QoS for the application by exploiting the channel diversities of MANET peers in a hybrid MANET/Cellular environment. The idea of using multiple communication channels to achieve diversity and resilience has been exploited in many layers of the protocol stack. Utilizing multiple cellular channels can potentially improve the system performance, and of course comes with costs. In spite of the processing overhead of decomposing and combining traffic of multiple channels, there are two questions that need to be answered. First, the peer nodes of the MANET are highly likely to be located in the same cell. Recruiting more nodes to have active data transactions will inevitably increase the cell load, therefore it is not apparent that the collective performance of the application will be improved. Second, in order to deal with momentarily bad conditions in certain channels, it is desirable to have more peer nodes to exploit the spatial diversity. However, communication among the peers will cause contentions and collisions, the effect of which will become worse as the number of active peers increase. We answer the first question by demonstrating that under prevalent scheduling policies it is possible to achieve better collective performance even if the peers are within the same cell. To answer the second question, we design an adaptive peer management scheme to optimally control the number of peers based on the performance gains and contention effects. A novel contention indicator is introduced based on the MAC layer retransmission rates and backoff timers.

Large scale sensor networks call for automatic deployment and maintenance. Mobile sensors are important for facilitating sensor deployment and maintaining coverage and communication during runtime. Hopping sensors are a class of mobile sensors with a bionic mobility design that are inspired by creatures, such as grasshoppers. Hopping may be propelled by a fuel engine, compressed air, or a spring. In order to move to a different location, a hopping sensor throws itself high and toward the destination direction. After landing, the sensor is able to reorient itself into the normal position. Hopping sensors are capable of maintaining mobility in terrain where wheeled mobility is not possible. Compared with wheeled sensors, the mobility of hopping sensors are nearly instant, but lack accuracy.

Hopping enabled mobility can be used to facilitate the sensor network deployment and maintain coverage and connectivity during runtime. In the lifetime of a sensor network, the sensing task and information flow may not be homogeneous in all areas of the sensor network. Since both sensing and information forwarding consumes energy, it often happens that the sensors in a certain area are depleted faster than other areas. Those areas that have depleted sensors are called sensing holes or sensing wells. A well planned deployment may allocate redundant sensors in the field, thus when sensing wells are detected, sensors can be migrated from those regions that have redundant sensors (referred as suppliers or sources) to sensing wells. We consider the problem of transporting a certain number of hopping sensors from multiple sources to a detected sensing well.

To facilitate sensing well detection and the matching of sources to the well, we organize the sensor network field as a set of clusters. Quorum or broadcast based approaches can be used to match the supplier and consumer clusters. We model the hopping inaccuracy using a multivariate normal distribution. The influence of air disturbance is also considered. In the transporting stage we employ cascaded movement to speed the migration and argue the distance between relay clusters is crucial in determining the routing path length and the consumption of the mobile capability of sensors. We propose two schemes to minimize the total number of hops needed to fill a certain sensing well, while at the same time maintaining the relocation capability of the whole network. One scheme uses upper and lower relay edge hop limits, while the other uses a balancing coefficient to construct a new optimization target dynamically. Simulation results indicate that both algorithms are effective in balancing the requirement of path optimality and maintaining the relocation capability of the network. The dynamic algorithm is also shown to be resilient to topological changes of the network.

1.4 Structure of the Content

The remainder of this document is structured as follows. The related research work is given in chapter 2. In chapter 3, the architecture of STRON is elaborated, along with simulation and experimentation results. Chapter 4 continues to present a QoS management system based on STRON and experimental results that verify the effectiveness of the QoS management system. Chapter 5 extends the STRON idea into TRSC: Transport service for Remote Sensing and Control over wireless networks. Simulation results and theoretical modeling of TRSC are also presented in this chapter. Chapter 6 presents a QoS provision system for remote sensing and control in heterogeneous environments. Simulation results are presented to verify the proposed approach. Chapter 7 proposes a scheme to reorganize a mobile sensing network whose nodes are mobile hopping sensors. A probabilistic model is proposed for hopping sensors and an algorithm is proposed to minimize the number of hops needed when migrating a certain number of sensors. The algorithm is evaluated through simulations. Finally, a summary and possible future work are outlined in chapter 8.

CHAPTER 2

Related Work

This chapter includes a comprehensive literature survey related with supporting remote sensing and control applications in both wired and wireless environments. We start by an introduction of previous research on Internet based teleoperation systems in section 2.1. Section 2.2 reviews the bandwidth and rate control mechanisms designed for traditional multimedia applications. In section 2.3 we study the formal definitions of QoS and the QoS requirement of teleoperation systems. The related work of QoS improvement through overlay networks and in MANETs is presented in section 2.4 and 2.5. Section 2.6 focuses on QoS support in heterogeneous environments and section 2.7 provides previous research concerning nodes relocation in self-organizing sensor networks.

2.1 Internet Based Teleoperation Systems

Internet based robot systems can be classified into three categories [56]: teleprogrammed, telesimulated, and real-time teleoperated. Teleprogrammed Internet based robots are the ones that require the operator to upload a plan or a set of commands for it to execute. This uploaded program is executed by the robot either autonomously or semi-autonomously. Telesimulated Internet based robots are systems that feed forward commands in real-time but the feedback is simulated. The simulated feedback can be corrected by actual feedback. Real-time teleoperated Internet-based robotic systems feed forward commands in real-time and feedback real-time sensory information. The feedback comes in several forms the most typical of which are video and force. Elhajj, et al [54] introduced force reflection as a haptic feedback for teleoperated robots. In some actual implementations, the reflected force is not the physical reflected force the robot received, but a function of the distance of the robot to the object, or other parameters. We focus our research on the real-time teleoperated robots systems.

Teleoperation systems allow people to control automatic systems operating in a remote site where it is inaccessible by the operators. For decades teleoperation systems are used in industries to perform hazardous, repetitive operations or any other tasks which machines can perform better or more economically than human beings. All these teleoperation systems use dedicated communication channels between the operator and the robot, which introduces higher costs and less flexibility. However, the growth of the Internet opens a new stage for teleoperation systems. Through the Internet, teleoperation systems can be deployed into the everyday life of common people in an economical and flexible way. The Internet not only enables the operator to control the robot in a remote site, but also transfers video, audio and haptic feedback information back to the operator. The operator can watch, hear, touch and feel the environment the robot is working in through the Internet. The video, audio and haptic feedback extends people's sensing ability from the approachable physical vicinity to the far reaching ends of the Internet. Through an Internet based teleoperation system, physicians can touch and feel patients to make examinations, buyers can touch and manipulate products displayed in an online store, and visitors can touch and feel the exhibits in a museum.

Figure 2.1 shows the structure of a teleoperation system. The operator manipulates a control device such as control shaft or joy stick to generate control commands

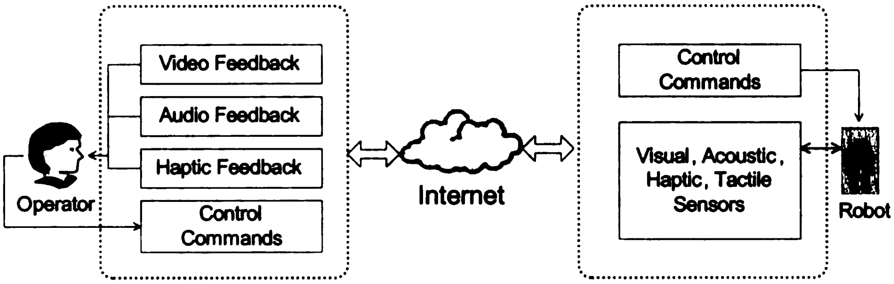


Figure 2.1. An Internet Based Teleoperation System

according to the feedback information. The operator can also send control commands in the form of text messages, voice command (useful, for example, when the operator is operating the robot through a mobile phone which does not have voice recognition capability), and video command (for example, gestures). The feedback information includes video, audio and haptic information. The robot is equipped with mechanical hands and other devices to perform operations, and also cameras, recorders and sensors to collect feedback information for the operator. At the robot side, control commands are fed into the robot to commit certain manipulations. A set of sensors (including cameras, microphones, etc.) gather information of the robot and its surrounding environment. The feedback information is sent back to the operator through the Internet. The operator and the robot form the global control loop of a teleoperation system. The Internet is serving as a communication channel within this control loop.

We call all the information flowing in a real-time teleoperation system supermedia [67]. Supermedia includes video, audio, haptic, temperature, control commands and other media. Supermedia differs from traditional multimedia in that a larger variety of media are involved, all of which are to be transmitted through a shared path in the Internet and each of them has a different quality of service requirement.

One of the major challenges of Internet based teleoperation is to make the unpredictability caused by the Internet to be transparent to the control loop of the system. Among all the uncertainties, time delay is one of the biggest obstacles to build a stable teleoperation control system. The performance difficulty caused by the Internet based teleoperation system is a result of using time as a reference for different system entities. The event based control model [56] has been proposed to combat the instability arising out of the random time delay in the control loop. In the event-based control approach, a non-time based reference is used. An event reference, which is a monotonically increasing parameter, is used to synchronize the operator and the robot. Using the event based control model, the system stability can be ensured.

However, the effective performance of highly dexterous tasks using the teleoperation system mainly depends on the quality of transfer of the supermedia streams involved in the system. A dynamic QoS based resource allocation scheme has been presented in [67] to address the problem from the application perspective. Fung, et al. introduced a Task Dexterity Index (TDI [67]) of the robotic task associated with each supermedia stream. A Task Dexterity Index (TDI) is generated for each data stream to describe the bandwidth requirement of the robotic task. With the help of TDI, bandwidth for all data streams involved in the teleoperation system is dynamically allocated during task execution.

2.2 Bandwidth and Congestion Control Mechanisms

Bandwidth is an important quality of service (QoS) factor in media stream transmission, including the media streams involved in a remote sensing and control system. Although TCP provides a reliable connection oriented transport service, it is not suitable to transmit in-time media streams. We must provide a new bandwidth and congestion control mechanism for the in-time media transmission. This chapter gives a brief review of the bandwidth and congestion control mechanisms found in traditional multimedia transmission solutions. We discuss other QoS factors in the following chapter. The bandwidth and congestion control function can be implemented in both network and transport layer. This chapter focuses on the transport layer based end-to-end control mechanisms since the current Internet supports the best-effort service model.

As shown in the research of Jain, et al [90], there are two main types of congestion control mechanisms: resource creation schemes that require the increase of the capacity of the resource, and demand reduction schemes, which reduce the demand on the resource. Bandwidth and congestion control can be implemented in both the network layer and transport layer. The network layer examples include the ICMP Source Quench [139], Explicit Congestion Notification (ECN, [145]), and Random Early Detection [63]. Transport layer based congestion control is more flexible in the case of end-to-end congestion control. Since the capacity of current network links is mostly static, the plausible mechanism would be the demand reduction mechanism.

According to the methods used to shape traffic, congestion control mechanisms can be categorized into rate based, window based and credit based mechanisms. TCP is a typical window and credit based congestion control protocol and TFRC (TCP Friendly Rate Control [79]) is a typical rate-based congestion control protocol. Congestion control protocols can also be categorized into sender based, receiver based and transcoder based mechanisms. Most mechanisms are sender based, such as TCP and TFRC. Receiver based congestion control mechanisms can be used in a multicast environment to alleviate the burden of the sending server, such as the rate control algorithm done by Smith, et al [163]. Transcoder based mechanisms take advantage of the multimedia codecs to adjust the encoding rate in accordance with the network condition.

In best effort networks, link capacity is often fixed, which means network congestion becomes the major reason of network rate changes. In this situation, the bit rate of the traffic transmitted in a certain path may give us enough information to know the congestion situation and take measures to alleviate the congestion. The routers or receiver can send back rate information, or any information related with rate measurement to the receiver. The transport entity of the receiver can adjust the packet injection rate based on the transmission rate received or calculated. By allocating an appropriate rate to each flow, the sender can keep the connection at the highest throughput while at the same time being able to avoid congestions.

2.2.1 ATM ABR Rate Control

One of the earliest rate based congestion control mechanisms appears in ATM (Asynchronous Transfer Mode [9]). ATM provides a connection oriented cell switched network service. Cells are fixed size (53 bytes) and virtual circuits are established for each network connection. There are several service classes that ATM can provide: Constant Bit Rate (CBR), Variable Bit Rate (VBR), Available Bit Rate (ABR) and Unspecified Bit Rate (UBR). Among the four classes, Available Bit Rate allows the users to specify a minimum cell rate and a maximum cell rate when the connection is to be established. During the transmission stage, the source transmits at a certain rate between the minimum and maximum rates. When congestion occurs, the source need to reduce the transmission rate to alleviate the congestion. Several rate-based congestion control mechanisms were proposed to control source rate for the ABR service in ATM [173], including Forward Explicit Congestion Notification (FECN [125]), Backward Explicit Congestion Notification (BECN [125]), Proportional Rate Control Algorithm (PRCA [80]), and Enhanced PRCA (EPRCA [152]). Although ATM has many different features compared with the Internet, some aspects of the congestion control are in common. The Explicit Congestion Control [145] of the Internet shares similar characteristics of ATM congestion control mechanisms.

2.2.2 Rate Adaptation Protocol

RAP (Rate Adaptation Protocol [149]) is a sender and rate based congestion control protocol. It is similar to TCP in that it is also an additive increase multiplicative decrease algorithm. This means when no loss happens the transmission rate is increased linearly. If congestion occurs, the transmission rate is decreased multiplicatively. In RAP, the transmission rate is controlled by inter-packet gap (IPG). The transmission rate S_i is thus defined by

$$S_i = \frac{PacketSize}{IPG_i}.$$

During the additive stage, IPG is adjusted by

$$IPG_{i+1} = \frac{IPG_i \times C}{IPG_i + C} \tag{2.1}$$

where C is a time constant. The result of Equation 2.1 is that the sending rate S_i is adjusted in the form of

$$\alpha = S_{i+1} - S_i = \frac{PacketSize}{C} \tag{2.2}$$

During multiplicative stage, S_i is updated by

$$S_{i+1} = \beta S_i, IPG_{i+1} = IPG_i/\beta \tag{2.3}$$

and β is set to 0.5 in the algorithm. Simulation shows RAP achieves the unreliable and TCP friendly transmission purpose.

2.2.3 General Additive Increase and Multiplicative Decrease

GAIMD (General Additive Increase and Multiplicative Decrease [195]) generalized TCPs window based congestion control mechanism. Instead of the increase-by-one and decrease-by-half strategy used in TCP, the sender increases its window size W by $\alpha/W(\alpha > 0)$ for each new ACK received when there is no congestion and reduces the window size to $\beta \times W(0 < \beta < 1)$ when congestion is detected. The parameters α and β are adjusted such that the flow is not as jittered as TCP but still remains TCP friendly. The ideal values for α and β are $\alpha = 0.31, \beta = 7/8$ when the GAIMD flow competes with TCP Reno flows and TCP SACK flows in both drop-tail and RED conditions.

2.2.4 Streaming Control Protocol

SCP (Streaming Control Protocol [24]) is also a modified window based congestion control protocol based on TCP. During slow-start stage and the time when congestion happens, SCP uses the mechanism as in TCP. However during the steady stage, the window size adjustment is customized based on the following equation in order to insure a smoother rate adjustment compared with TCP:

$$W_{SS} = r \times T_{rtt} + W_{\Delta} \tag{2.4}$$

where W_{SS} is the congestion window, r is the path capacity estimated during slow start stage and is updated periodically, T_{rtt} is the round trip time and W_{Δ} is an increment constant.

2.2.5 TCP Emulation at Receivers

TEAR (TCP Emulation at Receivers [150]) is a rate-based congestion control protocol in which the rate calculation is done at the receiver. The receiver gathers a bunch of congestion signals and calculates an appropriate receiving rate. This rate is fed back to the receiver and the receiver adjusts the sending rate based on this rate. Here is how the receiver calculates the receiving rate. The receiver uses a method similar to TCP to estimate the round trip time RTT and record the current congestion window $CWnd$. For a certain period of time (called epoch), a new receiving rate is calculated using $r = \frac{CWnd}{RTT}$. The epoch is defined as a period begins either when the transmission begins or congestions are detected and ends when a new epoch begins. The rate returned to the sender is the weighted average rate of the last k epochs.

2.2.6 TCP Friendly Rate Control

TFRC (TCP Friendly Rate Control [79, 65, 64, 132]) is an equation based congestion control mechanism and also one of the control mechanisms of DCCP (Datagram Congestion Control Protocol [100, 101]). DCCP provides congestion controlled flow of unreliable datagram and allows the application to choose different congestion mechanisms. The selection is done by using Congestion Control IDs (CCIDs). Presently two CCIDs are defined. CCID 2, which is TCP-like Congestion Control and provides an additive increase multiplicative decrease congestion control method with behavior modeled directly on TCP, including congestion window, slow start, timeouts and so on. CCID 3 denotes TFRC.

The TCP throughput equation used in TFRC is based on the TCP Reno equation from [132]:

$$X = \frac{s}{R\sqrt{\frac{2bp}{3}} + t_{RTO}(3\sqrt{\frac{3bp}{8}})p(1+32p^2)} \tag{2.5}$$

where

- X is the transmit rate in bytes/second.
- s is the packet size in bytes.
- R is the round trip time in seconds.
- p is the loss event rate, which is between 0 and 1.0, of the number of loss events as a fraction of the number of packets transmitted.
- t_{RTO} is the TCP retransmission timeout value in seconds, which is set to $4R$ in practice.
- b is the number of packets acknowledged by a single TCP acknowledgement.

Among these parameters, s, R, t_{RTO} and b are measured or calculated by the sender. p is measured by the receiver and sent back to the sender in the feedback packets. The research of Floyd, et al [64, 79] explains the measurement method in detail.

After a connection is set up between the sender and the receiver, the sender sends a stream of packets to the receiver at a controlled rate. When a feedback packet is received from the receiver, the sender changes its sending rate, based on the information contained in the feedback report. If the sender does not receive a feedback report for two round trip times, it cuts its sending rate in half. This is achieved by means of a timer called the no-feedback timer. To initialize the sender, the value of X is set to 1 packet/second and the no-feedback timer is set to expire after 2 seconds. The initial values for R and t_{RTO} are undefined until they are set when the feedback is received.

The receiver periodically sends feedback messages to the sender. Feedback packets are normally sent at least once per RTT. Or if the sender send less than one packet per RTT a feedback packet should be send for every data packet received. A feedback packet should also be sent whenever a new loss event is detected without waiting for the end of an RTT, and whenever an out-of-order data packet is received that removes a loss event from the history. Simulations [64] shows that TFRC has a smoother throughput variance compared with TCP and is remarkably fair when competing with TCP traffic.

2.2.7 Stream Control Transmission Protocol

Stream Control Transmission Protocol (SCTP) is defined in RFC2960 [165]. The initial motivation of SCTP is to provide a transport protocol to support Common Channel Signaling System No. 7 (SS7) on IP networks. Later SCTP turns out to be supporting high throughput network and multimedia transmission too. The major differences between SCTP and TCP include:

- SCTP supports multiple logical streams in an association (an association is the binding of several source destination pairs).
- SCTP is message oriented other than stream oriented compared with TCP.
- SCTP is designed for reliable data transfer but also include unreliable choices.

- SCTP provides mechanisms against denial of service attacks.

SCTP uses the slow start and credit window mechanisms to achieve congestion control. These are similar to TCP with little modification. Simulation shows SCTP emphasizes too much on TCP friendliness that it is too friendly in the face of congestions [23, 19]. Several works are dedicated to modify the congestion control mechanisms of SCTP [85].

2.2.8 Multicasting Rate Control

Multicasting multimedia applications need to deal with the heterogeneity of the Internet. Different multicast receivers have different computing and bandwidth capacity, thus the multicast application needs to differentiate these capacities and achieve the overall efficiency of the whole system. Since IP multicast is not aware of the capabilities of each receiver, the rate control should be addressed separately in the multicast system. Several approaches have been proposed to deal with the heterogeneity of the multicast system. In the simulcast approach [35], the receivers are divided into several groups according to their available bandwidth and the multimedia stream is transmitted according to the available bandwidth of each group. Video gateways [51] can also be introduced into the network. A video gateway resides in the network and transcodes the multimedia stream according to the available bandwidth.

Most multicast rate control mechanisms belong to either sender based approaches or receiver based approaches. In the sender based approaches [164, 20], the sender multicasts a single multimedia stream and adjusts the transmission rate based on the feedback from the receivers. In the receiver based approaches [35, 116, 179], the single multimedia stream is split into multiple segments which are transmitted across several multicast channels and the receiver decides to add or drop the multicast channels to achieve the best performance. The Layered Multicast Control Protocol (LMCP [164]) combines the sender and receiver based approaches. The receivers in

LMCP not only add or drop multicast channels according to their available bandwidth, but also dynamically approximate their available bandwidth and provide this information as feedback to the sender. The sender determines the optimal transmission rate according to the receiver feedback to optimize the performance of the multicast system.

Multicast with Cache (Mcache [146]) is a multicast mechanism proposed for video-on-demand applications. In this approach, regional cache servers are used to cache video clips which is sent to the receiver before the multicast of the sender begins. Some research has also been done on video multicast in large local area networks [159], where the variations of the CPU power other than the available bandwidth of the receivers are the constrains of the multicasting system.

2.3 QoS in Wired Networks and QoS Requirement of Internet Based Teleoperation Systems

2.3.1 What is QoS

The major motives of using the Internet as the transmission media for teleoperation systems is the low cost and ease of access. However, the intrinsic nature of the Internet imposes major challenges of transmitting supermedia streams between end teleoperation systems. The best effort service model of the Internet can work fairly well with non-real time information transmission, but the random time delay and packet loss becomes major problems in the real time based teleoperation systems.

Quality of Service (QoS) is a widely used term in the communication community to refer to a set of parameters that define the level of service provided by the communication channels. However, this is not a formal definition for QoS. The International Telecommunication Union (ITU) standard X.902 [86] refers to QoS as "A set of quality requirements on the collective behavior of one or more objects." The research of

Vogel, et al [181] provides a general survey of QoS issues for communication protocols, operating systems, multimedia databases, file servers and those issues concerned with human users. In this paper, QoS is defined as

> "the set of those technical and other parameters of a distributed multimedia system, which influence the presentation of multimedia data to the user, and in general the user's general satisfaction with the application."

Based on the research of Vogel, et al [181], QoS parameters can be classified into the following types:

- Performance oriented QoS parameters, including end to end delay, throughput of the communication channel, delay variance and throughput variance.
- Format oriented QoS parameters, including media format such as MPEG, resolution, frame rate, and also audio encoding format such as bit rate.
- Synchronization oriented QoS parameters, including coordination between different supermedia streams. For example, the synchronization between the beginning audio and video streams, the synchronization between the haptic feedback and the video, audio feedback.

2.3.2 QoS Provision in Traditional Internet

Network based QoS requires support from the network layer for the application operation. For current Internet, network support QoS is provided by the routing facilities. Several standards are proposed to support QoS in network layer, such as Integrated Service (IntServ [160]) and Differentiated Services (DiffServ [14]).

Integrated service introduces the definition of flow: a flow is a stream of packets with common source address, destination address and port number. IntServ requires routers to maintain state information for each flow and the routers determine the resources for each flow based on the resource availability. Traffic is classified into best-effort, controlled load and guaranteed service. RSVP (Resource Reservation Protocol [144]) is used to setup the integrated service. IntServ does not scale well since it requires the core Internet routers to maintain states for every individual flow.

Differentiated Services [14] are introduced to address the problems of IntServ. Instead of maintaining the states of each individual flow on all the routers, flows are aggregated into big flows that are classified on the routers. Each packet is marked as belonging to a certain kind of service and routers on the path examine the packet header to decide the service allocated to the packet based on its mark. Network based QoS requires the modification of the Internet forwarding architecture. Due to practical issues, most part of present Internet are not deployed with QoS enabled routers.

End system based QoS [119, 2, 206, 82] deals with the per-flow QoS management. For end systems that have access to QoS enabled network services, the end system should be able to interact with the network service provider to negotiate the QoS parameters. Abdelzaher, et al [2] proposed a communication subsystem architecture that satisfies the requirement of an end host in a QoS enabled network. Mehra, et al [119] presents a receiver based control system for allocating bandwidth among TCP flows according to user preference without changing the network infrastructure and the TCP stack at the sender side. In the supermedia transmission system, each supermedia stream corresponds to a traffic flow among the end systems. These streams have different QoS requirements on bandwidth, latency and jitter. End host based QoS can be used to manage the resource consumption of different supermedia streams.

2.3.3 QoS Requirement of Internet Based Teleoperation Systems

For the teleoperation systems, the relevant QoS parameters include latency, jitter, available bandwidth and packet loss rate. Latency is the time delay experienced by the data packets during transmission. The latency includes processing and transmission delay at the end system, queuing, processing and transmission delay at routers and switches, and propagation delay over the links. The queue delay is the time the data packet spent in the buffers of routers. If the path in the network experiences heavy load, the queue delay becomes longer. Thus the latency is influenced by the load of the underlying network. The unexpected delay in the network may impose challenges on the stability and synchronization of the teleoperation systems. An event-based controller design is introduced to alleviate the time delay problem in the Internet. The event-based control for teleoperation with haptic feedback is first introduced by Xi [190], and further studied in [191, 58].

Jitter is the variation of the transmission latency. Due to the variation of the queuing delay along the transmission path, the transmission latency experiences a random delay as the data reaches the end systems. Jitter not only causes challenges for the control mechanisms of teleoperation systems, but also influences the play back of feedback information. In order to remove the effects of jitter on video and audio transmission, a reconstruction filter is needed in the playback stage of the multimedia data. Each chunk of the data is stamped with a timestamp and a sequence number, and play-out of some particular packets or frames are delayed according to the network condition. A reconstruction filter is also used to mitigate the effects of jitter in the control system [128].

The bandwidth of a path is decided by the minimal bandwidth of the links among the intermediate nodes. Teleoperation systems are sensitive to bandwidth. The video and audio feedbacks need a minimal bandwidth to maintain a satisfactory display for the end users. The haptic feedback would lose its real-time capability and stability in the situation where bandwidth is scarce. In order to keep the bandwidth needs of supermedia streams, special bandwidth control algorithms should be designed.

Packet loss is the result of exceeding traffic over the network capacity. Two factors may influence the packet loss rate: the network load and the queuing policy. It is straightforward that overloaded network traffic tends to increase packet loss rate. Two most common queuing policies are drop tail and Random Early Detection (RED [63]). In drop tail queues, every incoming packet will be dropped when the queue length reaches a certain threshold. In RED queues, two thresholds are used to adjust the traffic. When the first threshold is hit, packets are dropped randomly. Only when the second threshold is hit, packets on the queue are totally dropped. Some networks can also enable Explicit Congestion Notification (ECN [145, 62]). By ECN the network can signal the end systems of congestion without dropping packets. Since packet loss is an important signal for the end host to know the situation of the network, the nature of the packet loss should be taken into account when designing bandwidth and congestion control allocation algorithms. The research of Floyd, et al [64] defines the notion of packet loss event rate other than packet loss rate considering the situation where a single congestion may cause a series of consecutive packet losses.

In the teleoperation system, the flow of the supermedia data is bidirectional. Both the operator and robot could be sender or receiver. Supermedia from the operator to the robot might include: text command (or commands translated from joystick movements), audio commands, and video commands (such as gestures). Supermedia from the robot might include all kinds of feedback: video, audio, force feedback and temperature information. For a certain kind of supermedia, there might be several data streams being transmitted. For example, the video feedback of the robot might include an overview of the robot movement as well as a close-up of the robot hand.

The objective of a good supermedia bandwidth allocation mechanism is to maximize the usable bandwidth for each type of supermedia while still maintaining a stable receiving rate for the receiver and at the same time be able to respond to network congestion at a certain level of promptness. However, different kinds of supermedia stream may require different transmission services. Some media types, such as text or binary commands, require reliable in-order transmission. We call this type in-order media (OM). Other media types, such as video and audio, may require high bandwidth in-time delivery with small end to end latency and little jitters, but reliability and packets order are not as important. We call this type in-time media (TM). The requirement for transmission of in-time media is basically similar to the requirement of traditional multimedia.

The transport layer of the Internet provides two kinds of transport service: the connection oriented TCP (Transmission Control Protocol) and datagram oriented UDP (User Datagram Protocol). TCP provides reliable transport service with flow control and congestion control. For the supermedia transmission, it is straightforward to use TCP as the transport layer protocol for in-order media. However, TCP is not designed to serve as a transport layer protocol for multimedia or the in-time media streams because of the following reasons [74]:

- TCP uses AIMD (Additive Increase and Multiplicative Decrease) as its flow and congestion control mechanism. Upon a packet loss, TCP decreases its window size sharply (some version of TCP decrease the transmission rate to the minimum). However, most multimedia streams require a minimum throughput and less jitter for useful operation. The sharp decrease of the transmission rate can cause unpleasant quality for the receiver.

- Video and audio streams can tolerate a certain rate of packet loss but have a high requirement of the timely delivery of the packets. Retransmission of lost packets is not necessary in most multimedia applications. Thus the reliable transmission service provided by TCP is not fit for multimedia applications.

- TCP introduces per packet acknowledgement in order to satisfy the reliability requirement. This can be a large portion of overhead for high bandwidth multimedia applications and multicast applications.

Due to above reasons most multimedia applications choose UDP as their transport layer protocol and implement the bandwidth and congestion control function in the application layer. For the same reason, novel transport service is needed for supermedia applications. More care should also be take to provide transport service for in-time media streams.

Player [120], Apple QuickTime [8] implement their own proprietary application layer congestion control mechanisms.

2.3.4 Encoding and Synchronization of Teleoperation Systems

This section is devoted to discuss some issues concerning the interaction of supermedia transmission and encoding that are not included in previous sections. Among different types of supermedia, video and audio streams have a relatively high requirement on the bandwidth, so the data of these streams are often compressed before being transmitted to the Internet. In order to cope with the packet loss, bandwidth and latency variation of the Internet, the compress algorithms have to deal with issues like synchronization, error correction, error concealment and rate-distortion (R-D) optimization. This solution can be integrated into the transcoder based congestion control mechanisms.

Originally the rate-distortion theory addresses the problem of determining the minimal amount of entropy R (information amount in information theory) that should be communicated over a channel such that the source can be reconstructed at the receiver given the distortion D. This theory was initially created by Claude Shannon and widely used in image and video encoding over the Internet. In a given media encoding mechanism, rate-distortion optimization can be carried out to dynamically adapt media rate according to the network status and media characteristics. Zhang, et al [200] proposes a resource allocation scheme to adapt media rate to the estimated network bandwidth using the rate-distortion function of each media layer under various network conditions. Zhai, et al [198] designs a rate-distortion optimization approach to determine the encoding mode for each packet and the channel coding rates in order to minimize the overall expected end to end distortion.

In the supermedia transmission system, we may define a distortion index for each supermedia stream. The relationship between the transmission rate, the task dexterity index and the distortion index can be explored to define a rate-distortion function. A series of rate-distortion functions can be applied into the bandwidth control mechanism to maximize the quality of the media displayable to the end users. Synchronization control in multimedia systems can be classified into two groups: preventive control and reactive control [84]. Preventive control tries to deal with synchronization problems before they really happen. Preventive approaches include using timestamp to identify each packet, and putting reorder buffer at the sender and receiver side. Reactive control concerns with the situation where desynchronization already happens in the transport layer. Reactive control mechanisms include pausing, skipping, discarding, shortening and extension of output duration.

2.4 QoS Improvement through Overlay Networks

An overlay network is composed of a set of IP-layer network paths, which are called overlay links. The end hosts of the network paths are overlay nodes. Packets may be routed among overlay nodes according to overlay link performance measurements. Since the setup of an overlay network does not require changing the underlying network infrastructure, many overlay networks are used to deploy emerging networking applications [1, 17, 99, 106, 156, 138, 142, 167, 189, 202, 201, 203, 148, 11, 170, 31, 91, 166, 212]. Other overlay networks are used to improve the performance of the Internet by circumventing the inefficiencies of BGP4 in the face of link and router failures [48, 47, 107, 108, 168, 208]. Common overlay networks include RON [6], QRON [108], SON [48], and PlanetLab. Some research focuses on implementing multicasting through overlay networks [199, 38]. Others focus on peer-to-peer file sharing systems [52, 98, 123, 71].

A Resilient Overlay Network [6, 5, 122] allows distributed applications to detect and recover from link failures or performance degradation, which may be much quicker than solely relying on the routing protocols. The RON nodes monitor the Internet paths among them and decide whether to allow the Internet to route the packets or relay the packets among RON nodes to achieve a better performance. In order to meet the reliable and fast transmission requirement of teleoperation systems, the STRON approach may utilize several overlay network paths supported by the overlay nodes. The use of parallel connections to increase throughput was extensively studied in the high performance computing community. Some research shows that when the network is under utilized, parallel connections can achieve fair utilization among common connections [77]. Statistics shows in large academic networks such as Abilene [3], most network links are usually under utilized.

In order for the STRON system to provide an effective transport service, the overlay paths should be physically as disjoint as possible. Currently there are several research efforts related with disjoint paths selection in overlay networks. Cui uses a probability model to represent the disjoint degree of two overlay paths [42]. Any pair of links of the two paths is given a probability of breaking at the same time. The disjoint degree of the two paths is thus in reverse proportion to the sum of the breaking probability of the link pairs of the two path. Finding the minimum sum has been proved to be an NP complete problem. This research presents an alternative approach to provide an approximation of the optimal case. In this approach, a shortest latency based path is selected as a primary path and another path is selected to be backup path based on its failure probability. The Control Overlay Protocol [18] divides the overlay nodes into regions. Each region has a super node. The super node probes its subordinate nodes to collect their disjoint information. Different super nodes coordinate to provide a set of approximately disjoint paths. A routing underlay approach for overlay networks [122] introduces a routing underlay that sits between the overlay network and the Internet. The routing underlay collects topological information from the Internet and this information is used in the overlay network to provide a more reliable and efficient service. This paper shows that by detouring into another AS, 1157 out of 1235 direct paths (93.7%) have at least one disjoint secondary path.

STRON uses forward error correction encoding to provide a reliable and fast transmission service, which is applicable for real time media. TCP provides the standard reliable transport service but it is not designed to serve the requirement of real time supermedia streams. UDP and its variant transport services, such as TCP Friendly Rate Control (TFRC [79]), Datagram Congestion Control Protocol (DCCP) and Stream Control Transmission Protocol (SCTP), are designed for multimedia transport but each has difficulty to provide reliable transport services by taking advantage of possible performance enhancements of overlay networks. SCTP also supports transport of the data over multiple paths but it requires the sender and receiver nodes to have multiple physical networking connections. RTP and RTCP provides support for real time application controls such as timing reconstruction, loss detection, security and content identification. However, they lack the congestion control and error recovery mechanisms that are essential for supermedia applications.

Other research work also uses overlay networks as a means to improve quality of service for multimedia applications. QRON (QoS-aware Routing in Overlay Networks [108]) is a general unified overlay network framework. One or more Overlay Brokers (OBs) in each Autonomous System (AS) form an Overlay Service Network (OSN). OSN provides services such as resource allocation and negotiation, overlay routing, and topology discovery. A hierarchical QoS-aware routing protocol is used to setup overlay paths between OBs that meet the QoS requirement of different applications. OverQoS [168] is an architecture to provide QoS using overlay networks.

Flows are aggregated into a bundle in order for OverQoS to exercise control over the available resources among the flows. Controlled-Loss Virtual Link (CLVL) is an abstraction of the physical overlay links to characterize the service received by a bundle. By policing at the entry and exit nodes of CLVL, a bound of the loss rate can be provided to the overlay applications. OverQoS does not provide a mechanism to reduce end-to-end latency, which is essential in supermedia applications. The method presented by Nguyen, et al [126] uses a *traceroute* based heuristic scheme to find redundant paths and transmit FEC encoded data over multiple paths to reduce packet loss. However, the *traceroute* approach is traffic intensive and the heuristic scheme may not always be able to find the optimal path set. Andersen, et al [7] examined the performance of transporting data over multiple overlay network paths. The measurement results showed that "the chances of losing two packets between the same hosts is nearly as high when those packets are sent through an intermediate node (60%) as when they are sent back-to-back on the same path (70%)". STRON may also use multiple overlay paths. However the FEC encoding and TCP friendly rate control of STRON make sure the latency between the end hosts is reduced even if the overlay network paths experience high packet loss rates.

2.5 QoS in MANETs

There are plenty of research work dedicated to solve the Quality of Service issues in mobile ad hoc networks (MANETs). MANETs has a lot of natures that infrastructure based networks do not have, such as variable link performance, node movements, and changing topology etc. Soft QoS [176] was proposed to release the QoS requirement in MANETs. After connections are setup, soft QoS permits a grace period during which the QoS specification may not be satisfied. Dynamic QoS approach was presented to meet the changing nature of MANETs. In the work of dynamic RSVP (dRSVP [121]), resource reservations represent ranges and applications adapt to a QoS level within the requested range.

There were efforts to support QoS in the MAC layer of MANETs, such as MACA [97] (Multihop Access Collision Avoidance), MACAW [13] and IEEE 802.11e, but we focus our research on QoS support in the network layer. Earlier MANET routing protocols does not support QoS. These protocols include Wireless Routing Protocol (WRP [153]), Ad hoc On-demand Distance Vector (AODV [137, 136]), Dynamic Source Routing (DSR [94]), Temporary Ordered Routing Algorithm (TORA [134]), Associativity Based Routing (ABR [172]), Signal Stability Routing (SSR [153]), and Dynamic Load-Aware Routing (DLAR [103]).

The goal of QoS aware routing is to find a path from the source to the destination while satisfying the QoS requirements. CEDAR, a Core-Extraction Distributed Ad hoc Routing algorithm [162] uses a set of ad hoc nodes called *core* to establish an QoS aware route from the source to the destination. The bandwidth availability information was propagated among core nodes using an link state protocol. Ticket based probing [33, 34] is one of the flooding based routing discovery algorithms. Since a complete optimal QoS routing protocol is very complicated, the ticket based routing algorithm assumes an imprecise state model and tries to reduce the amount of flooding routing messages by issuing logical tickets. Each probing message contains at least one ticket. When a probe arrives at a node, the tickets contained in the probe can be split to its neighbors. When one or more probes arrive at the destination node, the routing path is known and the networking information can be used to establish a quality aware path. The ticket based routing is again extended by Liao, et al to find a multi-path QoS routing scheme between the source and the destination [109]. The multi-path approach is able to find several paths to satisfy the QoS requirement when the available bandwidth in the ad hoc network is limited. One of the characteristics of MANETs is that the nodes are mostly energy constrained. Jia, et al proposed a method [92] to find a network topology that meets the QoS requirements while the maximum transmitting power of nodes is minimized. In the work of Zhu, et al [207], the route discovery mechanism of AODV was used to set QoS routes in a TDMA enabled MANET. In AQOR (Ad hoc QoS on-demand routing [193]), the source uses limited flooding for route establishment. The destination of the route was responsible for QoS violation detection and and the destination-initiated recovery process begins when a QoS violation is detected.

INSIGNIA [105] is an effort to design a cross-layer framework to support QoS routing in ad hoc network. INSIGNIA uses an in-band and soft-state based signaling protocol to support fast reservation, restoration and end-to-end adaptation of QoS parameters. FQMM (Flexible QoS Model for MANETs [192]) uses both the *per-flow* state property in IntServ and the *service differentiation* in DiffServ. In FQMM, the highest priority classes has per flow provisions while the other classes are given per class provisions.

Two notable multiple paths routing protocols over MANETs are AODVM [196] and Split Multi-path Routing (SMR [104]). AODVM is a multiple paths routing protocol based on Ad hoc On-Demand Distance Vector (AODV) Routing protocol. In AODV, duplicate RREQ packets are discarded by the intermediate nodes while only the first received RREQ message is processed. In AODVM, however, all received RREQ messages are recorded in the RREQ table at each intermediate node. However, the limitation of AODVM is that it cannot ensure a disjoint path is found if one exists. For example, in Figure 2.2, assuming we are routing from node A to node I, the algorithm cannot ensure that paths $AEBFI$ and $AGCHI$ can be found. In some cases, the algorithm may only be able to find a single path ($ABCDI$).

