EXPERIMENTAL INVESTIGATION OF AUDIO AND VIDEO QUALITY IN MULTI-VIDEO STREAMING ENVIRONMENTS

BY

JIGAR HARSHAD DOSHI

B.E., University of Madras, 2004

THESIS

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer Science in the Graduate College of the University of Illinois at Urbana-Champaign, 2006

Urbana, Illinois
Abstract

This work addresses the problems of sending live media, in the form of audio and video over heterogeneous networks. Specifically we address streaming multiple video streams in a wireless network and streaming audio in a multi-video tele-immersive environment.

In the wireless domain, we explore the use of a user level Bandwidth Manager as QoS control to aid the transfer of adaptive video. We implement this system and test it on a linux based wireless testbed. We test the performance of multi video streaming in this environment. The results shows the advantages as well as some of the disadvantages of user space QoS control.

We also create a auction based migration scheme to ensure that QoS control is always available in the ad hoc network. Since the ad hoc network is made up of strangers, we design an incentive based mechanism to ensure cooperation. Towards this end we design and implement a Vickerey auction based migration scheme.

In the multi-video tele-immersive environment, we investigate the addition of audio to facilitate remote collaboration. We develop a H.323 based conferencing application to achieve this. We also work on the problem of audio-video synchronization and design a simple scheme which has good results.
In Memory of my Grandfather Vadilal Kalidas Doshi
Acknowledgments

First and foremost I would like to express my deepest gratitude to my advisor Klara Nahrstedt. Without her efforts this thesis would not be possible. I am thankful for her efforts to help me in spite of her busy schedule. Her vision helped steer the project.

I would also like to thank Samarth Shah for his help and support. A large part of this work is based on his research work. His insights helped greatly in my project.

I have had the great fortune of working with some very smart people during the course of my thesis. I would like to thank all my colleagues in the MONET group especially Zhenyu Yang, Hoang Nguyen, Bin Yu, Wanmin Wu and Jin Liang with whom I interacted with in my work.

I would like to thank my family, especially my parents, Harshad and Yashumati Doshi for their constant support and encouragement.

A part of this thesis has been supported by a NSF grant (NSF CNS 05-51665. NSF SCI 05-49242 and NSF CNS 05-20182), However the opinions expressed here are solely my own and not of the funding agency.

This thesis is dedicated to my late grandfather, Vadilal Kalidas Doshi for being supportive in whatever I did and for leading by example.
# Table of Contents

List of Tables ........................................................................ viii

List of Figures ......................................................................... ix

Chapter 1 Introduction .......................................................... 1
  1.1 Environment ..................................................................... 1
  1.2 Problem Scope ................................................................... 1
  1.3 Outline ............................................................................. 2

Chapter 2 Related Work .......................................................... 3
  2.1 Adaptively Transmitting Video Over Wireless Networks ............. 3
  2.2 Migration of Bandwidth Manager in a Wireless Network ............... 4
  2.3 Tele-Immersion ................................................................. 5

Chapter 3 Bandwidth Management Framework ............................ 8
  3.1 Network Model .................................................................. 9
  3.1.1 Assumptions ................................................................. 9
  3.1.2 Bandwidth Management System Architecture ....................... 10
  3.2 Link Level Module ............................................................. 13
  3.3 Bandwidth Management Middleware ....................................... 14

Chapter 4 Adaptive Video Streaming over Wireless Networks .......... 15
  4.1 Overview of H.263 ............................................................. 15
  4.1.1 H.261: Video Coding and Decoding (CODEC) ..................... 15
  4.1.2 H.263: ........................................................................ 16
  4.1.3 Encoder/Decoder Description ........................................... 17
  4.1.4 Motion Estimation and Compensation ................................. 18
  4.1.5 DCT Transform ............................................................ 19
  4.1.6 Quantization ................................................................. 19
  4.1.7 Entropy coding ............................................................. 20
  4.1.8 Coding Control ............................................................ 20
  4.2 System Architecture ........................................................ 21
  4.2.1 Rate Adaptor ............................................................... 21
  4.2.2 Adaptive Video Encoder ............................................... 22
  4.2.3 Decoder ...................................................................... 22
  4.3 Implementation ............................................................... 23
  4.4 Problems and Solutions ..................................................... 23
  4.5 Evaluation ....................................................................... 24
  4.5.1 Experimental Testbed .................................................... 24
  4.5.2 Metrics ...................................................................... 24
  4.5.3 Results ....................................................................... 25
  4.6 Explanation ..................................................................... 27
List of Tables

3.1 Explanation of Notation used in Bandwidth Management Protocol .......................... 10
5.1 Various Auction Settings ......................................................................................... 32
List of Figures

1.1 Overview of the problem area ................................................................. 2

3.1 Bandwidth Manager in an adhoc network .................................................. 9
3.2 Bandwidth Management System Architecture .......................................... 11
3.3 Bandwidth Management Protocol ............................................................ 11
3.4 Transmission Sequence ................................................................. 13

4.1 Comparison of H.261 and H.263 .......................................................... 16
4.2 Encoding Steps and Block Diagram ......................................................... 17
4.3 Decoder Description .............................................................................. 20
4.4 Topology of Experimental Setup ........................................................... 24
4.5 Frame rate in topology 1 where multiple senders send to one receiver. Here two video streams are sent and the frame rate at the receiver is measured over time. The graph shows the frame rate of each stream and the average frame rate. The Average frame rate in this experiment is found to be 12 fps ................................................................. 25
4.6 Frame rate measured in topology one with five concurrent streams. Here the frame rate falls and the average frame rate measured is 8 FPS ................................................................. 26
4.7 Comparison of average frame rate comparison in topology 1 with two, three, four and five concurrent streams ........................................................................ 26
4.8 Comparison of average frame rate comparison in topology 2 with multiple independent streams ................................................................. 27
4.9 Comparison of average frame rate of four streams in topology 1 versus the average frame rate of four streams in topology 2 ................................................................. 28

5.1 Modified Max-min Fair Resource Allocation Algorithm .............................. 34
5.2 Algorithm to Scale Weights & Calculate Costs ........................................... 36
5.3 Time required to shift the Bandwidth Manager as the number of nodes in the network increase ................................................................. 37
5.4 Data overhead to shift the Bandwidth manager as the number of nodes in the network increase ................................................................. 38
5.5 In this topology two video streams are sent to the same receiver. The first two streams are sent with a BM migrated using a Vickerey Auction and the last two streams are sent with a static BM. The dips in throughput at around time instant 40 are due to the vagaries of the wireless channel. Again there is no noticeable difference in throughput ................................................................. 39
5.6 This graph shows the throughput of H.263 Video in terms of the number of frames per second received. In this topology two independent video streams are sent simultaneously to different receivers. The first two streams are sent with a BM migrated using a Vickerey Auction and the last two streams are sent with a static BM. There is no noticeable difference in throughput. The BM is migrated at around time instant 20 ................................................................. 40
5.7 Variance of Utility Function .................................................................. 41

6.1 Tele-Immersion System Architecture ..................................................... 47
6.2 Design Space for 3D Audio ..................................................................... 47
6.3 Wavefield Effect to regenerate 3D Audio ................................................. 48
6.4 VOIP Protocol Stack .............................................................................. 50
6.5 Call Graph of OpenH323, The shaded area represents the classes of interest for modification. 53
6.6 Class Diagram of OpenH323 protocol. 53
6.7 Screen Shot of OpenPhone Application. 53
6.8 Screen Shot of Packets Lost statistics. 54
6.9 Scatter Plot showing the arrival of Audio Frames and Video Frames. 55

7.1 Plot showing the arrival of audio frames when compared to the arrival of the median macro video frames. 58
7.2 Plot showing the arrival of audio frames when compared to the arrival of the last macro video frame. 59
Chapter 1

Introduction

1.1 Environment

The rapid increase of processing power and the proliferation of network connectivity has lead to the prevalence of streaming multimedia applications. The capabilities present today allow both delivery and display of multiple streams of audio and video over heterogeneous IP based networks and diverse operating systems.

However, the presence of new application areas and domains now present interesting problems in the reliable delivery of such media. In particular the delivery of media over lossy wireless channels presents many challenges. A media streaming application needs to adapt to the conditions of the time variant wireless channel. The vagaries of the wireless channel necessitate the use of a QoS control framework. Since such networks are formed in an ad hoc manner they necessitate a mechanism to migrate the functionality of the QoS control across the network to ensure availability.

The delivery of media in the next generation three dimensional tele-immersive environments is equally challenging. The sheer scale of the data involved and the loosely defined notion of a media unit pose interesting problems in the delivery of audio and the synchronization of audio and video.

1.2 Problem Scope

In this thesis we investigate multimedia streaming issues in two very different environments, the wireless 802.11 environment and the 3D tele-immersive Internet2 based environment. In the first part of the thesis we look at video streaming problems and the delivery of adaptive video over a wireless channel. The environment necessitates the use of QoS control in the form of a Bandwidth Manager. To ensure availability we create a scheme to migrate the Bandwidth Manager across the network. The second part of the thesis deals with the streaming of audio in a multi video tele-immersive environment and synchronization of audio with the video.
Existing approaches to delivering QoS in the wireless domain have worked at different layers from the MAC layer up to the application layer. In the domain of wireless video streaming existing research has worked on the transmission of previously compressed video like MPEG files, not live video. In a tele-immersive setting, the use of audio has not been fully explored. Synchronization of audio with 3D video is also an open area.