Split Multi-path Routing (SMR [104]) is a DSR (Dynamic Source Routing) based multiple paths routing approach. Unlike DSR, intermediate nodes in SMR do not keep a route cache and do not reply to RREQ messages. This ensures that the destination node receives all the possible routing messages so that it can select additional paths.

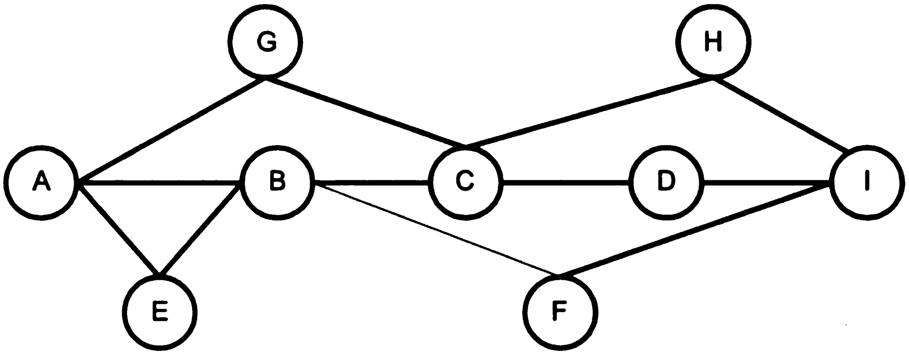


Figure 2.2. An Example for AODVM

Another difference with DSR is that duplicate RREQ messages are not discarded by intermediate nodes. Instead, intermediate nodes forward RREQ messages that are received through different incoming links and whose hop counts are not larger than previously received RREQ messages. The first RREQ received by the receiver denotes the primary path with the shortest delay. The destination then waits to receive more RREQ messages. The path that is maximally disjoint with respect to the primary path is selected as the secondary path.

FEC is traditionally used in the link layer to avoid unnecessary retransmissions. Some research uses FEC to improve the performance for a general wireless transport protocol [10, 104, 110, 111, 112, 126]. TCP with adaptive forward error correction (TCP-AFEC) [112] uses FEC to improve the performance of the TCP connection between the mobile host and the base station. FEC is also used in end-to-end TCP connections to reduce the number of retransmissions [10]. FEC was also used by McKinley, et al. [117] to build a reliable web resource multicasting service over WLANs.

Other research aims to build a general real time transport protocol using multi-flows, but FEC is not integrated into the transport protocol [115]. Some research also deals with using multiple paths for specific video encoding schemes [205]. The work of Ma, et al. [114] uses a hop-by-hop FEC check within ad hoc networks to improve the performance of multi-hop connections. However this approach imposes a large processing burden on intermediate ad hoc nodes, which is not power-optimal for mobile computing devices. M-MPR [43] and ReInForM [44] are examples that use multipath or FEC to improve the routing reliability in general wireless sensor networks. However these schemes have to trade transport efficiency for power conservation, which is a common constraint for sensor networks. In teleoperated applications, energy consumption are usually not crucial but transport efficiency and transmission latency are more important factors.

2.6 Cooperative Resource Sharing in Celluar Networks and MANETs

The integration of cellular networks and wireless LANs has drawn much attention from research communities. Some efforts have focused on developing multi-hop cellular networks by adding multi-hop relays into the cellular networks. The methods range from using a single wireless interface for both the relay and infrastructure mode [188], and using two interfaces to connect to the cellular data network and the mobile ad hoc network simultaneously. Some research [113, 143, 155, 210, 96] and our work fall into the second category. In the Unified Cellular and Ad-hoc Network Architecture (UCAN [113]), a relay proxy forwards packets from the base station to the clients with poor channel quality via high-bandwidth IEEE 802.11-based ad hoc links to improve throughput of the cellular network. A greedy and on-demand proxy discovery mechanism is introduced but the adversary effects of traffic contention caused by more proxies are not addressed. iCar [143] uses network operator managed relay proxies to divert bursty traffic from one congested cell to another idle cell. The scheme is not applicable to ad-hoc users that reside in one cell. CHUM (Cooperating ad Hoc network to sUpport Messaging [210]) is proposed as an approach to integrate 3G networks and ad hoc networks in a manner that significantly reduces 3G network costs to provide energy efficient solution for instant messaging and better QoS support by aggregation idle cellular links [209, 211].

Compared with those schemes, our work goes one step further by asking the question of under what condition the benefits of relay proxies outweigh their costs. We answer the question by measuring the contention caused by MANET peers through the MAC layer packet retransmission rate, and design a dynamic mechanism to control the number of relay proxies.

The value of spatial channel diversity in homogeneous environments has been studied from the physical layer to the network layer of different wireless communication systems [102, 158, 177, 204, 69]. A comprehensive overview of the studies can be found in [46]. Even in wired networks, the value of traffic diversity in the Internet has been studied and exploited to provide better QoS under multi-homing [171] or application layer overlay networks [6, 127, 30]. A key difference of that research and this one is that we target our design under the settings of MANET/cellular heterogeneous environments and emphasize the trade-off of diversity gains and contention avoidance. Our work is motivated by supporting remote sensing and control applications in pervasive environments, and special attention is given to reducing end-to-end latency and latency jitters. This is important to ensure system responsiveness and stability of the control loop.

2.7 Node Relocation in Mobile Sensor Networks

Much work has utilized the mobility of general mobile sensors to facilitate deployment, coverage maintenance and improving energy efficiency [161, 182, 41, 187, 183, 131, 32, 81, 66]. In the deployment stage, mobile sensors are used to adjust the coverage after being initially distributed to the sensing field. The general approach is to detect sensing holes and ensure that they are covered through sensor relocations. In [182], holes are detected using Voronoi diagrams and three protocols, namely vector based algorithm, Voronoi based algorithm and Minimax based algorithm are proposed to relocate sensors to fix the holes. In the work of [41], all sensor nodes are placed at a single point and a gradient based distributed algorithm is used to disperse the sensors into the covering area.

In the runtime stage, proper sensor distribution is essential in covering the target field. Sensor node density is also an important factor of efficient traffic forwarding. It has been shown in multi-hop ad-hoc communications, node density affects both the network throughput and latency [26, 75]. A cluster based approach is proposed in [187]. The objective of sensor relocation is to ensure the number of sensors per cluster are uniform. During the relocation, overall sensor movement is minimized to be energy efficient. Another cluster based approach is proposed by [183]. In [183], clusters are organized into grids and a quorum method is used to match sensing holes and those grids with excessive sensors. A cascaded movement method is used to transport sensors to cover the holes. The work of [131] models the mobile sensors as a virtual body and uses the notion of artificial potentials to maintain sensing coverage.

The use of hopping or flipping based sensors in sensor network deployment is also studied in [32]. The work models the individual movement of each flipping sensor and constructs a virtual graph to depict the sensing holes. A minimum-cost maximum-flow based solution is proposed to optimize both the network coverage and number of flips needed in the relocation process. However, the work in [32] does not explicitly consider the characteristics of hopping sensors we investigate in this research, and more importantly does not build the relocation scheme based on the understanding of the movement of large number of multi-hop sensors. The work of Wang, et al. [183] also comes close to our research in terms of the objectives and system settings. They follow a similar approach for migrating multiple wheeled sensors between a single source and destination within the sensor network. In contrast, we focus on the problem of relocating hopping based sensors, which have a different mobile and dynamical model compared with wheeled sensors. In addition, we consider the case where a single source cannot provide sufficient sensors and multiple sources are needed.

CHAPTER 3

Supermedia Transport over Overlay Networks

A teleoperation system is a typical example of a remote sensing and control system. In real-time Internet based teleoperation systems, the Internet is the communication medium through which the operator sends control commands to the robot and receives feedback. Teleoperation systems support robotic control commands, video, audio, haptic feedback, and other sensory information, which are called supermedia. Traditional transport services of the Internet may not be able to meet the timely transmission requirements and dynamic priorities of supermedia streams. This research aims to design an efficient and reliable transport service for teleoperation applications. Built upon multiple disjoint paths in overlay networks, Supermedia TRansport for teleoperations over Overlay Networks (STRON) uses forward error correction encodings to reduce end-to-end latency for the transmission of supermedia streams. Network routes and encoding redundancy may be adjusted dynamically to meet the QoS requirements of the supermedia streams in the face of networking performance degradation. At the cost of minimal encoding computation, the system achieves better performance in transporting supermedia streams than available transport services and at the same time remains friendly to other Internet traffic. Evaluations using PlanetLab available bandwidth traces show that STRON can significantly reduce latency.

The rest of this chapter is organized as follows. In section 3.1, the architecture of STRON is presented with a detailed design of the optimal path selection module. We discuss the simulation methodology and results in section 3.2. Section 3.4 provides the a summary.

3.1 System Architecture

3.1.1 System Overview

The basic idea of the Supermedia TRonspart over Overlay Networks (STRON) may be explained as follows. Each supermedia stream contains a series of messages generated by the teleoperation application in a variable or constant bit rate. The message is again chopped into packets with a certain size (such as the Maximum Transfer Unit or MTU) determined by the networking layer. Given p packets for a certain message that needs to be transmitted, STRON encodes the p packets into αp packets, where α ($\alpha \geq 1$) is called the stretch factor. The encoded data packets are scheduled to be transmitted over multiple overlay paths. As soon as the receiver collects $(1+\epsilon)p$ distinctive encoded data packets, the decoding algorithm can reconstruct the original data packets. Here ϵ is called the reception overhead. The reception overhead is zero for some encoding algorithms and a small number for others. A class of erasure codes that has this property is called a digital fountain code [21, 15, 151]. By using a digital fountain code, the transport protocol can provide a rather reliable transport service without using acknowledgments and retransmissions. The most common digital fountain codes are Reed-Solomon codes and Tornado codes [21]. Simulation using Reed-Solomon codes shows the encoding and decoding time for 100KB data (100 1KB-sized packets) are around 0.12 seconds and 0.054 seconds (the simulation is done on an AMD Duron 1.26GHz CPU with 512MB memory). However, the Tornado codes can decrease the encoding and decoding time by over 100 times [21]. For Reed-Solomon codes, the reception overhead is zero. For Tornado codes, the reception overhead is around 0.05. The advantage of the Tornado codes is that they trade off a small increase in the reception overhead for a big improvement of the encoding and decoding efficiency.

Figure 3.1 illustrates the mechanisms of transporting supermedia streams over multiple paths. Source data were encoded with stretch factor 2 and sent over two disjoint paths. Six packets were sent over path 1, which has a higher bandwidth and lower loss rate. Four packets were sent over path 2, whose bandwidth is low and the loss rate is high. As soon as the receiver collects 5 data packets, which are enough for the decoder to work, it signals the sender to stop sending packets and decodes the information. Because STRON eliminates the requirement of most retransmissions when packet loss occurs, it has great advantage over traditional transport protocols.

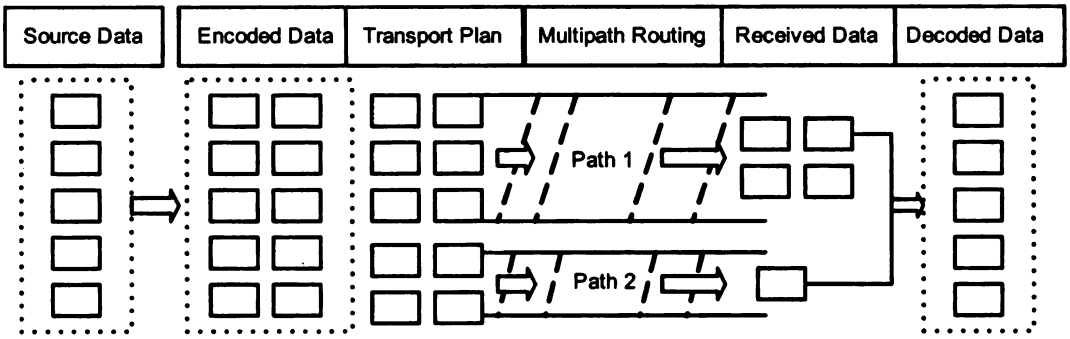


Figure 3.1. An Example of Transporting Messages over Multiple Paths

The system architecture is shown in Figure 3.2 and Figure 3.3. At the sender side, the supermedia streams are classified according to their roles in the teleoperation system by the Classifying System (CS). Each class of stream has its own QoS requirement. For teleoperation applications, the supermedia streams have three classes. Class 1 is for small but high frequency data streams, such as temperature and force feedback. These data streams have a low bit rate but whenever a data packet is available it requires timely delivery. We may not apply redundancy encoding to supermedia of this class and simply transfer copies of the data over several overlay network paths. Class 2 also works for low bit rate and high-frequency supermedia streams, such as the force feedback when multiple force detectors are present. For class 2 supermedia streams, the data generated during each sample interval cannot fit into one packet. Hence redundancy encoding is needed to provided a reliable transmission. However, since the amount of data being sent is not large, a high redundancy level may be used to ensure reliability and timeliness. Class 3 is designed to transmit high bit rate supermedia streams that do not have a strict requirement on reliability. Examples of class 3 supermedia streams include video and audio. A certain level of redundancy is used to transmit class 3 data over multiple overlay paths. We may also differentiate video and audio streams since sometimes audio streams are more sensitive to data loss than video streams. Differentiated treatments of each supermedia class are implemented in the γ updating algorithm of the Optimal Path Selection Module (OPSM).

The Sender-side Overlay Network Agent (SONA) is responsible for transporting each class of streams according to its QoS specifications. The Disjoint Path Search Module (DPSM) runs a disjoint overlay network path searching algorithm [42, 18, 122] through the connected overlay nodes. The paths found by DPSM may not be totally physically disjoint. Such paths may be called semi-disjoint paths. A disjoint degree shows the correlation of the network performance fluctuation of a pair of overlay paths. A higher disjoint degree may enhance the reliable transmission service of the system. The Network Measurement Module (NMM) is usually deployed in overlay nodes as a common service. Active measurements are launched periodically to probe the networking conditions. Some research efforts have been dedicated to making accurate measurements of the network. *Pathload* [88, 87, 37, 36, 89, 124, 129, 140] takes advantage of the fact that the one-way delays of a periodic packet stream show an increasing trend when the stream rate is larger than the available bandwidth. The

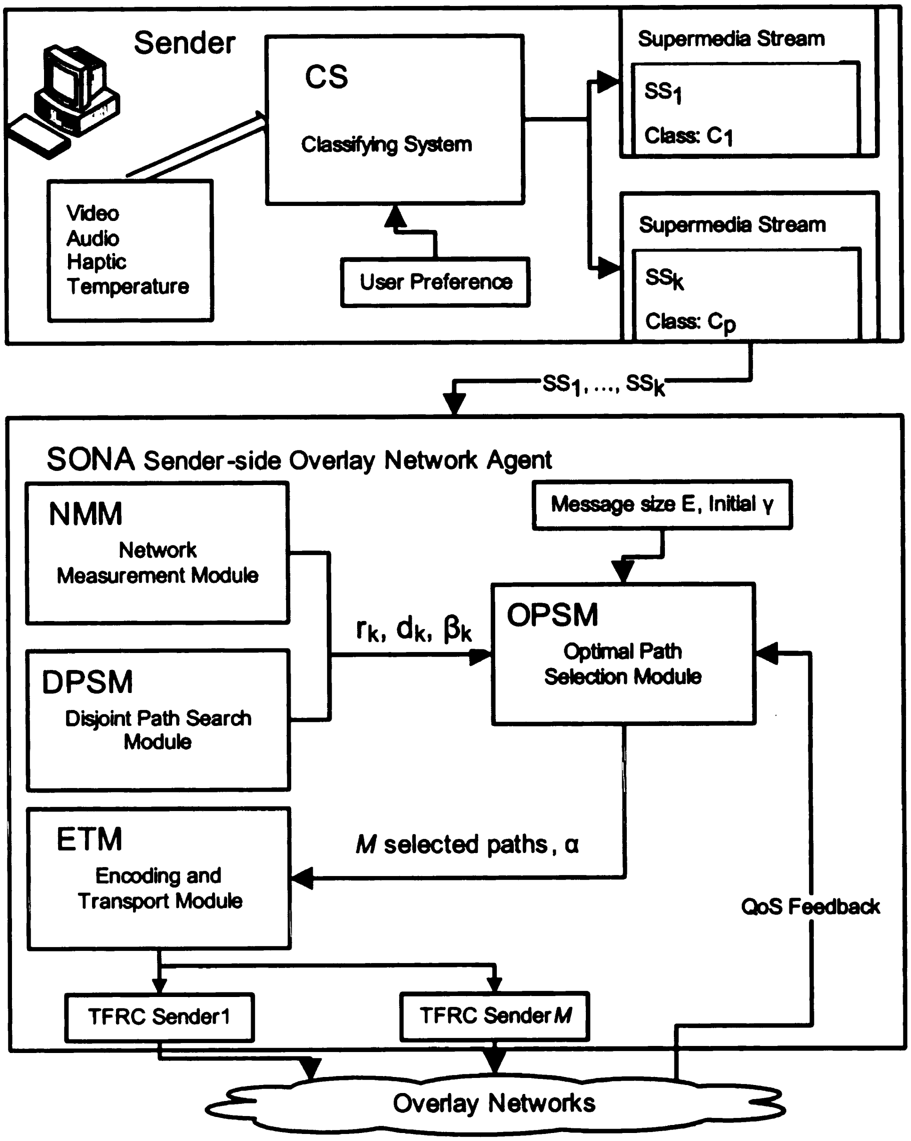


Figure 3.2. The Architecture of the Sender Side

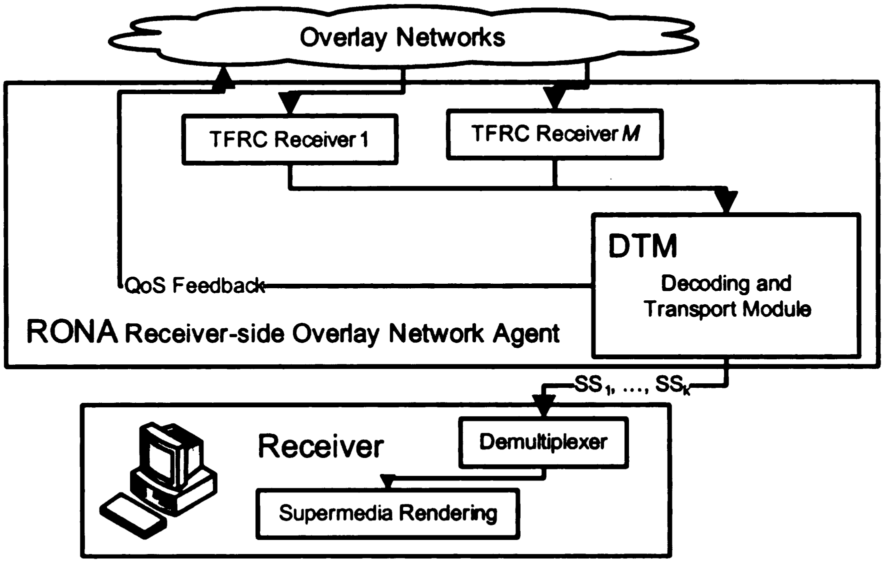


Figure 3.3. The Architecture of the Receiver Side

measurement results (available bandwidth r_i, single trip delay d_i and packet loss rate β_i) of the disjoint path set are fed into the Optimal Path Selection Module (OPSM).

The objectives of OPSM include: (1) find the optimal disjoint path set to be used as active transmission paths; (2) decide the amount of data sent over each active path; (3) decide the stretch factor according to the networking condition of the chosen active paths. In the case when the available disjoint paths set are not too large (which is true in common cases), a brute force method may be used to find an optimal subset of disjoint paths to minimize the overall transmission delay. The amount of data sent over each active path is determined according to the capacity of each path. The redundancy coefficient γ is a safety factor to make sure enough packets are received by the receiver under the presence of unpredictable loss patterns. The loss rate of each active path and γ jointly decides the stretch factor α. OPSM also receives QoS feedback from the receiver. The feedback tells the sender to what degree the redundancy in the encoding is effective or whether retransmission is needed for reliable transport services. Hence OPSM uses the feedback to decrease or increase the redundancy by calculating a new stretch factor α. The design of the OPSM is discussed in detail in the latter parts of this section.

The Encoding and Transport Module (ETM) encodes the supermedia streams using a stretch factor α and passes the encoded packets to the overlay transport service. The stretch factor α is found by the Optimal Path Selection Module. The encoding algorithm may be any of the digital fountain codes, such as Reed-Solomon codes [151] and Tornado codes [21]. TCP Friendly Rate Control (TFRC [79]) is used as the transport control mechanisms for each overlay path. TFRC provides a congestion control mechanism that is more suitable to multimedia applications than TCP while at the same time remains TCP friendly.

At the receiver side, the Decoding and Transport Module (DTM) of the Receiver-side Overlay Network Agent (RONA) decodes the packet streams received by the TFRC receivers and notifies the sender when enough packets have been collected for a certain message or a message is lost if a timeout occurs. The feedback to the sender also includes information telling the sender to what degree the redundancy in the encoding is effective so that the sender can adjust the redundancy level accordingly.

In the following section we discuss the design details of the Optimal Path Selection Module and the transport protocol.

3.1.2 Optimal Path Selection Module (OPSM)

The OPSM receives input from NMM and DPSM specifying the available disjoint overlay paths and the quality parameters of each overlay path. The goals of the OPSM is to find the optimal disjoint path set to be used as active transmission paths, and decide the amount of data sent over each active path and the stretch factor for the digital fountain encoding. The OPSM consists of two parts: the path selection submodule and the γ adaption submodule.

The Path Selection Submodule

We formulate the problem of optimal path selection as follows. Suppose N disjoint (or semi-disjoint) overlay paths were found in DPSM. For each path k ($k \in [1, N]$), the following parameters are given: the single trip delay d_k in terms of seconds, the average throughput r_k in terms of bytes per second, and the packet loss rate β_k. Each supermedia stream contains a series of messages generated by the teleoperation application as a variable or constant bit rate. The size of the message, which is the amount of effective data we need to transmit from the sender to the receiver, is E in terms of bytes.

We need to choose M disjoint paths out of the N given paths to minimize the latency between the sender and receiver. For each path i ($i \in [1, M]$) of the M path, we also need to calculate V_i, which is the amount of data injected into this path by the sender. The system redundancy coefficient γ shows how much redundancy the system has over unexpected packet loss. Assume E_i is the effective data gathered from path i that is used in the data construction process, and we have

$$E_i = \frac{V_i(1-\beta_i)}{\gamma}. \tag{3.1}$$

Assuming t is the time for the receiver to receive and reconstruct the original data, we have

$$t = \frac{V_i}{r_i} + d_i \tag{3.2}$$

and

$$V_i = (t - d_i)r_i \tag{3.3}$$

According to (3.1)

$$E_i = \frac{r_i(t-d_i)(1-\beta_i)}{\gamma} \tag{3.4}$$

We have

$$\sum_{i=1}^{M} E_i = E(1+\epsilon) \tag{3.5}$$

where ϵ ($\epsilon \in [0,1)$) is the reception overhead of the digital fountain code, or

$$E = \frac{\sum_{i=1}^{M}[(t-d_i)r_i(1-\beta_i)]}{(1+\epsilon)\gamma} \tag{3.6}$$

Solving t yields

$$t = \frac{E(1+\epsilon)\gamma + \sum_{i=1}^{M} r_i(1-\beta_i)d_i}{\sum_{i=1}^{M} r_i(1-\beta_i)} \tag{3.7}$$

To solve the problem, we may try set $s = \{(i_1, i_2, \cdots, i_M) | i_p \in [1,N], p \in [1,M]$, and for any $p \neq q, p \in [1,M], q \in [1,M], i_p \neq i_q\}$, which is a combination of set $[1,N]$ to minimize (3.7), which is

$$t = \frac{E(1+\epsilon)\gamma + \sum_{j \in s} r_j(1-\beta_j)d_j}{\sum_{j \in s} r_j(1-\beta_j)} \tag{3.8}$$

The most straightforward method is to enumerate all the $\binom{N}{M}$ sets to find the subset s of $[1,N]$ that minimizes (3.7). When N is small, which is true in most cases, this method works well. When N is large, more efficient algorithms are needed.

The volume that needs to be sent over a selected disjoint path i, which is V_i, may be found by using (3.3). The stretch factor α of the digital fountain encoding may be calculated as

$$\alpha = \frac{\sum_{i=1}^{M} V_i}{E} \tag{3.9}$$

A *transport plan* consists of a series of active overlay network paths, the QoS parameters of these paths, the redundancy factor γ, the stretch factor α and the number of packets (or the volume of data) to be sent over each active overlay path.

We have the following example to illustrate how the path selection process works. Assume the message size is 100 packets and the packet size is $1K$ bytes. Thus $E = 100KB$. We assume the redundancy coefficient γ is 2 and the reception overhead ϵ is 0.05. Assuming we have two disjoint paths to choose (shown in Table 3.1), we have the following possible strategies.

1. Choosing path 1 only, we have
$t = \frac{E(1+\epsilon)\gamma + r_1(1-\beta_1)d_1}{r_1(1-\beta_1)} = 0.4535.$

Table 3.1. Two Disjoint Paths

Path	r_k (KB/sec)	d_k (sec)	β_k
1	600	0.1	0.01
2	1000	0.2	0.005

2. Choosing path 2 only, we have
$t = \frac{E(1+\epsilon)\gamma + r_2(1-\beta_2)d_2}{r_2(1-\beta_2)} = 0.4111.$

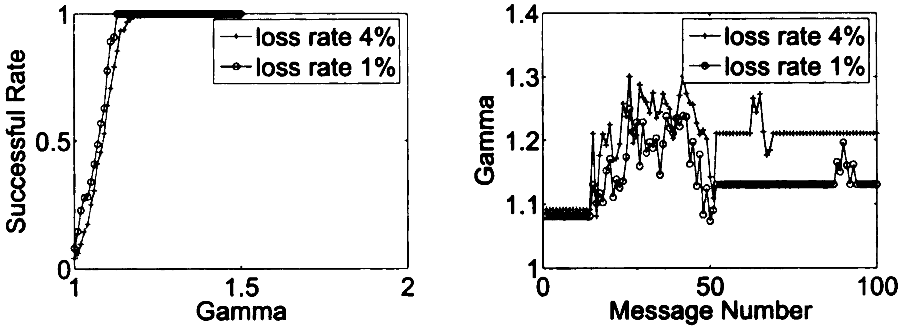
3. Choosing both paths, we have
$t = \frac{E(1+\epsilon)\gamma + r_1(1-\beta_1)d_1 + r_2(1-\beta_2)d_2}{r_1(1-\beta_1)+r_2(1-\beta_2)} = 0.2948.$

It is clear that strategy 3 is the best one. Given this strategy, we can calculate the V_i for each path according to (3.3): $V_1 = (t-d_1)r_1 \approx 117KB$ and $V_2 = (t-d_2)r_2 \approx 95KB$. The stretch factor α is $\frac{V_1+V_2}{E} = 2.1164$.

The γ Adaption Submodule

The redundancy coefficient γ is a user defined parameter specifying the amount of redundancy to be used in the system. The value of γ influences the efficiency and reliability of the system and should be adjusted dynamically when the quality of the overlay paths changes. The larger γ is, the more reliable the transport service. However, a larger γ requires more encoding and decoding overhead. Figure 3.4(a) shows the relationship of γ and the successful rate.

In the simulation scenario, two candidate disjoint overlay network paths were established (detailed description of the simulation methodology may be found in Section 3.2). The slower path *path0* has an available bandwidth of 1MB and a single trip delay of 100ms. The faster path *path1* has an available bandwidth of 10MB and a single trip delay of 100ms. In both cases, 100 supermedia messages were sent. Each message contains 50 packets whose size are 1000 bytes. The MMPP model was used to simulate the packet loss events (see Section 3.2.2 for details). A message is said to be transmitted successfully when the receiver is able to reconstruct the original message using the packets received over different overlay paths. In the first case, both overlay paths were using an MMPP error model with an expected packet loss rate of 4%. We see that $\gamma = 1.2$ ensures the transport to be 100% reliable under this network condition. In the second case, both overlay paths used an MMPP error model with an expected packet loss rate of 1%. A smaller γ was able to ensure that the transport is 100% reliable. Considering the difficulty of specifying the redundancy



(a) Effects of Gamma over Successful Rate

(b) Automatic Gamma Adaptation

Figure 3.4. Simulation of the Influence of γ

coefficient by the user and the fact that the quality of the network is always changing, we need an automatic γ updating algorithm. The γ adaption submodule updates the redundancy coefficient according to the quality of the overlay paths. Figure 3.5 shows the gamma adaptation algorithm. The variables **actual_succ_rate** and *uf_rate* are feedbacks received from the receiver by the sender. The target successful rate of this supermedia class is denoted as **target_succ_rate**. The variables **gamma_roof** and **gamma_floor** are the minimal and maximal value of γ for this supermedia class. The variables **diff_allowable**, **uf_floor** and **uf_roof** are other QoS parameters for this supermedia class and will be explained in the following paragraphs.

The variable **actual_succ_rate** is the ratio of the messages successfully delivered by the system up to the present. A message is regarded as delivered successfully if enough packets are received by the receiver before a timeout occurs and the original message is successful decoded by the decoder. The variable uf_rate means "under-full rate." The under-full rate is calculated by the receiver after a message is successfully decoded, otherwise the under-full rate is set to -1. The definition for uf_rate is

$$uf\_rate = \frac{(AllPkts - PktsSent) \cdot \frac{PktsReceived}{PktsSent}}{AllPkts} \tag{3.10}$$

where $AllPkts$ is the total number of packets after the message is encoded and may be calculated as

$$AllPkts = \lceil \alpha \cdot EffectivePkts \rceil. \tag{3.11}$$

α is the stretch factor and $EffectivePkts$ is the original data packets of the message.

Upon decoding, $PktsReceived$ should equal $EffectivePkts$. $PktsSent$ is the number of packets sent by the sender up to now. The receiver obtains this information from the packet ID of the last packet received. Although there may be some packets in transit or lost after the last packet was sent, those packets are negligible.

The variable uf_rate shows the percentage of the packets that would have been useless if they were received successfully by the receiver. Thus, uf_rate is a parameter that shows how much redundancy was "wasted" during previous transmissions. To keep the system stable, uf_rate is updated in the following way:

$$\begin{aligned} uf\_rate_{new} &= uf\_rate_{old} \cdot \alpha_{uf} \\ &\quad + uf\_rate_{last\_time} \cdot (1 - \alpha_{uf}) \end{aligned} \tag{3.12}$$

where α_{uf} is a stabling parameter.

The γ adaptation algorithm works as the following. If the actual successful rate of the supermedia stream is below the target successful rate by a difference more than `diff_allowable` or uf_rate is below the floor threshold and γ has not reached the roof value, γ is increased. Otherwise if the under-full rate is over the roof value (`uf_roof`) and γ is over the floor value, γ is decreased.

```
bool gamma_changed = false;
if ( (uf_rate < uf_floor ||
       target_succ_rate - actual_succ_rate
       > diff_allowable) &&
      gamma < gamma_roof) {
      increase gamma;
      gamma_changed = true;
}
else if ( uf_rate > uf_roof &&
      gamma > gamma_floor) {
      decrease gamma;
      gamma_changed = true;
}
if( gamma_changed)
      run path selection algorithm;
```

Figure 3.5. γ Adaptation Algorithm

The system adjusts the quality of service for different classes of supermedia by setting appropriate γ adaptation parameters for them. The default values of the γ adaptation parameters used in the simulation system are shown in TABLE 3.2. βavg is the average loss rate of the active overlay paths. The parameters for class 1 supermedia are trivial since class 1 supermedia may not need redundancy encoding but simply send duplicate data packets over different paths. If the quality of the overlay paths is stable, the value of γ will converge and the system reaches a stable state. Figure 3.4(b) shows the working process of the automatic γ adaptation

Table 3.2. The γ Adaptation Parameters

Class	2	3
Initial γ	$\beta avg + 2$	$\beta avg + 1$
target_succ_rate	100%	98%
gamma_roof	$\beta avg + 4$	$\beta avg + 3$
diff_allowable	0.000001	0.001
uf_roof	0.16	0.12
uf_floor	0.12	0.10

system. The simulation scenario is the same as the one described at the beginning of this section. In both cases, 100 class 2 supermedia messages were sent. After a certain number of transmissions, γ tends to converge to a stable value.

3.1.3 The Transport Protocol Design

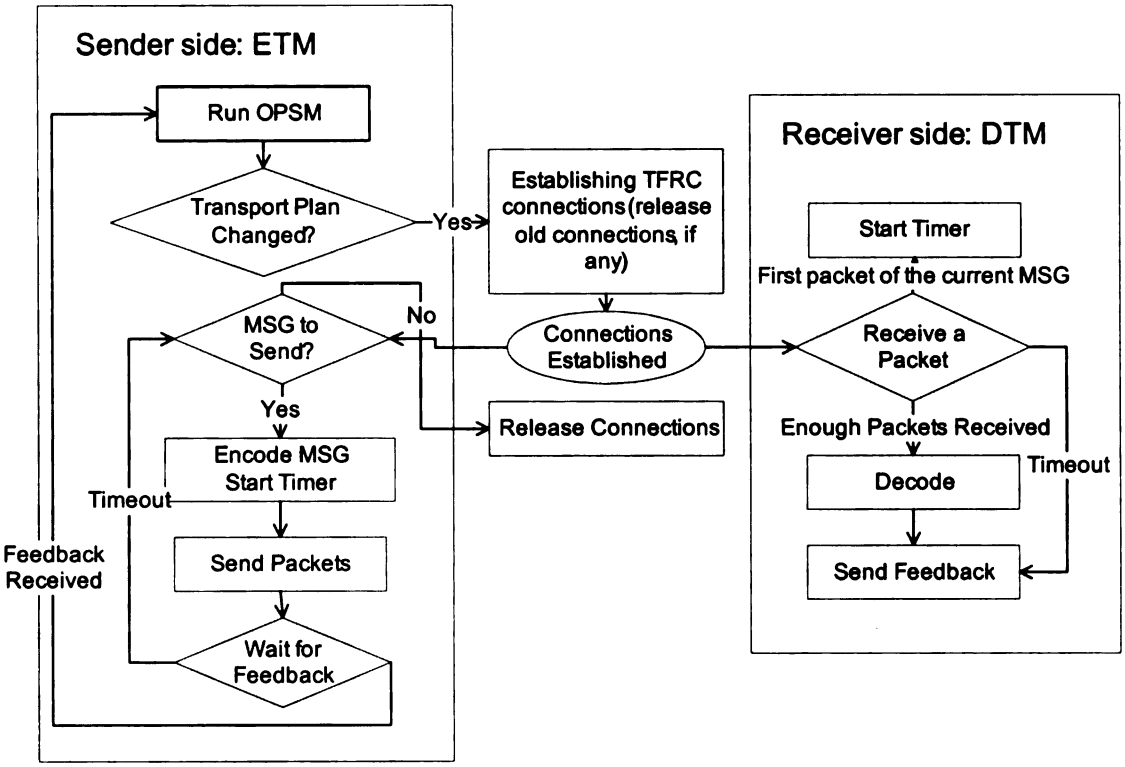


Figure 3.6. The Transport Protocol State Transitions

The state transitions of the transport protocol are shown in Figure 3.6. The two major transport entities are the Encoding and Transport Module (ETM) at the sender side and the Decoding and Transport Module (DTM) at the receiver side. The ETM calls OPSM when the system initializes and when a feedback is received. OPSM in turn gets inputs from NMM and DPSM, which provide the overlay network information (as seen in Figure 3.2). OPSM produces a transport plan using the overlay network information and the feedback. According to this transport plan, ETM initializes the TFRC connections and encodes the messages to be sent. Before sending the encoded packets, ETM starts a timer. If the timer timeouts before feedback is received, ETM stops waiting and tries to send a new message using the old transport plan. The value of the timer may be specified in the form of δt, where t is the speculated transmission time found in (3.8) and δ is an adjustable parameter. For certain classes of supermedia streams, a reliable transport service may be needed. In such cases, ETM will repeat transmitting the same message until a positive feedback is received.

The receiver side transport entity DTM starts a timer when the first packet of a new message is received. After enough packets were received for the current message (the number of effective packets required to decode the message is specified in the data packet, see Figure 3.7), ETM decodes the message and sends a feedback to the sender. If the timer timeouts before enough packets were received, a negative feedback (with the under-full rate set to -1) is sent back to the sender. Similar to the sender side timer, the value of the receiver timer may also be specified in the form of δt, where t is the speculated transmission time found in (3.8) and δ is an adjustable parameter. However, the sender side feedback timer should be a bit longer than the receiver timer considering the single trip delay of the feedback packet.

Since the feedback only contains a small amount of data, no redundancy encoding is needed. The feedback may be sent over the primary overlay path (the overlay path with the best quality) through the TFRC connection of that path. When none of the overlay paths has satisfactory quality, the feedback may also be sent with duplicate copies over several overlay paths. Besides the existing TFRC connection, an additional TCP connection can be setup to transmit the feedback data. The packet formats of STRON data and feedback are shown in Figure 3.7 and Figure 3.8. Stretch factor, uf_rate and **actual_succ_rate** are double float numbers. The sender and receiver should agree on a format of these numbers.

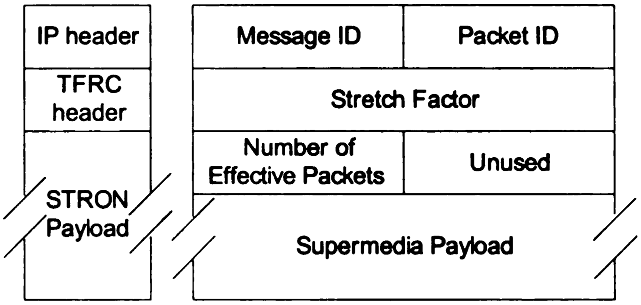


Figure 3.7. The Data Packet Format of STRON

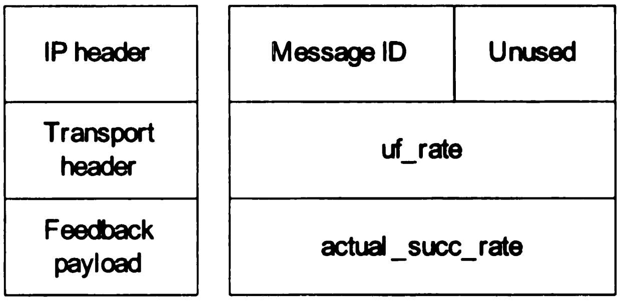


Figure 3.8. The Feedback Packet Format of STRON

3.2 Simulation

To evaluate the effectiveness and efficiency of the system, the ns2 network simulator was used to construct a simulation environment for the overlay network transport protocol. An Optimal Path Selection Module was implemented in the application layer of the simulator. An implementation of Reed-Solomon codes [15] was used to estimate the encoding and decoding overhead of the algorithm. We did not use Tornado codes in the simulation due to its proprietary nature.

3.2.1 Simulation Methodology

The OPSM was implemented in ns2 as an Application layer protocol. According to the overlay network information and the supermedia transport scenarios provided by the simulator input, the OPSM constructs a transportation plan. The TFRC module in ns2 was modified to support a supermedia payload. One unreliable TFRC connection is established over each chosen overlay path in the supermedia transport plan. The receiver side of the application collects successfully transmitted packets and carries out a simulated decoding process. The application layer of the simulator consists of the following interfaces:

- Adding paths: Add an overlay network path to the system. For each path, the system is provided with the available bandwidth r_i, single trip delay d_i and packet loss rate β_i.
- Removing paths: Remove a previously added overlay network path.
- Modifying paths: Modify the parameters of a previously added path.

After collecting simulation setup information from the user, the system application layer runs the path selection submodule using the initial γ adaptation submodule parameters to choose a set of paths as an active overlay path. The simulator scheduler starts sending supermedia messages at a specified time. For each message received, the receiver generates the QoS feedback needed for the γ updating algorithm. When a QoS feedback is received by the sender, the sender runs the gamma adaption submodule and path selection submodule, which yield a new transport plan.

3.2.2 Packet Loss Models

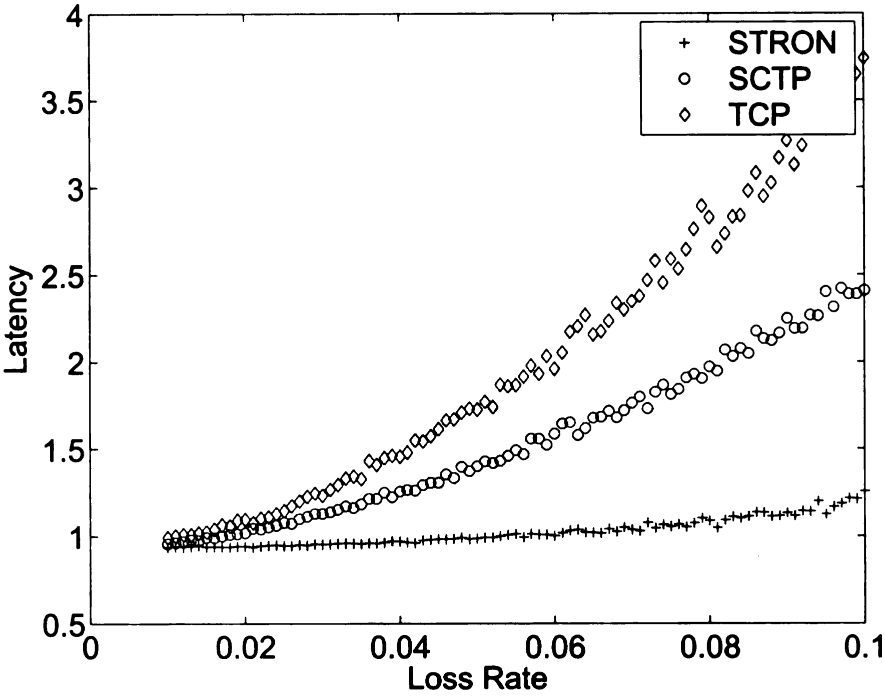


Figure 3.9. Latency vs Loss Rate: STRON, TCP and SCTP (with 1 path)

In order to simulate the behavior of the system under lossy network conditions, it is important to choose good packet loss models. Extensive research [76, 39, 194, 174, 175, 157, 93, 77, 61, 4] has been done on simulating packet loss of the Internet.

Trivial stochastic packet loss models drop a subset of packets according to a probabilistic process, such as uniform or exponential processes. A Poisson arrival process assumes the packet loss event to be independent and the loss event inter-arrival time conforms to an exponential distribution. Some measurements [194] show that the

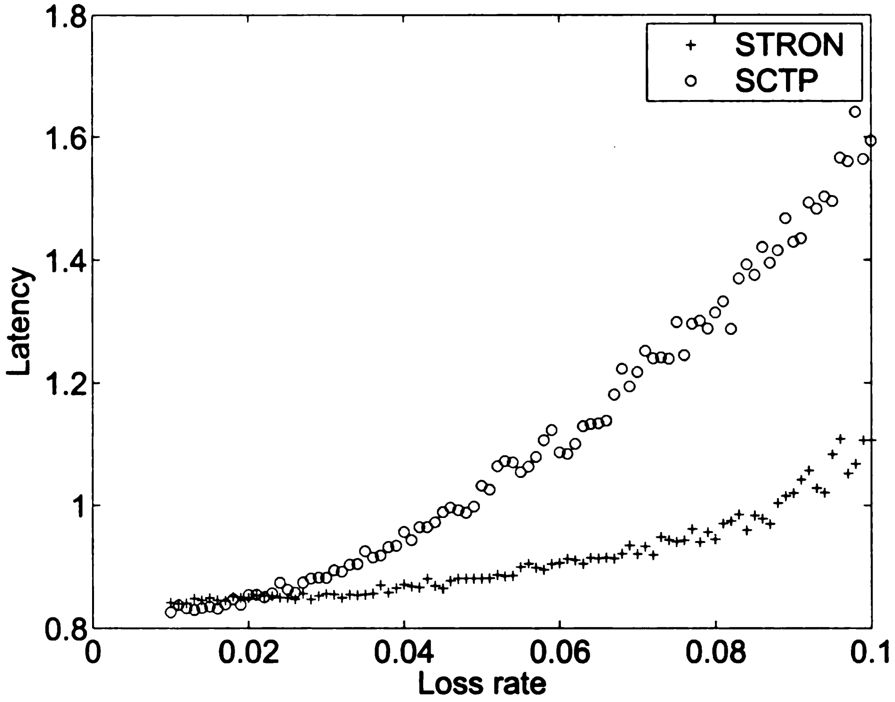


Figure 3.10. Latency vs Loss Rate: STRON and SCTP (with 2 Paths)

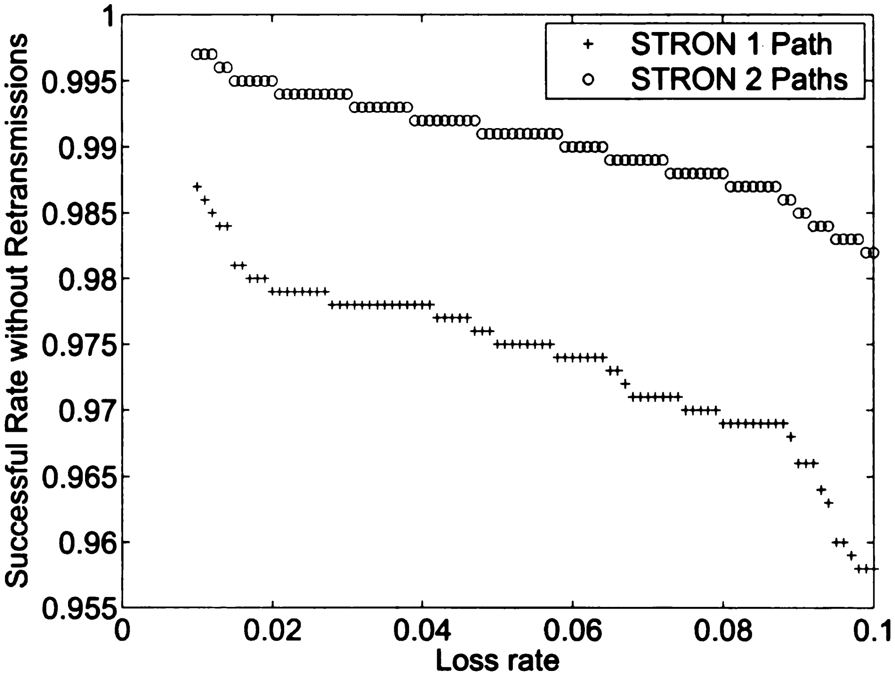


Figure 3.11. Successful Rate without Retransmissions

packet loss events of the Internet may be independent and fit an exponential distribution.

A more sophisticated error model uses a two-state continuous time Markov chain. The model has two states that have different packet dropping rates. One common configuration is to set one state to have a zero dropping rate, corresponding to a stable period of the network, and the other state have a certain dropping rate, corresponding to the lossy period of the network. The state sojourn time conforms to an exponential distribution. Although the Markov model retains memory of previous states, some work [93] found that the Markov model tends to underestimate the probability of consecutive packet losses.

To depict packet loss events more accurately, the Poisson process is modified into a Markov Modulated Poisson Process (MMPP). The MMPP is a doubly stochastic Poisson process whose arrival rate is given by an m-state irreducible continuous time Markov chain. The Markov chain should be independent of the Poisson process. The MMPP can be implemented in the ns2 simulator using the MultiState loss module.

3.2.3 Simulation with Variable Packet Loss Rates

In the following simulations, two candidate disjoint overlay network paths were established. The slower path *path0* has an available bandwidth of 1MB and a single trip delay of 100ms. The faster path *path1* has an available bandwidth of 10MB and a single trip delay of 100ms. We simulated the transmission of class 2 supermedia. Each message contains 50 packets, each of which is 1000 bytes. In the following two subsections, we compared the performance of STRON with TCP and SCTP under variable packet loss rates. Since TCP does not support multiple transmission paths, only *path1* is used in the simulation when TCP is involved in the comparison.

Unless explicitly specified, we used the exponential packet loss model in this simulation since the exponential model is more suitable to generate packet loss events according to a given packet loss rate. In the following simulation, the packet loss rate was increased from 0.01 to 0.1. We see from Figure 3.9 that while the packet loss rate increases, STRON performs better than TCP and SCTP. For example, when the loss rate is 0.05, STRON takes 0.98 seconds to transmit a 50KB message, while TCP takes 1.73 seconds and SCTP takes 1.40 seconds. An experiment with the Reed-Solomon codes shows with stretch factor 1.2, the encoding and decoding of 50KB data takes 0.014 and 0.002 seconds under the experimenting computer (an AMD Duron 1.26GHz CPU with 512MB memory). Thus overall STRON takes 0.996 seconds, with an improvement of 42% compared with TCP and 29% compared with SCTP. If Tornado codes were used as the digital fountain encoding method, the improvement would be even higher.

Unlike TCP, SCTP is able to support multiple transmission paths. In Figure 3.10 two paths were used for the transmission. The path *path0* has an MMPP packet loss model with expected packet loss rate 0.01. The path *path2* has an exponential packet loss model with the packet loss rate increasing from 0.01 to 0.1. Although when the packet loss rate is low, SCTP performs a little better than STRON, STRON may take better advantage of the overlay paths as the loss rate increases. For example, when the loss rate is 0.05, STRON takes 0.88 seconds to transmit a 50KB message, while SCTP takes 1.03 seconds. Using the same encoding and decoding time in the previous simulation, we have a 13% improvement with STRON over SCTP. The successful rates without retransmissions of STRON are shown in 3.11. It shows an additional overlay path helps improve the successful rate dramatically.

3.2.4 Simulation with PlanetLab Traces

In this part, we simulated the behavior of STRON using available bandwidth traces collected under PlanetLab. The available bandwidth traces were measured using *Pathload* [88] among three PlanetLab nodes that are located in Michigan State Uni-

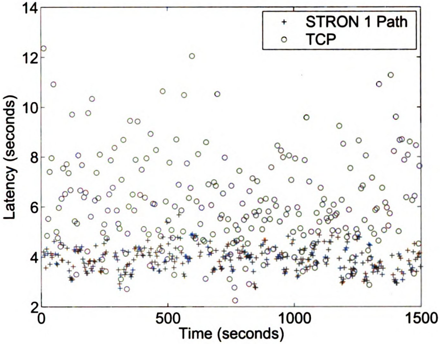


Figure 3.12. Comparison of STRON and TCP Using BW 1

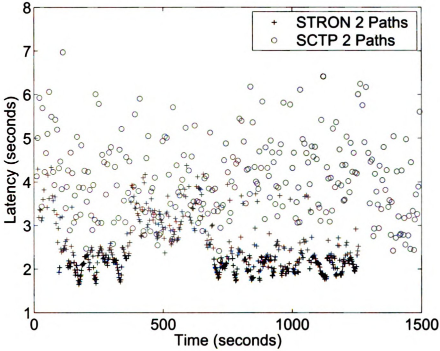


Figure 3.13. Comparison of STRON and SCTP Using BW 1 and BW 2

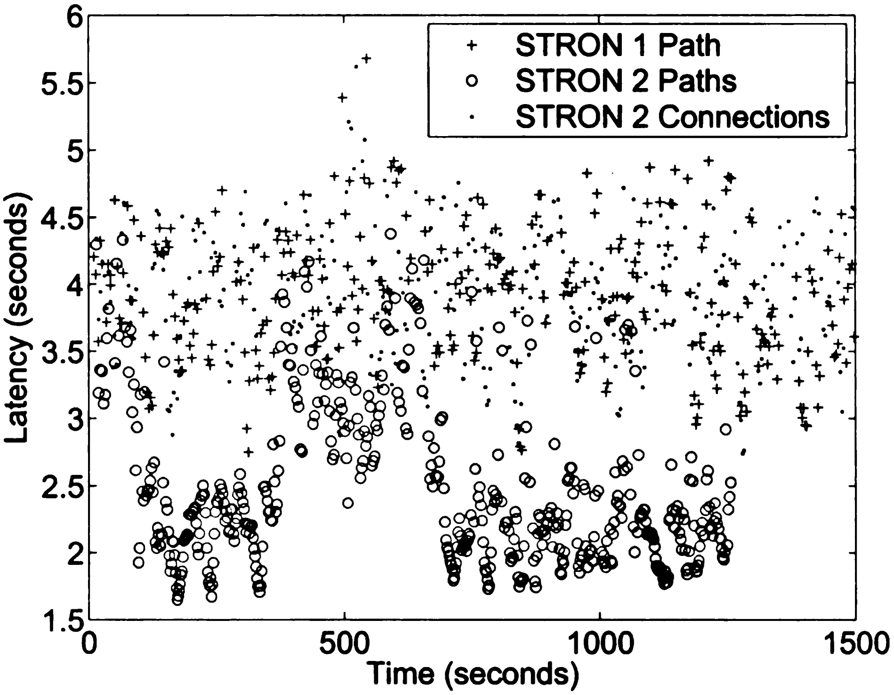


Figure 3.14. STRON under Different Conditions

versity (node MSU), Japan (node JPN) and Australia (node AUS), respectively, as shown in Figure 3.15. The measurements were done among the three nodes at the

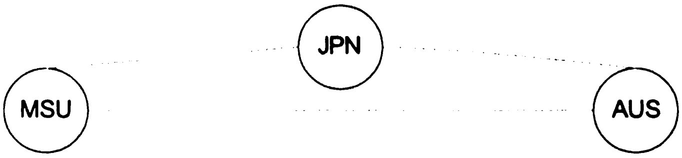
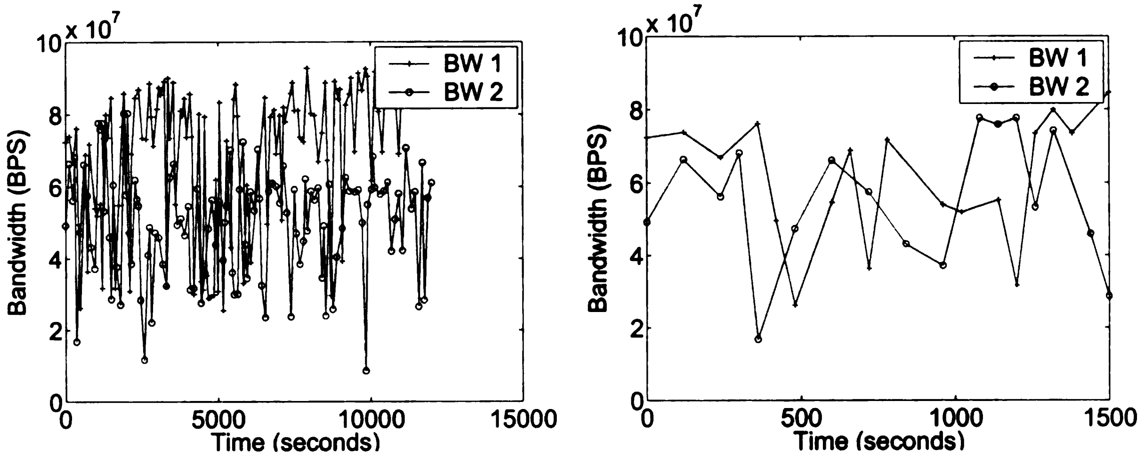


Figure 3.15. Three PlanetLab Nodes

same time. One *Pathload* probe started at the beginning of every minute. Most measurements finished within one minute. If *Pathload* does not converge within one minute, the launching of the next measurement was delayed until the beginning of the minute that goes right after the previous *Pathload* measurement finishes. The measurement results generated two bandwidth series as shown in Figure 3.16(a). $BW1$ was the available bandwidth between node MSU and AUS. $BW2$ was the minimum of the available bandwidth of the path between MSU and JPN, and the path between

JPN and AUS. $BW2$ can be regarded as the available bandwidth series of an overlay path between MSU and AUS that traverses JPN. Results of *traceroute* show that the paths of MSU-AUS and MSU-JPN-AUS are quite disjoint except for some overlaps on the continents of North America and Australia. A snapshot of the available bandwidth measurement (bandwidth series from 1 to 1500 second in Figure 3.16(b)) was used in the simulation. We modified the simplex-link object in ns2 to enable the bandwidth input. In the following scenarios, a series of supermedia messages were sent from node MSU to AUS. Each message has 250 original packets and the packet size is 1000 bytes. An MMPP packet loss model with expected packet loss rate 1% was assumed on all paths. Figure 3.12 shows the latency variance when the messages were sent over the direct connection using the available bandwidth series $BW1$. STRON achieves better performance and has less variance compared with TCP. Figure 3.13 shows the situation when both the direct path and overlay path are used by STRON and SCTP. In the simulation of Figure 3.14, we established two TFRC connections under the direct path (with available bandwidth $BW1$). The results show that two TFRC connections under one path do not necessarily decrease the latency compared



(a) The Available Bandwidth Series

(b) The Series Used in the Simulation

Figure 3.16. PlanetLab Available Bandwidth Series

with one TFRC connection under one path. However, a second overlay path can improve the performance of the system under the same packet loss conditions. The average and standard deviation of the latency under different schemes are shown in TABLE 3.3. Considering the encoding and decoding time of STRON (which are 0.071 seconds and 0.014 seconds for 250KB data on the experimental computer, see Section 3.2.3), in the case of one path, STRON has an improvement of 34% over TCP. Using the same encoding and decoding overhead, in the case of two paths, STRON has an improvement of 37% over SCTP. STRON has reduced the latency by 37% by using an additional transport path. STRON with 1 path and STRON with 2 paths have an average successful rate without retransmissions 98% and 99%, respectively.

Table 3.3. The Average and Standard Deviation of the Latencies

Simulation	Average	STD
STRON 1 path	3.9268	0.4947
STRON 2 paths	2.4604	0.5886
TCP 1 path	6.0985	1.8439
STRON 2 connections	3.9022	0.4899
SCTP 2 paths	4.0613	0.9387

3.3 Experimental Implementation and Testing

The advantages of the proposed algorithm are demonstrated using experiments with a teleoperated mobile manipulator robot.

3.3.1 Description of the Experimental Platform and Setup

Figure 3.17 shows the hardware structure of the Experimental test platform. It consists of a robot manipulator, a haptic device and a vision system connected through the Internet. The haptic device (phantom) is used to obtain velocity commands from the operator and render force feedback from the robot. The vision system is used for visual feedback from the robot and its surroundings.

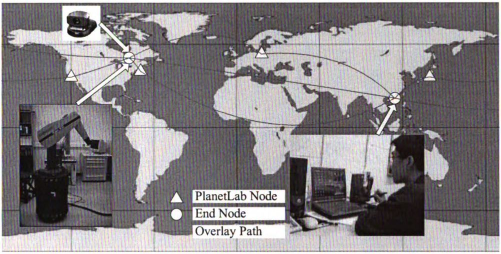


Figure 3.17. The Experimental Test Platform

The communication channel is composed of several nodes in the PlanetLab overlay network. The nodes were selected at myriad locations over the world from several distinct Autonomous Systems (AS). Multiple routes from the source to the destination for the various supermedia streams were implemented via the overlay network. The stretch factor α for the various supermedia streams was preset to be equal to the number of paths being used. The reception overhead factor, ϵ, was set to 0.

The robot system was located at Michigan State University (MSU), USA and the operator was located at Chinese University of Hong Kong (CUHK), Hong Kong, China. Overlay routing nodes for implementing the multiple path approach were selected from many different location over the world. Four disparate paths were heuristically chosen for the overlay routing links. The four paths chosen can be tabulated as 1) east coast of USA path, 2) west coast of USA path, 3) a path through Europe and 4) a path through Asia. Experiments were carried out utilizing 1, 2, 3 and 5 paths (including the default Internet path). In order to achieve repeatable operation to compare the results of the various experiments, the teleoperation task was simulated using a stream of actual teleoperation commands recorded in a previous trial. The task involved moving the robot end-effector in concentric circles in the

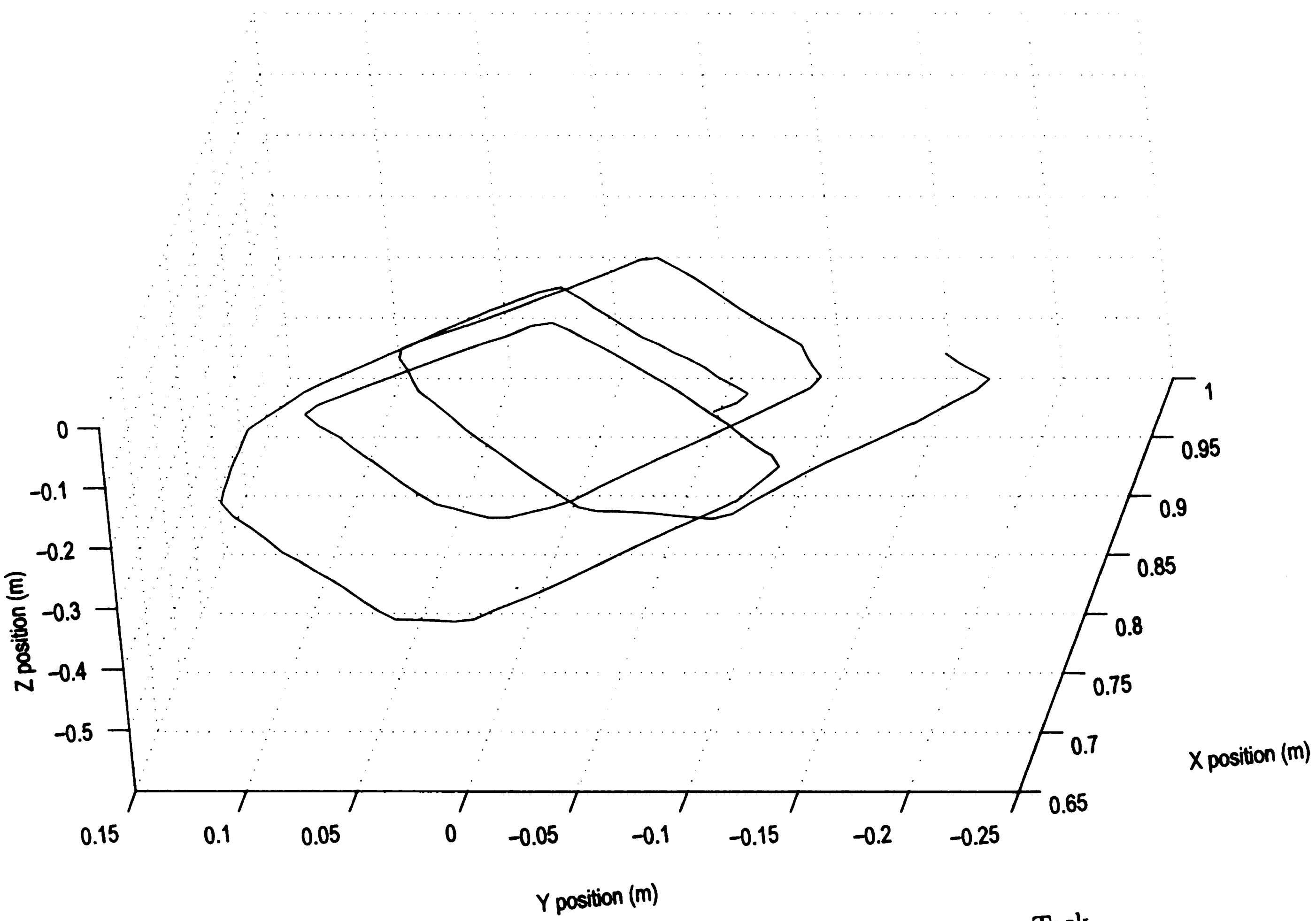


Figure 3.18. The Commanded Position for Teleoperation Task.

$X - Y$ plane as is shown in figure 3.18. This task has high dexterity requirements as is explained in [133]. The same set of commands was played back at the Hong Kong peer for each experimental run.

3.3.2 Experimental Data and Analysis

The teleoperation task was carried out for different number of paths used to transport the supermedia streams. The Round Trip Time (RTT) delay, motion of the robot and effective time to completion of the task are recorded and analyzed.

Experimental Results and Discussions

First, the teleoperation task was carried out using only the default route as a single data path between the operator and the robot. Next the proposed system using a total of up to 5 routes (4 routes via the overlay network and one default route), was used to route supermedia streams involved in the teleoperation task. Figure 3.19 depict a 1 minute time slice of typical sample runs for the two experimental cases implemented with and without the use of the proposed framework.

Figure 3.19 shows the measured round trip time delay trace during the task execution for both the trial runs. It can be seen that the RTT delay is significantly higher in the case of the single path. The proposed multiple paths via overlay networks approach significantly reduces the end-to-end latency in the control loop which results in efficient task execution.

Table 3.4 lists the various statistical quantities of interest for the various experiments conducted using the proposed approach. The mean RTT delay decreases as a function of the number of paths used. Further the variance of the RTT delay decreases quite rapidly with the increase in the number of paths used for the supermedia transport. This decrease in the variance of the RTT with the number of paths used is very significant as it improves the reliability and performance of the communication

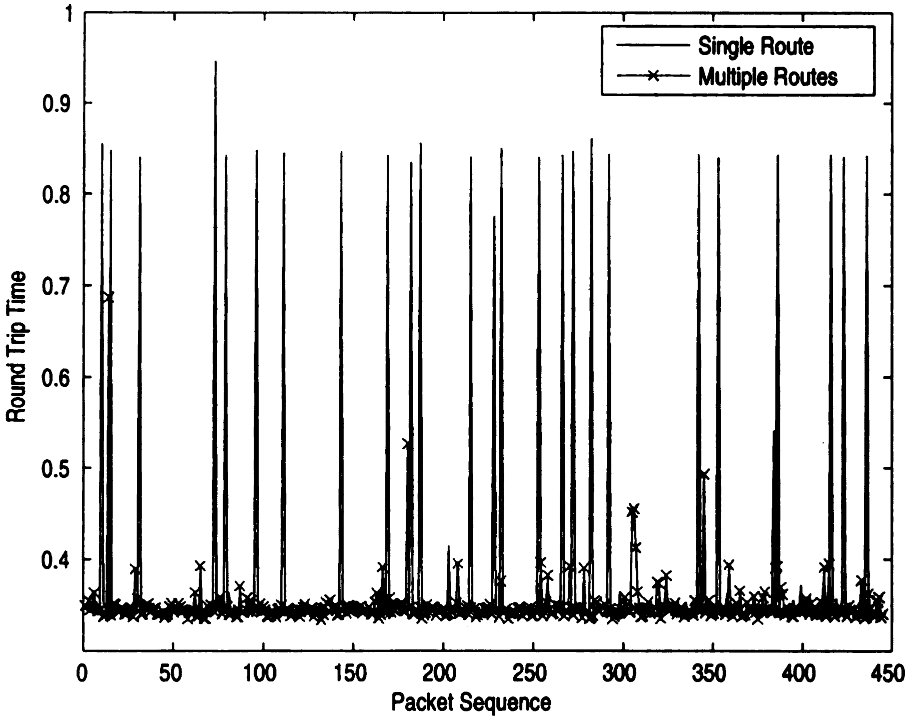


Figure 3.19. Round Trip Time (RTT) Delay with Drops.

channel used. The mean time per event is an indication of the overall efficiency of the system. A lower mean time per event indicates the system performs the task more efficiently. These results reinforce the idea that by using multiple overlay paths to provide the supermedia transport service, the data receiver will always take the data packet that arrives first from the fastest path, thus the round trip time of the packets is reduced. It is typical that network congestion or router capacity overflows occur in a restricted geographical area. The multiple paths provide a convenient and resilient way to circumvent the local networking performance degradation without introducing much overhead. Using multiple overlay paths does increase the networking traffic e.g., four overlay paths increase the networking traffic from about 1KBPS to 5KBPS when transmitting the latency critical data. However, this increase is not at all significant when compared to the streaming rates of audio and video data which are close to 500KBPS. What's more, the TFRC protocol in the STRON approach ensures the supermedia traffic is friendly to other traffic.

Table 3.4. Statistical Data For Experiments Conducted

No. of Paths	No. of Events	RTT (mean) (sec)	RTT (variance)	Execution time (sec)	Mean time per event (sec/event)
1	4014	0.4488	0.7256	1857.371	0.4627
2	4289	0.3949	0.1345	1789.598	0.4173
3	4194	0.3742	0.0473	1598.912	0.3811
5	5549	0.3580	0.0052	2047.581	0.3690

Table 3.5 indicates the percentage of time a particular path has the lowest latency. It shows the contribution of the individual paths to the overall supermedia transport system. It can be seen that the default route (Path #1) is the most used route. This is expected, however in the event of network congestion in the default path, various

other routes can be used to maintain low latency.

Table 3.5. Routing Percentage per Path

Number of Paths	Path #1 (%)	Path #2 (%)	Path #3 (%)	Path #4 (%)	Path #5 (%)
1	100	-	-	-	-
2	97.73	2.27	-	-	-
3	96.13	1.89	1.98	-	-
5	55.1	39.9	2.06	1.64	1.30

Experiments with Simulated Packet Loss

Due to the difficulty in capturing the characteristics of a network route in the limited time span of conducting the real-time teleoperation task, a packet loss simulation was also implemented in one of the sets of experiments reported. For this experiment, a 10 % packet loss across all paths in the system was simulated. The loss model is implemented using a stochastic packet loss model which drops a subset of packets according to an exponential processes. Some measurements [194] show that the packet loss events of the Internet may be independent and fit an exponential distribution.

Figure 3.20 shows the RTT delay trace during the task execution for both the trial runs. It can be seen from the figure that packet loss affects RTT delay for single routes more severely than the proposed approach. These results are in tune with the fact that the proposed method is more robust to the congestion and failure of individual network paths.

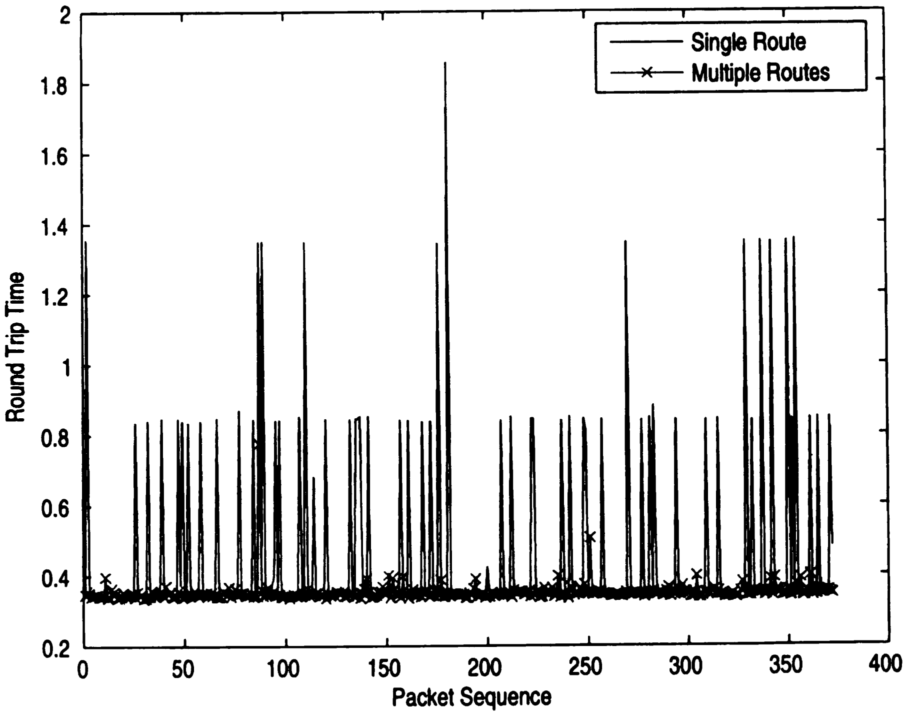


Figure 3.20. The Round Trip Time (RTT) Delay with Drops.

3.4 Summary

Designing a transport service for an Internet based teleoperation system is challenging because of diverse media forms in the application and the timely transmission requirement of supermedia streams. The end-to-end latency of the communication channel is crucial for the teleoperator to send commands to the robot and receive haptic feedback from the robot, both of which are important for the operator to maintain smooth control of the robot. The STRON system is built to reduce the end-to-end latency and improve other QoS parameters for teleoperation systems.

Disjoint overlay network paths and forward error correction encoding are used to improve the reliability and efficiency. An automatic redundancy factor adjustment system can update the redundancy level used to ensure the right amount of redundancy is used to make the transmission reliable. An optimal path selection module can adjust the transmission paths according to the dynamic network conditions.

The results of the simulation demonstrates that the supermedia transport system performs well under different network conditions. In the case of network paths with heavy packet loss rates, it has a significant improvement over TCP and SCTP.

CHAPTER 4

QoS Management of Supermedia Enhanced Teleoperation via Overlay Networks

In supermedia enhanced Internet based teleoperation systems, the data flowing between the operator and the robot include robotic control commands, video, audio, haptic feedback and other media types. The difference between an Internet based teleoperation system and other Internet applications are that (1) there are many media types involved in teleoperation systems and each of them has a particular Quality of Service (QoS) requirement; and (2) some media types are very latency sensitive. Overlay networks have been proposed to improve the QoS of teleoperation applications. However efficient use the overlay network resources and the distribution of these resources optimally to all supermedia streams remains an important problem. This research work aims to provide a framework of QoS management for teleoperation systems over overlay networks. The validity and performance of the system is evaluated using the PlanetLab overlay network. The usage of overlay networks to improve QoS of supermedia enhanced teleoperation systems can be illustrated in Figure 4.1. The experiments show that STRON was effective in reducing end to end latencies of teleoperation systems. However, considering that various supermedia streams have different QoS requirements, and all need to be transmitted over the same set of overlay paths, it is important to design a QoS management framework to assign the networking resource to the supermedia streams efficiently and optimally. This is one of the major problems we are going to solve in this research.

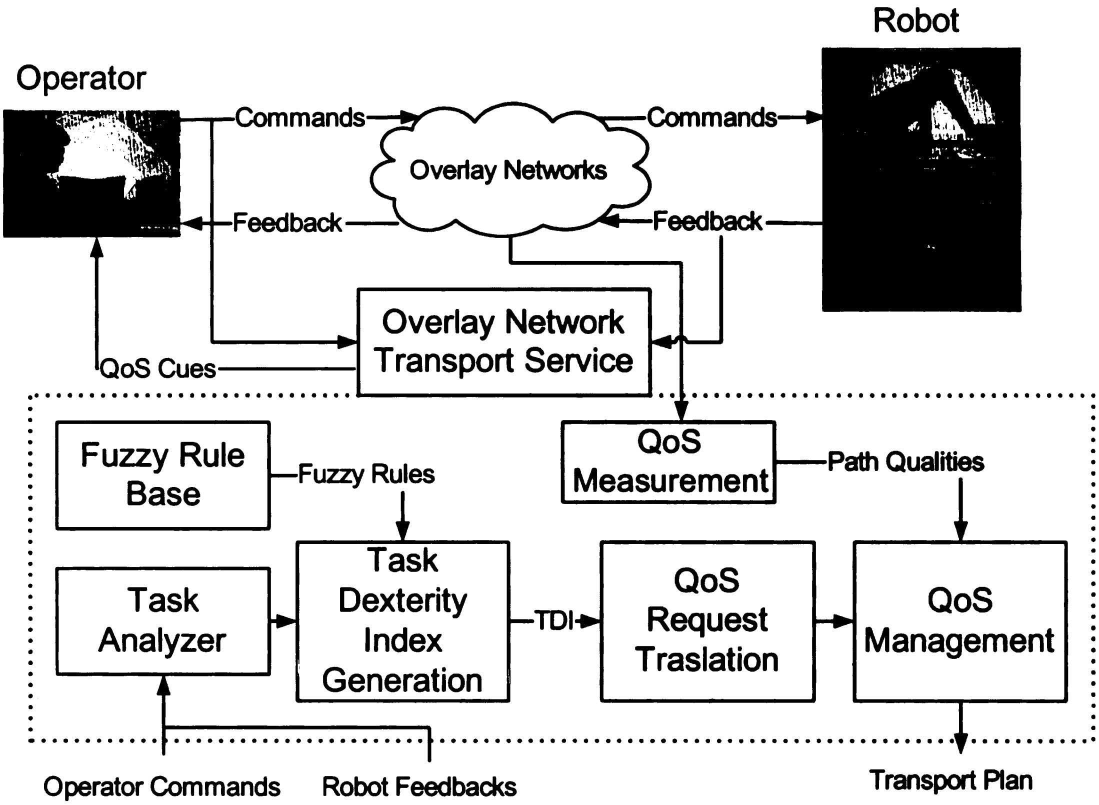


Figure 4.1. Usage of Overlay Networks to Improve the QoS

4.1 Task Dexterity Index Generation

In order for the overlay transport services module to allocate the networking resources optimally, a mechanism is needed to model the dexterity and resource requirement of robotic tasks. This mechanism is task dexterity index. An on-line task dexterity analyzer is proposed to analyze task dexterity requirements by Pan, et al. [133]. The operator commands, the information from the mobile manipulator and the sensory information are taken as inputs and a collection of task indicators is generated online, which represent the actual conditions of the task being performed. Using a fuzzy inference scheme, the task analyzer maps the task conditions represented by task indicators into TDI, which represents task dexterity requirements. The task indicator vector $\mathbf{u} \in R^M$ (set of all task indicators) is defined as:

$$\mathbf{u} = \{u_1, u_2, \cdots, u_M \mid \forall u_i \in U, \quad i = 1 : M\}^T. \tag{4.1}$$

where, M is the number of task indicators. The following is an example of the collection of task indicators which could be used for the tele-operated mobile manipulator:

- End-effector absolute motion on position:

  $$\Delta P_{arm} = \sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2}$$

  where Δx, Δy, Δz are the position transformation on x, y, and z axes in the world frame.

- End-effector absolute motion on orientation:

  $$\begin{aligned} \Delta \Theta_{arm} &= Abs(\arccos 0.5(Tr(R_{err}) - 1)) \\ R_{err} &= R_{des} R_{act}^{'}. \end{aligned}$$

  Here R_{err} is the product of the desired rotation matrix and the transpose of the actual rotation matrix, Tr is the trace of a matrix and *Abs* is the absolute value.

- Local curvature k and torsion τ of the trajectory, which are described by the previous and current commands.

- Operator commands. The characteristics (say acceleration and deceleration) of commands can indicate task conditions and the operator's intention.

- Status of the mobile manipulator (such as end-effector open/close, the external force, *etc.*).

- Sensory information about the environment (say obstacles, working environment temperature, *etc.*).

The Task Dexterity Index is used to mathematically describe the task dexterity requirements. The index is generated by a fuzzy mapping, which takes the task indicators as input variables:

$$Index = f(\mathbf{u}) = \frac{\sum_{j=1}^{M} a_j g_j(u_j)}{\sum_{j=1}^{M} a_j}. \tag{4.2}$$

where a_j is the relative weight of task indicator u_j in the task, and g_j is the mapping function of the task indicator u_j for the task. Equation (4.2) maps the task indicator value into the extent of the task dexterity requirement. The relative weights are assigned based on the physical meaning and importance of a particular task indicator.

4.2 Transport Service Architecture

The system architecture is shown in Figure 4.2. The design details of Sender-side Overlay Agent and Receiver-side Overlay Agent can be found in [25] [30]. At the sender side, the supermedia streams are classified according to their roles in the teleoperation system by a classifying system. Each class of stream has its own QoS requirement. Differentiated treatments of each supermedia class are implemented in the QoS Management Module (QMM).