The main contribution of the thesis in the wireless domain is an integrated cross-layer solution which allows the delivery of media over a wireless network. We use this framework to compress video using a H.263 encoder and deliver it over an ad hoc network. In teleimmersion, we add synchronized, live audio and stream it alongside multi-camera three-dimensional video.

1.3 Outline

The rest of the thesis is organized as follows. In Chapter 2, we look at the related work in streaming area. In Chapter 3, we cover the problem of adaptively transmitting video over a wireless network. In Chapter 4, we tackle the problem of ensuring availability if the centralized arbiter moves out of the network. In Chapter 5, we explore the problems of adding audio to a teleimmersive environment. In Chapter 6, we work on the synchronization of audio and video in a tele-immersive environment. An overview of the application areas covered is given in Figure 1.1.
Chapter 2

Related Work

IP based networks are best-effort, and not designed for the delivery of media like audio and video which have timing constraints. This has necessitated a number of approaches to stream media. We look at trying to extend these approaches to new challenging environments. Specifically we look at streaming video for wireless networks, and streaming audio over the fixed Internet along with 3D tele-immersive video. Both are extremely challenging and stress the networks in question.

We look at related work which covers the problems of transmitting media over wireless networks, auction based bandwidth management mechanisms and tele-immersive streaming.

2.1 Adaptively Transmitting Video Over Wireless Networks

The work which is most relevant to our work is the Dynamic Bandwidth Management for 802.11 Wireless Networks by Shah et al [30, 33] which looks at proportioning bandwidth among multiple flows which share the same network. Since the 802.11 network is best effort and does not have any delivery guarantees, achieving QoS guarantees on such networks becomes challenging. This work looks at giving loose bandwidth guarantees to media streams over single hop ad hoc networks. Since the network is highly variable, the Bandwidth Manager changes the resource allocation as the network state changes. We use the bandwidth management functionality outlined in this paper to calculate the video transmission rate. As the network state changes the Bandwidth Manager informs our application of the new bandwidth to send the video at. Since these papers forms the basis of our work, it is summarized in Chapter 3.

A related project is Stateless Wireless Ad Hoc Networks (SWAN)[4]. SWAN relies extensively on explicit congestion notification to regulate real time traffic. A stateless network protocol has the advantage that the space requirements at each node is considerably lower. In a single-hop ad hoc network, the number of flows is restricted and hence the information stored at each node is low. Therefore the advantages of a stateless protocol are not significant. The notion of bandwidth used in the paper does not take into account that different nodes in the network may have different instantaneous bandwidth. This is not accommodated in
the allocation of bandwidths. Thus the architecture proposed in [33] is superior for our intended application.

An interesting approach to bandwidth allocation is “Allocation of layer bandwidths and FEC’s for video multicast over wired and wireless networks” by Lee et al. [21] which looks at multi layered video and the allocation of optimal bandwidth and FEC to tackle a heterogeneous wireless environment. The system uses transcoding and proposes a dynamic approach to the allocation of layered forward error correction codes. The interesting aspect of this work is the use of transcoding to deliver the video on a last mile wireless link. FEC’s are allocated dynamically according to network state. While the work is interesting for hybrid networks, for our current focus which is adaptive delivery of video in a single hop ad hoc networks, such an approach is not required.

The MIT OxygenTV project has extensively studied sending adaptive video over wireless networks. Representative publications include [14], [15] and [13]. A big difference between the Oxygen TV project and our project is that it looks at the adaptive delivery of stored video content like MPEG 4 streams. Our project on the other hand, looks at the adaptive delivery of live video streams with the help of the H.263 encoder which is more challenging. The scenario envisaged by the Oxygen TV is video broadcast, we envisage a conferencing scenario which would allow video phone type services. Since the domains are so similar, the project has greatly influenced our work.

2.2 Migration of Bandwidth Manager in a Wireless Network

Since the bandwidth management facility consumes resources, We look at a migration scheme to ensure availability. However since the network is made up of strangers incentive schemes have to be introduced.

Two classical election algorithms for distributed systems are the Bully [17] algorithm and the Ring algorithm [2] which could be used to elect the new BM for migration. Both these algorithms expect honest parameters for our network. Election algorithms have also been proposed for multi-hop wireless ad hoc networks like those covered in [12] and [24]. But these algorithms look at holding secure elections in an insecure environment. There is no way to ensure a node’s honesty without incentive-based mechanisms.

Since the proposed scheme in Chapter 4 uses auction mechanisms based on Game theory, game theoretic mechanisms applied to wireless networks important to our work. The problems of cooperative access to the internet in an environment with strangers is covered by Zhu et al in [39]. Each node is assigned a pair wise reputation, which determines its connectivity. The approach followed by Lai et al. in [29] is to play a packet forwarding game among all nodes in the network. Each node is ranked with a reputation based on its packet forwarding behavior observed by nodes in the same neighborhood. Hence, a node with a bad reputation is
isolated. However these papers do not cover mechanisms to ensure node honesty.

The bandwidth management framework which is the basis of this work is covered in [30, 33]. The presence of one node which acts as a Bandwidth Manager is assumed. The schemes concentrate on fairness and do not prioritize allocation of bandwidth. Extensions of this work include a price based [31] approach which uses a payment scheme to regulate bandwidth usage. Another extension is iPass [19]. This work looks at providing economic incentives to forward packets in a hybrid ad-hoc and base station scheme. However both approaches do not touch on service provisioning in the network itself.

Auction schemes have been widely studied in economics. Our scheme borrows heavily from the work of Liu et al. [22] who have proposed and studied single round second price auctions. The wealth of work in Distributed Algorithm Mechanism Design (DAMD) [37] also guides our work. Together these papers give an overview of Vickerey auction scheme which we use. DAMD has been used to solve some ad hoc network and MANET problems. For example, Ad-Hoc VCG [5] is a MANET routing algorithm for networks with selfish agents. Clementi et al have looked at range assignments in ad hoc networks when the nodes are selfish [8].

There has been work on bandwidth allocation in communication networks using auctions [23]. This paper looks at auctioning the actual bandwidth used by the user. This differs greatly from our work where we auction the role of being a Bandwidth Manager.

### 2.3 Tele-Immersion

Tele-Immersion represents the next generation of electronic interaction. It enables users in physically remote spaces to interact with one another in a shared space that mixes both local and remote realities, and allows participants to share a mutual sense of presence. There has been concurrent work in teleimmersion which are represented by four main projects which are listed below.

The TeleImmersion project at the University of Pennsylvania [10] focuses on the study and development of systems that can scan wide-area dynamic scenes and create 3D view-independent representations. With such view independent representations geographically distributed sites can collaborate in real time in a shared simulated environment as if they were in the same physical room. A demo of the setup shown in Thinquest Live shows remote interaction with the help of head tracking equipment. However the main focus of the system is a virtual office where some of the users can be local whereas some of the users are remote. The research groups efforts have been summarized in [35]. A key challenge of their setup was to create an accurate immersive display such that there is a continuum between the local and the remote sites.

The Brown University Tele-Immersion project [36] is a part of the National Tele-Immersive Initiative,
whose other members include the University of North Carolina, Caltech and Cornell. The specific focus of the research at Brown University has been user interfaces and software infrastructure for tele-immersion.

The Office of the Future Project at the university of North Carolina at Chapel Hill is a part of the same initiative and has worked on a number of projects related to 3D Tele-Immersion. Some of the interesting projects include Group-TeleImmersion [20], where multiple camera views are stitched together to create a composite wide angle view. The stream is still two dimensional and no 3D reconstruction is performed like in our scheme. Another interesting project by the same group is the Immersive Electronic Book which creates 3D electronic books for surgical training [38].

The Coliseum system [7] by Hewlett Packard attempts at creating a tele-immersive system using a single computer. Instead of having a virtual tele-immersive room, a camera cluster attached to a PC is used for remote collaboration. Such a system can be used for better video conferencing, But does not allow for tele-immersive performances.

The University of Illinois at Chicago has worked extensively in the area of 3D Tele-Immersion. However their main research focus has been 3D Data Exploration [6]. Participants are represented by Avatars and can interact with a visualization of data.

The CMU Virtualized Reality project [26] uses a 3D room with a number of cameras which simultaneously capture the user from multiple angles. A flyby of the user can be created by switching multiple camera angles using composition. This system is offline and not meant for interactive performance.

Now we look at related work in the transmission of audio in a 3D Tele-Immersive environments. Cavernsoft G2 [28] is a toolkit which allows for the construction of virtual reality applications. The emphasis is on the building of high quality network applications. However, the toolkit lacks portability. Since our goal is building a TEEVE environment with COTS components, we needed components with emphasis on portability which the toolkit cannot provide.

We deal with buffering delays and play out delays in our system. An interesting related work is an efficient synchronization mechanism which adapts to dynamic networked state [27]. The proposed mechanism changes the playout buffer balancing interactivity and a smooth user experience. A disadvantage of [27] is that the adaptation mechanism proposed is relatively slow and does not always react rapidly to change in network state.

Another relevant paper which guides us is the human factor influences on synchronization. This was covered by Ralf Steinmetz’s seminal work on lip synchronization by the human perception of media synchronization [34]. A key result of this paper is that if the body view of a person is shown, then the constraints for lip synchronization are lower than just the head view. A tolerance of 140 ms is found to be tolerable.
difference for audio and video for the body view. Our system uses a full body view of the user and is thus subjected to lower synchronization constraints.