The Sender-side Overlay Network Agent (SONA) is responsible for transporting each class of streams according to its QoS specifications. The SONA consists of four submodules: the Disjoint Path Search Module (DPSM), the Network Measurement Module (NMM), the Optimal Path Selection Module (OPSM), and the Encoding and Transport Module (ETM).

The Disjoint Path Search Module (DPSM) runs a disjoint overlay network path searching algorithm [42] [18] [122] through the connected overlay nodes. A higher disjoint degree of the overlay paths ensures that if one of the paths experiences congestion or service outage, the performance of other paths will be minimally influenced.

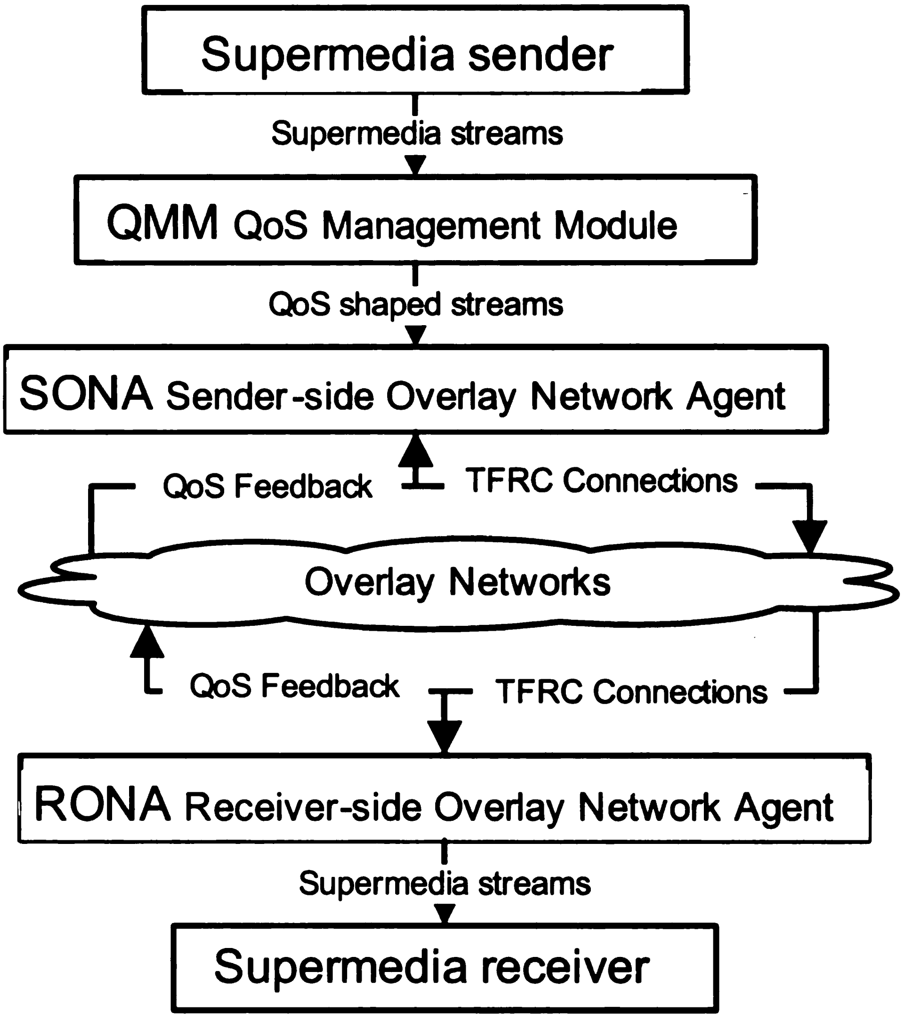


Figure 4.2. The Architecture of the Overlay Network Transport System

The Network Measurement Module (NMM) is usually deployed in overlay nodes as a common service. Active measurements are launched periodically to probe the networking conditions. The measurement results needed by SONA include, the available bandwidth, the single trip delay, and the packet loss rate. Some research efforts have been dedicated to making accurate measurements of the network. In our framework, we use *Pathload* as the available bandwidth measurement tool. *Pathload* [88] takes advantage of the fact that the one-way delays of a periodic packet stream show an increasing trend when the stream rate is larger than the available bandwidth. The measurement of single trip delay and packet loss could be trivial except that half of the round trip time is taken as the single trip delay for simplicity.

The objectives of OPSM are to find the optimal disjoint path set to be used as active transmission paths and decide the amount of data sent over each active path. The redundancy encoding parameters are also decided in OPSM. The design of the OPSM is discussed in detail in [25].

The Encoding and Transport Module (ETM) encodes the supermedia streams using redundancy encoding and passes the encoded packets to the overlay transport service. TCP Friendly Rate Control (TFRC [79]) is used as the transport control mechanisms for each overlay path. TFRC provides a congestion control mechanism that is more suitable to multimedia applications than TCP while at the same time remains TCP friendly.

At the receiver side, the Decoding and Transport Module (DTM) of the Receiver-side Overlay Network Agent (RONA) decodes the packet streams received by the TFRC receivers and notifies the sender when enough packets have been collected for a certain message or a message is lost if a timeout occurs. The feedback to the sender also includes information telling the sender to what degree the redundancy in the encoding is effective so that the sender can adjust the redundancy level accordingly.

In the following section we discuss the design details of the QoS management module and the transport protocol.

4.3 QoS Management Module (QMM)

The available bandwidth B in terms of bytes per second of the sender is reported by the Network Measurement Module of the sender side overlay agent. The purpose of the QoS Management Module is to allocate B for each supermedia stream according to their requirements. For each supermedia stream i ($i = 1, \cdots, M$) we have the following parameters.

- Stream Priority

  For each supermedia stream, a heuristic priority number p_i is calculated and updated dynamically based on the robotic tasks of the teleoperation application. A good way to calculate p_i is using the Task Dexterity Index (TDI) as introduced in [67].

- Stream Weight

  The weight w_i denotes the possible percentage of bandwidth of this stream over the whole bandwidth. w_i is related with the type of stream. For example, a video stream should have more weight than a text stream.

- Stream Floor Rate

  f_i is the minimum rate to make the stream i workable. For some types of media, such as video, if the bandwidth available to this this type of media drops below a certain level, the media will simply not be useful anymore.

- Stream Sending Rate

  The allocated sending rate r_i of each stream is decided by the following algorithm.

  If B satisfies all f_i, that is

$$B \geq \sum_{i=1}^{M} f_i \tag{4.3}$$

The remaining of the bandwidth is allocated according to the weight of the streams.

$$r_i = f_i + w_i \times \frac{B - \sum_{i=1}^{M} f_i}{\sum_{i=1}^{M} w_i} \tag{4.4}$$

If B cannot satisfy all f_i, that is

$$B < \sum_{i=1}^{M} f_i \tag{4.5}$$

The bandwidth is allocated according to the priority up to f_i for each stream. Sort the stream according to the priority in the descending order, we have

$$r_i = \min(f_i, \max(0, B - \sum_{j=1}^{i-1} f_j)) \tag{4.6}$$

Given the target rate r_i we need a packet scheduler to make sure the actual sending rate of each TM stream is r_i. We may use token bucket algorithm [154] to achieve this goal. For each TM stream i we have a token bucket B_i whose size is F_i. One token in the bucket corresponds one byte/second bandwidth the stream can use. The rate of token increase in the bucket is r_i and F_i is a number decided by w_i. F_i should at least be greater than the packet size. The stream can send a packet whose size is X if there are X or more than X tokens in the bucket, otherwise, the stream would have to wait until the bucket has the required number of tokens. Once a packet is sent out for a stream, a certain number of tokens are deducted from the corresponding token bucket. The token bucket algorithm is illustrated in Figure 4.3.

4.4 Experimental Implementation and Simulation

In this section we present the results of using the overlay based QoS framework to accomplish an obstacle avoidance and object pickup task through teleoperation. The number of overlay paths is dynamically adjusted based on the TDI during the teleoperation process. Also network simulator NS2 is used to simulate the behavior of the

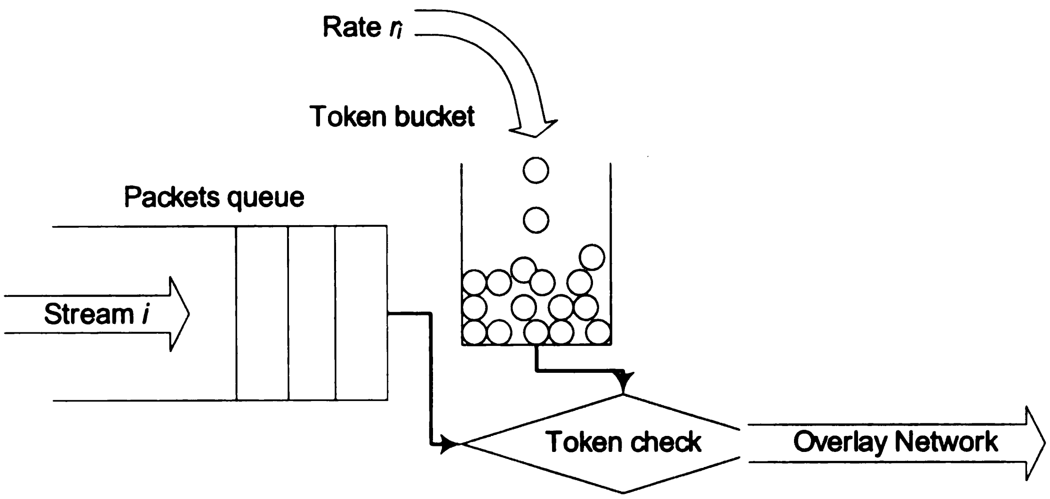


Figure 4.3. The Token Bucket Algorithm

token bucket algorithm in shaping the bandwidth consumption of several supermedia streams.

4.4.1 Experiment Setup

Our teleoperation system mainly consists of two parts: the mobile manipulator (Nomad XR400 and Puma 560) and the haptic device. The devices can be seen in Figure 4.5. Two types of haptic devices are available in our system: Phantom Premium 1.5A and Microsoft SideWinder Force Feedback Pro joystick (with three degree of freedom). Unless noted, Phantom is our default haptic device. In the experiments, the Phantom is used to send position commands to mobile manipulator and render force feedback from the mobile manipulator. Several kinds of sensors were installed in the mobile manipulator. In our experiment, laser sensors have been used to detect the obstacle and produce the TDI. In the proposed approach for reliable and efficient teleoperation, the PlanetLab nodes serve as overlay network relay agents to transmit the control command stream through selected overlay routes in addition to the default IP routes determined by the routing protocols. In the experiment, PlanetLab nodes are also used as relay nodes to simulate that the operator controls the robot in a remote location. A snapshot of one of the experiments (the experiment described in section 4.4.3) is shown in Figure 4.4.

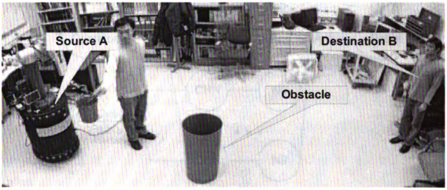


Figure 4.4. Experiment Snapshot

PlanetLab [138] is an overlay network used to design, evaluate and deploy geographically distributed network services. The sender side overlay agent and receiver side overlay agent are implemented and run on a set of preselected PlanetLab nodes. Our experiments can be classified into two categories. The first category of experiments are carried out to verify the operation efficiency of teleoperation tasks using the proposed transport service. In the second category of experiments, the operators are facilitated with QoS cues when performing certain teleoperation tasks, and the results are compared with the case when no QoS cues are present.

4.4.2 Experiments to Measure Operation Efficiency under the Transport Service

In this experiment we compared the operation efficiency of performing certain teleoperation tasks using the traditional transport service and the overlay network transport service (Figure 4.5). In the experiment we recruited a certain number of globally distributed PlanetLab nodes as overlay nodes to form a pool of ten overlay paths. As the TDI changes, from one to six paths out of the ten paths will be used for the teleoperation task. Periodical latency measurements are done over each path in the

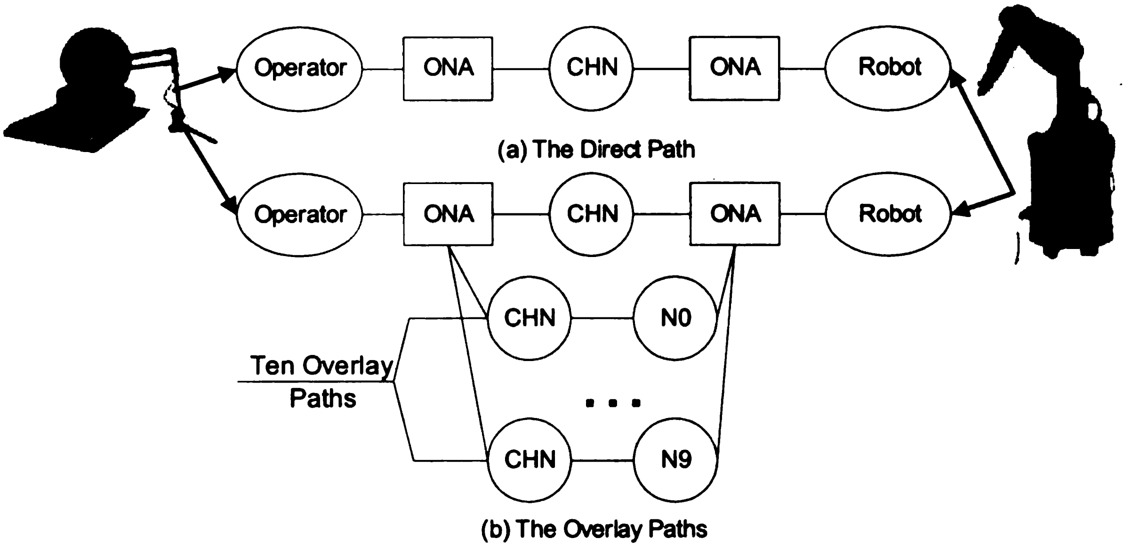


Figure 4.5. The Two Transport Schemes.

pool regardless of whether the path is active or not. All the overlay paths in the pool are ordered according to their QoS performance. This makes sure whenever another active path is needed, the path with the best performance will be chosen. The stretch factor α for the supermedia stream was preset to be equal to the number of paths being used. This will actually duplicate the sending data to each overlay paths, and as a result simplifies the process of redundancy encoding and decoding.

During the experiments, up to six paths will be allocated to the most dextrous tasks. The number six is chosen because according to our previous experiments [25], five or six paths are generally enough to reduce the latency and its variance in most cases. More overlay paths would increase the performance marginally but the gain will be very limited. The same teleoperation task is carried out under two transport schemes (Figure 4.5). Both the robot and the operator are located in the United States. In both schemes an overlay node in Beijing, China (denoted as CHN) is always chosen in the paths. This is to simulate that the teleoperation task is performed between two distant locations (China and USA). In scheme (a), the operator is connected to the robot through a direct Internet path. In scheme (b), ten overlay paths are served as an path pool (established by node $N0$ to $N4$ as in Figure 4.5. At a certain stage during the task, zero up to five overlay paths will be chosen as active overlay paths from the path pool depending on the task dexterity index. Supermedia traffic are transmitted through the default path and the active path pool based on the proposed QoS improvement approach. In the second scheme, ONA (Overlay Network Agent) performs the the overlay transport service operations of SONA (Sender-side Overlay Network Agent) and RONA (Receiver-side Overlay Network Agent), such as duplicating the packets and reconstructing them. The separation of ONA and the sender and receiver makes it possible to adapt to the overlay network service without modifying the sender and receiver software.

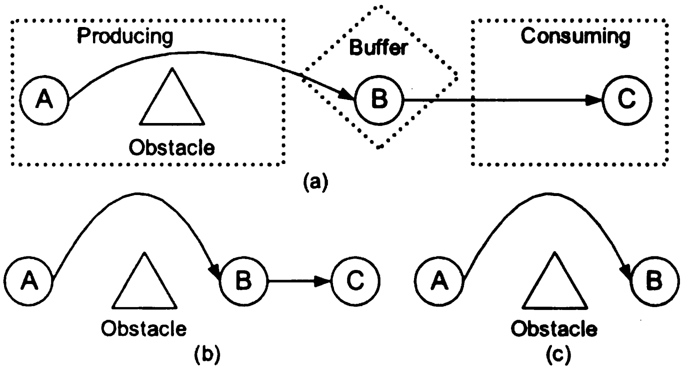


Figure 4.6. Experiment Scenarios.

In the experiment, the operator controls the mobile manipulator to move front point A to point B while avoiding an obstacle in the middle, accept a material piece and then move to point C (as shown in Figure 4.6(b)). The task can be divided into the following five subtasks.

- Subtask 1: move the robot toward the obstacle.
- Subtask 2: avoid the obstacle.
- Subtask 3: move the robot toward point B.
- Subtask 4: accept the object.

- Subtask 5: move the robot toward point C.

During the teleoperation experiment, TDI is produced online to reflect the dexterity of robotic tasks. The experiment is repeated several times for both the overlay paths case and the direct path case.

The RTT (Round Trip Time) comparison of using overlay paths and one direct path can be seen in Figure 4.7. In the overlay paths case, the RTT has an average of 1.04 and a standard deviation of 0.78. The corresponding statistics of the direct path case are 1.52 and 1.47. The results show that the additional overlay paths are very effective in reducing the RTT. More importantly, the variance of RTT is also reduced greatly.

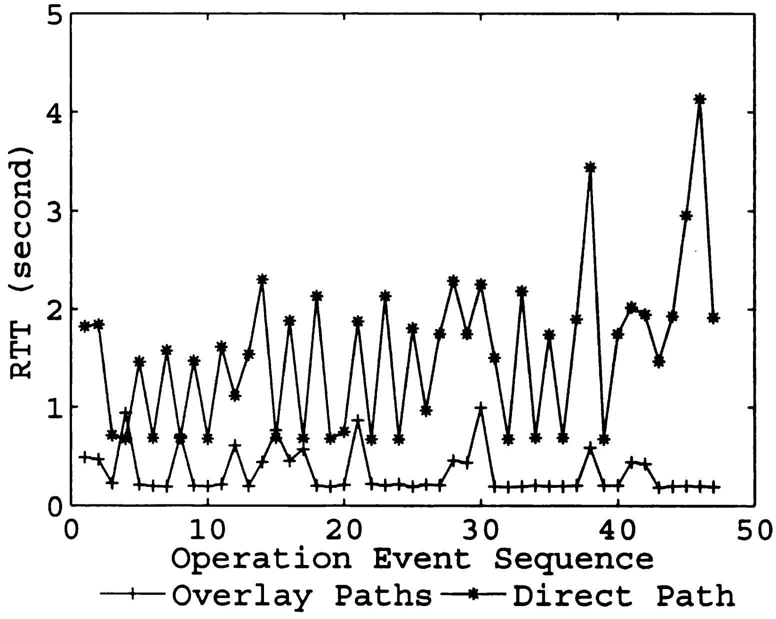


Figure 4.7. RTT of the Two Schemes

The average task finishing time (repeating the experiment 6 times) of the two schemes are shown in Table 4.1.

Figure 4.8 shows the results of RTT, TDI and active number of paths when using the overlay transport service to accomplish the teleoperation task. TDI was generated

Table 4.1. Time Measurements For Experiments Using Different Transport Services (second)

Category	Direct	Overlay	Improvement
Subtask 1	26.19	21.05	24.4%
Subtask 2	30.50	21.89	39.3%
Subtask 3	35.34	29.42	20.1%
Subtask 4	29.83	21.66	37.7%
Subtask 5	13.27	9.41	41.0%
Total	135.12	103.44	30.6%

through the fuzzy logic system taking the distance of the robot to the nearest object as input. TDI shows the dexterity requirement while operating the robot. We can observe for subtask 2 (from event sequence number 24 to 40) and subtask 4 (from event sequence number 72 to 82) the TDI increased to accommodate the task dexterity. As the TDI changes, the number of active overlay paths is automatically adjusted. As a result, the RTT tends to decrease when more active paths are used in the transmission. More obvious is that the variance of RTT is reduced significantly when the number of active paths increase.

In another teleoperation task, the operator controls the mobile manipulator to accept a material piece and transport it to another place. The robot base tracks (one track of each type of experiments) are plotted in Figure 4.9. We can see that the overlay path experiment gives a smoother track than the direct path experiment. The changes of Y distance (one plot of each type of experiments) during the teleoperation process is shown in Figure 4.10. The experiment using overlay paths completes the task earlier than the experiment with the direct path only.

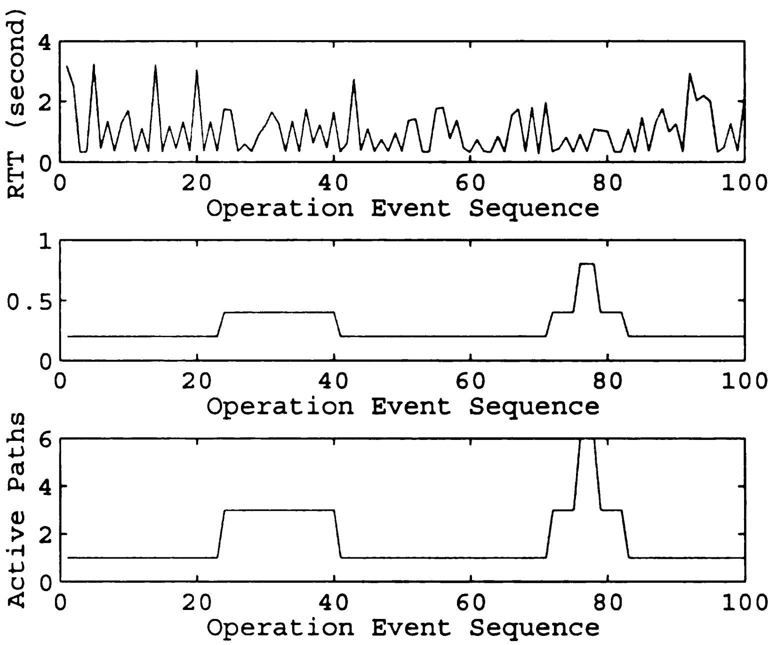


Figure 4.8. The Results of Using Six Paths (including the Default Path)

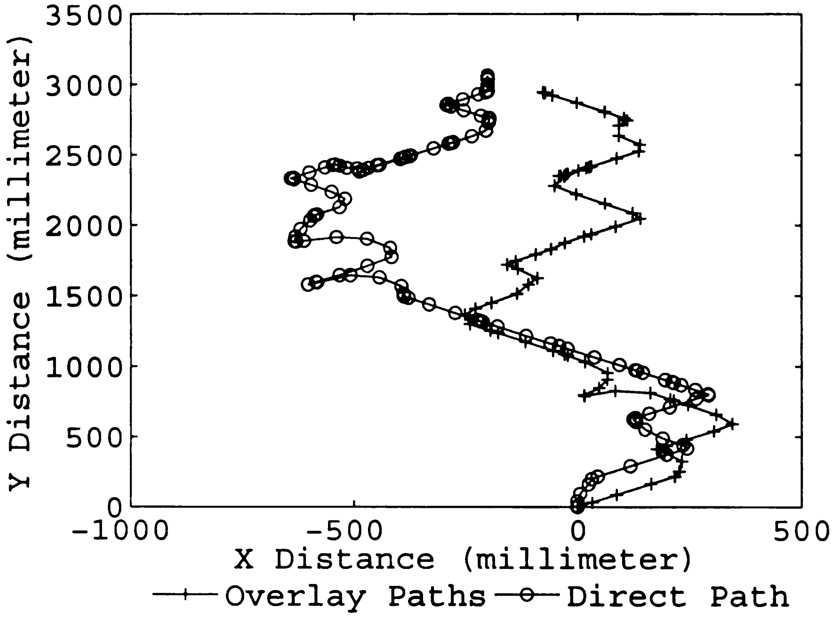


Figure 4.9. Robot Base Tracks

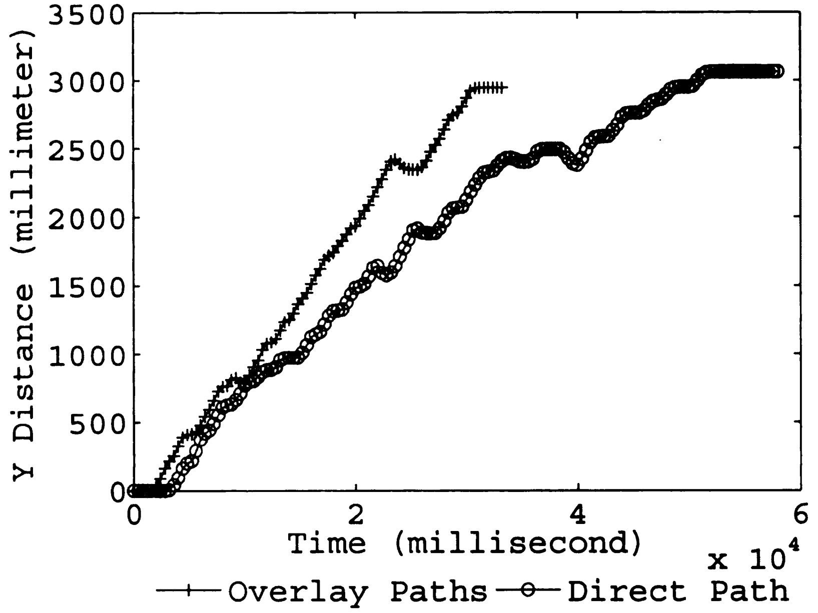


Figure 4.10. Y Distance of the Robot Base Tracks vs Time

4.4.3 Experiments to Measure the Effects of Facilitating the Operator with QoS Cues

In addition to measuring all the available overlay path in the path pool, the Network Management Module also gathers QoS information of the paths in the active transport plan. The QoS information is sent back to the operator in the form of a series of QoS cues. The operator can benefit from the QoS cues directly or through a QoS cue processing fuzzy logic system. Experienced operators may interpret the QoS cues and choose from alternative strategies when planning the tasks, or choose from alternative devices when performing the same task, depending on the quality of the transport service. In the QoS cue processing fuzzy logic system, task planning and implementation strategies may be recommended to the operator based on a fuzzy logic rule base and the current QoS cue values and Task Dexterity Indices (TDI) [67]. A fuzzy logic rule base contains a set of linguistic rules. A linguistic rule $\mathcal{R}_j$ with n

inputs and one output can be written as,

$$\text{IF } X_j \text{ is } \tilde{F}_1^j \text{ AND } \ldots\ X_n \text{ is } \tilde{F}_n^j \text{ THEN } Y \text{ is } \tilde{G}_j$$

where X_i are input variables, $\tilde{F}_1^j$ are antecedent fuzzy sets, Y is output variable and $\tilde{G}_j$ is consequent fuzzy set of rule $\mathcal{R}_j$. Examples of linguistic rules are:

- IF *TDI* is *LOW* AND *BANDWIDTH* is *GOOD*
  THEN *DEVICE* is *JOYSTICK*
- IF *TDI* is *HIGH* AND *BANDWIDTH* is *GOOD*
  THEN *DEVICE* is *PHANTOM*.

The final result can be computed using common fuzzifier, defuzzifier and inference rules [184].

We designed a series of experiments to verify that providing the operator with QoS cues of the network will help improving the operation efficiency of the teleoperation task. One of the challenges is to be able to have similar and predictable QoS conditions (such as congestion) throughout different experiments. In order to have predictable and observable congestion during each experiment, we modified the overlay network agent (ONA) to generate artificial congestion. For the experiments whose results are compared, we make same active artificial congestion intervals during the operation process. The active artificial congestion intervals are short (usually half a minute) and distributed randomly during the operation. During the active artificial congestion phase, the ONA drops the data packets following an exponential distribution with a fixed rate (5% in the experiments). Some measurements [194] show that the packet loss events of the Internet may be independent and fit an exponential distribution.

Two types of experiments were carried out to verify the effectiveness of QoS cues. In the first type, we designed a teleoperation task that consists of several subtasks. The actual order of implementing the subtasks can be adjusted by the operator based on QoS cue observations. In the second type of experiments, the operator is given a choice of different operation devices and the decision of which to use is made based on task requirement and QoS conditions.

Using QoS Cues to Facilitate Task Planning

In this experiment, the operator operates the robot to simulate a classical producer-consumer process. In the producer-consumer problem, the producer produces a certain type of objects and put them in a buffer. The consumer will consume the objects from the buffer. In the teleoperation experiment, the producing-consuming task is decomposed into the following subtasks.

- Subtask 1: simulate the producing phase by operating the robot to transfer an object from point A (the source site) to point B (the buffer) while avoiding an obstacle during the trip (as shown in Figure 4.6(a)).
- Subtask 2: simulate the consuming phase by operating the robot to transfer an object from point B to point C (the destination site).

The final goal of the task is to produce and consume as many objects as possible during a given time, or produce and consume a certain number of objects as fast as possible. Initially we can set the buffer to be nonempty, thus the operator may have the choice of performing either subtask 1 first or subtask 2 first. Since the buffer has a certain size and can hold several objects, the operator also has the choice of performing either subtask during the process unless the buffer is drained or full. The producing phase (subtask 1) is more difficult than the consuming phase (subtask 2). The operator may utilize QoS cues in planning the operating strategy.

In the first experiment setup, the operator is asked to produce and consume five objects. The buffer has a size of three and initially has one object. The experiments are repeated many times for the following three scenarios:

1. There is no artificial congestion.
2. The ONA generates artificial congestion but the operator is not notified with the QoS cues.
3. The ONA generates artificial congestion and the operator is notified with the QoS cues.

The time used to produce and consume five objects for the three scenarios are 270.5 seconds, 346.9 seconds and 298.7 seconds. With the help of QoS cues, the operator can improve the completion time by 13.9% under the same networking conditions.

In the second experiment setup, the operator is given ten minutes to produce and consume as many as possible objects. Under the three scenarios, the number of objects produced and consumed are 12, 9, and 11. The number of objects achieved under artificial congestion with the help of QoS cues is almost as good as the case when no artificial congestion are generated. We can infer from the experiments that QoS cues can alleviate the impact of bad networking conditions to a certain extent.

Using QoS Cues to Choose Operation Devices

In this experiment, the operator operates the mobile manipulator to move from point A to point B while avoiding an obstacle in the trip (as shown in Figure 4.6(c)). A snapshot of the experiment is also shown in Figure 4.4. However, instead of using Phantom as the only teleoperation device, the operator can also use an additional Microsoft force feedback joystick to perform the operation. The joystick can issue velocity commands and is more suitable when the robot needs to travel long distance. However when the network condition is not good enough the risk of mal-manipulation when using the joystick is higher than using Phantom. Also Phantom can perform better than the joystick when doing fine manipulation movements since Phantom has more degrees of freedom and finer control. Facilitated with QoS cues, the operator can choose with operating device to use depending on the task and the network conditions. For example, if the network condition is good and the robot is not near to an obstacle, the operator may choose to use the joystick.

The experiments are repeated many times for the following three scenarios (all with artificial congestion):

1. The operator can only use Phantom.
2. The operator can only use the joystick.
3. The operator can use both Phantom and the joystick but is not facilitated with the QoS cues.
4. The operator can use both Phantom and the joystick and is facilitated with the QoS cues.

For the sake of time keeping we break the task into the following subtasks.

- Subtask 1: move the robot toward the obstacle.
- Subtask 2: avoid the obstacle.
- Subtask 3: move the robot toward point B (the destination).

Table 4.2. Time Measurements For Experiments Using Different Devices (second)

Category	Scen. 1	Scen. 2	Scen. 3	Scen. 4
Subtask 1	24.22	30.21	28.54	19.38
Subtask 2	64.51	71.35	61.98	50.12
Subtask 3	53.11	32.17	36.25	32.61
Total	141.84	133.73	126.77	102.11

The average task finishing time (repeating the experiment 6 times) of the four scenarios are shown in Table 4.2. In Scenario 4, the operator was able to choose between Phantom and the joystick based on QoS cues. This scenario has the shortest completion time, which has an improvement of 28% compared with scenario 1. Generally the joystick achieves better control when the robot is moved over relatively long distances but not as good as Phantom when doing fine manipulations, such as avoiding an obstacle. This can be observed through the fact that subtask 3 in scenario 2 took the shortest time but subtask 2 in scenario 2 took the longest time. With the help of QoS cues the operator can choose the right control device that fits the task best and improve the operation efficiency.

4.4.4 Simulation of the QoS Management Module

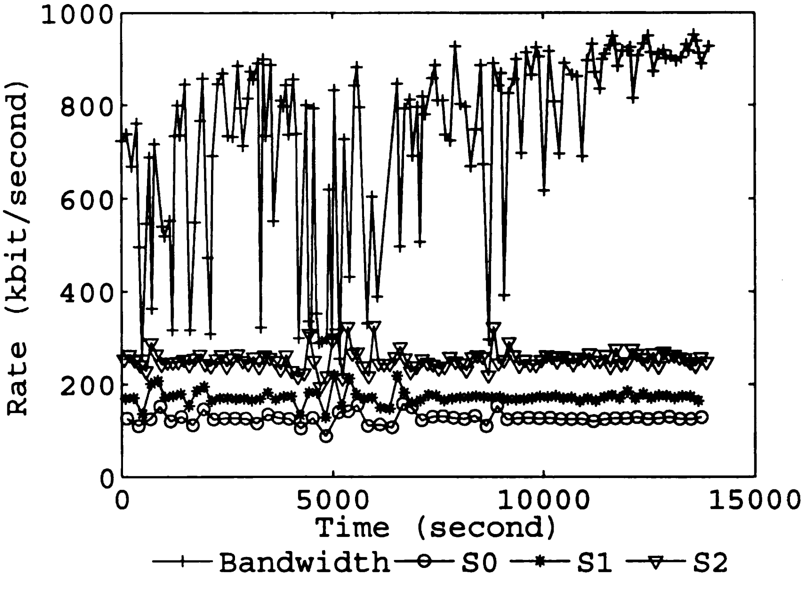


Figure 4.11. Effects of the Token Bucket Algorithm

In the overlay network based transport service framework, the QoS management module is to allocate the network resource to different supermedia streams (as detailed in section 4.3. To reflect the resource variance of the real network, available bandwidth traces were collected between two remote PlanetLab nodes using *Pathload* [88]. The bandwidth trace is further normalized to meet the simulation needs. The simplex-link class in NS2 is modified to enable the bandwidth input.

In the simulation scenario, three supermedia streams $s0$, $s1$, and $s2$ are scheduled to be transmitted. The weights of the three streams are 2, 3, and 5. The priorities of the three streams descend in the order and the floor rates of the streams are 10, 20, 30 KB accordingly. We can see from the simulation results (Figure 4.11) that the token bucket algorithm is effective in partitioning the available bandwidth according to the QoS specifications of each supermedia stream.

4.5 Summary

A QoS Management framework for supermedia enhanced teleoperation systems is presented. This framework aims to provide an efficient QoS control mechanism to allocate networking resources for different supermedia streams. Overlay networks are used to reduce the end to end latency of teleoperation applications and hence improve the operation efficiency and stability.

The architecture of the transport service based on overlay networks is described. The architecture supports dynamically allocate overlay paths based on the task dexterity index generated according to the dexterity of the robotic task. A QoS management model is designed using the token bucket algorithm to ensure the networking resources are allocated efficiently over multiple supermedia streams. The state transitions of the transport protocol are also discussed.

Experiments using multiple overlay nodes on PlanetLab show that the overlay network is effective in reducing the round trip time and its jitter when the dexterity of the task demands a higher quality of service. Simulations are also done to verify the effectiveness of the bandwidth allocation module of the QoS management framework. The results show that the token bucket algorithm can efficiently partition the bandwidth according to the requirement of each stream.

CHAPTER 5

Remote Sensing and Control in MANETs

In remote sensing and control applications, such as bilateral teleoperated systems, the signal transmissions between an operator and the slave manipulators have different QoS requirements in comparison to traditional network traffic. The challenge for supporting teleoperated systems over wireless networks is more difficult in comparison to wired networks. The media streams involved in teleoperated systems differentiate themselves from other media types in that they require both reliable and smooth delivery. Reliable delivery requires that the transport service have TCP style semantics. By being smooth, the transport service should be able to deliver the control and sensing data with both the average latency and the standard deviation of the latency bounded and reduced. We conducted numerous teleoperated experiments using our system. Our teleoperation experiments and simulations show that traditional transport services have great difficulty meeting the latency requirements of delivering sensing and control data over wireless networks. This research work proposes an improved Transport service for Remote Sensing and Control (TRSC). The service reduces the end-to-end latency and the standard deviation of the latency (jitter) for real-time reliable media in mobile ad hoc networks by using forward error correction encoding and multiple network paths. Theoretical modeling and simulation using NS2 shows the approach performs well under different wireless scenarios.

Mobile Ad hoc Networks (MANETS) are becoming a supporting framework for applications that enable people to acquire and disseminate information freely. Remote control and sensing are another class of applications that may find their way to wireless networks. A typical application that uses remote control and sensing is a bilateral teleoperation system. An IP based network is an ideal candidate for the communication channel between the operator and the robot due to its flexibility and interoperability. IP based MANETs further make a teleoperation system more resilient and adaptive to different environments, such as battle fields or emergency rescue scenarios.

A typical MANET based teleoperation system may be illustrated as in Figure 5.1. A multi-functional robot is controlled by a human operator through the ad hoc wireless network. The operator issues control commands to the robot and receives feedback information from the robot. The feedback information includes audio, video, haptic and temperature, etc. The working environment of the robot may be deployed with a number of sensors, which may be video cameras, recorders, infrared sensors, laser sensors, temperature sensors, etc. In a mobile surveillance network, mobile computer nodes equipped with multiple functional sensors may recognize, characterize and track certain objects. Information of the target objects in the form of image, binary data and even audio and video need to be exchanged. The mobile nodes also need to coordinate the tracking movements in order for the target to be covered by at least one of the nodes. The information exchange is also time sensitive depending on the movement features of the target.

The media streams involved in these applications may be classified into two groups. The first group, which includes video and audio, features traditional multimedia streams. The second group, which includes control commands and binary feedback (force, temperature etc.), is described as *Real-time Reliable Media* or RRM.

RRM differs from traditional non-real-time reliable data. The typical non-real-time reliable data are FTP or HTTP traffic supported by TCP. TCP serves the

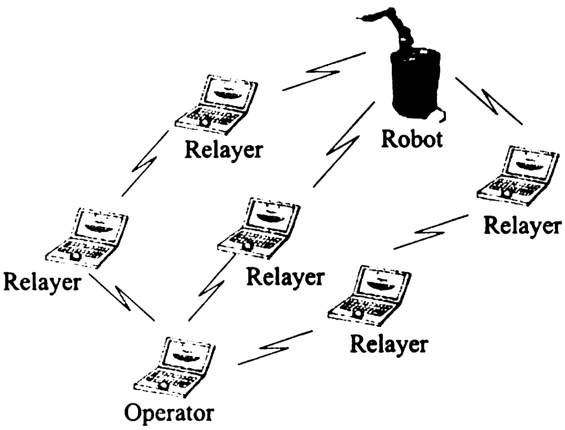


Figure 5.1. An Teleoperation System Over Mobile Ad hoc Networks

non-real-time reliable transport service well with a conservative congestion control scheme. However, the AIMD (Additive Increase and Multiplicative Decrease) nature of TCP causes undesirable jitter for RRM applications in the face of congestion. The segment based acknowledgment scheme may also increase end-to-end latency, which may impact the performance of RRM applications. Under wireless environments, the impact will become worse.

UDP and its variant transport services, such as TCP Friendly Rate Control (TFRC [79, 65, 132]), Datagram Congestion Control Protocol (DCCP [101, 100]) and Stream Control Transmission Protocol (SCTP [19, 23, 85, 165]), are designed for multimedia transport. However, each has difficulty providing reliable transport services, which is crucial to RRM. RTP and RTCP provide support for real time application controls but they lack the congestion control and error recovery mechanisms that are essential for RRM applications.

We conducted a series of teleoperation experiments using wired networks. These experiments gave us insight of the QoS requirements of the RRM streams. In our experiments, we controlled a mobile robot based upon Nomad XR400 and Puma 560 controllers using the Phantom desktop haptic device. Using our teleoperated system, if the end-to-end latency standard deviation becomes larger than 0.3 second, the operator would have difficulty maintaining smooth control of the slave manipulator. However, in an ad hoc network, the latency variance may be much larger under TCP. Using NS2, we simulated a MANET of 50 mobile nodes equipped with Orinoco 802.11b network cards with data rate of 11 Mbit/sec and a transmission range of 60 *m*, which are scattered across a 400 *m* × 400 *m* rectangular area. A modified version of the Random Waypoint Model [197] was used to determine the initial position and the moving patterns of these mobile nodes, with one second pause time and 4 m/s as the maximal moving speed. A series of 1000 *byte* packets were sent from a selected sender to a certain receiver to test the end-to-end latency variance. The result is that the end-to-end latency standard deviation may be up to 2.5 seconds in the simulated scenario. This is far beyond the acceptable range of a teleoperation system.

5.1 System Architecture

The improved transport service for RRM applications takes advantage of multiple disjoint ad hoc routes and FEC encoding to improve the quality of service. Current ad hoc routing protocols are extended to support finding disjoint routes among the ad hoc nodes. Digital fountain [21] encoding is used to reduce the necessities of retransmissions of current transport protocols. The encoding redundancy level is adjusted automatically according to the network conditions and the QoS requirement of the application. The transport protocol is built over TFRC to ensure its friendliness with other networking applications. A measurement module provides the networking parameters needed for the application.

The system consists of the following modules (Figure 5.2).

- Multi-path Routing Module (MRM). Responsible for finding multiple paths between the source and the destination. The paths are preferably node disjoint.
- Network Measurement Module (NMM). Responsible for measuring and report-

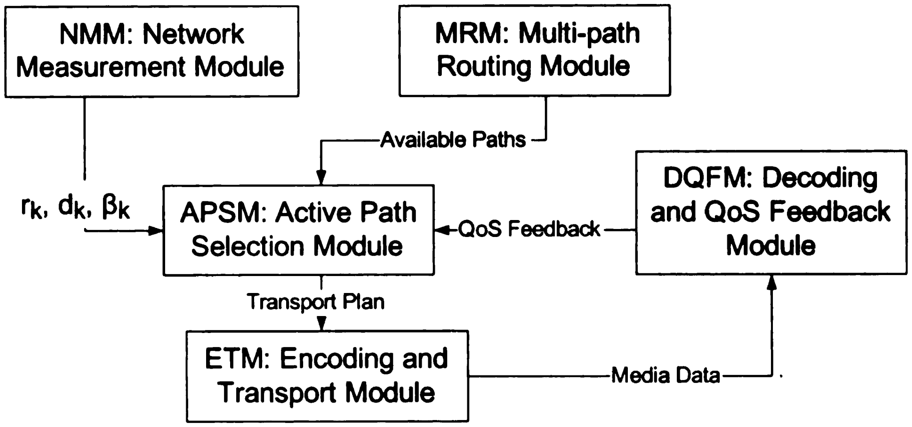

NMM: Network Measurement Module
MRM: Multi-path Routing Module
Available Paths
r_k, d_k, β_k
APSM: Active Path Selection Module
QoS Feedback
DQFM: Decoding and QoS Feedback Module
Transport Plan
ETM: Encoding and Transport Module
Media Data

Figure 5.2. Architecture of the Reliable Transport Service over Ad hoc Networks

ing the QoS parameters of the paths.

- Active Path Selection Module (APSM). Responsible for selecting the optimal path set that will be used in the transmission. APSM also decides the amount of data sent over each active path and the stretch factor according to the networking conditions of the chosen active paths.

- Encoding and Transport Module (ETM). Encodes the transport messages and transmit the messages using TFRC connections.

- Decoding and QoS Feedback Module (DQFM). Resides at the receiver and is responsible for decoding and QoS feedback.

The transport layer message for each media stream is chopped into packets with a certain size (such as the Maximum Transfer Unit or MTU) determined by the networking layer. Assuming the message for a certain stream contains p packets, ETM encodes the p packets into αp packets, where α ($\alpha \geq 1$) is referred as the stretch factor. The encoded data packets are scheduled to be transmitted over multiple paths. As soon as the receiver collects $(1 + \epsilon)p$ distinctive encoded data packets, the decoding algorithm can reconstruct the original message. Here ϵ is referred as the reception overhead. The reception overhead is zero for some encoding algorithms and a small number for others. A class of erasure codes that has this property is called a digital fountain code [21]. By using a digital fountain code, the transport protocol can provide a rather reliable transport service without using acknowledgments and retransmissions. The most common digital fountain codes are Reed-Solomon codes and Tornado codes [21]. Simulation using Reed-Solomon codes shows the encoding and decoding process does not introduce significant overhead under common computing environments (Section 5.3). Moreover, the Tornado codes can decrease the encoding and decoding time by over 100 times [21]. Tornado codes are not used in our simulation due to their proprietary nature.

5.1.1 Multi-path Routing Module (MRM)

In order to provide a reliable and efficient transport service for RRM applications, multiple paths may be used in the ad hoc network. Compared with single path routes, multiple paths may provide fault tolerance when one routing path experiences performance degradation. By splitting the traffic over different paths, QoS parameters such as throughput, delay, and performance jitter may be improved. Moreover, to ensure fault tolerance and reduce media contention, it is desirable for the multiple paths between the sender and receiver to be disjoint. Paths can be disjoint in two senses: node-disjoint and link-disjoint. Node-disjoint paths are more desirable in that they can be more fault tolerant and contention free.

The issue of finding multiple disjoint paths in an ad hoc network can be challenging. We extend the SMR protocol to choose multiple disjoint paths in the RRM application transport framework.

As in DSR, the sender initializes the path-finding process by sending RREQ messages to its neighbors. Besides the original fields of the DSR RREQ message (the address of the initiator, the address of the target, a sequence number, and a route record), the RREQ message in MRM also contains path measurement parameters, as described in Section 5.1.2. Unlike DSR, intermediate nodes do not keep a route cache and do not reply to RREQ messages. This ensures the destination node receives all the possible routing messages. Also, duplicate RREQ messages are not unconditionally discarded by intermediate nodes. Instead, intermediate nodes forward RREQ messages that are received through different incoming links and whose hop counts are not larger than previously received RREQ messages. Since the source route is appended in each of the RREQ messages, the destination is able to choose a group of disjoint paths according to the quality of service requirement of the application.

5.1.2 Network Measurement Module (NMM)

After MRM has found a group of available disjoint paths, it is important to decide if some paths perform better than others. Thus, we can allocate the data streams over different paths according to their performance. The QoS parameters for each path include available bandwidth r_i, packet loss rate β_i and one-way trip delay d_i.

The available bandwidth of an ad hoc link can be measured using the method presented in [211].

$$r_i = R_{capacity} - \sum_{J \in N(I)} R_{consumed}(J) \tag{5.1}$$

According to equation 5.1, the available bandwidth of one node I could be computed as the ad hoc network capacity minus the total consumed traffic in I's neighbor $N(I)$. The available bandwidth of a path could be computed as the minimal of the available bandwidth of the nodes the path traversed.

For the packet loss rate, we assume that most packet loss events are caused by the physical layer transmission rather than packet dropping by the relay nodes. This assumption is reasonable since the relay nodes are only responsible for forwarding for a small number of nodes in a moderate sized ad hoc network. We assume that each node is capable of Signal-to-Interference plus Noise Ratio (SINR) measurement. If independent Rayleigh fading from symbol-to-symbol within a packet is assumed, we can map the SINR to a symbol error rate through some modulation analysis. For example, if the 2^m-ary symbols are transmitted with 2^m-ary noncoherently decoded orthogonal frequency shift keying, the symbol error rate is [141, 112]:

$$SER = \sum_{l=1}^{2^m-1} \frac{(-1)^{l+1}\binom{2^m-1}{l}}{1+l+lSINR} \tag{5.2}$$

And the frame error rate (FER), or the packet loss rate is given by:

$$FER = 1 - \sum_{i=0}^{(N-K)/2} \binom{N}{i} SER^i (1-SER)^{(N-i)} \tag{5.3}$$

If we assume block fading from packet to packet, similar calculations can be done accordingly. Details can be seen in [141].

If we assume the packet loss events are independent at the hops of one path, we can have the packet loss rate of a path with k hops given by:

$$\beta_i = 1 - \prod_{i=1}^{k} (1 - FER_i) \tag{5.4}$$

In most cases, the one-way trip delay d_i of path i may be taken as half of the round trip delay of the path. The measurement of the round trip delay of a path is straightforward. One possible way might be to piggyback the measurement information to the routing messages. Other measurement information may be passed back to the sender through the feedback mechanism, which will be discussed in the following sections.

5.1.3 Active Path Selection Module (APSM)

The active path selection module selects some of the disjoint paths found in the MRM according to the QoS parameters of these paths found in the NMM. The selected paths serve as the active transmission paths for the RRM application.

The problem of optimal path selection may be formulated as follows. Suppose N paths were found in MRM. For each path k ($k \in [1, N]$), the following parameters are given: the one-way trip delay d_k in terms of seconds, the average throughput r_k in terms of bytes/second, and the packet loss rate β_k. Each media stream is chopped into messages as a variable or constant bit rate. The size of the message, which is the amount of effective data we need to transmit from the sender to the receiver, is E in terms of bytes.

We need to choose M paths out of the N given paths to minimize the latency between the sender and receiver. For each path i ($i \in [1, M]$) of the M path, we also need to calculate V_i, which is the amount of data injected into this path by the sender. The system redundancy coefficient γ shows how much redundancy the system has over unexpected packet loss. The actual active path selection algorithm is the same as Section 3.1.2 so it will not be repeated here.

The redundancy coefficient γ is a user defined parameter specifying the amount of redundancy to be used in the system. The value of γ influences the efficiency and reliability of the system and should be adjusted dynamically when the quality of the overlay paths changes. The larger γ is, the more reliable the transport service is. However, a larger γ requires more encoding and decoding overhead. In the system implementation, γ is initialized to an empirically determined value and adjusted through the QoS feedback process (Section 5.1.5).

5.1.4 Encoding and Transport Module (ETM)

The encoding module encodes the supermedia streams that need to be transmitted using forward error correction to ensure the quality of service. The encoded packets are allocated to the selected path and transmitted using an end-to-end transport service. One transport service candidate is TFRC since it is designed to transmit multimedia data while remaining friendly to existing TCP traffic.

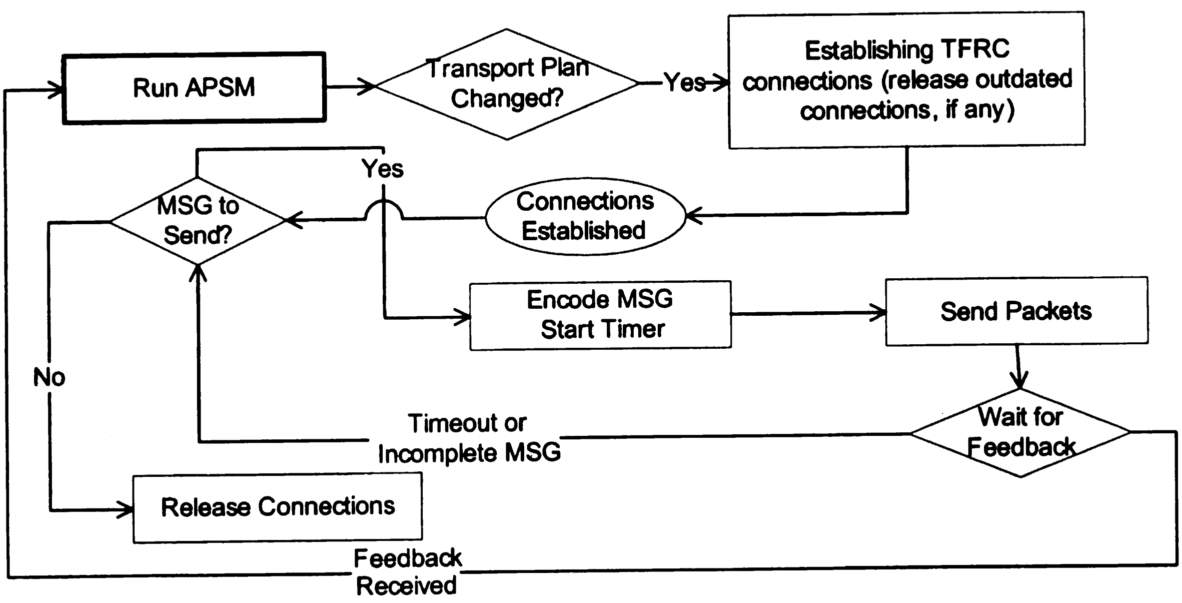


Figure 5.3. The Transport Protocol Design

The state transitions of the sender side transport protocol are shown in Figure 5.3. The active path selection module produces a transport plan using the network measurement information and the feedback. According to this transport plan, TFRC connections are initialized and the messages are encoded using the given redundancy parameters. As the packets are being sent, the sender starts a timer. If the timer timeouts before any feedback is received, the sender stops waiting and tries to send a new message using the old transport plan. The value of the timer may be specified in the form of δt, where t is the speculated transmission time found in the active path selection algorithm, and δ is an adjustable parameter.

The receiver side starts a timer when the first packet of a new message is received. After enough packets were received for the current message (the number of effective packets required to decode the message is specified in the data packet), the receiver decodes the message and sends feedback to the sender. If the timer timeouts before enough packets were received, a negative feedback is sent back to the sender.

5.1.5 Decoding and QoS Feedback Module (DQFM)

The Decoding and QoS Feedback Module collects the data packets from different paths and decodes them. QoS feedbacks that show the effectiveness of the redundancy encoding are sent back to the sender. A timer at the receiver side also starts when the first packet of a message is received. If the timer timeouts and the receiver has not collected enough packets, a negative feedback is sent back to the sender, otherwise a QoS feedback packet is sent back. The QoS feedback includes QoS parameter adjustments made based on the new transmission, including the redundancy coefficient γ. Similar to the sender side timer, the value of the receiver timer may also be specified in the form of δt, where t is the speculated transmission time and δ is an adjustable parameter. However, the sender side feedback timer should be a bit longer than the receiver timer considering the one-way trip delay of the feedback packet.

Since the feedback only contains a small amount of data, no redundancy encoding is needed. The feedback may be sent over the primary path (the path with the best quality) through the TFRC connection of that path. When none of the overlay paths has satisfactory quality, the feedback may also be sent with duplicate copies over several paths. The adjustment of the the redundancy coefficient γ can be done using a threshold based algorithm; the details are presented in [29].

5.2 Analytic Modeling of the System Performance

In this section we model the actual effective data transmitted by each active path. Due to instability and mobility of mobile wireless networks, the effective traffic carried by each path could be larger than the value calculated through the transport plan. We will show in this section that the actual effective traffic of each path will still be bounded by the total effective data.

As discussed in Section 5.1.1, the multiple paths used by the transport system are ideally node disjoint. For a given transport plan that has M active disjoint paths, consider that path j is one of the disjoint paths from the source node S to the destination node D. Let N_i be the ith ($i = 1, 2, ..., k_j$, where k_j is the number of nodes in path j) node in the path. The lifetime of a wireless node will expire when the node cannot be contacted by its neighbors. The node may have moved, been turned off by the user, or consumed its power. We represent the lifetime of wireless node N_i with an exponentially distributed random variable $\mathbb{L}_{N_i}$. Assume that $\mathbb{L}_{N_i}$ are independent and identically distributed exponential random variables, each with mean l. Knowing the expected transmission time t calculated through equation 3.8, we can calculate the probability of failure of each path j during the transmission.

We assume that if the lifetime of one node expires, the path where the node resides will fail. In the following parts we will use blackboard alphabets to denote random variables and the corresponding upper-case alphabets to denote their values. The probability density function of $\mathbb{L}_{N_i}$ is given by

$$f(L) = \frac{1}{l} e^{-\frac{L}{l}} \qquad (L \geq 0). \tag{5.5}$$

The probability that this node fails during the transmission is

$$\begin{aligned} F &= \mathrm{P}\{\mathbb{L}_{N_i} < t\} = \int_0^t f(L) dL \\ &= \int_0^t \frac{1}{l} e^{-\frac{L}{l}} dL = 1 - e^{-\frac{t}{l}} = 1 - e^{-\frac{1}{\tau}}. \end{aligned} \tag{5.6}$$

In equation 5.6, $\tau = \frac{l}{t}$ may be referred as *availability coefficient*, which shows the ratio of node average lifetime versus the time used to transmit a message. The probability that path j fails during the transmission is

$$\bar{F}_j = 1 - (1 - F)^{k_j}. \tag{5.7}$$

To simplify the analysis we assume that the average path length is k, thus equation 5.7 may be written as

$$\bar{F} = 1 - (1 - F)^k = 1 - e^{-\frac{k}{\tau}}. \tag{5.8}$$

Assume that $\bar{\mathbb{M}}$ is the random variable that denotes the number of failed paths during the transmission. The probability that $\bar{\mathbb{M}}$ equals $\bar{M}$ may be given by the following equation, where M is the total number of paths in the transport plan.

$$\mathrm{P}\{\bar{\mathbb{M}} = \bar{M}\} = \binom{M}{\bar{M}} \bar{F}^{\bar{M}} (1 - \bar{F})^{M - \bar{M}} \tag{5.9}$$

The effective traffic that needs to be carried by each unfailing path may be expressed as a random variable $\mathbb{E}$, which is a function of random variable $\bar{\mathbb{M}}$.

$$\mathbb{E}(\bar{\mathbb{M}}) = \frac{E(1 + \epsilon)}{M - \bar{\mathbb{M}}} \tag{5.10}$$

In equation 5.10, E is the amount of effective traffic and ϵ is the reception overhead of the digital fountain code (section 5.1.3).

The expectation of $\mathbb{E}$ is

$$\begin{aligned}
\mathrm{E}[\mathbb{E}] &= \sum_{\bar{M}=0}^{M-1} \mathbb{E}(\bar{\mathbb{M}})\mathrm{P}\{\bar{\mathbb{M}} = \bar{M}\} \\
&= E \sum_{\bar{M}=0}^{M-1} \frac{1+\epsilon}{M-\bar{M}} \binom{M}{\bar{M}} (1 - e^{-\frac{k}{\tau}})^{\bar{M}} e^{-\frac{k}{\tau}(M-\bar{M})}.
\end{aligned} \tag{5.11}$$

Let

$$\zeta = \sum_{\bar{M}=0}^{M-1} \frac{1+\epsilon}{M-\bar{M}} \binom{M}{\bar{M}} (1 - e^{-\frac{k}{\tau}})^{\bar{M}} e^{-\frac{k}{\tau}(M-\bar{M})} \tag{5.12}$$

to be a *partition coefficient* showing the average ratio of effective traffic carried by each path. We have

$$\mathrm{E}[\mathbb{E}] = E\zeta. \tag{5.13}$$

In a wireless environment in which the nodes are relatively stable, that is, the availability coefficient τ is high enough, the partition coefficient ζ should converge to $\frac{1}{M}$. We simulated the change of ζ and the results are plotted in Figure 5.4. In the simulation scenario, the reception overhead ϵ is chosen to be 0 and the average path length k is chosen to be 10. When τ is too small, the network is not stable enough to finish transmitting the message. This situation is handled by retransmission as described in section 5.1.4. Under the settings of the simulation, when τ is over 5, full messages can be received and the value of ζ begins to converge. The converging points of ζ are 0.2 and 0.1, when $M = 5$ and $M = 10$, respectively.

5.3 Simulation Methodology and Results

To evaluate the effectiveness and efficiency of the system, the NS2 network simulator was used to construct a simulation environment for the real-time media transport protocol.

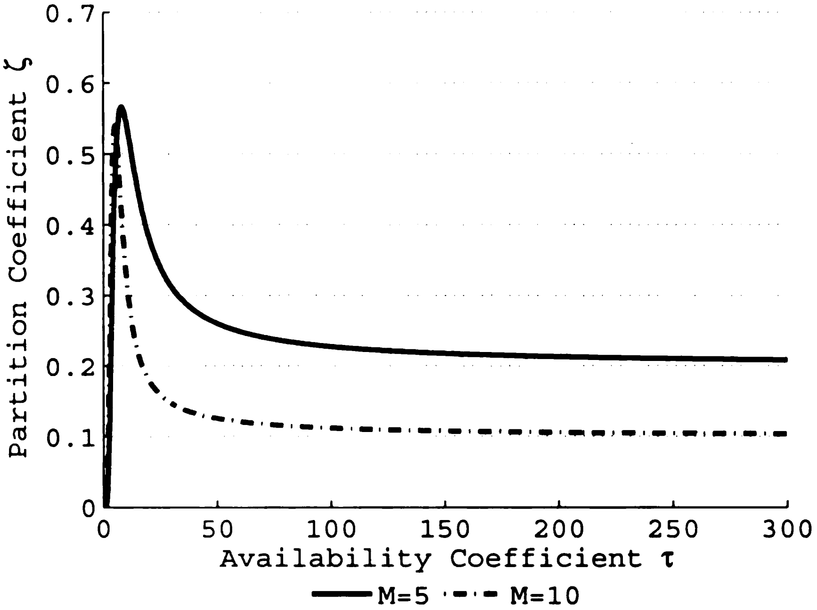


Figure 5.4. The Convergence of ζ

An ad hoc routing protocol was extended to provide two paths between the media sender and receiver. An Active Path Selection Module was implemented in the application layer of the simulator. An implementation of Reed-Solomon codes [15] was used to estimate the encoding and decoding overhead of the algorithm. We did not use Tornado codes in the simulation due to their proprietary nature. The TFRC module in NS2 was modified to support having a payload. One TFRC connection was established over each chosen path in the transport plan. The receiver side of the application collected successfully transmitted packets and carried out a simulated decoding process. In the simulation, the available bandwidth r_i was set to equal to the total link bandwidth since the real-time application was assumed to be the only application running on the network. For the sake of simplicity, the packet loss and the one-way trip delay were set to fixed values without losing generality of the algorithm.

In the mobile network scenario, the mobile nodes were scattered into a $400m \times 400m$ rectangular area. A modified version of Random Waypoint Model [197] was used to determine the initial position and the moving patterns of these mobile nodes. The old Random Waypoint Model was found to fail to provide a steady state in that the average nodal speed consistently decreases over time. In the modified Random Waypoint Model, the minimal speed of the mobile nodes was set to a nonzero value (0.1 meter/second in the simulation) so that a steady state could be reached. Each node moved to a random spot within the area, rested for a certain period and then headed for another random spot, resulting in continuous changes in the topology of the underlying network. Unless otherwise stated, the pause times of the mobile nodes were set to be one second.

Each mobile node was simulated to work with the Orinoco 802.11b network card, which has a data rate of 11 $Mbit/sec$ and a transmission range of 60 m. A series of 50 messages with a size of 1000 bytes were sent from one mobile node to another during the simulation period, which was set to be 100 seconds.

Two types of experiments were carried out on the simulation platform. In both types of simulations, we measured the end-to-end latency of sending the specified number of messages from the sender to the receiver, the standard deviation (STD) of the latency, as well as the goodput ratio of both of the TRSC and TCP cases. Many simulation runs were conducted and the results are averages of the simulations runs. For TRSC, the goodput ratio is the effective traffic divided by the total amount of traffic sent out by the sender, which is encoded using FEC. For TCP, the goodput ratio is the amount of net traffic (which is the same as the effective traffic of TRSC under the same simulation settings) divided by the total TCP traffic sent out by the sender. The total TCP traffic consists of the net traffic and the traffic retransmitted under TCP semantics. When calculating the latency of TRSC, we took into account the encoding and decoding times of the packets. An experiment with the Reed-Solomon codes shows that with stretch factor 1.2, the encoding and decoding of 50KB data takes 0.014 and 0.002 seconds on the experimental platform (an AMD Duron 1.26GHz

CPU with 512MB memory).

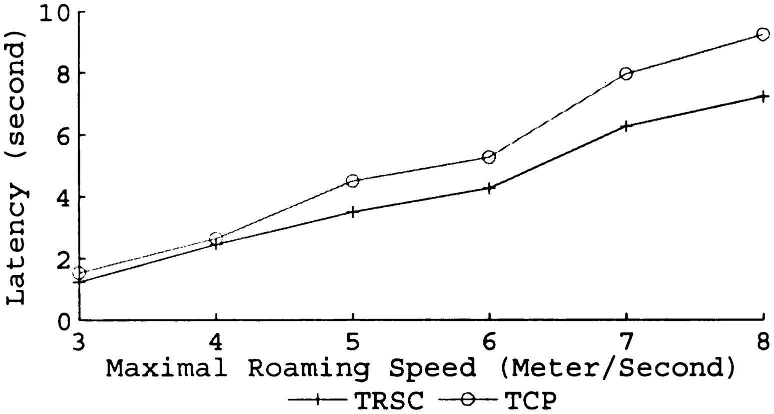


Figure 5.5. Comparison of the Latency of TRSC and TCP under Different Maximal Speeds

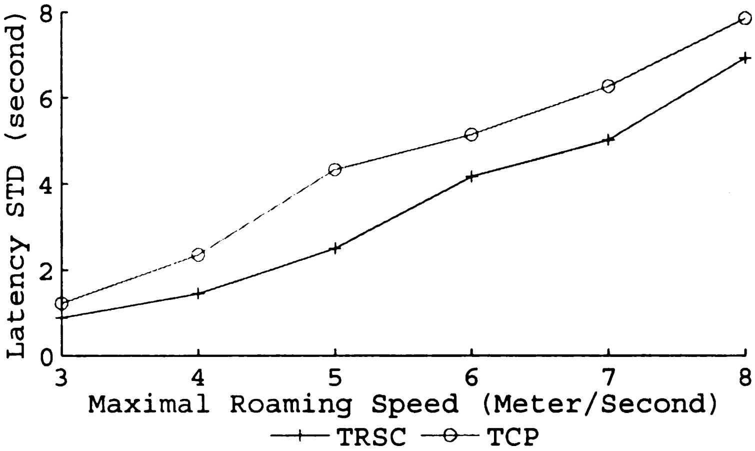


Figure 5.6. Comparison of the Latency Standard Deviation of TRSC and TCP under Different Maximal Speeds

First, we fixed the number of nodes in the simulation and changed the maximal roaming speed of the Random Waypoint Model. In the simulations, the number of nodes were chosen to be 50 and the maximal roaming speed changed from 3 m/s to 8 m/s. The end-to-end latency and latency standard deviation of sending the given number of messages using TRSC and TCP is shown in Figures 5.5 and 5.6. Of all the speed modes, the average latency for TRSC and TCP were 4.16 *s* and 5.19 *s*. TRSC had an improvement of 19.9% compared with TCP. The goodput ratio of TRSC and TCP are shown in Figure 5.7. The averages of the goodput ratio were 91.5% and 93.2%, respectively. We can see that TRSC reduced the latency by 19.9% but only had a 1.7% goodput ratio decrease.

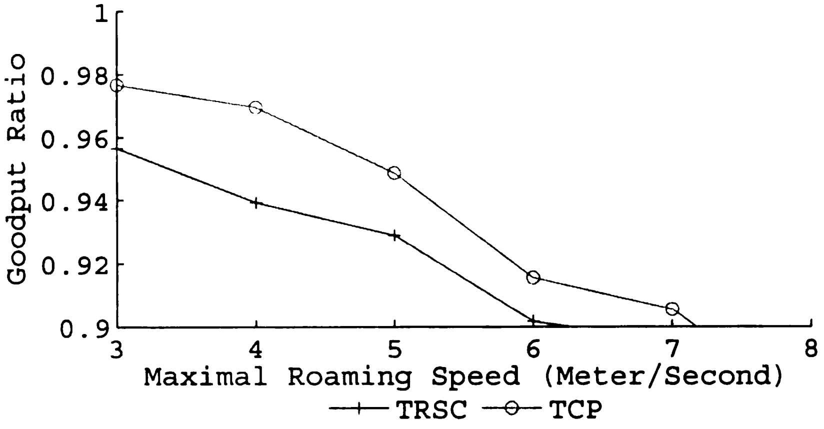


Figure 5.7. Comparison of Goodput Ratio of TRSC and TCP under Different Maximal Speeds

Second, the maximal roaming speed was fixed to 3 *m*/*s* and the number of nodes in the simulation changed from 50 to 100. The end-to-end latency and latency standard deviation of sending the given number of messages using TRSC and TCP are shown in Figures 5.8 and 5.9. Since the sender and receiver were the only traffic generators in the network, an increase in the number of nodes improves the chance of finding a viable and good quality path between the sender and the receiver. Of all the cases with different numbers of nodes, the average latency for TRSC and TCP are 0.87 *s* and 1.30 *s*. TRSC has an improvement of 32.7% compared with TCP. Compared with the case when only the maximal speed changes, TRSC performs even better when the number of nodes is large, in which situation TRSC can take advantage of more quality paths. The goodput ratios of TRSC and TCP are shown in Figure 5.10. The averages of the goodput ratios were 96.2% and 97.9%, respectively. We see that TRSC reduces the latency by 32.7% and has a 1.74% goodput ratio decrease compared with TCP.

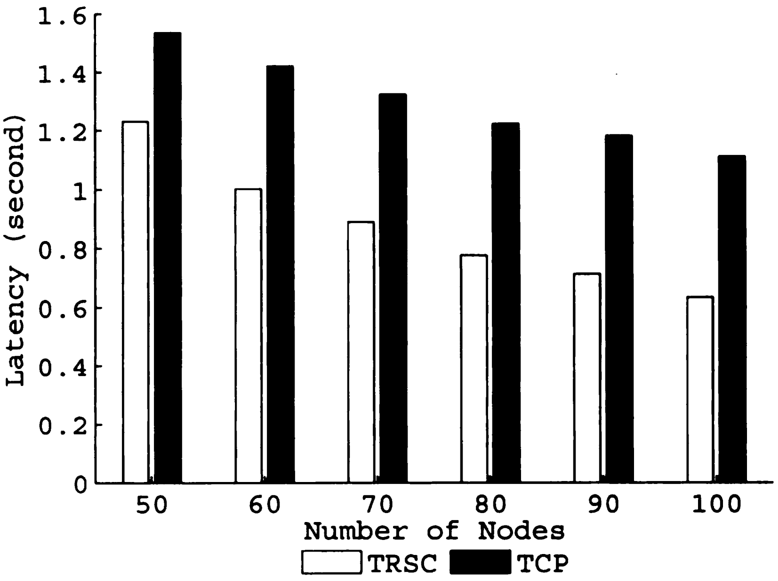


Figure 5.8. Comparison of the Latency of TRSC and TCP with Different Number of Nodes

The maximum number of nodes used in the simulation is 100. We also did some simulations for TRSC and TCP when fewer nodes are involved. It shows that as long as the number of mobile nodes is larger than 10, TRSC has significant advantages over TCP in reducing latency and latency jitter. We can infer that TRSC works better in the scenarios where even more mobile nodes are involved. In MANETs deployed for remote sensing and control applications, the number of nodes are often large. In those circumstances TCP often performs poor in terms of latency and latency variance due to broken routes and frequent retransmissions. TRSC, on the other hand, can reduce the latency and latency variance through the usage of multiple paths routing and forward error correction encodings.

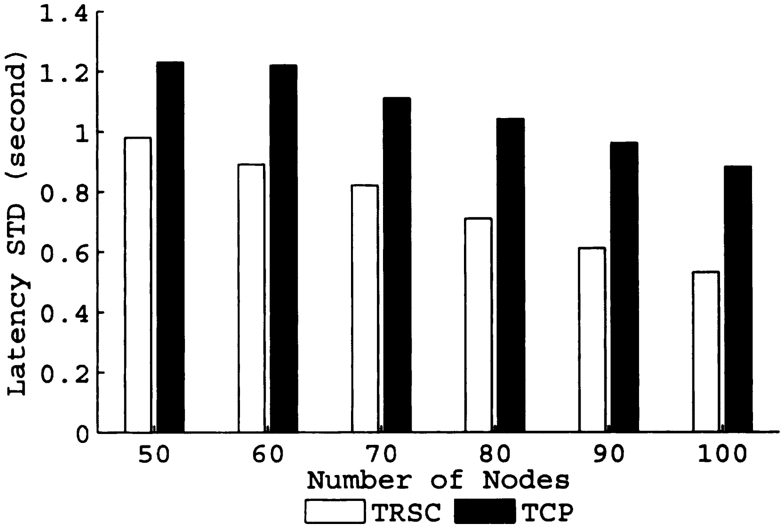


Figure 5.9. Comparison of the Latency Standard Deviation of TRSC and TCP with Different Number of Nodes

5.4 Summary

Real-time wireless applications, such as teleoperation systems, require some media streams to be reliably transmitted between end hosts with the latency to be low and have little jitter. Traditional transport services either provide reliable content based transport services (TCP) or unreliable real time transport services (UDP and its variants). However none works well with real-time reliable media (RRM). This research proposed an improved real-time reliable transport service (TRSC) over mobile ad hoc networks. The proposed approach utilized multiple disjoint paths and forward error correction to reduce the end-to-end latency and the jitter of the latency. Probabilistic approaches are used to modeled the traffic load of each path. Simulation results show that in comparison to TCP, TRSC performs well in reducing end-to-end latency and jitter with only marginal goodput ratio decrease.

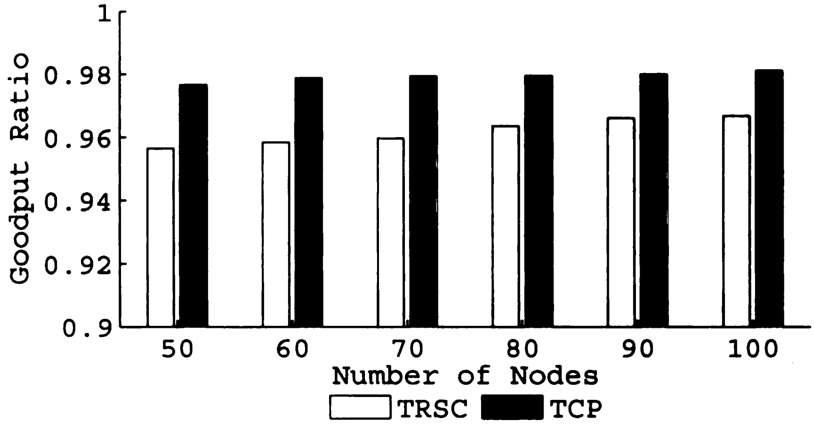


Figure 5.10. Comparison of Goodput Ratio of TRSC and TCP with Different Number of Nodes

CHAPTER 6

QoS Provision for Remote Sensing and Control in Heterogeneous Environments

In a heterogeneous wireless network environment, the network nodes are equipped with both wireless local area network (WLAN) and cellular interfaces. It is desirable to combine the higher bandwidth of a WLAN and the ubiquitous nature of cellular networks to provide the transport service for QoS demanding applications, such as remote sensing and control systems. Common solutions exploit the diversity of cellular channels by recruiting multiple WLAN peers to work as proxies for a MANET node that needs to communicate through the cellular network. One important problem when utilizing channel diversity in a heterogeneous wireless network environment is to balance the trade-off of channel diversity gains with MANET peer contentions. A greater number of proxies will inevitably increase the MANET contention and possibly QoS degrading of the cell, and fewer proxies may not be able to provide sufficient diversity. This research aims to build a dynamic mechanism to determine the appropriate number of proxies to use. The mechanism uses the MAC layer retransmission rate as an indicator of the contention level in MANET. Simulation using cellular channel fading models and prevalent cell scheduling policies indicate the scheme can provide better QoS through channel diversity by using the appropriate number of proxies.

The development of cellular communication technologies makes it possible to deploy ubiquitous applications with higher QoS requirements. However, the data rate and QoS of cellular networks are not comparable to wired networks or wireless local area networks (WLAN). In a hybrid WLAN/cellular environment, the mobile hosts are equipped with both WLAN and cellular interfaces, referred as dual-mode nodes. WLAN technologies such as IEEE 802.11 suite of protocols have limited coverage. However the availability of relatively high bandwidth WLANs can be a good complement to the cellular link. By recruiting WLAN peers to form a mobile ad hoc network (MANET) and share their cellular channels, a mobile node can achieve the level of QoS that can never be met alone.

An instance of an application that could take advantage of dual mode connectivity is a remote sensing and control system [29]. Figure 6.1 shows an example of such a system. The wireless control terminal is equipped with both WLAN and cellular interfaces. The control terminal also has the capabilities of issuing control commands such as pan, tilt or zoom of visual sensors, or moving the base of mobile sensors. Sensory information, such as video and audio, will also be received and rendered by the control terminal. The control commands and sensory feedbacks, including possibly different types of media types, are referred to as supermedia [25]. In this dissertation, the control terminal may also be referred to as a control client or a client.

The communication between the control terminal and the sensors may traverse different types of communication channels. Usually the communication path will include cellular links with the base station (BS), wired networks (such as the Internet) and wired or wireless local area networks, to which the sensors attach. We focus on the communication issues of cellular link. Particularly, video, audio and other sensory feedbacks are transmitted from the BS to the control terminal, referred as the downlink. Control commands are sent from the control terminal to the BS, referred as the uplink. Prevalent cellular networks, such as CDMA2000, exhibit asymmetric bandwidth levels over the uplink and downlink. The downlink channel is usually controllable by the BS, and may have greater bandwidth capabilities. This could be a blessing for remote sensing and applications since the sensory information usually has higher bandwidth demands. Nevertheless, the QoS provided by a single cellular link may still not be able to meet the needs of the application. Moreover, the requirement of high responsiveness and control loop stability of remote sensing and control applications often demands stable channel quality as well as higher bandwidth. This implies the end-to-end latency and latency variance (jitters) are important QoS metrics besides throughput.

The rest of this chapter is organized as follows. In section 6.1, we examine the scheduling policies of prevalent cellular data networks and simulate the possible bandwidth gains when using more than one simultaneous connecting proxies. An adaptive peer management scheme is presented in section 6.2. We simulated the performance of the proxy group control scheme under the NS2 simulator, and the results are presented in section 6.3. Section 6.4 provides the conclusions.

6.1 Model of Bandwidth Sharing in Cellular Networks

Recruiting peer proxies to share their cellular bandwidth involves introducing more active cellular network users. In most cases, all the users will reside in the same cell. As a result, the workload of the cell will increase. Since most cellular data networks are packet switch networks, heavier workload will inevitably influence the amount of bandwidth that each user can get. Therefore it is not straightforward that bandwidth sharing will increase the throughput of the control terminal. However, due to small scale fading of the radio signal, different cellular users receive different data rates even if they share the same large scale physical environments. Cellular bandwidth sharing can take advantage of this fact to ensure the control terminal receives stable QoS even if the radio signal of its single channel experiences the fading effects.