The system that we use makes extensive use of Voice over IP standards to deliver low latency audio. The audio setup uses RTP to transfer the audio. This is covered in RFC 2498.
Wireless Networks have become ubiquitous and wireless interfaces are now found on a variety of computing devices. Earlier wireless interfaces were available primarily on portable computers. But now wireless interfaces are found on a variety of consumer electronic devices like PDA’s, cell phones, portable game consoles, home media players, media extenders and wireless media players. These devices can connect to networks in either infrastructure mode or Ad Hoc mode. An Ad Hoc network allows these nodes to communicate with each other in an arbitrary fashion and without the presence of an access point. Thus the Ad Hoc mode enables these devices to interact with each other in airports and other locations where an access point is either unavailable or protected.

Single hop Ad Hoc networks also have found wide applications in every day scenarios like meetings, collaborative work groups and conferences. Here people meet up without any prior plan and may not have access to infrastructure based networks and thus can readily use single hop ad hoc networks. With the ubiquity of new consumer electronic devices, one can envision joining a game with somebody who is within radio range even if the user is not known. Another application is streaming music and video from a computer to a device attached to the home entertainment system. The popularity of VoIP implies that wireless VoIP devices can be prime items of deployment on such networks.

The transfer of such media over wireless networks constitutes a challenging problem because of three factors.

1. Stringent QoS requirements of multimedia application.
2. Bursty nature of multimedia traffic.
3. Unreliable and dynamic nature of the wireless channel. In particular the effects of contention and congestion of the wireless channel.

Many approaches have been proposed to solve this problem at various layers of the networking stack. In particular there have been many approaches at the MAC layer like those proposed in [9, 25, 11]. Our approach to solve this problem is to nominate a centralized arbiter at the application layer which proportions
resources among various multimedia streams. Such a design takes advantage of cross-layer interaction by allowing the multimedia application to factor in the link layer measurements to guide the sending rate. This arbiter can be easily deployed over existing networks and requires minimal overhead. A complete description of the framework is given in [33]. A brief overview is given below and forms the basis of the work presented in the next chapters.

3.1 Network Model

3.1.1 Assumptions

![Bandwidth Manager in an adhoc network](image)

Figure 3.1: Bandwidth Manager in an adhoc network

The bandwidth management scheme is designed for a wireless network consisting of heterogeneous computers and devices connected together over the IEEE 802.11 MAC layer. Each node in the network is assumed to be within the transmission range of every other node. Hence, only one node can transmit at a time over the channel. Since every node is within the transmission radius of every other node, routing is singlehop. This is shown in the Figure 3.1 The IEEE 802.11 MAC protocols DCF, which is the one relevant to our network model, does not have a provision for a fixed transmission schedules. A node can send when it senses that the channel is not busy. A binary exponential backoff mechanism resolves collisions that might occur as a result of nodes transmitting at random times. Moreover, any node in the network can transmit to any other node directly without using the base-station as an intermediary hop.

We assume a network has a set of flows $F$. Each flow $g \in F$ is uniquely identified by its source IP address,
source port number, destination IP address and destination port number. We call this unique identifier the flow-id of the flow. A new flow $f$ registers with the BM before beginning its transmission. The application initiating flow $f$ has a minimum bandwidth requirement $B_{\text{min}}(f)$ and a maximum bandwidth requirement $B_{\text{max}}(f)$. The flow $f$ also has an estimate of the total network bandwidth $B_p()$. At the time of registration, it specifies its minimum and maximum CTP requirements, $p_{\text{min}}(f)$ and $p_{\text{max}}(f)$, to the BM. In response, the BM adds flow $f$ to set $F$ and allocates it a certain channel time $p_a(f)$, when the flow is admitted. Flow $f$ then uses this allotted CTP $p_a(f)$ to calculate its transmission rate. It transmits using this transmission rate until either it stops or until a new $p_a(f)$ value is allotted to it. A new $p_a(f)$ could be allotted to it when there is a change in the channel characteristics or in the network traffic characteristics. We assume that the flows in the wireless network are well-behaved and co-operative, i.e. they will refrain from exceeding their allotted channel share (eating into other flows’ share) and will release any channel share allotted to them when they stop.

### 3.1.2 Bandwidth Management System Architecture

The architecture of the bandwidth management system consists of three major components as shown in Figure 3.2:

a. The Rate Adaptor (RA) at the application or middleware layer,

b. the per-node Total Bandwidth Estimator (TBE) at the MAC-layer and

c. the Bandwidth Manager (BM), which is unique in the entire single-hop wireless network.

Our system takes advantage of cross-layer interaction between the application/middleware and link layers.
Figure 3.2: Bandwidth Management System Architecture

Figure 3.3: Bandwidth Management Protocol.
**Rate Adaptor (RA):** The Rate Adaptor (RA) regulates a flow’s bandwidth consumption in accordance with its allotted CTP. The RA converts a flow’s bandwidth requirements into CTP requirements, communicates this to the BM, and obtains an allotted CTP for this flow from the BM. It then controls the transmission rate of each flow depending on its allotted CTP. The RA is implemented separately as a module and is linked to the application at run-time. It thus functions as middleware, just below the application layer, and shapes the applications traffic.

**Total Bandwidth Estimator (TBE):** The per-node Total Bandwidth Estimator is co-located with the IEEE 802.11 protocol at the MAC layer. It estimates the total network bandwidth $B_p(f)$ for each flow $f$ sourced at the node it resides on. $B_p(f)$ is what flow $f$ perceives to be the total bandwidth of the network at a particular time. In other words, at a particular instant in time, $B_p(f)$ is equal to the theoretical maximum capacity of the channel (1, 2, 5.5 or 11 Mbps for IEEE 802.11) minus the bandwidth lost due to channel errors, caused by fading, interference and contention experienced by flow $f$’s packets, at that instant. The physical channel errors and contention at a particular instant in time is estimated from the errors and contention experienced in recent history. Note that the TBE is per-node whereas it performs total bandwidth estimation per-flow sourced at the node it resides on.

The TBE continuously measures the total perceived bandwidth for each flow. It periodically passes this up to the RA of the flow at the higher layers. The RA of a flow $f$ uses it in the translation of flow $f$’s bandwidth requirements to its CTP requirements. When the total bandwidth $B_p(f)$ perceived by flow $f$ changes, the channel time requirements calculated using $B_p(f)$ also change. The TBE informs the RA of the new $B_p(f)$. The RA may now need to re-negotiate on behalf of flow $f$ with the BM, using flow $f$’s new CTP requirements that are calculated with the new $B_p(f)$ estimate. Since CTP allotted to flow $f$ is directly related to its share of total network bandwidth, if a flow perceives the total network bandwidth as having decreased, its share of the bandwidth will also decrease. This may cause it to fall substantially below its minimum bandwidth requirements. Hence the re-negotiation.

**Example:** Assume a flow $f$ in a 2 Mbps wireless network has minimum bandwidth requirement 300 Kbps and perceives total network bandwidth of 1.5 Mbps. (That is, the flow $f$ perceives this to be the total capacity of the 2 Mbps channel.) Assume further that the CTP allotted to it is 20%, thus ensuring it just meets its minimum bandwidth requirement. If the total network bandwidth, as perceived by $f$, decreases to 1.2 Mbps due to an increase in physical channel errors or contention, then the 20% channel time is no longer sufficient for the flow to meet its minimum bandwidth requirement. Its RA must then re-negotiate for at least a 25% of the channel time. Similarly, if a flow perceives the total network bandwidth to have increased, it must release any excess share of the channel it has been allotted, so that some other flow can
use it.

**Bandwidth Manager (BM):** The BM performs admission control at the time of flow establishment and bandwidth redistribution at the time of flow teardown. Admission control involves revocation of some channel time from existing flows and re-allocation of this portion to the new flow. The BM also performs re-negotiation either when some flow detects a change in its perceived bandwidth or when its traffic characteristics change.

The BM admits a flow only if it can allot *at least* its minimum CTP requirement. Otherwise, the flow is rejected. The remaining channel time as yet unallotted after all the admitted flows' minimum channel time requirements are satisfied, is allotted on a *max-min fair* basis. Each flow receives whatever CTP is allotted to it by the max-min fair algorithm, in addition to its minimum CTP request, which is automatically guaranteed when it is admitted.

### 3.2 Link Level Module

![Transmission Sequence](image)

Figure 3.4: Transmission Sequence

The Link level module monitors the network and estimates the available bandwidth. Determining the channel quality being experienced by a certain uplink or downlink flow is vital to our solution. The rate allocated by the BM to a flow depends on the channel quality it experiences. Our channel quality estimation mechanism comprises two parts:

1. Channel capacity estimation
2. Channel loss rate estimation.

Channel capacity estimation involves measuring the impact of contention, fading and interference effects on a flow transmissions, and hence on its maximum available throughput.

Increased contention causes an increase in the time a host must wait for the medium to become idle. This increases the time interval $t_r - t_s$ in Figure 3.4, and decreases channel bandwidth perceived by this
flow packets. Signal fading effects cause bit errors in the individual frames (RTS/CTS/DATA/ACK), and necessitate RTS or DATA retransmissions. This results in larger $t_r$ ts intervals and smaller perceived channel bandwidth. If the retransmit limit is reached for a flow packet, the packet is dropped, resulting in a wastage of the flow channel time, and an increase in its loss rate. Our channel quality estimation mechanism works as follows. Assume a flow attempts to transmit $k$ packets in a time interval $T$. Assume that $j$ of these attempts result in successful packet transmissions while $k - j$ packets are dropped by the MAC protocol. Assume that the flow has packet-size $S$ at the MAC layer. The channel capacity perceived by the flow $f$ is calculated as:

$$L(f) = 1/(k - j)$$

Both estimates are computed every time interval $T$ and running averages with history decay are kept. Channel quality estimation is done for every flow at the MAC layer of the flow source. This is necessitated because fading, contention and interference effects are different for different flows, depending on the location of their end-points. Channel quality estimation is done using data packets; no probing traffic is injected. We have assumed unidirectional flows (e.g. UDP) and constant packet sizes. In practice, different estimates are kept for different packet size ranges. Bidirectional flows are considered as separate uplink and downlink flows.

The Link layer monitor is implemented by modifying the Orinoco driver in Linux. The solution is tested with the Linux 2.6 kernel on Fedora Core 2. The driver is modified such that when a packet is ready to be transmit a timer is started and when the control returns from the hardware the timer is stopped. Additional data is collected as noted above and the channel capacity $B_p(f)$ is calculated as per the above procedure.

### 3.3 Bandwidth Management Middleware

The Bandwidth Management middleware forms a layer that negotiates with the BM. The functions implemented by the middleware are:

a. Setup  
b. Renegotiation  
c. Monitoring  
d. Rate Adaptation and Translation

The middleware layer is implemented as a thread which the adaptive media application talks to. The middleware layer receives notifications from the centralized BM and translates the rate given into a QoS level that the adaptive media application can use in order to change the media delivery rate.
Chapter 4

Adaptive Video Streaming over Wireless Networks

In this chapter we use the bandwidth management framework described in the previous chapter to aid the transmission of H.263 video. Towards this goal we create an adaptive H.263 encoder and a decoder. The wireless channel characteristics covered earlier have a detrimental effect on the quality and sustainability of the throughput. Thus bandwidth management becomes necessary for multi-video streaming.

The system is constructed at multiple layers with link layer bandwidth measurement, bandwidth management middleware and adaptive H.263 coding at the user level. It takes in a bandwidth estimate from the link layer and negotiates with a BM to get an estimate of the channel time it is allowed to occupy. Based on this estimate the encoder is adapted to send video at a specified frame rate and quantization limit.

The special characteristics of wireless channels also force design modifications of the H.263 encoder. The loss rates of wireless channels have special effects on media like motion compensated video that have temporal dependencies. The high error rate coupled with temporal dependence causes the propagation of errors over a long period of time. Thus error recovery mechanisms need to be built in to the encoder.

4.1 Overview of H.263

4.1.1 H.261: Video Coding and Decoding (CODEC)

H.261 is video coding standard by the ITU. It was designed for data rates which are multiples of 64Kbit/s, and is sometimes called p x 64Kbit/s. These data rates suit ISDN lines, for which this video codec was originally designed for.

The coding algorithm is a hybrid of inter-picture prediction, transform coding, and motion compensation. The data rate of the coding algorithm was designed to be able to be set to between 40 Kbits/s and 2 Mbits/s. INTRA coding where blocks of 8x8 pixels each are encoded only with reference to themselves and are sent directly to the block transformation process. On the other hand INTER coding frames are encoded with respect to another reference frame. The inter-picture prediction removes temporal redundancy. Motion vectors are used to help the codec compensate for motion. To remove any further redundancy in the
transmitted bitstream, variable length coding is used.

H.261 supports two image resolutions, QCIF (Quarter Common Interchange format) which is (144x176 pixels) and CIF (Common Interchange format) which is (288x352). The video multiplexer structures the compressed data into a hierarchical bitstream that can be universally interpreted. The hierarchy has four layers:

1. Picture layer: corresponds to one video picture (frame)
2. Group of blocks: corresponds to 1/12 of CIF pictures or 1/3 of QCIF
3. Macroblocks: corresponds to 16x16 pixels of luminance and the two spatially corresponding 8x8 chrominance components.
4. Blocks: corresponds to 8x8 pixels.

4.1.2 H.263:

<table>
<thead>
<tr>
<th>Picture format</th>
<th>Luminance pixels</th>
<th>Luminance lines</th>
<th>H.261 support</th>
<th>H.263 support</th>
<th>Uncompressed bit rate (Mbit/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10 frames/s</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Grey</td>
</tr>
<tr>
<td>SQCIF</td>
<td>128</td>
<td>96</td>
<td>Yes</td>
<td>Yes</td>
<td>1.0</td>
</tr>
<tr>
<td>QCIF</td>
<td>176</td>
<td>144</td>
<td>Yes</td>
<td>Yes</td>
<td>2.0</td>
</tr>
<tr>
<td>CIF</td>
<td>352</td>
<td>288</td>
<td>Optional</td>
<td>Optional</td>
<td>8.1</td>
</tr>
<tr>
<td>4CIF</td>
<td>704</td>
<td>576</td>
<td>Optional</td>
<td>Optional</td>
<td>32.4</td>
</tr>
<tr>
<td>16CIF</td>
<td>1408</td>
<td>1152</td>
<td>Optional</td>
<td>Optional</td>
<td>129.8</td>
</tr>
</tbody>
</table>

Figure 4.1: Comparison of H.261 and H.263

The H.263 standard, by the International Telecommunications Union (ITU), supports video compression (coding) for video-conferencing and video-telephony applications. H.263 was developed to stream video at bandwidths as low as 20K to 24K bit/sec and was based on the H.261 codec. As a general rule, H.263 requires half the bandwidth to achieve the same video quality as in the H.261. As a result, H.263 has replaced H.261. The coding algorithm of H.263 is similar to that used by H.261, however with some improvements and changes to improve performance and error recovery. Half pixel precision is used for motion compensation whereas H.261 used full pixel precision and a loop filter. Some parts of the hierarchical structure of the datastream are now optional, so the codec can be configured for a lower datarate or better error recovery.
There are now four optional negotiable options included to improve performance: Unrestricted Motion Vectors, Syntax-based arithmetic coding, Advance prediction, and forward and backward frame prediction similar to MPEG called P-B frames.

The differences between the H.261 and H.263 coding algorithms are listed in the figure 4.1

Following the adoption of the first version of the H.263 standard, further enhancements were proposed, leading to Version 2 (H.263+) of the standard. H.263+ provides 12 new negotiable modes and additional features that improve video quality and increase robustness and functionality of the target video coding systems.

4.1.3 Encoder/Decoder Description

![Figure 4.2: Encoding Steps and Block Diagram](image)

All versions of H.263 support five standardized picture formats: sub-QCIF (88x72), QCIF (176x144), CIF (352x288), 4CIF (704x576), and 16 CIF (1408x1152). The luminance component of the picture is sampled at these frame resolutions, while the chrominance components, $C_b$ and $C_r$, are downsampled. Each picture in the input video sequence is divided into macroblocks, consisting of four luminance blocks of 8 pixels x 8 lines followed by one $C_b$ block and one $C_r$ block, each consisting of 8 pixels x 8 lines. A group of blocks (GOB)
is defined as an integer number of macroblock rows, a number that is dependent on picture resolution. For example, a GOB consists of a single macroblock row at QCIF resolution.

Video coding aims at providing a compact representation of the information in the video frames by removing spatial redundancies that exist within the frames, and also temporal redundancies that exist between successive frames. Spatial redundancy is removed using the Discrete Cosine Transform (DCT) to remove spatial redundancies, and motion estimation and compensation to remove temporal redundancies. When a source frame is coded using the DCT, the encoder is said to be operating in the intra coding mode, and the corresponding encoded frame is called an I picture. In the case where temporal prediction is used, the encoder is said to be operating in the inter coding mode, and the corresponding encoded frame is called a P-picture. A block diagram for a typical encoder is given in Figure 4.2

4.1.4 Motion Estimation and Compensation

Often video frames that are close in time are also similar. Therefore, when encoding a video frame, it would be judicious to make as much use as possible of the information presented in a previously encoded frame. This temporal redundancy forms the basis of motion compensation. One approach to achieve this goal is to simply consider the difference between the current frame and a previous reference frame, and encode the difference or residual. When the two frames are very similar, the difference will be much more efficient to encode than encoding the original frame. In this case, the previous frame is used as an estimate of the current frame. A more sophisticated approach to increase coding efficiency is to work at the macroblock level in the current frame, instead of processing the whole frame all at once as described above. The process is called motion compensation, or more precisely, motion compensated prediction, and is based on the assumption that most of the motion that the macroblocks undergo between frames is a translational motion. This approach attempts to find for each 16x16 luminance block of a macroblock in the current frame, the best matching block in the previous frame. A search window is usually defined and bounds the area within which the encoder can perform the search for the best matching block.

The motion of a 16x16 block of a macroblock is represented by a motion vector that has two components, the first indicating horizontal displacement, and the second indicating vertical displacement. Different criteria could be used to measure the closeness of two blocks. The most popular measure is the Sum of Absolute Differences (SAD). One approach to find the best matching macroblock is to evaluate the SAD at every pixel location within the specified search window. This approach is called full search or exhaustive search, and is usually computationally expensive but on the other hand yields good matching results. A more computationally efficient approach is to restrict the search to only few points in the search area, where
there is a high likelihood of finding a good match. These search points are defined by predicted motion vectors that are calculated based on motion vectors of previously encoded macroblocks.