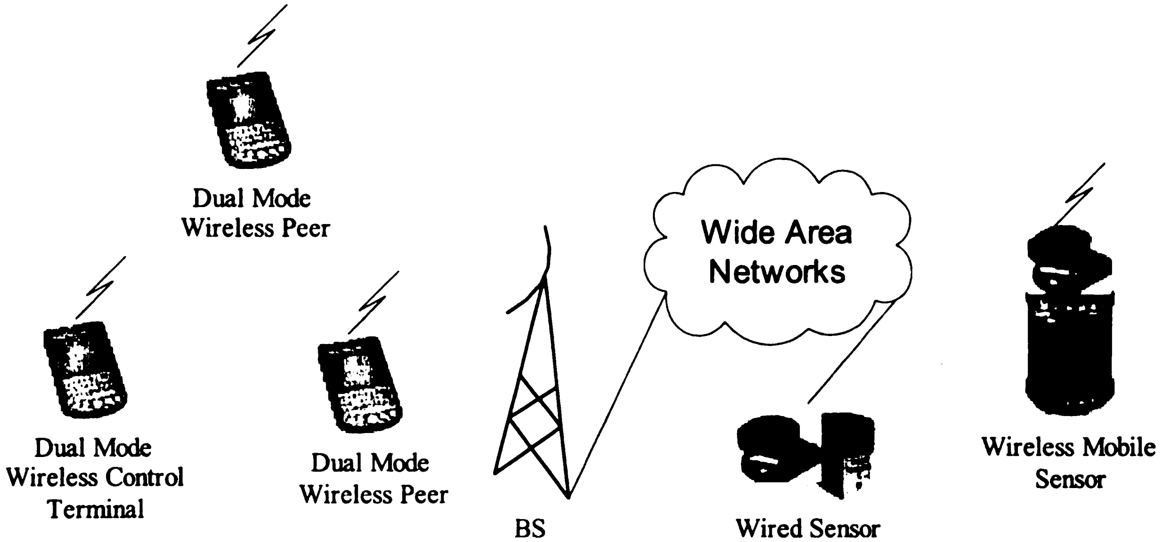


Figure 6.1. A Remote Sensing and Control System in Heterogeneous Environments

Since the radio signal received by the terminal is the sum of many signals received from different directions, the instantaneous received signal strength may fluctuate even if the terminal moves over very small distances. When the receiver moves only a fraction of a wavelength, the received signal power may vary by up to three or four orders of magnitude (30 or 40 dB) [147]. For cellular frequencies in the range of 1 GHz to 2GHz, the effective movements can be less than a meter. Such movements are very common for cellular users and as a result the data rate received by the users will have dramatic change due to small scale fading effects. In the following parts, we simulate the effects of bandwidth sharing in a typical cellular network. We consider a CDMA based cellular data network.

In a cellular network, the forward channel (or downlink channel as referred in some literature), which is from the base station to the user, is often controlled by a scheduler in the base station. The reverse channel, is an asynchronous channel where all users contend for the usage. The relation between the forward rate and power of a mobile node in a CDMA network can be given as [95, 180]:

$$\phi = (\frac{E_b}{P_I})(\frac{R}{SINR}), 0 \leq \phi \leq 1 \tag{6.1}$$

where ϕ is the fraction of the base station power required to support a data rate R, E_b is the received energy per data bit, P_I is the total interference power, and $SINR$ denotes signal to interference plus noise ratio. The quantity E_b/P_I depends on the physical layer frame error rate (FER) and is often maintained below a threshold. Base station power control refers to changing ϕ to maintain a rate R given a certain $SINR$, and rate control refers to changing R given a certain $SINR$ to maintain a fixed ϕ value. It has been proved that in continuous bandwidth conditions, time multiplexing is superior to code multiplexing in CDMA networks [95]. This policy is also adopted in CDMA2000 HDR (High Data Rates, also known as 1xEvolution Data Only or 1xEV-DO) [185]. In CDMA2000 HDR, the scheduler at the base station allocates the whole capacity of the channel to one user at a time at the maximum power, which corresponds to $\phi = 1$. In order to decide which user to allocate the bandwidth to at each time step, proportional fairness scheduling policy is often used [12].

Assume that the channel of a given cell is shared among N users through fixed time slots. The length of one time slot is denoted by τ_l. The time slot length is 1.66 ms in EV-DO and 1.25 ms in EV-DV (Evolution Data and Voice). The input to the scheduler includes a rate vector $R^\tau = (r_1^\tau, \cdots, r_N^\tau)$, which is the achievable data rate r_i^τ for user i during the next time slot τ. R^τ is determined by the channel conditions of each user. For users residing in the same cell, R^τ will be mainly determined by the small scale fading effects of each user channel.

The proportional fairness policy always allocates the channel to the user that has the largest ratio of the next time slot achievable rate over its previously achieved rate. Let k denote the user that is picked for the next time slot, and then it holds

$$\frac{r_k}{A_k} = \max_{j \in \{1, \cdots, N\}} (\frac{r_j}{A_j}) \tag{6.2}$$

where

$$A_j(\tau) = (1 - \alpha)A_j(\tau - 1) + \alpha r_j. \tag{6.3}$$

Equation (6.3) calculates the previously achieved rate $A_j(\tau)$ at time step τ for user j. $\alpha (0 < \alpha < 1)$ is a coefficient to control the influence of history rates on the calculation of the achieved rate.

We can see that the proportional fairness policy favors two types of users:

1. The nodes that have good achievable rates (or channel qualities).

2. The nodes that have smaller previous achieved rates.

Through bandwidth sharing among cellular users, more than one nodes are recruited to serve the current application. Thus the chance of the node (or nodes) that contribute to the current application being scheduled in the next time slot is increased. This takes advantage of the first type of users favored by the scheduling policy. Also compared with being served by a single node, the application is served by a collection of nodes. As a result, the previous achieved rate of each of the collected node will be smaller compared with the case when only one node serves the application. This exploits the second type of users.

We simulated the effects of using multiple proxies in a cellular network under a Gaussian fading channel with 1.2% variance. The simulation consists of a proportional fairness scheduling module and an wireless channel fading simulation model. Following previous literature [185], we used Gaussian fading to model the channel characteristics. Similar results can be obtained by using Rayleigh fading or Rician fading models. For some simulation scenarios, cross traffic generators (CTGs) are introduced. CTGs cannot be recruited as proxies but they will remain as cellular users in the scheduling algorithm. It is assumed that all cellular users require a class of 1 MBPS rate, and the actual achievable rate of each user is determined through the fading channel simulation.

When multiple cellular proxies are available in the simulation, the achieved rate is calculated by averaging the collective rate of those proxies through the simulation time span. For a single client (taken as the case where there is one proxy, which is the client itself) case, the achieved rate is calculated by average the rate of the particular client over the simulation time span.

The changing trend of the achieved cellular rate versus the number of recruited proxies is shown in Figure 6.2. The curves starting with P denote the achieved rate with proxies, and the curves starting with S denote the achieved rate without proxies. S curves are not influenced by the number of recruited proxies and their fluctuation comes from the channel fading effect of different simulation runs. S curves are plotted for comparison purposes. The numbers following each curve legend denote different simulation scenarios. In scenario 1, there are no CTGs. And there are 5, 10, and 15 CTGs in scenarios 2, 3, and 4, respectively. We can see that even a few number of proxies are even quite effective in improving the achieved rates. CTGs impact the performance of both single client case and the case with proxies. An increase in the number of proxies can reduce the impact of CTGs. This can be further illustrated in another set of simulations as shown in Figure 6.3.

As the number of CTGs increases, the achieved rate of the cases with more proxies tend to decrease slower, compared with the cases where there are no or fewer proxies. When no CTG is present, the achieved rates of single client case, the cases with 5 proxies and 20 proxies are 0.917, 0.993, and 0.994 MBPS, respectively. The cases with proxies have an improvement of 8.2% and 8.4% over the single client case, respectively. When 10 CTGs are present, the achieved rates of single client case, the cases with 5 proxies and 20 proxies are 0.119, 0.354, and 0.676 MBPS, respectively. The cases with proxies have an improvement of 198% and 469% over the single client case respectively. In practice when the number of CTG is large, it tends to be difficult to recruit more proxies.

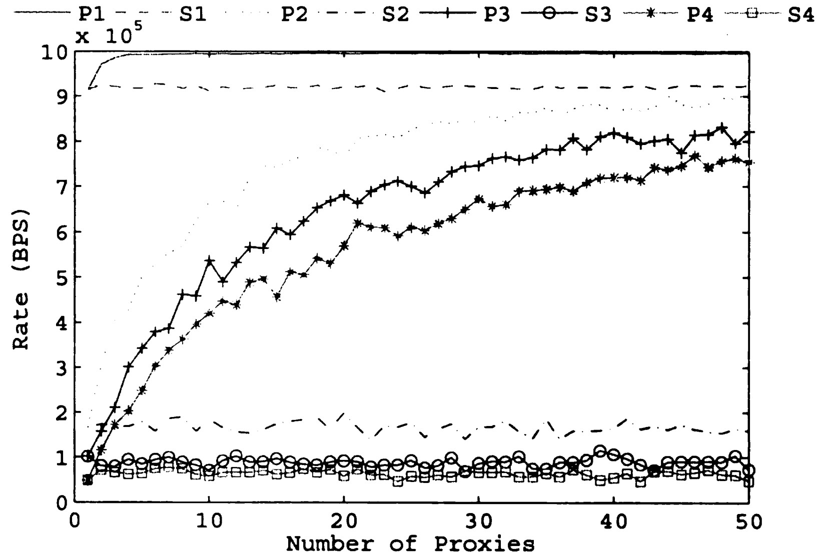


Figure 6.2. Cellular Rate vs. the Number of Proxies

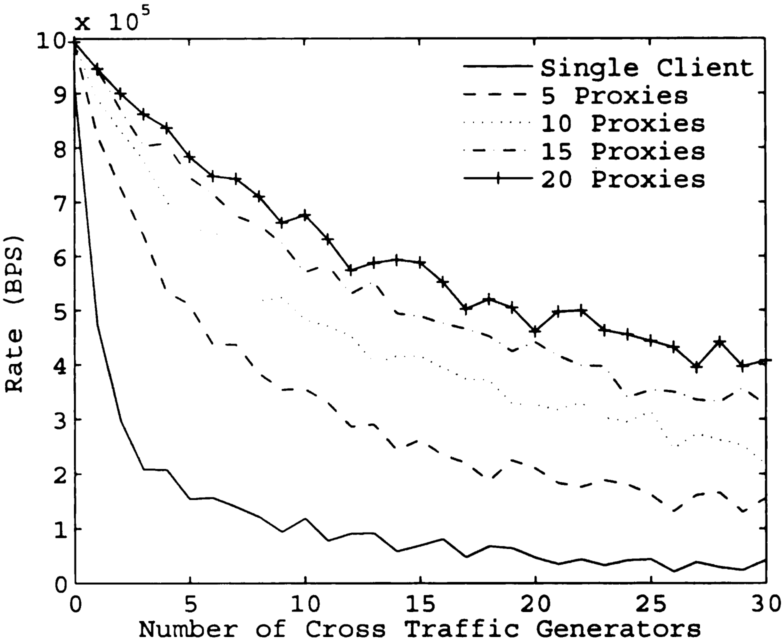


Figure 6.3. Cellular Rate vs. the Number of Cross Traffic Generators

6.2 A Dynamic Multi-proxy Selection Algorithm

The use of multiple cellular network relay proxies in the MANET can help improve the network QoS of the sender. However, it is also possible that multiple transmission flows increase the traffic load of the network and cause further congestion for the wireless network. For the cellular network, we can assume generally the BS will enforce an admission control policy to prevent the system load from exceeding its capacity. For the MANET, however, it is difficult to impose such a policy. Moreover, since all cellular proxies need to communicate with the control client through its WLAN interface, media contention and congestion in the MANET will have serious performance impact on the client.

In this section, we devise a dynamic contention detection and multi-proxy selection scheme to solve this problem. The basic idea is to have the control client monitor the contention level of the WLAN (indicated by the link layer retransmission rate) and the cellular bandwidth gains. Using those parameters, the control client estimates the transmission latency for a given traffic block. By calculating the derivative of the estimated latency over the number of proxies, the control client knows whether to increase, decrease, or maintain the number of proxies.

In a WLAN with multiple paths that originate from the sender, the control client is the most possible victim when the increased traffic load causes media layer contentions and interferences. Also, it is shown that a few more hops in the MANET will decrease the QoS dramatically [204]. Thus we only consider the impact of additional proxies on the first node of the multi-path, which is the control client. It is also safe for us to assume that the majority of the proxies are only one hop away.

When the number of paths is so large that excessive contention occurs, the QoS of the whole application degrades. This problem is closely related with the capacity of MANETs. It has been shown [75] that in a MANET with n nodes, the throughput $\lambda(n)$ obtainable by each node for a random destination is $\Theta(\frac{W}{\sqrt{n \log n}})$, given the rate of each node is W bits per second. Numerical simulations also show that the throughput of an IEEE 802.11 infrastructureless network will decrease once the number of node-pairs that are exchanging traffic is over 15 [50]. In our case of a MANET/cellular network, the impact of node density could be even worse since the control client is the bottleneck of the MANET communication.

Of all the costs associated with using multiple cellular proxies, the potential MANET contention caused by the proxies is a major constraint of the system performance. It is important to find a measure to describe the QoS degrading when excessive contention occurs. Unlike wired networks, where increased RTT is a good sign of built up queues because of congestion in routers, RTT is not a good measure because the major reason for QoS degrading in wireless networks is not due to built-up queues in the relayers, but packet loss or link layer retransmissions caused by increased media contention. Instead of using RTT as a metric of overloaded traffic, we develop a metric based on the link layer retransmission rate detected by the

MANET senders to monitor the condition of the media channels. Both media collision and channel noise can cause link layer retransmissions. Unlike media collisions, channel noise can be regarded as independent of the number of MANET users. Thus the retransmission rate can be taken as a measure of the degree of media contention in a MANET. By modeling the 802.11 media channel as an $M/G/1/PS$ system, it is shown that the collision probability increases logarithmically with the number of users [118]. In reality, the rate of retransmissions caused by collisions should be constantly monitored to facilitate the proxy selection process.

By taking the link layer retransmission rate as a collision indicator and using the retransmission rate and the cellular network bandwidth model proposed in section 6.1, we propose a scheme to dynamically adjust the number of proxies to achieve optimal system latency for a given traffic block. The multi-proxy selection algorithm can be illustrated in Figure 6.4.

Considering the asymmetric nature of the cellular link and the targeted application, the multi-proxy selection algorithm is designed to optimize the performance of downlink traffic. The major traffic flow will be from the BS to proxies, and then from proxies to the client. The system will bootstrap by recruiting a constant number of proxies. After that, the multi-proxy selection algorithm illustrated in Figure 6.4 is invoked periodically to maintain the optimal number of proxies in the active proxy group. The proxy group control algorithms determine whether to increase, decrease, or maintain the number of nodes in the proxy group. We assume the number of members in the proxy group is denoted by k. To support the proxy group control algorithm, each node i in the proxy group maintains two metrics, the measured link layer retransmission rate β_i and the cellular rate c_i. More details about retransmission rate measurement are discussed later. The achievable cellular rate can usually be obtained from the BS [95]. Periodically the retransmission rate β_i and the cellular rate c_i are sent to the client by being piggybacked to data packets.

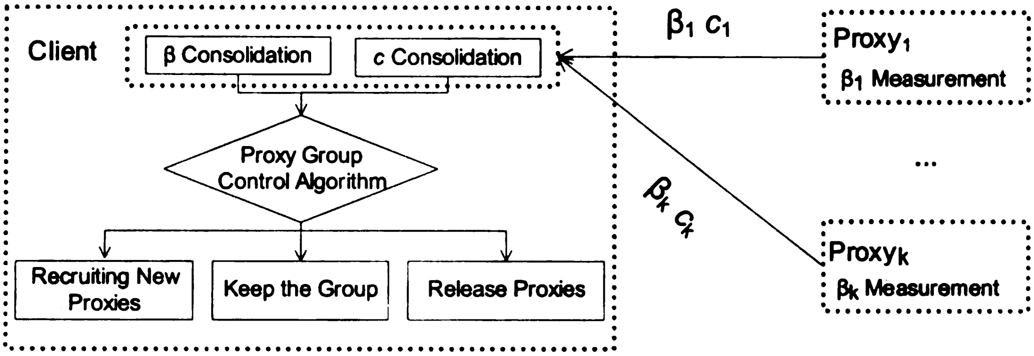


Figure 6.4. The Multi-proxy Selection Process

The β consolidation module processes the individual retransmission rate of each proxy to arrive at a collective retransmission rate of the proxy group and the client. As discussed at the beginning of this section and proved in [118], the collective retransmission rate is a monotonically increasing function of k. We denote it as $\beta(k)$ or β when there is no confusion. A simple consolidation process is to set $\beta(k)$ to be the arithmetic mean of all β_i. A more sophisticated method can take into account the amount of traffic carried by each proxy and use a weighted average. The c consolidation module calculates the collective cellular rate c by adding together the cellular rate c_i of each proxy i. Assuming the number of recruited proxies does not exceed the capacity of the given cell, as shown in the cellular bandwidth sharing model of section 6.1, c is a monotonically increasing function of k. Thus the collective cellular rate c can also be noted as $c(k)$ to reflect the dependence.

Assume the amount of a traffic block to be sent is n and the data rate of the ad hoc network is r. We can take r as the achievable bandwidth of the shared MANET media. For example, for IEEE 802.11b, the achievable bandwidth could be 2 MBPS or 11 MBPS depending on the networking conditions. The latency to transmit the traffic block is composed of two parts, the latency from the BS to the proxies and the latency from the proxies to the control client in the MANET. Before reaching the BS, the traffic has to traverse other communication channels, for example, the Internet. The latency of those communication channels can be optimized too but is beyond the scope of this research. The time used by the proxies to obtain the traffic from the base station can be approximated as $t_1(k) = \frac{n}{c(k)}$. For the latency in the WLAN, we model the impact of the retransmission rate over the latency. Given n as the net traffic to be sent and the link layer retransmission rate β, the actual traffic to be sent is $\frac{n}{1-\beta(k)}$. And the latency for transmitting the traffic in the WLAN is $t_2(k) = \frac{n}{(1-\beta(k))r}$.

The total latency is a performance indicator of the system given k proxies:

$$t(k) = t_1(k) + t_2(k) = \frac{n}{c(k)} + \frac{n}{(1-\beta(k))r}. \tag{6.4}$$

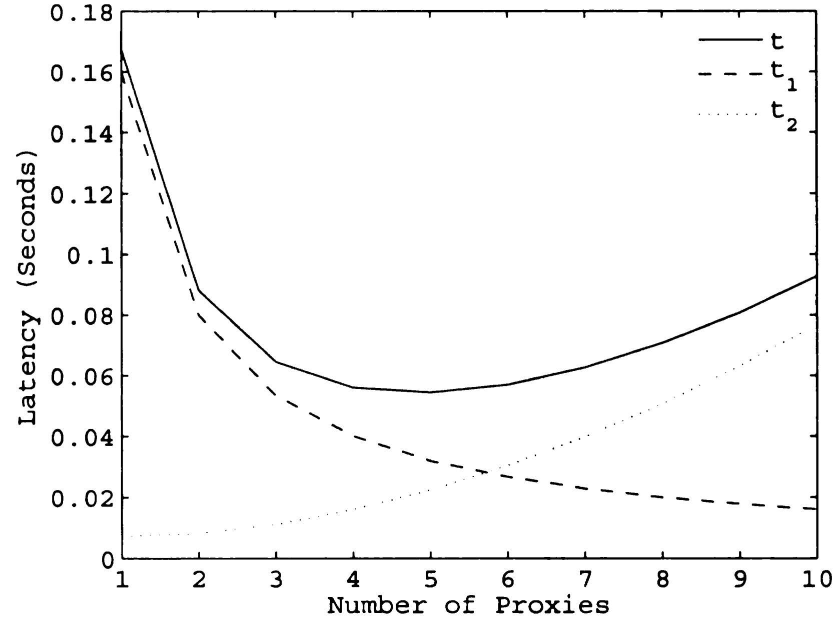


Figure 6.5. Simulated Latency vs. the Number of Proxies

The relationship of t, t_1 and t_2 versus k can be simulated as shown in Figure 6.5. β is simulated through a monotonically increasing function from $[1, +\inf)$ to $(0, 1)$. The parameter c is simulated through a monotonically increasing function from $[1, +\inf)$ to $(1.0e6, +\inf)$, which denotes 1 MBPS channel for each cellular connection. The parameter n is $8e4$ bits, which denote 10 1000-byte packets. The parameter r is $11e6$ to denote an IEEE 802.11b channel. The result shows when k is around 5, the latency is minimized. We can also observe that $t(k)$ is a convex function, so its extremum can be located through the derivative of k.

We obtain the derivative of k for equation (6.4):

$$t'(k) = \frac{n\beta'(k)}{r(1-\beta(k))^2} - \frac{nc'(k)}{c^2(k)}. \tag{6.5}$$

β' and c' can be calculated using numerical methods. Depending on the sign of t', we can decide whether to increase or decrease the number of proxies. In practice, t' is compared with a small positive threshold value δ. If $|t'| \leq \delta$, k is near optimal and we keep the number. If $t' < -\delta$, it means k value is on the left of the optimal value and we need to increase it. If $t' > \delta$, it means k value is on the right of the optimal value and we need to increase the number of proxies.

In reality, $\beta(k)$ or $c(k)$ may not be strictly monotonically increasing functions. Changes of physical environments may cause $\beta(k)$ or $c(k)$ to fluctuate. In that situation, the local extremum indicated by $t'(k)$ may not be the global extremum. We reduce the possibility of the local extremum by randomizing the amount of increment or decrement imposed on k. For example, if k needs to be increased, a random number is drawn from $[1, K]$ to be the increment value, where K is an upper bound of the increment. When t' indicates a near optimal k, a small amount of randomized increment or decrement can also be applied to avoid local extremum. As noted before, the maximum of k is limited by the capacity of the current cell. We assume that the proxies being recruited are facilitated with mechanisms to be notified of a capacity overflow. Otherwise, a upper bound of k is set for the algorithm.

The proxy hunting process can be based on any broadcasting based mechanisms as proposed by previous work [113, 211]. Failure recovery processes will also be invoked when one or more of the active proxies fail. In order to reduce the overhead of hunting for new proxies, when k needs to be decreased the node IDs of the released proxies can be cached as candidate proxies in a local pool of the client. Later when more proxies are needed, the proxies in the local pool are first probed and used. The candidate proxies are also preferably required to maintain connection information for the relay traffic if connection oriented transport protocols such as TCP are used to support reliable traffic. This further reduces the connection setup overhead. Alternatively, connectionless reliable transport services can also be used. Examples include forward error correction based connectionless reliable transport services, such as [30].

6.3 Simulation Results

We simulated the performance of the multi-proxy selection algorithm using the cellular network bandwidth model discussed in section 6.1 and the MANET/802.11 modules of NS2 ([130], version 2.29).

In the simulation, the MAC layer of NS2 802.11 is modified to support retransmission rate measurement. The measurements consider both retransmissions and the backoff timer of retransmissions. Successive retransmissions after the first retransmission are assigned more weight.

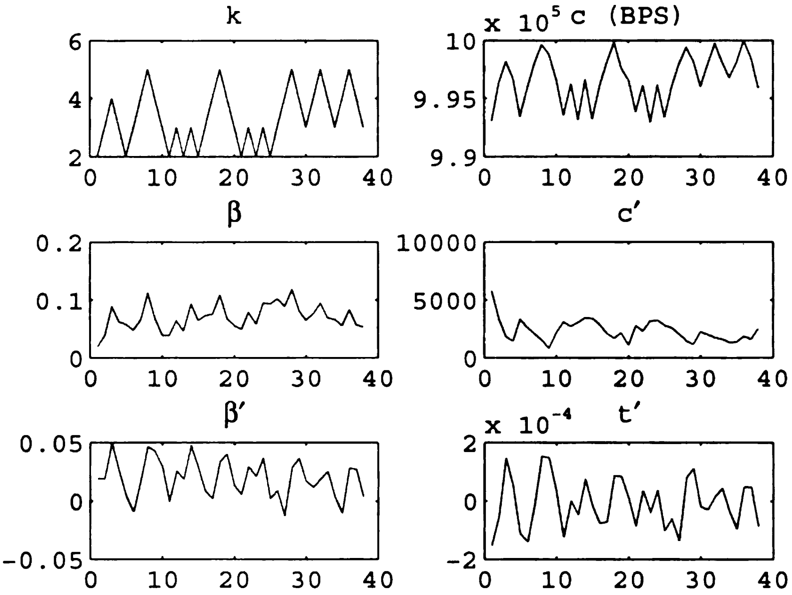


Figure 6.6. Parameters of Proxy Group Control with no CTGs

In the MANET scenario, the simulation area is a 400*m* × 400*m* rectangle. The client node is placed in the center of the area and the potential proxy nodes scatter around it within the power range. Each mobile node was simulated to work with the Orinoco 802.11b network card, which has a data rate of 11 *Mbit*/*sec* and a trans-

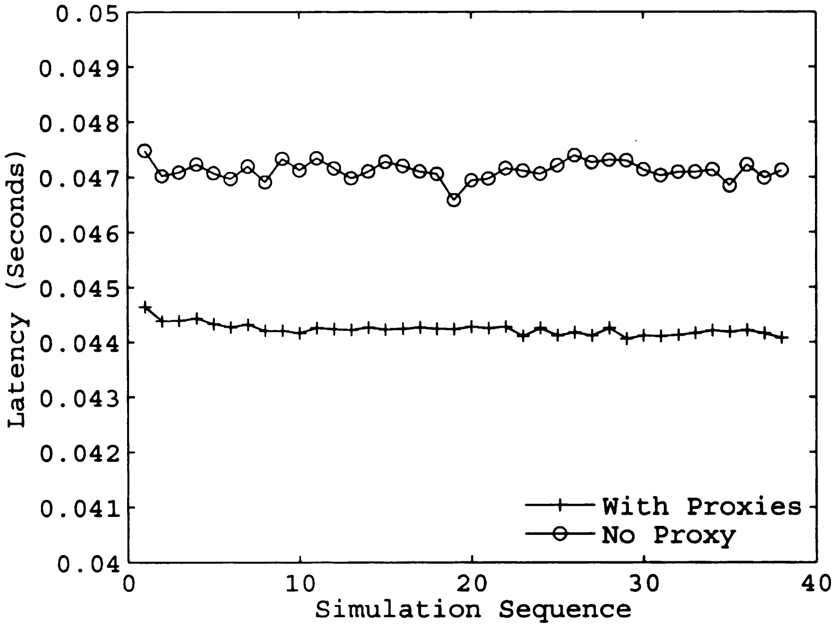


Figure 6.7. Latency Comparison When There Is no CTGs

mission range of 60 m. For the cellular network, each cellular interface requests an average of 1 MBPS rate. We used the on/off traffic pattern to measure the performance of the proxy selection algorithm. At each simulation step, the client requests a traffic block of 5K bytes. As in section 6.1, we simulated the effects of using multiple proxies in a cellular network under a Gaussian fading channel with 1.2% variance. In the proxy group control algorithm, we use a simple linear k adjustment algorithm, that is, increasing or decreasing k by one upon each t' indicator is calculated. More sophisticated algorithms can be used in real implementations to reduce the effect of local extremum as discussed in section 6.1.

The simulations are carried out under the situation when there are no Cross Traffic Generators (CTGs) in the cellular network and when there are CTGs in the cellular network. Figure 6.6 shows the parameters of proxy group control with no CTGs. In the figure, the X axis denotes the simulation event sequence. For each simulation event interval, QoS indicator t' is calculated and k is updated. The Y axis of each subgraph denotes the parameters as indicated by the title of each subgraph. In the given simulation conditions, k is stabilized around 4, although the upper bound of k is set to be 10. The collective cellular rate c is able to benefit from channel diversities and maintain an average rate of 9.96 MBPS. It is also clear that as k increases or decreases, β and c also increases or decreases. The derivative of latency, t', which is calculated according to Equation (6.5), serves as a good cue to control the number of proxies k. The latency comparison of the cases with and without proxies is shown in Figure 6.7. All the simulation parameters are the same except that in the case where there is no proxies k is always set to 1. The average latency for 5 K bytes on/off traffic in the two cases are 44.2 and 47.1 microseconds. The standard deviations are 0.108 and 0.166 microseconds. The improvements when proxies are used for latency and its standard deviation are 6.1% and 35%. The use of proxies are particularly effective in reducing end-to-end latency variance.

Figure 6.8 shows the parameters of proxy group control with CTGs. In the simulation, 5 CTGs in the cellular network are introduced. We observe that in the simulation with CTGs, the proxy group control process takes more simulation steps to converge (about 10 steps in the simulation, while in the case without CTGs, the converging time is almost negligible). The reason is that when CTGs are present, more proxies are needed to meet the requirement of QoS indicator. We can also see that the average number of proxies required is also increased from around 4 with no CTGs to around 12 with CTGs. The converging trend is also shown in the latency comparison (Figure 6.9). Once k stabilizes, the average latency for 5 K bytes on/off traffic in the two cases are 47.4 and 83.3 microseconds. The standard deviations are 0.971 and 1.73 microseconds. The improvements when proxies are used for latency and its standard deviation are 43% and 44%. When CTGs are present, using proxies are even more effective in reducing end-to-end latency and its variance.

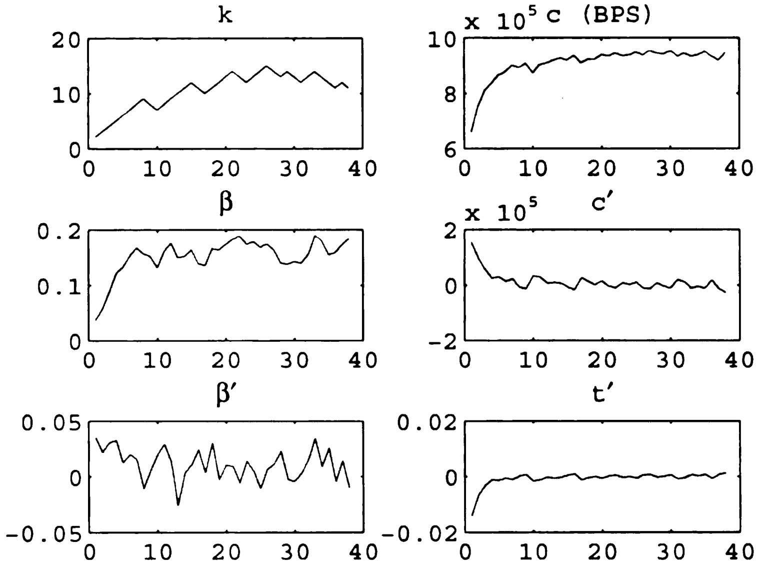


Figure 6.8. Parameters of Proxy Group Control with CTGs

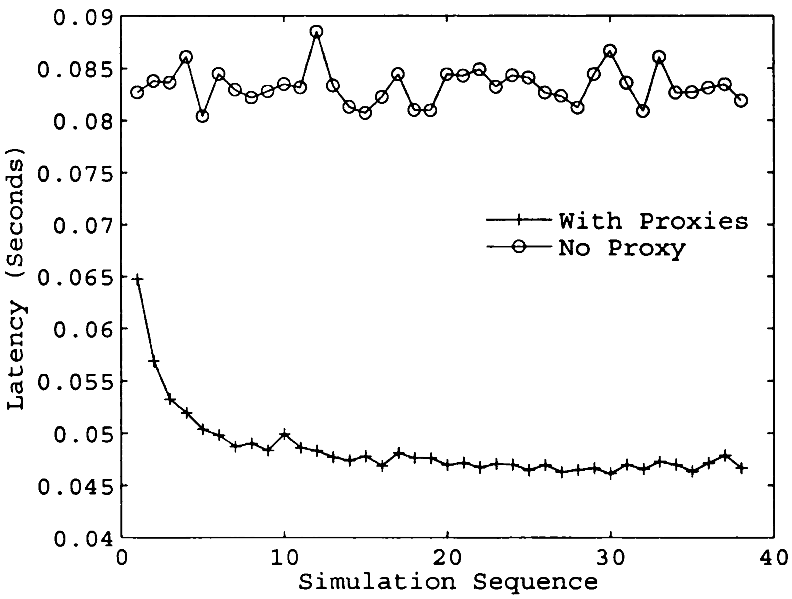


Figure 6.9. Latency Comparison When There Are CTGs

6.4 Summary

We aim to provide acceptable QoS for remote sensing and control applications by exploiting the channel diversities of MANET peers in a hybrid MANET/Cellular environment. Utilizing multiple cellular channels can potentially improve the system performance, and of course comes with costs. Compared with previous schemes, our work goes one step further by asking the question of under what condition the benefits of relay proxies outweigh their costs. We answer the question by measuring the contention caused by MANET peers through the MAC layer packet retransmission rate, and design a dynamic mechanism to control the number of relay proxies.

The model evaluation under common fading channels using prevalent scheduling algorithms indicate that utilizing channel diversity has great potential of improving the system performance. Built upon the cellular bandwidth sharing model we designed a dynamic proxy group control algorithm that will automatically adjust the number of proxies based on the cellular network bandwidth gain and MANET contentions. Simulations under NS2 indicates the proposed scheme is effective in reducing end-to-end latency and latency variance. When cross traffic generators are present, the gains obtained by multiple proxies are even more impressive.

CHAPTER 7

Sensor Network Reorganization through Hopping Relocation

Hopping sensors are a class of mobile sensors whose mobility design is inspired by creatures such as grasshoppers. Such sensors are able to maintain mobility in harsh terrains but may lack the movement accuracy of those sensors that are powered by wheels. Hopping sensors may also be susceptible to the influence of wind.

Mobility powered by the concepts of hopping are widely discussed in the area of planetary exploration [40, 78]. Similar units are constructed for robotic perimeter surveillance systems and self-healing minefield [60]. The work of Feddema, et al. [60] described a prototype minefield hopping robot. The prototype is 12 cm in diameter and 10 cm in length. It can make 100 hops without refueling and can hop as high as 3 m. The robot is also equipped with a radio, a steering system and ranging sensors.

In this research, we assume that the hopping sensors are capable of adjusting the hopping direction. The hopping range could be adjustable, but we do not make that assumption. We generally assume a fixed propelling force for hopping. In reality, the hopping range may vary due to the physical load of the sensor, different terrain conditions, and local aerodynamic settings. The landing area can be modeled with a normal distribution centered around the projected destination. In order to facilitate routing, the sensors are also assumed to have a localization capability.

Compared with wheeled mobile sensors, hopping sensors have the following characteristics:

- The movements of hopping sensors are not as accurate as wheeled sensors. Thus the movements of the sensors should be modeled probabilistically. When transporting multiple sensors from one place to another, we may need to consider the possibility that some sensors may have a skew track and thus miss the destination.

- The movement of hopping are more susceptible to air disturbance than wheeled mobility. Although local air disturbance is difficult to predict, it is not as difficult to measure and predict large scale air disturbance or wind. Using such measurements the hopping directions and number of hops needed to reach a destination can be adjusted dynamically.

- Hopping sensors may need a charging period before engaging in hopping, but after that the hopping can happen nearly instantly. Compared with wheeled sensors, which may need considerable time to move to the destination, hopping sensors may have fewer coverage disruptions caused by mobility.

We examine the opportunities and challenges of deploying low cost hopping sensors and utilizing their mobility to ensure coverage and maintain energy efficiency within a sensing field. We focus on the problem of transporting a number of hopping sensors from multiple sources to a destination. Probabilistic methods are used to contain the movement inaccuracies along the hopping course. We also consider the impact of wind under an aerodynamic setting. Two transport schemes are designed to minimize the number of hops needed while considering other constraints, such as sustaining the capability of relocating sensors within the whole network. In one scheme we use upper and lower hopping limits to apply the network mobility constraints. The other scheme uses a balancing coefficient to construct a new optimization target to meet the requirement of path optimality and network mobility dynamically. Both schemes are simulated and evaluated using the physical and dynamic parameters of the prototype minefield hopping robots. The results show that both schemes work well regardless of the wind factors, while the dynamic scheme is also shown to be resilient to topological changes of the network.

7.1 System Model

Hierarchical models are widely used in sensor network design. The work of Zou, et al. [213] assumed that a cluster head was available to coordinate the sensor deployment based on virtual forces. Wang, et al. [183] also used a grid based quorum method to match the source and destination grids. Clustering techniques are also used to facilitate efficient routing in wireless ad hoc networks, such as zone routing protocols [135]. In our hopping sensor network model, we assume that the sensors are organized into clusters. A cluster head assumes the responsibility of evenly distributing the sensors, detecting sensor deficiency, and detecting redundant sensors within the cluster. The cluster head may be a more powerful node in terms of energy and computing capacity compared with other nodes. However, it is also possible that the cluster head responsibility is rotated among peers and the head is elected using head election algorithms [83].

There are well established methods to detect redundant sensors in a sensor network based on computing Voronoi diagrams [22]. The clusters that are detected to have redundant sensors mark themselves as sources and identify themselves through the cluster head over the sensor network. The problem of detecting sensing holes or sensing wells are also studied [182, 70, 59]. Ghrist, et al. [70] propose using an algebraic topological invariant, referred as homology to detect sensing holes. The work of Fang, et al. [59] detects and circumvents sensing holes by means of identifying stuck nodes. The sensing holes also mark and identify themselves over the sensor network.

Matching the sources and holes could be complicated in the presence of multiple sources and holes. The work of Chellapan, et al. reduces the matching problem in the sensor deployment stage as a multi-commodity maximum flow problem [32]. We assume that hopping sensor migrations are necessary only in the runtime stage and the occurrence of sensing holes are not highly frequent. We take the approach of filling the sensing holes one by one. In order to maximize the migration efficiency, it is necessary to recruit multiple sources to provide redundant sensors. Thus we consider the problem of migrating hopping sensors from multiple sources to one destination. All the sources and the hole add their presence information to their identities. The matching process can be done by piggybacking the presence information to routing protocols. A more sophisticated method, using grid based quorum, is also presented in [183].

7.1.1 Normal Distribution Model of Inaccurate Hopping

Compared with wheeled mobile sensors, hopping sensors lack the accuracy of movement. This is a design trade off between more functionality and lower cost for mass deployment. Hopping sensors can also maintain mobility in harsh terrains. The orientation ability of the sensors are calibrated and measured before deployment. We use a multivariate normal distribution model to determine the landing accuracy of multiple sensors that hop together. Based on the model, we derive the upper bound of the number of hops needed to migrate a number of sensors to a target location. The result is also used to derive the number of sensors that can reach the consumer from the sources for a given demand when astray sensors are considered. A sensor is regarded as astray when its landing point is outside a range of the projected point, as will be explained later in the writing.

In a two dimensional case, the landing accuracy is characterized by the displacement between the targeted location, represented by vector $\mathbf{T}$, and the actual landing location represented by vector **L**. The displacement vector **D** can be represented as $\mathbf{D} = \mathbf{T} - \mathbf{L}$. **D** is modeled by a two dimensional normal distribution with mean $(0, 0)$, standard deviation (σ, σ), and correlation ρ. Note that we used a nondiscriminating standard deviation vector for the two dimensions. These parameters can be obtained based on sensor samples using maximum likelihood estimation methods. The probability density function of **D** in terms of (x, y) can be given as

$$f(x, y) = \frac{1}{2\pi\sigma^2\sqrt{1-\rho^2}} \exp\left(-\frac{x^2 + y^2 - 2\rho xy}{2\sigma^2(1-\rho^2)}\right). \tag{7.1}$$

In order to model the uncertainty of the landing location, we define an acceptable landing area as a torus S around the targeted location. As shown in Figure 7.1, the radius of the torus $n\sigma$ is determined by multiplying a factor n to the standard deviation σ. For the migration distance that requires one hop only, the acceptable landing area lies within the target location of the sensor. For migrations that need more than one hop, if the sensor lands within the acceptable landing area, the sensor is recharged for another hop towards the target; otherwise the sensor is deemed as astray and will seek to join a local cluster.

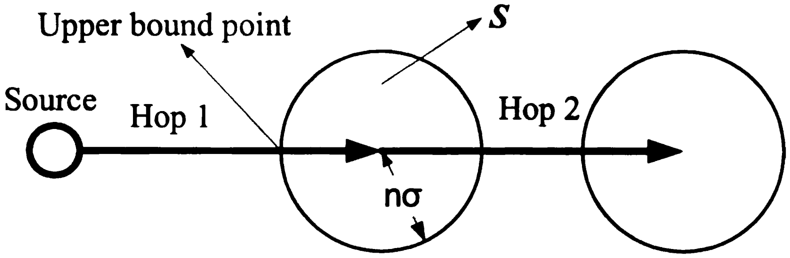


Figure 7.1. Modeling the Hopping Accuracy Using Normal Distribution

The probability that the hopping sensor lands in the acceptable landing area S can be represented as

$$P(S) = \iint_S f(x, y)\,dx\,dy. \tag{7.2}$$

It is known that Equation (7.2) does not have a closed form solution. Numerical methods can be used to calculate the probability [68].