For any given macroblock, and once the search for the best matching block is done, the latter is used instead of the original macroblock to construct the prediction frame. Once all the macroblocks in the prediction frame are identified, the difference between the prediction and the actual frame is computed to produce the residual frame to be encoded.

4.1.5 DCT Transform

The 8x8 DCT is used to decorrelate the 8x8 blocks of original pixels (in intra coding) or motion compensated difference pixels (in inter coding). The DCT transform concentrates the energy of input samples into a small number of transform coefficients, which are easier to encode than the original samples. In addition to its relatively high correlation and energy compaction capabilities, the 8x8 DCT is simple, efficient, and amenable to software and hardware implementations. A popular algorithm for implementing the 8x8 DCT is that which consists of eight-point DCT transformation of the rows and the columns, respectively. The quantization is a key determinant of the compression factor and different quantization rates are used to let the video adapt to the current bandwidth.

4.1.6 Quantization

The quantization is a significant source of the compression in the encoder bit stream. Quantization takes advantage of the low sensitivity of the eye to reconstruction errors related to high frequencies as opposed to those related to low frequencies. Quick high frequency changes can often not be seen, and may be discarded. Slow linear changes in intensity or colour are important to the eye.

Therefore, the basic idea of the quantization is to eliminate as many of the nonzero DCT coefficients corresponding to high frequency components. Every element in the DCT output matrix is quantized using a corresponding quantization value in a quantization matrix. The quantizers consist of equally spaced reconstruction levels with a dead zone at zero. In baseline H.263, quantization is performed using the same step size within a macroblock by working with a uniform quantization matrix. Except for the first coefficient of an intra block which is coded using a step size of eight, even quantization levels in the range from 2 to 62 are allowed. The quantized coefficients are then rounded to the nearest integer value. The net effect of the quantization is usually a reduced variance between the original DCT coefficients. Another important effect is a reduction in the number of nonzero coefficients.
4.1.7 Entropy coding

![Decoder Description Diagram]

Entropy coding is performed by means of variable-length codes (VLCs), and is used to efficiently represent the estimated motion vectors and the quantized DCT coefficients. Motion vectors are first predicted by setting their component values to median values of those of neighbouring motion vectors already transmitted: the motion vectors of the macro blocks to the left, above, and above right of the current macro block. The difference motion vectors are then VLC coded.

As for the quantized DCT coefficients, they are first converted into a one-dimensional array for entropy encoding by an ordered zigzag scanning operation. The resulting array contains a number of nonzero entries and probably many zero entries. Hence, the array can be represented as a number of segments stitched together, where each segment contains one or more (or no) zeros followed by a nonzero coefficient. To efficiently encode the whole array, each segment is assigned a code word, with the most frequent segments getting the code word with the least number of bits, and the least frequent segments getting the code word with the highest number of bits. The code word is generated based on three parameters (LAST, RUN, LEVEL). The symbol run is defined as the distance between two nonzero coefficients in the array (i.e. the number of zeros in a segment). The symbol LEVEL is the nonzero value immediately following a sequence of zeros. The symbol LAST, when set to 1, is used to indicate the last segment in the array.

4.1.8 Coding Control

Coding control allows switching, at the macroblock level, between the intra and inter coding modes. The H.263 standard does not specify how to perform coding control. If a macroblock does not change significantly with respect to the reference picture, then the encoder may skip encoding that macroblock and the decoder will simply repeat the macroblock located at the subject macroblock spatial location in the reference picture.
A block diagram of a typical decoder is shown in Figure 4.3.

In the case of an intra coded macroblock, the encoder performs only the inverse quantization and inverse DCT operations to reconstruct the original macroblock. The reconstructed macroblock is then used in the reconstructed frame. In the case of an inter coded macroblock, the decoder performs the inverse quantization and inverse DCT operations on the DCT coefficients corresponding to the prediction residual. The decoder also uses the information in the motion vectors to find the best matching macroblock in the previous reconstructed frame. The latter is then added to the residual to reconstruct the original macroblock. Once the complete frame is reconstructed, it is then stored for use when decoding the subsequent frame.

4.2 System Architecture

The architecture of the video application consists of the following parts:

a. The Rate Adaptor (RA) at the application or middleware layer
b. The per-node Total Bandwidth Estimator (TBE) at the MAC-layer
c. The Bandwidth Manager (BM), which is unique in the entire single-hop wireless network.
d. The Adaptive Video Encoder at the application layer.
e. Video Decoder

Most of the components have been covered in the previous chapter. Here we cover only the Rate Adaptor and the Adaptive Video Encoder as they have been uniquely developed for our system.

4.2.1 Rate Adaptor

The Rate Adaptor forms the part of the application which continuously interacts with the BM and the link layer module. Initially the RA reads the video to make an estimate of the minimum and maximum quality it can send the video at. It converts this quality into a minimum and maximum CTP metric which is the proportion of channel time that this flow can occupy. This CTP metric is communicated to the BM. The BM gives an allocation of CTP which the RA converts into a quality metric for the video. As more flows join the network, the BM may change the bandwidth allocation. This is then communicated this to the RA using a gratuitous reply message. The RA then changes the quality metric at which the video is sent. Thus the RA needs to interact continuously with the BM.

The Rate Adaptor also continuously monitors the link layer. This is done for two reasons:
1. The CTP represents the proportion of channel time that the flow should occupy and the actual bandwidth that can be used by the media application to stream video depends on the actual link layer bandwidth seen. Thus the link layer estimate is used in converting the CTP metric into a quality metric.

2. If the channel capacity changes greatly then the CTP allocated earlier does not make semantic sense. In such a case the RA re-negotiates with the BM to get a new CTP value.

4.2.2 Adaptive Video Encoder

At the core of the proposed system is an adaptive media delivery system which adapts the video to the currently seen bandwidth. The video encoder is a modified H.263 encoder and decoder combination which is based on the TMN encoder framework. The encoder takes in raw YUV frames and compresses them into H.263 frames that are transmitted over the network to the decoder which simply decodes them and displays them.

Since the throughput of TCP streams over wireless networks is seriously limited, UDP is used to transfer the streams over the wireless network. This simple mechanism has a serious draw back. Unlike TCP, UDP does not guarantee a lossless channel and the loss rates of the wireless channel is high. Thus special mechanisms have to be introduced in order to compensate for the losses of the wireless channel.

The Adaptive Video Encoder needs to be able to respect the bandwidth allocated to it by the BM. However, we deal with video whose bandwidth consumption is variable. To compensate for this the VBR video stream is profiled at different frame rates and different quantization thresholds and the quality is changed appropriately to meet the bandwidth requirements. If the bandwidth of the video stream still exceeds the allocated requirement, frames are dropped. However this is done on a small granularity of a second so the user does not feel the disruptive effects of the frame drop.

4.2.3 Decoder

The simple decoder reads from a socket, performs H.263 decoding and displays it. However because of the higher error rate of the wireless channel, the decoder had to be modified. The modified decoder now returns a message to the sender if the earlier frame was received with an error. Thus the encoder forces the next frame to be an I-Frame. Without this scheme a single frame error can propagate across the video stream.
4.3 Implementation

The system is implemented as a multi threaded modular system. At the core of the system is the adaptive encoder which takes in raw YUV frames and converts them into a compressed H.263 stream. The compressed H.263 stream is then sent over the wireless channel to the decoder which decodes and displays it. The encoder takes in an allowed bandwidth value from the rate adaptor middle ware and uses the static profile to decide its target rate. It then adapts the video to the static rate. If the bandwidth is exceeded, the excess frames are dropped. The driver for the orinoco wireless card is modified to perform total estimation and it forms the kernel module. The RA takes in the available bandwidth estimate from the kernel module and is responsible for negotiation with the BM. It translates the required bandwidth into a required channel time proportion which is then communicated to the BM. The BM allocates a CTP based on the number of flows in the network and congestion in the network. This CTP is returned to the RA which translates it to a bandwidth value. This value is then sent to the Adaptive Video Encoder. Most importantly, the RA described in the previous section can be abstracted as a middleware service, which any adaptive media application can use for wireless media delivery.

The profiler samples the video at various frame rates and quantization thresholds and forms a static profile of the video. This approach has been preferred because even on the Internet video is transcoded statically into various target profiles for streaming. This method is better than dynamically forming a predictive system which would take in a lot of overhead and whose efficacy is doubtful.

4.4 Problems and Solutions

Most of the problems with wireless video revolved around the high loss rates of the wireless channel. After a certain threshold the wireless channel starts dropping packets at a very high rate and also starts corrupting the content of individual packets itself. Since the H.263 stream is motion compensated, errors in individual packets can cascade into big errors. Thus new techniques need to be introduced for error correction.

The simple scheme that we used for error correction was sequence number based feedback to detect missing frames along with creating I-Frames on a regular basis. Thus whenever a loss occurs in a wireless channel the next frame sent is an I-Frame to reduce the error propagation due to motion compensation.