Based on the probability that the hopping sensor lands in the acceptable landing area, we derive the upper bound of the number of hops needed for a migration distance l. Assuming the distance covered by one sensor hop is r, the lower bound of the number of hops N needed to cover distance l is

$$N = \frac{l}{r}. \tag{7.3}$$

The upper bound of N, however, is determined by the situation when, for every hop, the sensor lands at the farthest point from the target and on the direct line segment between the source and the target location, as indicated in Figure 7.1. This is equivalent to reducing the hopping range of the sensor from r to $r - n\sigma$. Thus the upper bound of the number of hops given distance l and hopping range r can be given by

$$N_u = \frac{l}{r - n\sigma}. \tag{7.4}$$

Assume we know the number of sensors demanded by the target cluster and the number of sensors that are tasked to the current source is E_t. As will be discussed in section 7.2.1, the consumer estimates the number of astray sensors and demands some additional sensors to cover those sensors. For each hop, the number of hopping sensors will decrease due to straying. For the first hop, the number of hopping sensors is E_t. For the second hop, the number is $E_t P(S)$, etc. Thus the total number of hops needed for all sensors is

$$H = \sum_{i=0}^{N_u - 1} E_t P^i(S) = E_t \sum_{i=0}^{N_u - 1} P^i(S) = \frac{1 - P^{N_u}(S)}{1 - P(S)} E_t. \tag{7.5}$$

The calculation of the probability that the hopping sensor lands in the acceptable landing area S may be computationally intensive when using numerical methods. Thus $P(S)$ can be precalculated based on the predetermined parameters σ, ρ and n. For noncritical cases, $P(S)$ can even be approximated by its one dimensional counterpart.

7.1.2 Aerodynamical Model of Hopping under Air Disturbance

Unlike wheeled movement, the motion of hopping is more susceptible to air disturbance. We need to estimate the impact of air disturbance so that we can adjust the hopping orientation properly when wind is detected. We also model the influence of air disturbance on the hopping range of the sensors so that we can obtain an accurate estimation of the number of hops needed to traverse a certain path. A good estimation of the number of hops is crucial since one of the criteria that the route planning process considers is the number of hops needed to travel between clusters.

We assume the hopping sensor has a fixed propelling impetus. In order for the sensor to have the longest horizontal hopping range, the sensor should jump towards a tilted upward direction. When there is no air disturbance, the optimal upward angle that provides the longest horizontal range is 45°. This can be easily proved using the principles of gravity. Assume the hopping velocity is $\mathbf{v}$ and it is α up from the horizontal direction, as shown in Figure 7.2. The flying time of the sensor is

$$t = \frac{2|\mathbf{v_v}|}{g} = \frac{2|\mathbf{v}|\sin\alpha}{g}. \tag{7.6}$$

In Equation (7.6), $\mathbf{v_v}$ ($|\mathbf{v_v}| = |\mathbf{v}|\sin\alpha$) is the velocity of the sensor in the vertical direction and g is the amount of acceleration caused by gravity. Assuming a flat terrain, the horizontal distance traveled by the sensor in the air is

$$r = t|\mathbf{v_h}| = t|\mathbf{v}|\cos\alpha = \frac{|\mathbf{v}|^2\sin 2\alpha}{g}. \tag{7.7}$$

In Equation (7.7), $\mathbf{v_h}$ ($|\mathbf{v_h}| = |\mathbf{v}|\cos\alpha$) is the velocity of the sensor in the horizontal direction. Maximizing r in terms of α yields $r = \frac{|\mathbf{v}|^2}{g}$ when $\alpha = \frac{\pi}{4}$. Thus the magnitude of the initial horizontal velocity of the sensor is $|\mathbf{v_h}| = |\mathbf{v}|\cos\frac{\pi}{4} = \frac{\sqrt{2}}{2}|\mathbf{v}|$.

We assume that the air disturbance only influences the horizontal velocity of the sensor. The result of the influence will be reflected on both the direction of hopping

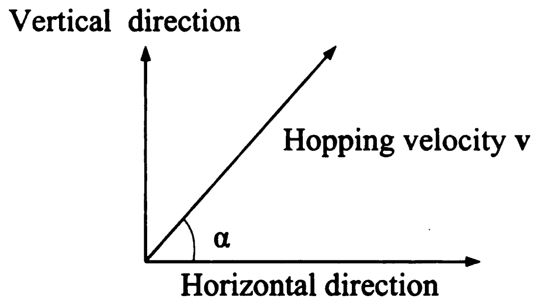


Figure 7.2. Velocity of the Hopping Sensor

and the magnitude of the horizontal velocity. If the velocity of the wind in the horizontal direction is $\mathbf{v_w}$, the real velocity of the hopping sensor in the horizontal direction is $\mathbf{v_{hreal}} = \mathbf{v_h} + \mathbf{v_w}$, as shown in Figure 7.3. In order to target the sensor for a certain direction, we have to adjust the direction of the initial hopping velocity to compensate for the influence of the air disturbance. If the direction of the targeted location is given in $\mathbf{v_{hreal}}$, the hopping direction of the sensor can be given as

$$\mathbf{v_h} = \mathbf{v_{hreal}} - \mathbf{v_w}. \tag{7.8}$$

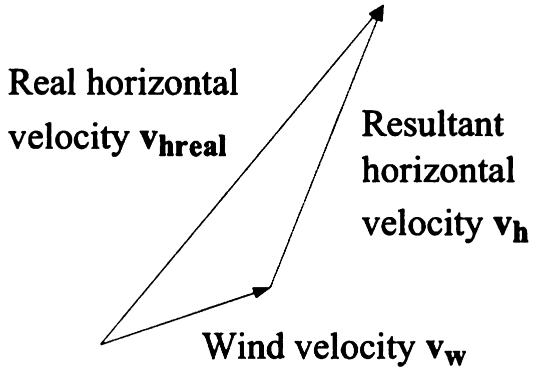


Figure 7.3. Horizontal Velocity of the Hopping Sensor under Air Disturbance

We also worry about the number of hops needed to traverse a distance. Under the influence of air disturbance, the number of hops needed for a certain distance will also change. When there is no wind, the hopping range of the sensor can be given in Equation (7.7). Since the wind only influences the horizontal velocity of the sensor, the flying time of the sensor remains the same when there is air disturbance. Thus the hopping range of the sensor with wind can be represented as

$$r_w = t|\mathbf{v_{hreal}}|. \tag{7.9}$$

In practice, we use the proportion of the hopping range under air disturbance to the normal hopping range to facilitate calculation of the number of hops needed. According to Equation (7.7) and Equation (7.9), we have

$$\frac{r_w}{r} = \frac{|\mathbf{v_{hreal}}|}{|\mathbf{v_h}|}. \tag{7.10}$$

Assuming the number of hops needed for a trip of length l under the normal condition is N, the number of hops needed under air disturbance N_w can be obtained through

$$\frac{N_w}{N} = \frac{\frac{l}{r_w}}{\frac{l}{r}} = \frac{r}{r_w} = \frac{|\mathbf{v_h}|}{|\mathbf{v_{hreal}}|} = \frac{|\mathbf{v_h}|}{|\mathbf{v_h} + \mathbf{v_w}|}. \tag{7.11}$$

In conclusion, when the air disturbance generates a horizontal velocity of $\mathbf{v_w}$, the hopping direction of the sensor can be given using Equation (7.8), while the number of hops needed for a fixed range compared with the normal case can be given using Equation (7.11). When considering the upper bound of the number of hops needed (denoted as N_{wu} and N_u), Equation (7.11) is modified to

$$\frac{N_{wu}}{N_u} = \frac{\frac{l}{r_w - n\sigma}}{\frac{l}{r - n\sigma}} = \frac{r - n\sigma}{r_w - n\sigma} = \frac{|\mathbf{v_h}|t - n\sigma}{|\mathbf{v_{hreal}}|t - n\sigma}. \tag{7.12}$$

Given Equation (7.6) and $\alpha = \frac{\pi}{4}$, Equation (7.12) is also easily calculable. Similar to Equation (7.5), the upper bound of the total number of hops for all sensors $H_w$with air disturbance can be given by

$$H_w = \frac{1 - P^{N_{wu}}(S)}{1 - P(S)} E_t, \tag{7.13}$$

where E_t is the number of sensors that are tasked to the current source.

In order to obtain the wind information needed by the model, anemometers are deployed within the sensor network. Those sensors equipped with anemometers periodically broadcast information about the wind power and direction. In extreme cases, anemometers can also be equipped on each sensor. The behavior of the air flow is bursty and changes fast in most atmospherical and geographical settings. The model implied in Equation (7.13) is only applicable to the situations where the wind flow has a predictable and relative stable nature. Examples of such environments are indoor spaces where air disturbance is caused by ventilation systems, or outdoor settings where the air flow is stable and predictable. For settings where the air flow follows random changes, a pre-deployment can be used to measure the air flow impacts. If the air flow impact is assumed to follow a normal distribution model, the method used in section 7.1.1 can be augmented to cover the upper bound of the number of hops needed in such situations.

7.2 Route Planning in Hopping Sensor Migrations

Route planning involves migrating the required number of sensors from multiple sources to the destination cluster. The process includes identifying supplying and consuming sensor clusters, matching those clusters and selecting the optimal route to migrate the sensors.

7.2.1 Matching of the Consumer and Suppliers

Once a cluster detects a sensing well inside its cluster, it signals the presence of its sensing well to the network through an SREQ message requesting additional sensors and identifies itself as a consumer. At the same time, the clusters with redundant sensors also signals their presence through an SREP message and identify themselves as suppliers. The SREQ message specifies the number of sensors needed and the SREP message specifies the number of sensors that can be provided by the supplier. Location information is also supplied in the messages. The matching process can be done through either broadcasting or quorum based approaches.

In a grid quorum based approach [183], the clusters are formed so that a cluster grid exists in the sensor field. In a grid, an SREQ message is only multicasted into its own row (column). Likewise, SREP messages are multicasted into the columns (rows) where the suppliers reside. The intersected clusters of the row and those columns are responsible for matching the suppliers and consumers. The work of [183] only supports matching one supplier with one consumer. In order to extend the approach to support multiple suppliers, multiple intersected clusters will perform the matching task. The intersected clusters of the row and those columns will have to send the matching information to the consumer. The consumer finally acknowledges to each potential supplier and determines the actual suppliers. Since sensors might go astray in the migration process, the consumer chooses the suppliers by considering the astray rate.

Assume that for supplier i ($i = 1 \cdots M$), the number of sensors it can provide is p_i and the distance between supplier i and the consumer is l_i, the number of sensors that can reach the consumer can be estimated as

$$E_{est} = \sum_{i=1}^{M} p_i P^{\frac{l_i}{r-n\sigma}}(S). \tag{7.14}$$

In Equation (7.14), $P(S)$ is the rate of the sensors that land in the acceptable range, as defined in Equation (7.2). $\frac{l_i}{r-n\sigma}$ is the number of hops needed between the consumer and supplier i, which is similar to Equation (7.4). When considering the wind factor, the number of hops estimation can be done with the aid of Equation (7.10). It is noted that the number of hops estimated is based on the assumption that the sensors migrate from the supplier to the consumer directly, without using relay clusters. The concept of relaying is explained in section 7.2.2. When relay clusters are used, the number of hops needed might be larger. In the latter case, an augmenting factor should be applied to Equation (7.14).

Assuming the number of sensors the consumer needs is d, we compare d and E_{est}. If $E_{est} \geq d$ the suppliers can satisfy the consumer and a matching process can be concluded. Otherwise the consumer has to wait for additional suppliers.

If multiple SREQ messages are issued in the network, policies can be made to decide which clusters to be served first. For example, priority levels can be indicated in the SREQ message and the request with highest level is honored first. Aged requests will have their priority levels increased in order to prevent starvation of initially lower level requests. Those requests that cannot be satisfied currently are cached by each supplier. When multiple suppliers are involved, deadlocks may happen when all the available suppliers are engaged into different consumers but none of the consumers have enough sensors to start sensor migration. When designing operation protocols of the supplier, consumer and matcher, mechanisms are developed to avoid or resolve such deadlocks. One method is to enforce non-duplicate priority levels and timeout mechanism in the matching process.

7.2.2 Finding the Optimal Migration Path

In this section we discuss the algorithms to migrate the hopping sensors from multiple suppliers to the consumer cluster. We assume the routing planning process is performed at the supplier clusters. Each cluster has a topological view of the network.

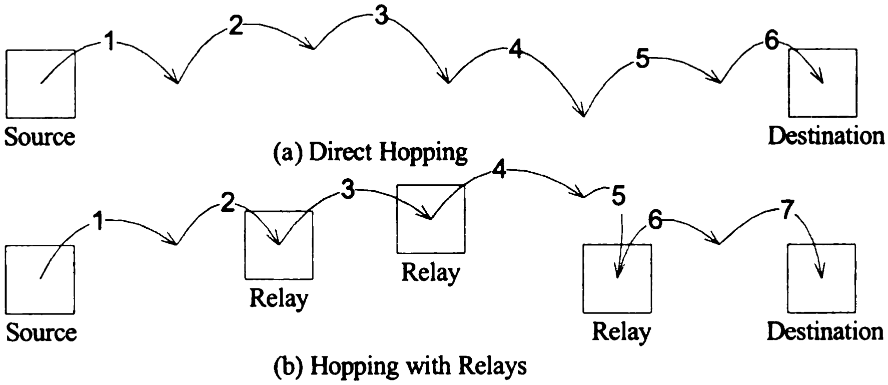


Figure 7.4. Two Hopping Strategies

Two possible migration strategies can be used. The first strategy is to migrate the sensors directly from the supplier to the consumer, as shown in Figure 7.4(a). The second strategy, however, uses intermediate clusters as relay clusters. Sensors can hop simultaneously starting from the supplier and the relay clusters, as illustrated by Figure 7.4(b). One cluster does not need to have redundant sensors to be able to serve as a relay cluster since the migrated sensors will soon be superseded by the sensors migrated from the preceding cluster in the cascading chain. However, a cluster may not be available to serve as a relayer if the majority of its nodes are engaged in sensing or communicating activities.

Cascaded hopping or hopping with relays of the second strategy was also proposed by Wang, et al. [183] to migrate wheeled sensors. Since sensors in the relay clusters can start hopping simultaneously, the time to finish the migration is shorter compared with direct hopping. The second strategy does not need to require the hopping sensors to be able to hop a long range. It is also beneficial in terms of maintaining the capability of mobilizing the majority of the sensors. Although sensors can be assumed to be able to make multiple hops, the number of hops one sensor can make (denoted as k_{hard}) is not unlimited before refueling or recharging. For example, the prototype proposed by Feddema, et al [60] has a limit of 100 hops before refueling.

In the first strategy, the hopping capability of the sensors throughout the network could be depleted unevenly, which actually reduces the dynamic relocation capability of the network. However, the first strategy wins by reducing the total number of hops needed in some cases. Direct hopping can always guarantee a straight route but hopping with relays cannot. Furthermore, direct hopping gives sensors more freedom to adjust their hopping direction on the fly if the landing point is off the route, which will ultimately save the number of hops. For example, in Figure 7.4(b), hop number 5 and 6 could have been combined into one hop if the relay is not used. Thus it is desirable to use straight hopping for relatively short migration paths but using relay hopping for longer paths. For relay hopping, the tradeoffs require us to set a limit on both the maximum and minimum distance between the relay clusters. We set the upper limit k to maintain the relocation ability of the network. A lower limit j is set to avoid unnecessary hops introduced by excessive number of relays.

The notion of short or long here, is related with k_{hard}, the number of hops one sensor can make before refueling. k_{hard} is referred as the hard limit. The soft limit k is set so mobility of the sensors are kept evenly throughout the network.

We use the topology and availability information of the clusters to construct a graph. The vertices set is composed of the clusters. Initially, all the available clusters, including the suppliers and the consumer, are connected with edges. The weight of the edges are set to be the estimated number of hops needed, as indicated in Equations (7.4) and (7.12). Our objective is to find an optimal path to migrate the sensors from the suppliers to the consumer.

Algorithm 1 MinHops($G, W, S, t, j, k, k_{hard}$)

```
$(G', W')$ ← Delete edges whose weights are outside $[j, k]$ in $W$
$(G'', W'')$ ← Delete edges whose weights are larger than $k_{hard}$ in $W$
for all $s$ in $S$ do
    $p$ ← Dijkstra($G', W', s, t$)
    if $p = \emptyset$ then
        $p'$ ← Dijkstra($G'', W'', s, t$)
        if $p' = \emptyset$ then
            Failed for source $s$
        else
            Output results from $p'$
        end if
    else
        Output results from $p$
    end if
end for
```

In the first route planning algorithm (Algorithm 1), we first apply the upper and lower limits of the edge weights by deleting the edges whose weights fall outside the range of $[j, k]$. Then Dijkstra algorithm (Algorithm 2) is called for the sources to find the shortest paths from the suppliers. If for some sources the shortest path cannot be

Algorithm 2 $p = \text{Dijkstra}(G, W, s, t)$

```
for all $v$ in $V(G)$ do
    $d(v) \leftarrow +\infty$ {Initialize the distance vector}
    $p(v) \leftarrow$ undefined {Initialize the previous node vector}
end for
$d(s) \leftarrow 0$ {The source has 0 distance to itself}
$C \leftarrow \emptyset$ {Initialize the checked node set}
$Q \leftarrow V(G)$ {Copy the vertex set into a working set}
while $Q \neq \emptyset$ do
    $u \leftarrow ExtractMin(Q)$ {Extract the vertex with minimum value in $d$}
    $C \leftarrow C \cup \{u\}$
    if $u = t$ then
        return
    end if
    for all edge $(u, v)$ do
        if $d(u) + W(u, v) < d(v)$ then
            $d(v) \leftarrow d(u) + W(u, v)$
            $p(v) \leftarrow u$
        end if
    end for
end while
if $!t \in C$ then
    $p \leftarrow \emptyset$
end if
```

found, we release the limits, but apply the upper hard limit, and run the algorithm again. If the shortest path still cannot be found for some suppliers, we conclude the supplier is separated from the consumer and the consumer is informed to search for new suppliers. Algorithm 2 takes the graph G, edge weight set W, source node s and destination node t as inputs and returns vector p. Vector p specifies the previous node chain along a found path starting from the destination. Using p, the shortest path can be easily constructed. Algorithm 1 takes the graph G, edge weight set W, source node set S, destination node t and the limits (j, k, k_{hard}) as inputs.

The algorithm stated above tries to maintain even mobility consumption among clusters and minimal number of hops using manually set limits. However, for the clusters whose edges are within the limits, the mobility consumption and number of hops of the edges can still vary from edge to edge. This is more obvious when the topology of the graph changes. In order to migrate the sensors from the suppliers to the destination optimally, and at the same time maintain even mobility throughout the clusters, we consider the following goals for a desirable path.

1. The total number of hops along the path should be small.
2. The motion should be distributed evenly among the clusters. That is, the maximum number of hops of the edges along the path should be small. However, the minimum number of hops of the edges should not be too small in order to avoid unnecessary relay clusters.

In order to meet the conditions stated above, we modify the path searching algorithm by adding an additional adjusting process. In the adjusting process, we try to minimize a fraction of the sum of the weights along the path and a fraction of the difference of the maximum and minimum weights of the edges along the path. Since the maximum and minimum weight of the edges along the path is only known after the path is found, we run Algorithm 2 several times until we find the optimal solution.

The second algorithm is described in Algorithm 3. The algorithm takes the graph G, edge weight set W, source node set S, destination node t and the hard limit k_{hard} as inputs. The lower and upper soft limits are not needed. For each source node, the algorithm first runs Algorithm 2 to get a shortest path solution where the only constraint is the hard limit. After that, the solution is refined through several iterations. The minimization target becomes a fraction of the sum of the weights along the path and a fraction of the difference of the maximum and minimum weight of the edges along the path, which is expressed as $(1-\gamma)w_{sum} + \gamma(w_{max} - w_{min})$ as shown in line 11. Here $\gamma\,(0 \leq \gamma \leq 1)$ is a coefficient to determine the fraction of the difference of the maximum and minimum weight of the edges to be considered in the algorithm. The sizes and availability of the clusters, and distances between clusters, are all factors to be considered when tuning the coefficient. The algorithm terminates when it cannot find a better solution given the constraints and the last known good solution is taken as the final solution.

Algorithm 3 MinHopsExt(G, W, S, t, k_{hard})

```
$p' \leftarrow \emptyset, (w'_{sum}, w'_{max}, w'_{min}) \leftarrow (+\infty, +\infty, 0)$
$W' \leftarrow$ Delete edges whose weights are larger than $k_{hard}$ in $W$
for all $s$ in $S$ do
  $(G_1, W'_1) \leftarrow (G, W')$
  loop
    $p \leftarrow$ Dijkstra$(G_1, W'_1, s, t)$
    if $p \neq \emptyset$ then
      $w_{sum} \leftarrow$ Get the sum of edge weights using $p$
      $(e_{max}, e_{min}) \leftarrow$ Get the edges with maximum and minimum weight using $p$
      $(w_{max}, w_{min}) \leftarrow$ Get the weights of $(e_{max}, e_{min})$
      if $(1-\gamma)w_{sum} + \gamma(w_{max} - w_{min}) < (1-\gamma)w'_{sum} + \gamma(w'_{max} - w'_{min})$ then
        $(w'_{sum}, w'_{max}, w'_{min}) \leftarrow (w_{sum}, w_{max}, w_{min})$
        $p' \leftarrow p$
        Delete edges $e_{max}$ and $e_{min}$ in $W'_1$ and $G_1$
      else
        Use the solution in $p'$ and break loop
      end if
    else
      Use the solution in $p'$ and break loop
    end if
  end loop
end for
```

7.3 Performance Evaluations

In order to validate the proposed models and algorithms, we simulated a hopping sensor network environment under Matlab. A topology generator is used to generate the simulation environment in a $1000m \times 1000m$ rectangular area. The simulation environment can be regarded as a sensor network where some clusters are available to serve as relays, while others are not. The available clusters, as well as the source and destination clusters, are plotted. Two clusters are connected if the number of hops needed to migrate between them is less than k_{hard}. Figure 7.5 shows a simulation topology. The dotted lines connect available clusters and the path between a possible supplier and consumer is indicated by a solid line.

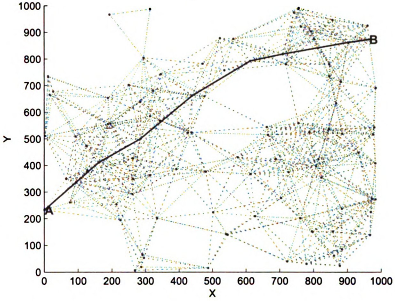


Figure 7.5. A Simulated Network Topology

The parameters regarding hopping sensor dynamics and sensor distribution are shown in Table 7.1. We derive the hopping range, initial velocity and hops capacity based on the Feddema prototype minefield hopping sensor model [60]. We assume zero correlation of the X and Y direction in the landing model.

A series of simulations are performed to evaluate the performance of the proposed route planning strategies. We first evaluated the performance of Algorithm 3 where the path optimality and remaining network mobility are controlled through the setting of soft hop limits. Considering the fact that most clusters may be involved in sensing

Parameter	Value
Number of Total Hops Capable without Refueling (k_{hard})	100
Hops Capable per Sensor Initially	100
Magnitude of Initial Horizontal Velocity ($\|\mathbf{v_h}\|$)	$7m/s$
Hopping Range (r)	$3m$
Sensors per Cluster Initially Deployed	200
Acceptable Landing Area Radius ($n\sigma$)	$0.6m$
Acceleration due to Gravity (g)	$9.8m/s^2$

Table 7.1. Parameters Used in Hopping Sensor Simulation

tasks and not available as relays, the simulation topology is generated as a sparse graph. In a sparse environment, it is safe to use a small lower hop limit j without jeopardizing the optimality of the path since the degree of each node is small.

In the first simulation we fixed the lower limit j to be 1 and evaluated the number of hops needed to migrate a group of sensors and the mobility capacity of the network after the migration when the upper hop limit k changes (Figure 7.6).

The path optimality is measured using the average number of hops consumed by each migrated sensor (referred as $\bar{H}_m$, Figure 7.6(a)). In the simulation, a group of 10 sensors ($E_t = 10$) are requested by the consumer. Thus $E_t\bar{H}_m$ is the total number of hops needed in the migration. The remaining mobility of the network is measured using the standard deviation (STD) of the number of remaining hops per sensor throughout the network (Figure 7.6(b)).

In the simulation we evaluated the metrics under three circumstances, the normal condition, the condition with a positive wind force and the condition with a negative wind force. Considering the migration path from A to B in Figure 7.5, a positive force is modeled as a wind velocity that is 45° counterclockwise of the positive X axis. Likewise, a negative force is modeled as a wind velocity that is 225° counterclockwise of the positive X axis. In either case, the magnitude of the wind velocity $|\mathbf{v_w}|$ is set to be 1/10 of the magnitude of the initial horizontal hopping velocity $|\mathbf{v_h}|$.

The simulation results of Figure 7.6 indicates that the value of the upper soft limit k has a significant influence on the path optimality and remaining network mobility.

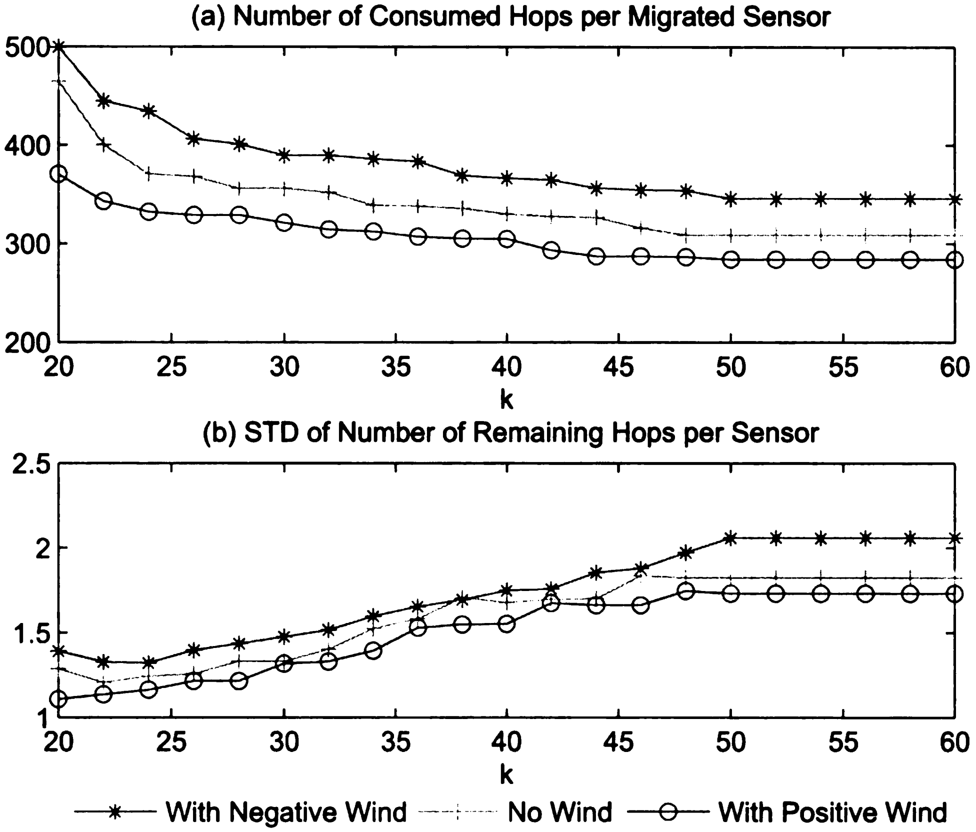


Figure 7.6. Number of Consumed Hops per Migrated Sensor and Mobility Metrics Vs. Upper Soft Limit k

Wind factors also slightly influence the results, but not as dramatic as k.

Larger k will yield better paths with fewer hops, but the STD of the number of remaining hops per sensor is also larger, which indicates imbalance of the mobility distribution throughout the network. The number of remaining hops per sensor also confirms the trend with a steep curve. When k is over a certain value (50 in this case), the parameters will not change since the constraints of the topology is reached. When k is too small, however, a path might not be found due to the topology constraints. It is also notable that under different wind conditions the influence of k follows the same trend. This indicates that we can set a k value and expect that it works well under different wind conditions. In practical deployments, the k value can be determined using simulations while taking into account various network and dynamic parameters. A tradeoff is to be made between obtaining the optimal path and maintaining an even network mobility distribution among sensors.

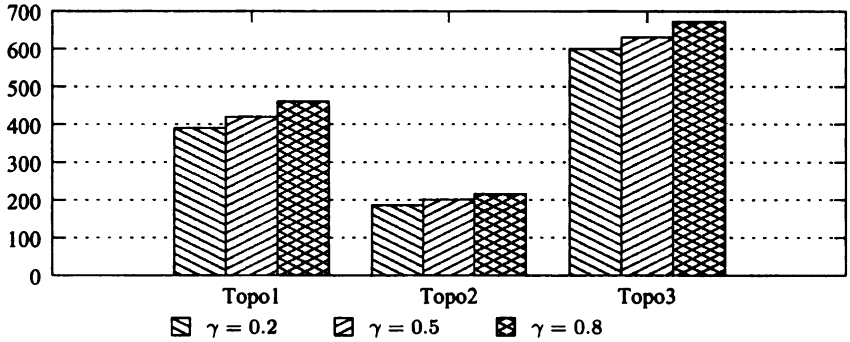


Figure 7.7. Number of Consumed Hops per Migrated Sensor ($\bar{H}_m$) under Different Topologies and γ

In the second simulation we evaluated the performance of Algorithm 3, where a dynamic approach is used to balance the need of path optimality and network mobility distribution. The advantage of this algorithm over Algorithm 1 is that the arbitrary hopping limits are not needed anymore. Instead, a scaling coefficient γ is used to determine the optimization target (as shown in Line 11 of Algorithm 3). In this simulation we evaluated the affect of the γ coefficient over the average number of hops consumed in a path and the mobility metrics of the network after the migration is performed.

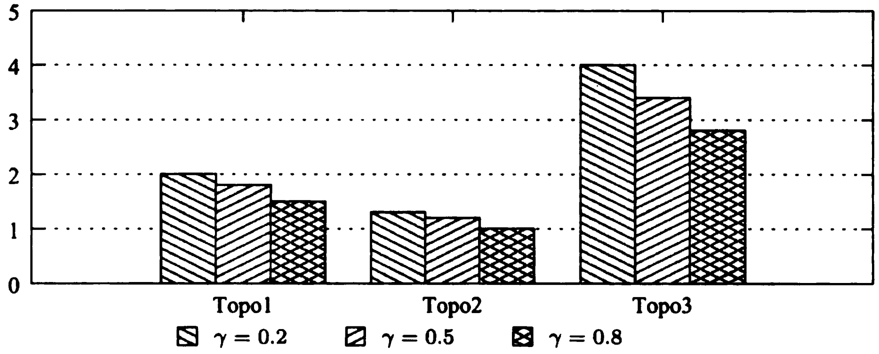


Figure 7.8. STD of Number of Remaining Hops per Sensor under Different Topologies and γ

As revealed in the first simulation, under different wind conditions, the path optimality and mobility metrics follow the same trend line as the hop limits change. Thus we did not simulate the cases with different wind influences when evaluating the performance of adjusting the hop limits dynamically using the γ coefficient. Unlike the upper and lower hopping limits, we argue that the dynamic method is more resilient to topological changes of the network. To validate this, we simulated the algorithm over three different topologies, namely Topol (the same as the topology used in the first simulation), Topo2, and Topo3, as indicated in the Figures 7.7 and 7.8. The results show that under different topologies, the coefficient has comparable balancing influences.

7.4 Summary

Hopping sensors are more adaptable to harsh terrains compared with wheeled mobile sensors. Low cost and large number of hopping sensors can be deployed into both military and civilian environments where dynamic and adaptable sensing capabilities are needed. Hopping sensors pose the challenges of being inaccurate while moving and susceptible to the influence of air disturbances.

We studied the multi-hop landing of the sensors under a multivariate normal distribution model and obtained an upper bound of the number of hops needed given the physical distance of the source and destination. The bound is further refined by considering the aerodynamics of air disturbances. Under such premises we focused on the problem of migrating a number of hopping sensors from multiple source clusters to a destination in a sensor network. We argue that cascaded relay hopping can speed up the migration but frequent relays may introduce unnecessary hops. We also want to conserve the dynamic relocation capability of as many clusters as possible. To achieve these goals we devised two algorithms. One uses upper and lower relay edge hop limits, while the other uses a balancing coefficient to construct a new optimization target dynamically. The two schemes are simulated using the physical and dynamic parameters of the Feddema prototype minefield hopping robots [60]. Simulation results indicated that both algorithms are effective in balancing the requirement of path optimality and maintaining the relocation capability of the network. The dynamic algorithm is also shown to be resilient to topological changes of the network. Further research includes implementing the route planning algorithms in a distributive fashion and studying the strategies and impacts of relocating the hopping sensors within clusters.

CHAPTER 8

Conclusions and Future Work

8.1 Conclusions

Remote sensing and control applications are a class of applications that will proliferate along with the development of communication technologies. By enabling the users to perform sensing and control tasks in remote locations, the Internet is turned from the traditional information highway to the "action highway". However, the media streams involved in teleoperated systems differentiate themselves from other media types in that they require both reliable and smooth delivery. Reliable delivery requires that the transport service have TCP style semantics. By being smooth, the transport service should be able to deliver the control and sensing data with both the average latency and the standard deviation of the latency bounded and reduced.

In this dissertation, a Supermedia TRansport over Overlay Networks (STRON) framework to improve the QoS of teleoperation systems over wide area networks is proposed. Multiple overlay paths are used to ensure reliability and timeliness of the transport service. Part of the work is published in [30, 28, 25]. A class of forward error correction encoding is used to eliminate the requirement of acknowledgments in typical reliable transport protocols. A QoS management framework is developed to allocate bandwidth among different supermedia streams and support efficient controlling the remote manipulators using QoS cues and sensor feedbacks, which is also published in [27]. A Transport service for Remote Sensing and Control (TRSC) over Wireless Networks is also proposed for MANET based applications. TRSC extended the STRON approach by building a reliable and efficient transport service based on multiple path routing protocols in MANETs. The proposed approaches were verified using experiments, network simulators and theoretical models and published in [29].

The possible challenges and problems of supporting remote sensing and control in cellular networks and pervasive environments are also studied. Supporting remote sensing and control applications in pervasive computing environments poses more challenges compared with traditional computing environments. We proposed an approach to utilize a cooperative local area wireless network to support data transport for remote sensing and control applications in pervasive computing environments. A novel mechanism is developed to determine the right number of proxies to use based on the MAC layer retransmission rate and is published in [26].

We also examine the organization structures of the networks that support remote sensing applications. Particularly we studied a type of sensor network that is cable of self-reorganization through sensor relocations powered by hopping. Hopping sensors are able to maintain mobility in harsh terrain but may lack the movement accuracy of those sensors that are powered by wheels. We focus on the problem of transporting a number of hopping sensors from multiple sources to a destination. Probabilistic methods are used to contain the movement inaccuracies along the hopping course. We also consider the impact of wind under an aerodynamic setting. Two transport schemes are designed to minimize the number of hops needed while considering other constraints, such as sustaining the capability of relocating sensors within the whole network. In one scheme we use upper and lower hopping limits to apply the network mobility constraints. The other scheme uses a balancing coefficient to construct a new optimization target to meet the requirement of path optimality and network mobility dynamically. Both schemes are simulated and evaluated using the physical and dynamic parameters of the prototype minefield hopping robots. The results show that both schemes work well regardless of the wind factors, while the dynamic scheme is also shown to be resilient to topological changes of the network.

8.2 Future Work

In the future work of the research, the proposed approaches to support remote sensing and control over wide area networks and wireless networks can be further refined and verified. Moreover, it is promising to explore the challenges and possible solutions to provide quality of service support for remote sensing and control applications in pervasive computing environments, as well as problems regarding novel self-organizing mobile sensor networks. Particularly, the possible researching directions are listed as follows.

- A pervasive computing environment often consists of different types of end devices and hosts. The heterogeneity of the system will invalidate the design principles of traditional homogeneous distributed systems. For example, the power and computing constraints of small devices may require the application to use more efficient algorithms. Thus one of the research topics might be to design the transport protocols and algorithms to accommodate the special needs of controllers, manipulators and sensors.

- In a pervasive computing environment, applications are desirably expected to be context aware and integrated well with other devices and surroundings. This requirement implicates many reliability, quality of service, invisibility and security issues when designing an application. For remote sensing and control applications, it will be an interesting topic to explore the methods that enable remote sensing and control devices to cope with frequent location changes and at the same time keep the impact on quality of service of the application to be minimal.

- In a heterogeneous environment with cellular and wireless LAN networks, the potential of using spatial diversity can be further exploited to dynamically adjust the traffic that are sent over each relay proxies. A network QoS measuring model can be deployed to actively monitor the capacity of each relay channels and allocate the amount of traffic that will be sent over each channel accordingly.

- Relocating mobile sensors powered by hopping is important to maintain the connectivity and coverage of a sensor network. The current centralized path planning approach can be extended to support distributed planning and decision making to further reduce the possibility of failure and increase the resilience of sensor network structure. Future study can also explore the cases when multiple destination clusters need to be filled with adequate sensors. In that situation a priority based system can be developed to reach an optimal route planning strategy.

BIBLIOGRAPHY

[1] 6Bone. 6Bone: testbed for deployment of IPv6. *http://www.6bone.net/*, 2003.

[2] Tarek Abdelzaher and Kang G. Shin. End-host Architecture for QoS-Adaptive Communication. In *IEEE Real Time Technology and Applications Symposium*, 1998.

[3] Abilene. Indiana University Abilene NOC Weathermap. *http://loadrunner.uits.iu.edu/weathermaps/abilene/*.

[4] E. Altman, K. Avrachenkov, and C. Barakat. A Stochastic Model of TCP/IP with Stationary Random Losses. *Proceedings of ACM SIGCOMM, Stockholm, Sweden*, 2000.

[5] D. Andersen, H. Balakrishnan, M. Kaashoek, and R. Morris. The case for resilient overlay networks. In *Proceedings of the 8th Annual Workshop on Hot Topics in Operating Systems (HotOSVIII)*, 2001.

[6] David Andersen, Hari Balakrishnan, Frans Kaashoek, and Robert Morris. Resilient Overlay Networks. In *18th ACM Symposium on Operating Systems Principles*, 2001.

[7] David G. Andersen, Alex C. Snoereny, and Hari Balakrishnan. Best-Path vs. Multi-Path Overlay Routing. In *IMC'03*, Miami Beach, Florida, USA, 2003.

[8] Apple. QuickTime. *http://www.apple.com/quicktime/*.

[9] ATMForum. ATM Forum. *http://www.atmforum.com/*.

[10] Luca Baldantoni, Henrik Lundqvist, and Gunnar Karlsson. Adaptive End-to-End FEC for Improving TCP Performance over Wireless Link. In *ICC 2004*, 2004.

[11] S. Banerjee, Bobby Bhattacharjee, and C. Kommareddy. Scalable application layer multicast. *ACM SIGCOMM, Pittsburgh, PA*, 2002.

[12] P. Bender, P. Black, M. Brob, R. Padovani, N. Sindhushayana, and A. Viterbi. CDMA/HDR: a Bandwidth Efficient High Speed Wireless Data Service for Nomadic Users. *IEEE Communications Magazine*, 2000.

[13] Vaduvur Bharghavan, Alan Demers, Scott Shenker, and Lixia Zhang. MACAW: A Media Access Protocol for Wireless LAN's. In *SIGCOMM*, pages 212–225, 1994.

[14] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An Architecture for Differentiated Services. *RFC 2475*, 1998.

[15] Johannes Blomer, Malik Kalfane, Richard Karp, Marek Karpinski, Michael Luby, and David Zuckerman. An xor-based erasure-resilient coding scheme. *Technical report, International Computer Science Institute, Berkeley, California*, 1995.

[16] K. Brady and T. J. Tarn. Internet-Based Remote Teleoperation. In *Proceedings of the 1998 IEEE International Conference on Robotics and Automation*, pages 644–650, Leuven, Belgium, 1998.

[17] R. Braynard, D. Kostic, A. Rodriguez, J. Chase, and A. Vahdat. Opus: An Overlay Peer Utility Service. *Proc. IEEE OpenArch '02*, 2002.

[18] Rebecca Braynard and Amin Vahdat. Masking Failures Using Anti Entropy and Redundant Independent Paths. In *SOSP Work in Progress Presentation*, 2001.

[19] Rob Brennan and Thomas Curran. SCTP Congestion Control: Initial Simulation Studies. In *Proceedings of the International Teletraffic Congress (ITC 17, Brazil)*, 2001.

[20] I. Busse, B. Deffner, and H. Schulzrinne. Dynamic QoS Control of Multimedia Applications Based on RTP. 1996.

[21] John W. Byers, Michael Luby, Michael Mitzenmacher, and Ashutosh Rege. A Digital Fountain Approach to Reliable Distribution of Bulk Data. In *SIGCOMM*, 1998.

[22] Bogdan Carbunar, Ananth Grama, Jan Vitek, and Octavian Carbunar. Redundancy and Coverage Detection in Sensor Networks. *ACM Transactions on Sensor Networks*, 2(1):94–128, 2006.

[23] Armando L. Caro, Jr., Keyur Shah, Janardhan R. Iyengar, Paul D. Amer, and Randall R. Stewart. SCTP and TCP Variants: Congestion Control Under Multiple Losses: Technical Report. Technical report, Computer and Information Science, University of Delaware, 2003.

[24] Shanwei Cen, Calton Pu, and Jonathan Walpole. Flow and congestion control for Internet streaming applicaitons. In *Multimedia Computing and Networking*, 1998.

[25] Zhiwei Cen, Amit Goradia, Matt Mutka, Ning Xi, Wai-keung Fung, and Yunhui Liu. Improving the Operation Efficiency of Supermedia Enhanced Internet Based Teleoperation via an Overlay Network. In *Proceedings of 2005 IEEE International Conference on Robotics and Automation*, Barcelona, Spain, 2005.

[26] Zhiwei Cen and Matt Mutka. QoS Provision for Remote Sensing and Control in Heterogeneous Environments. In *Proceedings of the Third IEEE International Conference on Mobile Ad-Hoc and Sensor Systems (MASS 2006)*, Vancouver Canada, 2006.

[27] Zhiwei Cen, Matt Mutka, Yang Liu, Amit Goradia, and Ning Xi. QoS Management of Supermedia Enhanced Teleoperation via Overlay Networks. In *Proceedings of 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2005)*, Edmonton, Alberta, Canada, 2005.

[28] Zhiwei Cen, Matt Mutka, Danyu Zhu, and Ning Xi. An Overlay Network Transport Service for Teleoperation Systems. *Technical Report MSU-CSE-05-01, Department of Computer Science and Engineering, Michigan State University*, 2005.

[29] Zhiwei Cen, Matt W. Mutka, Danyu Zhu, and Ning Xi. Improved Transport Service for Remote Sensing and Control over Wireless Networks. In *Proceedings of 2nd IEEE International Conference on Mobile Ad-Hoc and Sensor Systems (MASS 2005)*, Washington, DC, USA, 2005.

[30] Zhiwei Cen, Matt W. Mutka, Danyu Zhu, and Ning Xi. Supermedia Transport for Teleoperations over Overlay Networks. In Raouf Boutaba, Kevin Almeroth, Ramon Puigjaner, and et al., editors, *Proceedings of NETWORKING 2005: 4th International IFIP-TC6 Networking Conference*, volume 3462/2005, University of Waterloo, Waterloo Ontario Canada, 2005.

[31] Yatin Chawathe, Steven McCanne, and Eric A. Brewer. RMX: Reliable Multicast for Heterogeneous Networks. *INFOCOM*, 2000.

[32] Sriram Chellappan, Xiaole Bai, Bin Ma, and Dong Xuan. Sensor Networks Deployment Using Flip-Based Sensors. In *Proceedings of 2nd IEEE International Conference on Mobile Ad-Hoc and Sensor Systems (MASS 2005)*, Washington, DC, USA, 2005.

[33] Shigang Chen and Klara Nahrstedt. Distributed QoS Routing with Imprecise State Information. In *Proceedings of 7th IEEE International Conference on Computer, Communications and Networks (ICCCN'98)*, pages 614–621, Lafayette, LA, 1998.

[34] Shigang Chen and Klara Nahrstedt. Distributed Quality-of-Service Routing in Ad-Hoc Networks. *IEEE Journal on Special Areas in Communications*, 17(8):1–18, 1999.

[35] S.Y. Cheung, M. Ammar, and X. Li. On the Use of Destination Set Grouping to Improve Fairness in Multicast Video Distribution. In *IEEE INFOCOM*, 1996.

[36] A. Chobanyan, M. W. Mutka, V. S. Mandrekar, and N. Xi. Modeling Available Bandwidth for an Efficient QoS Characterization of a Network Path. In *Proceedings of NETWORKING 2005, 4th International IFIP-TC6 Networking Conference*, pages 316–327, 2005.

[37] Alexander Chobanyan, Matt Mutka, Zhiwei Cen, and Ning Xi. One Way Delay Trend Detection for Available Bandwidth Measurement. In *Proceedings of 2005 IEEE GLOBECOM Conference*, St Louis, MO, USA, 2005.

[38] Y. Chu, S. G. Rao, and H. Zhang. A Case for End System Multicast. *Proc. ACM SIGMETRICS*, 2000.

[39] I. Cidon, A. Khamisy, and M. Sidi. Analysis of packet loss process in high-speed networks. *EEE Transactions on Information Theory*, IT-39(1):98–108, 1993.

[40] Mirko Confente, Claudio Cosma, Paolo Fiorini, and Joel Burdick. Planetary Exploration Using Hopping Robots. In *7th ESA Workshop on Advanced Space Technologies for Robotics and Automation 'ASTRA 2002'*, ESTEC, Noordwijk, The Netherlands, 2002.

[41] Jorge Cortes, Sonia Martinez, Timur Karatas, and Francesco Bullo. Coverage control for mobile sensing networks. In *Proceedings of the 2002 IEEE International Conference on Robotics and Automation*, Washington, DC, 2002.

[42] Weidong Cui, Ion Stoica, and Randy H. Katz. Backup Path Allocation Based on a Correlated Link Failure Probability Model in Overlay Networks. In *10th IEEE International Conference on Network Protocols (ICNP'02)*, Paris, France, 2002.

[43] Swades De, Chunming Qiao, and Hongyi Wu. Meshed multipath routing with selective forwarding: an efficient strategy in wireless sensor networks. *Computer Networks*, 43(2003):481–497, 2003.

[44] B. Deb, S. Bhatnagar, and B. Nath. ReInForM: Reliable Information Forwarding Using Multiple Paths in Sensor Networks. In *The 28th Annual IEEE Conference on Local Computer Networks (LCN)*, 2003.

[45] Brian P. DeJong, J. Edward Colgate, and Michael A. Peshkin. Improving Teleoperation: Reducing Mental Rotations and Translations. In *Proceedings of the 2004 IEEE International Conference on Robotics and Automation*, New Orleans, LA, 2004.

[46] SUHAS N. DIGGAVI, NAOFAL AL-DHAHIR, A. STAMOULIS, and A. R. CALDERBANK. Great Expectations: The Value of Spatial Diversity in Wireless Networks. *PROCEEDINGS OF THE IEEE*, 92(2):219–270, 2004.

[47] Z. Duan, Z. Zhang, and Y. T. Zhou. Bandwidth Provisioning for Service Overlay Networks. *Proc. SPIE ITCOM Scalability and Traffic Control in IP Networks (II) July '02*, 2002.

[48] Zhenhai Duan, Zhi-Li Zhang, and Yiwei Thomas Hou. Service Overlay Networks: SLAs, QoS and Bandwidth Provisioning. *10th IEEE International Conference on Network Protocols (ICNP '02)*, 2002.

[49] N.G. Duffield, T.V. Lakshman, and D. Stiliadis. On Adaptive Bandwidth Sharing with Rate Guarantees. In *INFOCOM*, 1998.

[50] Ken Duffy, David Malone, and Douglas J. Leith. Modeling the 802.11 Distributed Coordination Function in Non-saturated Conditions. *IEEE COMMUNICATIONS LETTERS, VOL. 9, NO. 8, AUGUST 2005*, 9(8), 2005.

[51] E.Amir, S. McCanne, and H. Zhang. An Application-level Video Gateway. In *ACM Multimedia*, 1995.

[52] eDonkey. *http://www.edonkey2000.com/*.

[53] I. Elhajj, H. Hummert, N. Xi, Y. H. Liu, and W. J. Li. Synchronization and control of supermedia transmission via the Internet. *International Symposium on Intelligent Multimedia, Video and Speech 2001*, pages 320–323, 2001.

[54] Imad Elhajj, Henning Hummert, Ning Xi, and Yun hui Liu. Real-Time Haptic Feedback in Internet-Based Telerobotic Operation. In *Electro/Information Technology Conference*, Chicago, 2000.

[55] Imad Elhajj, Jindong Tan, Ning Xi, Wai Keung Fung, Yun hui Liu, Tomoyuki Kaga, and Toshio Fukuda. Multi-Site Internet-Based Cooperative Control of Robotic Operations. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Japan, 2000.

[56] Imad Elhajj, Ning Xi, Wai Keung Fung, Yun hui Liu, Tomoyuki Kaga, and Toshio Fukuda. Supermedia in Internet-Based Telerobotic Operations. In *MMNS 2001*, 2001.

[57] Imad Elhajj, Ning Xi, and Yun hui Liu. Real-Time Bilateral Control of Internet-Based Teleoperation. In *The Third World Congress on Intelligent Control and Automation*, 2000.

[58] Imad Elhajj, Ning Xi, and Yun hui Liu. Real-Time Control of Internet Based Teleoperation with Force Reflection. In *IEEE International Conference on Robotics and Automation, San Francisco*, 2000.

[59] Qing Fang, Jie Gao, and Leonidas J. Guibas. Locating and Bypassing Routing Holes in Sensor Networks. In *INFOCOM 2004*, 2004.

[60] John T. Feddema and David Schoenwald. Decentralized control of cooperative robotic vehicles. *IEEE Transactions on Robotics and Automation*, 18(5):852–864, 2002.

[61] W. Fischer and K. Meier-Hellstem. The Markov-modulated Poisson Process (MMPP) Cookbook. *Performance Evaluation, Vol 18*, 1992.

[62] S. Floyd. TCP and Explicit Congestion Notification. *ACM Computer Communication Review*, 24(5):10–23, 1994.

[63] S. Floyd and V. Jacobson. Random Early Detection Gateways for Congestion Avoidance. *ACM Transactions on Networks*, 1(4):397–413, 1993.

[64] Sally Floyd, Mark Handley, Jitendra Padhye, and Joerg Widmer. Equation-Based Congestion Control for Unicast Applications. In *SIGCOMM 2000*, 2000.

[65] Sally Floyd, Eddie Kohler, and Jitendra Padhye. Profile for DCCP Congestion Control ID 3: TFRC Congestion Control. *draft-ietf-dccp-ccid3-03.txt, work in progress*, 2003.

[66] Jonathan Friedman, David C Lee, Ilias Tsigkogiannis, Sophia Wong, Dennis Chao, David Levin, William J Kaiser, and Mani B Srivastava. RAGOBOT: A New Platform for Wireless Mobile Sensor Networks. In *Proceedings of the 1st IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS 2005)*, 2005.

[67] Wai Keung Fung, Ning Xi, Wang-tai Lo, BooHeon Song, Yu Sun, and Yun hui Liu. Task Driven Dynamic QoS based Bandwidth Allocation for Real-time Teleoperation via the Internet. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2003.

[68] A. Genz. Numerical Computation of Multivariate Normal Probabilities. *J. Comp. Graph. Stat.*, 1992.

[69] David Gesbert, Mansoor Shafi, Da-shan Shiu, Peter J. Smith, and Ayman Naguib. From Theory to Practice: An Overview of MIMO Space-Time Coded Wireless Systems. *IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS*, 21(3), 2003.

[70] Robert Ghrist and Abubakr Muhammad. Coverage and Hole-detection in Sensor Networks via Homology. In *The Fourth International Symposium on Information Processing in Sensor Networks (IPSN'05)*, UCLA, Los Angeles, California, USA, 2005.

[71] Gnutella. *http://www.gnutella.com/*.

[72] Amit Goradia, Zhiwei Cen, Ning Xi, and Matt Mutka. Modeling and Design of Mobile Surveillance Networks Using a Mutational Analysis Approach. In *Proceedings of 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2005)*, Edmonton, Alberta, Canada, 2005.

[73] Amit Goradia, Ning Xi, Mathew Prokos, Zhiwei Cen, and Matt Mutka. Cooperative Multi-Target Surveillance Using a Mutational Analysis Approach. In *Proceedings of 2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2005)*, Monterey, California, USA, 2005.

[74] Amit Gupta. Rate Based Flow and Congestion Control Schemes. *Department of Computer Science, University of Arizona*, 2002.

[75] Piyush Gupta and P. R. Kumar. The Capacity of Wireless Networks. *IEEE Transactions on Information Theory*, 46(2):388–404, 2000.

[76] T. Hacker, B. Athey, and B. Noble. The End-to-End Performance Effects of Parallel TCP Sockets on a Lossy Wide-Area Network. *Proceedings of the 16th International Parallel and Distributed Processing Symposium (IPDPS)*, 2002.

[77] Thomas J. Hacker, Brian D. Noble, and Brian D. Athey. The Effects of Systemic Packet Loss on Aggregate TCP Flows. *Conference on High Performance Networking and Computing*, 2002.

[78] Eric Hale, Nathan Schara, Joel Burdick, and Paolo Fiorini. A minimally actuated hopping rover for exploration of celestialbodies. In *Proceedings of IEEE International Conference on Robotics and Automation, 2000. ICRA '00*, pages 420–427, San Francisco, CA, USA, 2000.

[79] Mark Handley, Sally Floyd, Jitendra Padhye, and Joerg Widmer. TCP Friendly Rate Control (TFRC): Protocol Specification. *RFC 3448, Proposed Standard*, 2003.

[80] M. Hluchyj. Closed-loop rate-based traffic management. Technical report, Technical Report 94-0211R3, ATM Forum Traffic Management subworking group, 1994.

[81] Andrew Howard, Maja J Mataric, and Gaurav S Sukhatme. Mobile Sensor Network Deployment using Potential Fields: A Distributed, Scalable Solution to the Area Coverage Problem. In *Proceedings of International Symposium on Distributed Autonomous Robotics Systems (DARS)*, Fukupka, Japan, 2002.

[82] Pai-Hsiang Hsiao, H.T. Kung, and Koan-Sin Tan. Active Delay Control for TCP. In *IEEE Globecom '01*, 2001.

[83] J. Ibriq and I. Mahgoub. Cluster-based routing in wireless sensor networks: Issues and challenges. In *International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS'04)*, pages 759–766, San Jose, California, USA, 2004.

[84] Yutaka Ishibashi, Shuji Tasaka, and Hiroki Ogawa. A Comparison of Media Synchronization Quality among Reactive Control Schemes. In *INFOCOM*, 2001.

[85] A. Ishtiaq, Y. Okabe, and M. Kanazawa. Modified Congestion Control of SCTP over Wide Area Networks. In *Proceedings of the 2nd International Conference on Information Technology for Application (ICITA 2004)*, 2004.

[86] ITU/ISO. Reference Model of Open Distributed Processing - Part 2: Descriptive Model. *Draft International Standard 10746-2, Draft ITU-T Recommendation X.902*, 1994.

[87] Manish Jain and Constantinos Dovrolis. End-to-end available bandwidth: Measurement methodology, dynamics, and relation with TCP throughput. *Proceedings of SIGCOMM*, 2002.

[88] Manish Jain and Constantinos Dovrolis. Pathload: A Measurement Tool for End-to-end Available Bandwidth. In *Proc. of Passive and Active Measurement Workshop*, Fort Collins, CO, 2002.

[89] Manish Jain and Constantinos Dovrolis. End-to-end available bandwidth: Measurement methodology, dynamics, and relation with TCP throughput. *ACM/IEEE Transactions on Networking*, 2003.

[90] R. Jain. Congestion Control in Computer Networks: Issues and Trends. *IEEE Network Magazine*, 4(3):24–30, 1990.

[91] J. Jannotti, D. K. Gifford, K. L. Johnson, M. F. Kaashoek, and Jr. J. W. O'Toole. Overcast: Reliable Multicasting with an Overlay Network. *Proc. 4th USENIX OSDI*, 2000.

[92] Xiaohua Jia, Deying Li, and Dingzhu Du. QoS Topology Control in Ad Hoc Wireless Networks. In *IEEE INFOCOM*, 2004.

[93] W. Jiang and H. Schulzrinne. Modeling of Packet Loss and Delay and Their Effect on Real-Time Multimedia Service Quality. *10th International Workshop on Network and Operating System Support for Digital Audio and Video*, 2000.

[94] D. B. Johnson and D. A. Maltz. Dynamic Source Routing in Ad-Hoc Wireless Networks. *Mobile Computing*, pages 153–181, 1996.

[95] Niranjan Joshi, Srinivas R. Kadaba, Sarvar Patel, and Ganapathy S. Sundaram. Downlink scheduling in CDMA data networks. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, Boston, Massachusetts, United States, 2000.

[96] S.-S. Kang and Matt W. Mutka. A Mobile Peer-to-Peer Approach for Supporting Multimedia Content Sharing using 3G/WLAN Dual Mode Channels. *Wireless Communications and Mobile Computing Journal, Special Issue on WLAN/3G Integration for Next-Generation Heterogeneous Mobile Data Networks*, 5(6):633–645, 2005.

[97] P. Karn. MACA - A new channel access method for packet radio. In *ARRL/CRRL Amateur Radio 9th Computer Networking Conference*, pages 134–140, 1990.

[98] Kazaa. *http://www.kazaa.com/us/index.htm.*

[99] Angelos D. Keromytis, Vishal Misra, and Dan Rubenstein. SOS: Secure Overlay Services. In *SIGCOMM'02*, Pittsburgh, Pennsylvania, USA, 2002.

[100] Eddie Kohler, Mark Handley, and Sally Floyd. Designing DCCP: Congestion Control Without Reliability. *Technical Report, http://www.icir.org/kohler/dcp/dccp-icnp03s.pdf*, 2003.

[101] Eddie Kohler, Mark Handley, Sally Floyd, and Jitendra Padhye. Datagram Congestion Control Protocol. *draft-ietf-dccp-spec-04.txt, work in progress*, 2003.

[102] J. Nicholas Laneman, David N. C. Tse, and Gregory W. Wornell. Cooperative Diversity in Wireless Networks: Efficient Protocols and Outage Behavior. *IEEE TRANSACTIONS ON INFORMATION THEORY*, 50(12):3062–3080, 2004.

[103] S. Lee and M. Gerla. Dynamic load-aware routing in ad hoc networks. In *IEEE International Conference on Communications*, volume 10, pages 3206–3210, 2001.

[104] S.-J. Lee and M. Gerla. Split Multipath Routing with Maximally Disjoint Paths in Ad Hoc Networks. *IEEE International Conference on Communications*, 10, 2001.

[105] Seoung-Bum Lee, Gahng-Seop Ahn, Xiaowei Zhang, and Andrew T. Campbell. INSIGNIA: An IP-Based Quality of Service Framework for Mobile ad Hoc Networks. *Journal of Parallel and Distributed Computing*, 60(4):374–406, 2000.

[106] Z. Li and P. Mohapatra. HostCast: A New Overlay Multicasting Protocol. *Proc. IEEE Int. Communications Conference (ICC)*, 2003.

[107] Z. Li and P. Mohapatra. Impact of Topology on Overlay Routing Service. *IEEE INFOCOM*, 2004.

[108] Z. Li and P. Mohapatra. QRON: QoS-aware Routing in Overlay Networks. *Tthe Special Issue on Service Overlay Networks in the IEEE Journal on Selected Areas in Communications*, 2004.

[109] Wen-Hwa Liao, Yu-Chee Tseng, Shu-Ling Wang, and Jang-Ping Sheu. A Multi-Path QoS Routing Protocol in a Wireless Mobile Ad Hoc Network. *Telecommunication Systems*, 19(3-4):329–347, 2002.

[110] Haejung Lim, Kaixin Xu, and Mario Gerla. TCP Performance over Multipath Routing in Mobile Ad Hoc Networks. In *IEEE ICC'03*, Anchorage, Alaska, 2003.

[111] Shunan Lin, Yao Wang, Shiwen Mao, and Shivendra Panwar. Video transport over ad hoc networks using multiple paths. In *Proceedings of The 2002 IEEE International Symposium on Circuits and Systems (ISCAS)*, volume 1, pages 57–60, Scottsdale, AZ, 2002.

[112] Benyuan Liu, Dennis L. Goeckel, and Don Towsley. TCP-Cognizant Adaptive Forward Error Correction in Wireless Networks. In *IEEE INFOCOM*, 2002.

[113] H. Luo, R. Ramjee, P. Sinha, L. Li, and S. Lu. UCAN: A Unified Cellular and Ad-Hoc Network Architecture. *MOBICOM2003*, pages 353–367, 2003.

[114] Rui Ma and Jacek Ilow. Reliable Multipath Routing with Fixed Delays in MANET Using Regenerating Nodes. In *28th Annual IEEE International Conference on Local Computer Networks*, Bonn/Konigswinter, Germany, 2003.

[115] Shiwen Mao, Dennis Bushmitch, Sathya Narayanan, and Shivendra S. Panwar. MRTP: A Multi-Flow Realtime Transport Protocol for Ad Hoc Networks. In *Proceedings of the Fall IEEE Vehicular Technology Conference*, Orlando, Florida, 2003.

[116] S. McCanne, V. Jacobson, and M. Vetterli. Receiver Driven Layered Multicast. In *SIGCOMM*, 1996.

[117] Philip K. McKinley, Chiping Tang, and Arun P. Mani. A Study of Adaptive Forward Error Correction for Wireless Collaborative Computing. *IEEE Transactions on Parallel and Distributed Systems*, 13(9):936–947, 2002.

[118] Kamesh Medepalli and Fouad A. Tobagi. System Centric and User Centric Queueing Models for IEEE 802.11 based Wireless LANs. In *Proceedings of IEEE Broadnets, 2005*, 2005.

[119] Puneet Mehra, Christophe De Vleeschouwer, and Avideh Zakhor. Receiver-driven bandwidth sharing for TCP. In *IEEE INFOCOM*, San Francisco, USA, 2003.

[120] Microsoft. Microsoft Windows Mediaplayer. *http://windowsmedia.com/*.

[121] M. Mirhakkak, N. Schult, and D. Thomson. Dynamic quality-of-service for mobile ad hoc networks. In *International Conference on Mobile Computing and Networking*, pages 137–138, Boston, Massachusetts, 2000.

[122] Akihiro Nakao, Larry Peterson, and Andy Bavier. A Routing Underlay for Overlay Networks. In *SIGCOMM*, Karlsruhe, Germany, 2003.

[123] Napster. *http://www.napster.com/*.

[124] Jiri Navratil and R. Les. Cottrell. ABwE :A Practical Approach to Available Bandwidth Estimation. In *Passive and Active Measurements (PAM) Workshop*, 2003.

[125] P. Newman. Traffic Management for ATM Local Area Networks. *IEEE Communications Magazine*, 32(8):44–50, 1994.

[126] Thinh Nguyen and Avideh Zakhor. Path Diversity with Forward Error Correction (PDF) System for Packet Switched Networks. In *IEEE INFOCOM*, 2003.

[127] Thinh Nguyen and Avideh Zakhor. Path Diversity with Forward Error Correction (PDF) System for Packet Switched Networks. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 1, pages 663 – 672, 2003.

[128] G. Niemeyer and J. E. Slotine. Towards Force-Reflecting Teleoperation Over the Internet. In *IEEE International Conference on Robotics and Automation*, 1997.

[129] Hiroki Nishikawa, Takuya Asaka, and Tatsuro Takahashi. ABdis: Approach to Estimate Available Bandwidth Distribution Using a Multi-rate Probe. In *The second International Conference on Communication and Broadband Networking*, 2004.

[130] NS. VINT Project U.C. Berkeley/LBNL, NS2 Network Simulator. *http://www-mash.cs.berkeley,edu/ns/.*

[131] Petter Ogren, Edward Fiorelli, and Naomi Ehrich Leonard. Cooperative Control of Mobile Sensor Networks: Adaptive Gradient Climbing in a Distributed Environment. *IEEE Transactions on Automatic Control*, 49(8):1292–1302, 2004.

[132] Jitendra Padhye, V. Firoiu, D. Towsley, and J. Kurose. Modeling TCP Throughput: A Simple Model and its Empirical Validation. In *Proceedings of ACM Sigcomm*, Vancouver, Canada, 1998.

[133] Long Pan, Amit Goradia, and Ning Xi. Dynamic Multi-objective Optimal Task Distribution for Tele-Operated Mobile Manipulators. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sendai, Japan, 2004.

[134] V. Park and M. Corson. Temporally-ordered routing algorithm (TORA). *Version 1 functional specification, Internet Draft*, 1998.

[135] M.R. Pearlman and Z.J. Haas. Determining the optimal configuration for the zone routing protocol. *IEEE Journal on Selected Areas in Communications*, 17(8):1395–1414, 1999.

[136] Charles E. Perkins, Elizabeth M. Belding-Royer, and Samir R. Das. Ad-hoc on-demand distance vector routing. *IETF Draft, draft-ietf-manet-aodv-13.txt, Work in Progress*, 2003.

[137] Charles E. Perkins, Elizabeth M. Royer, and Samir R. Das. Quality of Service for Ad Hoc On-Demand Distance Vector Routing. *IETF Draft, draft-ietf-manet-aodvqos-00.txt, Work in Progress*, 2000.

[138] PlanetLab. PlanetLab: An open platform for developing, deploying, and accessing planetary-scale services. *http://www.planet-lab.org/.*

[139] J. B. Postel. Internet Control Message Protocol. *RFC 792*, 1980.

[140] Ravi Prasad, Manish Jain, and Constantinos Dovrolis. Effects of Interrupt Coalescence on Network Measurements. In *Passive and Active Measurements (PAM) Conference*, 2004.

[141] J. Proakis. Digital Communications: Third Edition. *McGraw-Hill, New York*, 1995.

[142] Qbone. *http://qbone.internet2.edu/.*

[143] C. Qiao and H. Wu. icar: An intelligent cellular and ad-hoc relay system. *Proceedings of IEEE IC3N*, 2000.

[144] Ed. R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin. Resource ReSer-Vation Protocol (RSVP) –Version 1 Functional Specification. *RFC 2205*, 1997.

[145] K. Ramakrishnan, S. Floyd, and D. Black. The Addition of Explicit Congestion Notification (ECN) to IP. *RFC 3168*, 2001.

[146] Sridhar Ramesh, Injong Rhee, and Katherine Guo. Multicast with Cache (Mcache): An Adaptive Zero-Deplay Video-on-Demand Service. In *INFOCOM*, 2001.

[147] Theodore S. Rappaport. *Wireless communications : principles and practice.* Prentice Hall PTR, 2002.

[148] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and Scott Shenker. A Scalable Content Addressable Network. *Proc. of ACM SIGCOMM*, 2001.

[149] R. Rejaie, M. Handley, and D. Estrin. RAP: An end-to-end rate-based congestion control mechanism for realtime streams in the internet. In *INFOCOM*, New York, NY, 1999.

[150] Injong Rhee, Volkan Ozdemir, and Yung Yi. Tear: Tcp emulation at receivers – flow control for multimedia streaming, technical report. Technical report, Department of Computer Science, North Carolina State University, 2000.

[151] L. Rizzo. Effective Erasure Codes for Reliable Computer Communication Protocols. *Computer Communication Review*, 1997.

[152] L. Roberts. Enhanced prca (proportional rate-control algorithm). Technical report, Technical Report 94-0735R1, ATM Forum Traffic Management subworking group, 1994.

[153] Elizabeth M. Royer and C.-K. Toh. A Review of Current Routing Protocols for Ad-Hoc Mobile Wireless Networks. *IEEE Personal Communications Magazine*, pages 46–55, 1999.

[154] Sambit Sahu, Philippe Nain, Don Towsley, Christophe Diot, and Victor Firoiu. On Achievable Service Differentiation with Token Bucket Marking for TCP. In *Proc. ACM SIGMETRICS'00*, Santa Clara, CA, 2000.

[155] N. Salem, L. Buttyan, J. Hubaux, and M. Jakobsson. A charging and rewarding scheme for packet forwarding in multi-hop cellular networks. In *ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, 2003.

[156] Kevin Savetz, Neil Randall, and Yves Lepage. *MBONE: Multicasting Tomorrow's Internet.* IDG, http://www.savetz.com/mbone/ edition, 1998.

[157] Steven L. Scott and Padhraic Smyth. The Markov Modulated Poisson Process and Markov Poisson Cascade with Applications to Web Traffic Modeling. *Bayesian Statistics, Volume 7, Oxford University Press*, 2003.

[158] A. Sendonaris, E. Erkip, and B. Aazhang. User cooperation diversity-Part I: System description. *IEEE Transactions on Communications*, 51(11):1927–1938, 2003.

[159] Sergio D. Servetto, Rohit Puri, Jean-Paul Wagner, Pierre Scholtes, and Martin Vetterli. Video Multicast in (Large) Local Area Networks. In *INFOCOM*, 2002.

[160] S. Shenker and J. Wroclawski. General Characterization Parameters for Integrated Service Network Elements. *RFC 2215*, 1997.

[161] Gabriel T. Sibley, Mohammad H. Rahimi, and Gaurav S. Sukhatme. Robomote: A Tiny Mobile Robot Platform for Large-Scale Sensor Networks. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA2002)*, 2002.

[162] P. Sinha, R. Sivakumar, and V. Bharghavan. CEDAR: a Core Extraction Distributed Ad hoc Routing Algorithm. In *IEEE INFOCOMM '99*, 1999.

[163] Hugh M. Smith, Matt W. Mutka, and Diane T. Rover. A Feedback Based Rate Control Algorithm for Multicast Transmitted Video Conferencing. *Journal of High Speed Networks*, 7(3-4):259–279, 1998.

[164] Hugh M. Smith, Matt W. Mutka, and Eric Torng. Bandwidth Allocation for Layered Multicasted Video. In *ICMCS*, 1999.

[165] R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, and V. Paxson. Stream Control Transmission Protocol. *RFC2960*, 2000.

[166] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications. *Proc. ACM SIGCOMM*, 2001.

[167] Ion Stoica, Daniel Adkins, Shelley Zhuang, Scott Shenker, and Sonesh Surana. Internet Indirection Infrastructure. In *Proceedings of ACM SIGCOMM 2002*, 2002.

[168] L. Subramanian, I. Stoica, H. Balakrishnan, and R.H.Katz. OverQoS: Offering Internet QoS using Overlays. *Proc. HotNet-I Workshop, October 2002*, 2002.

[169] Bernhard Suter, T.V. Lakshman, Dimitrios Stiliadis, and Abhijit Choudhury. Design Considerations for Supporting TCP with Per-flow Queueing. In *INFOCOM*, 1998.

[170] S.Y.Shi and J.S.Turner. Routing in Overlay Multicast Networks. *Proc. IEEE Infocom '02*, 2002.

[171] S. Tao, K. Xu, Y. Xu, T. Fei, L. Gao, R. Guerin, J. Kurose, D. Towsley, and Z.-L. Zhang. Exploring the performance benefits of end-to-end path switching. In *Proceedings of ICNP 2004*, Berlin, Germany, 2004.

[172] Chai-Keong Toh. Associativity-Based Routing For Ad-Hoc Mobile Networks. *Wireless Personal Communications*, 4(2):1–36, 1997.

[173] Warren Keith Toomey. *A Rate-Based Congestion Control Framework for Connectionless Packet-Switched Networks.* PhD thesis, University of New South Wales, 1997.

[174] V. Tsaoussidis and H. Badr. TCP-Probing: Towards an Error Control Schema with Energy and Throughput Performance Gains. *Proc. Of the 8th Conference on Network Protocols, ICNP 2000*, 2000.

[175] V. Tsaoussidis, H. Badr, K. Pentikousis, and X. Ge. Energy/Throughput Tradeoffs of TCP Error Control Strategies. *Proceedings of IEEE Symposium on Computers and Communications*, 2000.

[176] A. Veres, A. T. Campbell, M. Barry, and L. H. Sun. Supporting Service Differentation in Wireless Packet Networks Using Distributed Control. *IEEE Journal of Selected Areas in Communications*, 2001.

[177] Evangelos Vergetis, Roch Guerin, and Saswati Sarkar. Realizing the Benefits of User-Level Channel Diversity. *ACM SIGCOMM Computer Communication Review*, 35(5):15–28, 2005.

[178] Atul Verma, Hemjit Sawant, and Jindong Tan. Selection and Navigation of Mobile Sensor Nodes Using a Sensor Network. In *Third IEEE International Conference on Pervasive Computing and Communications (PERCOM'05)*, pages 41–50, Kauai Island, Hawaii, 2005.

[179] L. Vicisano, L. Rizzo, and J. Crowcroft. TCP-like Congestion Control for Layered Multicast Data Transfer. In *INFOCOM*, 1998.

[180] A. J. Viterbi. *CDMA: Principles of Spread Spectrum Communication.* Addison-Wesley, 1995.

[181] Andreas Vogel, Brigitte Kerherv, Gregor v. Bochmann, and Jan Gecsei. Distributed multimedia applications and quality of service: a survey. In *1994 conference of the Centre for Advanced Studies on Collaborative research*, 1994.

[182] Guiling Wang, Guohong Cao, and Tom La Porta. Movement-Assisted Sensor Deployment. In *IEEE INFOCOM*, 2004.

[183] Guiling Wang, Guohong Cao, Tom La Porta, and Wensheng Zhang. Sensor Relocation in Mobile Sensor Networks. In *IEEE INFOCOM*, 2005.

[184] L. X. Wang. *Adaptive Fuzzy Systems and Control: Design and Stability Analysis.* Prentice Hall, 1994.

[185] Cedric Westphal. Monitoring Proportional Fairness in CDMA2000 High Data Rate Networks. In *IEEE GLOBECOM 2004*, 2004.

[186] J. Wroclawski. The Use of RSVP with IETF Integrated Services. *RFC 2210*, 1997.

[187] Jie Wu and Shuhui Yang. SMART: A Scan-Based Movement-Assisted Sensor Deployment Method in Wireless Sensor Networks. In *IEEE INFOCOM*, 2005.

[188] X. Wu, S.-H. Chan, and B. Mukherjee. MADF: A Novel Approach to Add an Ad-hoc Overlay on a Fixed Cellular Infrastructure. In *Proceedings of IEEE WCNC*, Chicago, IL USA, 2000.

[189] Xbone. *http://www.isi.edu/xbone/.*

[190] Ning Xi. *PhD Thesis: Event-Based Planning and Control for Robotic Systems.* PhD thesis, Washington University, 1993.

[191] Ning Xi and T. J. Tarn. Action Synchronization and Control of Internet Based Telerobotic Systems. In *IEEE International Conference on Robotics and Automation*, 1999.

[192] H. Xiao, W. K. G. Seah, A. Lo, and K. C. Chua. A flexible quality of service model for mobile ad-hoc networks. In *IEEE VTC2000-spring*, Tokyo Japan, 2000.

[193] Qi Xue and Aura Ganz. Ad hoc QoS on-demand routing (AQOR) in mobile ad hoc networks. *Journal of Parallel and Distributed Computing*, 63:154–165, 2003.

[194] M. Yajnik, S. Moon, J. Kurose, and D. Towsley. Measurement and Modeling of the Temporal Dependence in Packet Loss. *Proceedings of IEEE INFOCOM'99*, 1999.

[195] Yang Richard Yang and Simon S. Lam. General aimd congestion control, technical report tr-00-09. Technical report, Departmentof Computer Sciences, The University of Texas, 2000.

[196] Zhenqiang Ye, Srikanth V. Krishnamurthy, and Satish K. Tripathi. A Framework for Reliable Routing in Mobile Ad Hoc Networks. In *INFOCOM*, 2003.

[197] Jungkeun Yoon, Mingyan Liu, and Brian Noble. Random Waypoint Considered Harmful. In *IEEE INFOCOM 2003*, 2003.

[198] F. Zhai, R. Berry, T. N. Pappas, and A. K. Katsaggelos. A rate-distortion optimized error control scheme for scalable video streaming over the Internet. In *Proc. IEEE ICME*, 2003.

[199] B. Zhang, S. Jamin, and L. Zhang. Host Multicast: A Framework for delivering Multicast to End Users. *Proc. Infocom '02*, 2002.

[200] Qian Zhang, Wenwu Zhu, and Ya-Qin Zhang. Resource Allocation for Multimedia Streaming Over the Internet. *IEEE Transactions on Multimedia*, 3(3), 2001.

[201] B. Y. Zhao, L .Huang, A. D. Joseph, and J. D. Kubiaotowicz. Tapestry: A Resilient Global-scale Overlay for Service Deployment. *IEEE Journal on Selected Areas in Communications*, 2004.

[202] Ben Y. Zhao, Ling Huang, Anthony D. Joseph, and John D. Kubiatowicz. Exploiting Routing Redundancy Using a Wide-area Overlay. *U. C. Berkeley Technical Report UCB/CSD-02-1215*, 2002.

[203] Ben Y. Zhao, Ling Huang, Jeremy Stribling, Sean C. Rhea, Anthony D. Joseph, and John D. Kubiatowicz. Tapestry: A Resilient Global-scale Overlay for Service Deployment. *IEEE Journal on Selected Areas in Communications, Vol 22, No. 1*, 2004.

[204] Dongmei Zhao and Terence D. Todd. Cellular CDMA Capacity with Out-of-Band Multihop Relaying. *IEEE TRANSACTIONS ON MOBILE COMPUTING*, 5(2):170–178, 2006.

[205] Jian Zhou, Huai-Rong Shao, Chia Shen, and Ming-Ting Sun. Multi-path Transport of FGS Video. *13th International Workshop on Packet Video*, 2003.

[206] X. Zhou, C.-Z. Xu, and J. Wei. Harmonic proportional bandwidth allocation for service differentiation on streaming servers. Technical report, CIC-03-05, Wayne State University Technical Report, 2003.

[207] Chenxi Zhu and M. Scott Corson. QoS routing for mobile ad hoc networks. In *INFOCOM*, 2002.

[208] D. Zhu, M. Gritter, and D. R. Cheriton. Feedback Based Routing. *Proc. ACM HotNet-I, 2002*, 2002.

[209] Danyu Zhu, Matt Mutka, and Zhiwei Cen. Using Cooperative Multiple Paths to Reduce File Download Latency. In *Proceedings of 2005 IEEE GLOBECOM Conference*, St Louis, MO, USA, 2005.

[210] Danyu Zhu and Matt W. Mutka. Promoting Cooperation Among Strangers to Access Internet Services from an Ad Hoc Network. *Pervasive and Mobile Computing*, 1(2):213–236, 2005.

[211] Danyu Zhu, Matt W. Mutka, and Zhiwei Cen. QoS Aware Wireless Bandwidth Aggregation (QAWBA) by Integrating Cellular and Ad-Hoc Networks. In *Proceedings of First International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks (QSHINE'04)*, pages 156–163, Dallas, Texas, 2004.

[212] Shelley Q. Zhuang, Ben Y. Zhao, Anthony D. Joseph, Randy Katz, and John Kubiatowicz. Bayeux: An Architecture for Scalable and Fault-tolerant Wide-area Data Dissemination. *Eleventh International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV 2001)*, 2001.

[213] Yi Zou and Krishnendu Chakrabarty. Sensor Deployment and Target Localization in Distributed Sensor Networks. *ACM Transaction on Embedded Computing Systems*, 3:61–91, 2004.