More complex techniques can be explored like forward error correction information in the frame itself. However these techniques affect the static profile created and also need to be regulated dynamically based on the loss rate of the channel that is seen.
4.5 Evaluation

4.5.1 Experimental Testbed

We test the system on a wireless testbed, consisting of 8 notebooks running a Fedora Core distribution of Linux. The wireless testbed contains nodes operating in the Ad Hoc mode. The nodes in the testbed are kept static. The nodes are configured to be in the same subnet in the ad hoc mode; hence any node in the network can communicate with any other node. The BM is kept running on one node of the network and all the nodes communicate with this BM.

4.5.2 Metrics

The metric which will be used in all the experiments will be the frame rate of the video. This has been chosen because this is the parameter which is most easily observed by the end user. The frame rate achieved can show properties of the system such as throughput, fairness and interference.

In our experiments we send a video sequence from one node to another and monitor the frame rate at the receiver over time. Once the sequence is over, the application quits. Thus we can get an idea of the average throughput just by looking at the completion times. We test the application in two topologies, shown in Figure 4.4. In the first topology video is sent between two laptops from one sender to one receiver. In this experiment the number of streams is increased and the frame rate achieved is observed. In the second topology multiple independent streams from different nodes are sent and the frame rate is observed. In the first topology the variations of the environment (i.e. the wireless channel) affect each stream equally. However, the frame rate is limited by the processing power of the node sending the data. In the second
In topology 1, we send multiple streams from one sender to one receiver and measure the frame rate. The experiment begins with two concurrent streams and scales up to five concurrent frames beyond which the frame rate fell to a very low level. Since the H.263 encode process is processor intensive, the frame rate in topology 1 would depend on CPU effects in addition to network effects. The graph in Figure 4.5 seems to show high temporal variance in the frame rate. This is caused due to the vagaries of the wireless channel. However, it can be seen that fair allocation of bandwidth is achieved as the throughput for stream 1 and 2 seem to follow each other.

As the number of video streams is increased to five in figure 4.6, one can clearly see the processor contention. The average frame rate drops considerably from Figure 4.5. We can also see the fairness achieved by the BM. The frame rate of every stream drops equally. A comparison of the frame rate when the number of streams increases is shown in 4.7. We can clearly see the drop in frame rate. The drop in average throughput per stream is shown by the completion times. The graph in Figure 4.8 shows the frame rate comparison in
Figure 4.6: Frame rate measured in topology one with five concurrent streams. Here the frame rate falls and the average frame rate measured is 8 FPS

Figure 4.7: Comparison of average frame rate comparison in topology 1 with two, three, four and five concurrent streams
topology 2 where the streams are being transmitted independently. Unlike figure 4.7, The drop in throughput is not as noticeable. However since multiple streams are sent over a wireless channel, they all contend with each other and the variance in frame rate is higher. Thus we can see that the dominant effect in topology one is processor contention, While the dominant effect in topology two is processor contention. This is shown clearly in Figure 4.9 where the frame rate when four streams are sent is compared. The throughput when all streams originate from the same computer is considerably lesser than when they originate independently.

4.6 Explanation

The experiments show the efficiency of the bandwidth management scheme. In all cases the bandwidth was proportioned fairly. However the experiments also show the limits of a user space approach. The frame rate varies pretty erratically in all cases due to wireless channel variations. Hence this scheme should be coupled with a mac layer scheme like [11] to control the variance in frame rate.

It is also interesting to note the processor contention due to the computational complexity of the encoding application. A plausible scenario where live video streaming can be applied is video conferencing for embedded devices like cell phones. However the results clearly show that the process is very processor intensive and may not work in such a resource constrained environment.

![Average Frame rate when the number of streams are increased and are transmitting Independently](image)

Figure 4.8: Comparison of average frame rate comparison in topology 2 with multiple independent streams
Comparing Frame rate of multiple streams by 1 sender and multiple streams of many senders

Four Stream Average with 1 sender
Four Stream Average with multiple senders

Figure 4.9: Comparison of average frame rate of four streams in topology 1 versus the average frame rate of four streams in topology 2
Chapter 5

Bandwidth Manager Migration using Vickerey Auctions

5.1 Introduction

The problem with a centralized bandwidth management approach presented earlier is that the Bandwidth Manager becomes critical to the operation of the network. If the Bandwidth Manager decides to leave the network then its function needs to be migrated so as to ensure availability. The node which represents the Bandwidth Manager expends valuable power in performing the functions assigned to it especially since it cannot go into sleep mode which is crucial to save power in such devices. Thus any election and then assignment of bandwidth management functionality cannot be permanent and would need to be shifted around the network. This is precisely the scope of this chapter.

The Ad Hoc network envisaged consists of nodes which are strangers to each other and in a way are similar to peer to peer networks. Research in peer to peer networks has shown that majority of the nodes do no uploading and free-ride on the network [3]. In this case each node looks for its personal gain which makes it suboptimal to share files. Such a situation is commonly referred to as “The Tragedy of The Commons” [18]. Thus we feel that similar nodes in the ad hoc network may try to avoid becoming the Bandwidth Manager as it is resource heavy. Hence we need an incentive mechanism to reward the node which is the Bandwidth Manager. We thus present an auction metaphor where nodes bid to be the Bandwidth Manager and the winner is paid some differentiated service reward.

Towards this end, we use a Vickerey Auction to choose the Bandwidth Manager. The Vickerey Auction, also known as a second price auction is a simple auction scheme which has a few desirable properties. The optimal strategy for a node in a Vickerey Auction is to bid honestly. This is advantageous to us as it ensures that no node lies about its resources. It is also done in a single step and efficient and as shown later by simulation extremely lightweight.
5.2 Target Scenario

The target scenario can be described as follows. Let us consider a number of multimedia devices which are currently operating in Ad Hoc mode. Media devices and portable game consoles fall into this category. Such devices can communicate with similar devices which also operate in ad hoc mode. Thus they form a network which can be used for sharing multimedia files or playing interactive games. Since such devices have dedicated functionality to exchange multimedia data, they can use specific MAC protocols which make soft guarantees of QoS. Examples of such protocols are [9, 25] and the 802.11e standard [11].

A problem with MAC layer approaches is the fact that although these schemes make soft guarantees of service differentiation, a media stream at the application layer decides which service class it belongs to. There exist adaptive codecs for both video and audio which can change the quality which is transmitted based on the network state. However the application does not know the state of the network and the state of overload. A mechanism is required through which an application can decide which service class it belongs to. Also there is a problem of admission control when some streams are rejected because the network is overloaded. Here the MAC layer approaches are not very effective. In either case an application layer construct is more effective to allow for fair sharing of bandwidth between the different devices. Such a construct is the bandwidth management middleware which was described in chapter three. The middleware allows a multimedia stream to indicate the maximum and minimum bandwidth required and performs max-min fair allocation after admission control.

The bandwidth management middleware communicates with the MAC layer to communicate the differentiated service priority of the multimedia stream. However in such a system it is obvious that a node can get better throughput by never being the Bandwidth manager. Since the bandwidth management layer is completely in the user space, a user might be able to disable the functionality of being a Bandwidth Manager. This can be done by modifying the parameters used for election. Thus an incentive mechanism must be introduced which compensates for the loss in resources due to the functionality of being the Bandwidth Manager.

To get around these problems we have to reward the node which is the Bandwidth Manager. The winner of the election gets a prioritized channel time proportion. In the bandwidth management scheme, each node in the wireless channel applies for and is given CTP based on the current state of the network. In the modified scheme, the node which is the Bandwidth Manager is given a priority to the allocation of CTP before other nodes are allocated CTP. This would convert the FCFS based scheme of [33] into a priority based approach.

This scheme of letting nodes bid for CTP priority based on the cost of hosting the BM is an auction.
However, such auctions can be subject to manipulations by the nodes taking part in it. To prevent manipulations of the auction scheme we use Vickerey based auction mechanism. This scheme has a provably optimal strategy which is covered in [22]. The optimal strategy for a node in these networks is to quote its true valuation. We proceed as follows, First we give an introduction to Auctions, followed by a description of the migration mechanism, concluding with a performance analysis.

5.3 Auctions

An auction consists of an auctioneer and potential bidders. The commonly seen auctions include English auction, first-price sealed-bid auction, Dutch auction and Vickrey auction.

**English Auction:** In English auction, the auctioneer starts with the reserve price and proceeds to solicit successively higher bids from the bidders until no one raises the bid. The highest bidder is the winner and pays the price it bid. The dominant strategy for one agent in English auction is to continuously raise its bid until it wins or it reaches the maximum price it is willing to pay for that item. A noticeable feature of English auction is that it is usually multi-round and the time and communication overhead is proportional to the difference between the starting price and the price at which the item is sold. However, it does allocate the item to the bidder with the highest valuation, who is the only bidder willing to outbid all other bidders.

**First-Price, Sealed-Bid Auction:** In first-price sealed auction, each bidder submits one bid in ignorance of all other bids to the auctioneer, who determines the highest bid and sells the item to that bidder for the bidding price. This kind of auction can be executed in one-round and thus is communication-saving. However, since each agentid is based on her private valuation and prior beliefs of others valuations, the item is not always awarded to the party who values it most.

**Dutch Auction:** In Dutch auction, bidding starts at an extremely high price and is progressively lowered until a buyer claims an item by calling nw. The winner pays the price at the current price. Dutch auction preserves maximal privacy, ie, only the highest bid is revealed. However, like English Auction, it is multi-round, and like first-price, sealed-bid auction, one agentid is strategically based on its private valuations and its beliefs of others valuation.

**Vickrey Auction:** Similar to the first-price sealed auction, Vickrey auction is sealed and executed in one-round . The highest bidder is the winner, but pays a price that is equal to the second-highest bid [37]. Vickrey auction has a very fundamental feature: the dominant strategy for every bidder is to bid her true valuation. Thus Vickrey auction always rewards the item to the bidder who values it most and thus realizes SCF.
Table 5.1: Various Auction Settings

From Table 5.3 that lists the features of the four previously mentioned auctions, we can see that only Vickrey auction and English auction have dominant strategy and realize SCF. Furthermore, Vickrey auction only requires single-round execution. Thus, from the perspective of both economic incentive and communication overhead, Vickrey auction is the best mechanism for service allocation in our scenario.

5.4 Choosing a Bandwidth Manager via Vickerey Auctions

To allocate the service of Bandwidth Management within the wireless network we use an auction scheme. It is worth noting that although an auction hosted by a service provider does find the service seeker that values a service most, it does not contribute to our purpose of social service allocation, i.e., serving a service request least expensively, which is essentially to allocate the service request to the service provider that can do so. Therefore, auctions are hosted by service seeking agents, leading to reverse Vickrey auction: the bidder (i.e., service provider) with the lowest instead of the highest bid is the winner. Bids submitted by service providers are determined by the cost for serving a service. As the cost reflects suitability of a service provider serving a service better than simply the load, auction-based service allocation is better than its non auction-based counterpart because it finds the cheapest way to serve a service via economic payoff. Auction based service allocation also achieves load balancing because heavy load leads to higher bid and thus less possibility of winning the auction. As mentioned above, service allocation via reverse Vickrey auction in ad hoc networks has three desired properties:

1. It has a dominant strategy, thus it is simple to implement.

2. The agents are motivated to bid. The winner gets a payoff which equals to the difference between its valuation and the second lowest bid; the losers lose nothing (i.e., payoff = 0).

3. The service is always allocated efficiently: service (request) is allocated to the provider that serves it least expensively.
5.4.1 The parameters used for bidding

The proposed framework and auction mechanism are generic. The reason is that they work with any parameters used for categorizing the nodes. However, for the purpose of this paper, the parameters that are used for bidding are:

1. **Bandwidth** This represents the bandwidth seen by the node with higher being better. This statistic is obtained from the device driver using the algorithms proposed in [30]

2. **Power Left** The power left is measured as the percentage of battery time left.

3. **Stability** The stability is measured by the number of disconnections that the particular node has seen.

4. **Trust_{ij}** In a cooperative short lived environment like an ad hoc network. There may be free riding by other nodes therefore trust can be used as a parameter to choose a bandwidth manager that is more prone to be honest. Trust can be quantified based on the interactions of the nodes with each other. A mechanism to update trust is given in [39]. Using this the value for Trust_{ij} can be calculated.

5.4.2 Bidding Mechanism

Once the bandwidth manager decides that it needs to be shifted, it sends out an elect message. The resource allocator which is currently interacting with the bandwidth manager gets the elect message and forwards the N-Vector containing the measured values of these parameters.

Using a Cost Function $F(X_i)$ these N values are converted into a cost value which represents the cost to be a bandwidth manager. The node which has the lowest cost is chosen to be the bandwidth manager. The mechanism used is a reverse Vickerey auction. Here the winner is paid according to the cost of the second lowest bidder. The utility function for the winner is the difference between the its bid and the second lowest bid. This is used in our bandwidth manager migration scheme. The payment, which is covered later, is scaled according to the cost of the second lowest bidder.

5.4.3 Bandwidth Manager Migration Technique

The migration of the bandwidth manager is a relatively straightforward procedure. After finishing the auction, The winner is informed and the winning RA launches the bandwidth manager which is situated on the node. If the node does not have a copy of the bandwidth manager, then the binary is transmitted over the wireless network. Information is transmitted to the bandwidth manager including history information which helps in scaling of weights, which is discussed later. Also transmitted is information about the time
that the current node was a bandwidth manager. This helps the new bandwidth manager decide how long it should continue as a bandwidth manager before sending out an election request.

After the winning node launches the bandwidth manager, all the other nodes in the network are informed about the winning bandwidth manager and they renegotiate with the new node with their CTP requirements. Noting that the new Bandwidth Manager has priority in the CTP allocation, the other nodes are allocated their requirements only after the Bandwidth Manager’s requirements are met.

5.4.4 Payment Mechanism

Max – min fair resource allocation algorithm
Input: channel time: \( p_{\text{rem}} \); set of requests: \( p_{\text{newmax}}[f] \)
Output: set of allocations: \( p_{\text{mm}}[f] \)

\[\text{proc Max-min}(p_{\text{rem}}, p_{\text{newmax}}[f]) \equiv\]
\[R := \{\}; // \text{set of satisfied flows}\]
\[N := \text{size of}(p_{\text{newmax}}[f]);\]
\[p_{\text{mm}}[f] := 0;\]
\[\text{if } (f == \text{BM}) \text{ then}\]
\[p_{\text{mm}}[f] := \text{min}(p_{\text{newmax}}[f], \text{VickereyPayment});\]
\[p_{\text{rem}} := p_{\text{rem}} - p_{\text{mm}}[f];\]
\[\text{if } (p_{\text{mm}}[f] == \text{VickereyPayment}) \text{ then}\]
\[R := R + \{f\};\]
\[\text{fi}\]
\[\text{fi}\]
\[\text{while } (\text{true}) \text{ do}\]
\[\text{total satisfied = 0;}\]
\[\text{foreach } f \in R \text{ do}\]
\[\text{total satisfied} += p_{\text{mm}}[f];\]
\[\text{od}\]
\[CA := (p_{\text{rem}} - \text{total satisfied})/(N - \text{size of}(R));\]
\[\text{stop} := \text{true};\]
\[\text{foreach } f \notin R \text{ do}\]
\[\text{if } (p_{\text{newmax}}[f] < CA) \text{ then}\]
\[R := R + \{f\};\]
\[p_{\text{mm}}[f] := p_{\text{newmax}}[f];\]
\[\text{stop} := \text{false};\]
\[\text{fi}\]
\[\text{od}\]
\[\text{if } (\text{stop}) \text{ then}\]
\[\text{foreach } f \notin R \text{ do}\]
\[p_{\text{mm}}[f] := CA;\]
\[\text{od}\]
\[\text{break};\]
\[\text{fi}\]
\[\text{od}\]

Figure 5.1: Modified Max-min Fair Resource Allocation Algorithm

Once the node wins the auction it is paid using a priority based channel time proportion rewarding scheme. Recall that the main function of the bandwidth manager is to arbitrate the amount of channel time
proportion that each node gets. Thus using the Vickerey payment scheme each node is given a reservation on the channel time proportion scheme. It can use this anytime during its role as a bandwidth manager.

The payment scheme is as follows. The algorithm to determine the allocation of the CTP is given in [33]. The modified algorithm is presented in 5.1 and does the following. Before the remaining bandwidth is split between the rest of the nodes, the bandwidth is allocated to the node which is currently the bandwidth manager. The node hosting the bandwidth manager is allocated a bandwidth of \( \min(vickereypayment, requestedbandwidth) \) before the loop starts.

The auction payment is converted into a priority CTP according to the previous payment and the CTP payment made by the parent. This is done using the equation:

\[
CTP = \frac{prev.CTP}{previousvickereypayment} \times (currentvickereypayment)
\]

The initial payment can be decided administratively or is set to a fair amount i.e. \( \frac{1}{\text{expectednumberofnodes}} \). For the experimentation it had been set to the fair amount.

An area of further research is converting the simple scheme given above into more accurate payoff schemes. In particular the existing algorithm for CTP allocation which is FCFS and fair in nature has to be converted into a priority based scheme. By doing this even if a node does not use up its Vickerey payment it can continue to use it even after the time it is a bandwidth manager.

### 5.4.5 Scaling Weight Factors

The N-Vector which is sent to the bandwidth manager is scaled using the weighing factors to get the cost function. The Cost function is Calculated using the simple weighted calculation

\[
\text{cost} = \sum_{i=1}^{n} W_i \times X_i
\]

The weights decide the importance of each parameter in the network. In a bandwidth constrained network, it may be important to choose a node which sees high bandwidth. Similarly in a power constrained network, it may be important to choose a node that has a high percentage of battery time remaining. These weights can be set administratively. In the proposed scheme we have set up a weight update algorithm which changes the weights of the parameters based on the variation of the usage statistics. As explained in the previous section, when the bandwidth manager is migrated, History information is transmitted. This information characterizes the use of these resources when the parent in question was the bandwidth manager and the historical values of these resources.
The algorithm to do weight scaling is given in Figure 5.2. The scheme used is a history predictor with a history of 1 generation. However if more generations are introduced, the scheme becomes more accurate but at the expense of transmitting more information.

Scaling Weights algorithm
Input: Set of Measured Weights : Measured_Weights[f];
Set of History Weights : History_Weights[f]
Output: Set of Scaled Weights : Weights[f]

proc Scale(Measured_Weights, History_Weights[f]) ≡
X := {}; //set of New Weights
N := size_of(Measured_Weights[i]);
SUM := 0;
AVERAGE := 0;
foreach i ∈ Measured_Weights do
    X[i] = (Measured_Weights[i] − History_Weights[i])/History_Weights[i];
    SUM := SUM + X[i];
od
AVERAGE := SUM/N;
foreach i ∈ Measured_Weights do
    X[i] := X[i] − AVERAGE; Weights[i] := History_Weights[i] + X[i];
    od
.

proc Calculate_COST() ≡
Input: Set of Values : Values[f];
Set of Measured Weights : Measured_Weights[f];
Set of History Weights : History_Weights[f]
Output: Set of Costs : Costs[f]
Weights := Scale(Measured_Weights, History_Weights);
foreach f ∈ Costs do
    Costs[f] := 0;
    foreach i ∈ Weights do
        Costs[f] := Costs[f] + Weights[i] * Values[i];
    od
od
.

Figure 5.2: Algorithm to Scale Weights & Calculate Costs

5.5 Implementation and Testing

5.5.1 Experimental Testbed

To test the system we first simulate it and then implement it on an actual wireless testbed. The system is simulated with 30 nodes operating on the loopback interface. The loopback interface represents an ideal channel with no losses. However, we use this interface only to test the scalability of the scheme. We then test the system on the wireless testbed of the last chapter.

The video application developed in the previous chapter is used in the experiments. The Rate Adaptor
is modified so as to participate in the Bandwidth Manager election. If the node in question is allocated the responsibility of hosting the new Bandwidth Manager, then the RA starts a copy of the Bandwidth Manager locally. All the nodes then re-negotiate with the new Bandwidth Manager.

![Graph showing time required to shift bandwidth manager as the number of nodes increase](image)

Figure 5.3: Time required to shift the Bandwidth Manager as the number of nodes in the network increase

### 5.5.2 Scenarios

In the first scenario all the nodes are simulated by running on the same computer. They communicate over the loopback interface. After an election a new Bandwidth Manager is spawned and all the nodes communicate with the new Bandwidth Manager. This scenario is used to measure the time taken for BM migration and also the data overhead as the number of nodes are increased.

We then run the Bandwidth Manager migration over an actual testbed and measure the time taken to migrate as well as the overhead.

To show that the Bandwidth Manager migration is lightweight, we show the throughput of the video streaming application in two topologies. In the first scenario two independent video streams are sent to two different receivers. In the second topology two streams are sent to the same receiver. In both cases the number of frames of video received per second per stream is measured.
5.5.3 Metrics

Two key properties of the system are tested. First, we test the time taken to migrate the Bandwidth Manager as the number of nodes increases. The time is measured from the point at which the election is started to the point at which the election completes and all the nodes negotiate with the new Bandwidth Manager.

Second, we test the data overhead for conducting the election. This overhead measure includes all requests and replies sent to elect a new Bandwidth Manager.

We then show the throughput of the video application in terms of frames per second achieved in the first topology where the receivers are different. This is tested with a static Bandwidth Manager and a Bandwidth Manager which migrates around the network after a Vickerey Auction. We attempt to show that the migratory Bandwidth Manager(BM) has no effect on the throughput achieved. We repeat the experiment in another simple topology where the video streams are being sent to the same receiver.

5.5.4 Results

Figure 5.5.1 shows the time taken in seconds to shift the Bandwidth Manager in the simulation. As seen in the figure the time taken from the time the election is started to elect and shift the Bandwidth Manager in a network with 30 nodes is 2 seconds. This is negligible when one considers the fact that the bandwidth
management functionality will be hosted at a node for quite a long time. Even if the Bandwidth Manager is hosted at a node only for 5 minutes, the election overhead amounts to 0.6% of the time the node is a Bandwidth Manager which is negligible.

Next, in Figure 5.5.1 the amount of total data sent to conduct the election is studied. Even at 30 nodes the total amount of data sent is less than 30 KB, thus the total election overhead per node is less than 1 kilobyte. This is negligible considering the mega-bit bandwidth of the wireless channels. On the actual wireless testbed, the time required to shift the Bandwidth Manager for a network of 6 nodes in the Ad Hoc network is only 0.6 seconds. Again assuming that the BM functionality would be hosted at a node for more than 5 minutes, this represents a 0.002% of the time a node is a Bandwidth Manager. This is negligible. The amount of data transmitted per node for the election in a wireless network is exactly identical to the simulation. In general the values measured practically do not differ greatly from the simulation because the amount of data transmitted is really low.

From the throughput experiments(Figures 5.5 and 5.6) we see that the throughput of the video is unaffected by the Bandwidth Manager migration. The throughput in both the topologies is similar for the case where the Bandwidth Manager is migrated and when it is kept static. From the graphs, it is also

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{FPS_of_Video_with_a_static_Bandwidth_Manager_and_a_migrating_Bandwidth_Manager.png}
\caption{In this topology two video streams are sent to the same receiver. The first two streams are sent with a BM migrated using a Vickerey Auction and the last two streams are sent with a static BM. The dips in throughput at around time instant 40 are due to the vagaries of the wireless channel. Again there is no noticeable difference in throughput.}
\end{figure}
clear that the Bandwidth Manager migration does not cause any pause in the video transmission. Thus the Bandwidth Manager migration ensures the availability of QoS control while not affecting the transmission of the media content. However if a node in question concurrently renegotiates with a Bandwidth Manager when the migration process is active, the node would need to wait till the new BM is installed.

5.5.5 Explanation

Thus from the experiments and the simulation it is clear that the Vickerey Auction mechanism to elect the new Bandwidth Manager consumes very little resources and is an ideal fit for the target scenario. System availability (which was the main driver to migrate the Bandwidth Manager) remains 100% as the nodes continue to send data to each other even when the election is in progress. Thus we feel that this scheme is an ideal solution for the problem.
5.6 Proof of the Optimality of Vickerey Auctions

In this section we will prove the optimal strategy in a Vickerey Auction is for the bidders to bid their true valuations. The work is adapted from [16].

Let us consider that a quantity $Q$ of a resource is to be shared among several players. Auction is a mechanism consisting of players submitting bids, i.e. declaring their desired share of the resource and the price they are willing to pay for it, and the auctioneer who allocates shares of the resources based on their bids.

Let $I = \{1, \ldots, K\}$ be the set of players competing for the resource $Q$. We define the $k$-players bid as the
vector \( s_k = (q_k, p_k) \in S_k \equiv [0, Q] \times [0, \infty) \), where \( q_k \) is the amount of resource that the player is bidding for, and \( p_k \) is the price per unit of resource he is willing to pay. A bid profile \( s \) is a \( K \times 2 \) array, that contains the bids of the players that participate in the game.

\[
s = \begin{pmatrix}
  s_1 \\
  s_2 \\
  s_3 \\
  s_4
\end{pmatrix} = \begin{pmatrix}
  q_1 & p_1 \\
  q_2 & p_2 \\
  q_3 & p_3 \\
  q_4 & p_4
\end{pmatrix} \quad (5.1)
\]

We will now introduce the following notation, which is really helpful in order to present all subsequent propositions and lemmata:

\[
qs = s \begin{pmatrix}
  1 \\
  0
\end{pmatrix}, \quad ps = s \begin{pmatrix}
  0 \\
  1
\end{pmatrix}, \quad \text{so that} \quad s = (ps, qs) \quad (5.2)
\]

We similarly denote \( k \)-players quantity and price in the following manner

\[
qs_k = s_k \begin{pmatrix}
  1 \\
  0
\end{pmatrix} = (q_k, p_k) \begin{pmatrix}
  1 \\
  0
\end{pmatrix} = q_k, \quad ps_k = s_k \begin{pmatrix}
  0 \\
  1
\end{pmatrix} = (q_k, p_k) \begin{pmatrix}
  0 \\
  1
\end{pmatrix} = p_k, \quad s = (ps_k, qs_k) \quad (5.3)
\]

and the opponent bid profile as:

\[
s_{-k} = \begin{pmatrix}
  s_1 \\
  \ldots \\
  s_k-1 \\
  s_k+1 \\
  \ldots \\
  s_k
\end{pmatrix}, \quad \text{so that} \quad s = (s_k, s_{-k}) \quad (5.4)
\]

The allocation rule \( A(s) \) is a function that describes the procedure of dividing the resource among players and assigning prices according to their bids. The allocation is done by the auctioneer. \( A(s) \) is formally defined as shown below:
\begin{equation}
A : S^K \rightarrow S^K, \text{ where } S^K = \prod_{k=1}^{K} S \text{ and } S = [0, Q] \times [0, \infty) \tag{5.5}
\end{equation}

The k-th row of the matrix \( A(s) \) is the allocation to player k, which consists of the amount of resource the player actually gets and the price he pays for it.

\[
A_k(s) = (qA_k(s) \quad pA_k(s))
\tag{5.6}
\]

We say that an allocation rule \( A(s) \) is feasible if both the following conditions are satisfied: The sum of all the amounts allocated to players does not exceed the available resource \( Q \).

\[
\sum_{k=1}^{K} qA_k(s) \leq Q \tag{5.7}
\]

Every player is never given more quantity than he asks for, and the price he has to pay does not exceed the one declared through his bid.

\[
A(s) \leq s \tag{5.8}
\]

where the operator less-or-equal (\( \leq \)) denotes element by element comparison.

Each player’s objective is described by a function \( u_k(s) \) which is called utility function, or simply utility. This function takes as argument a bid profile and a returns a value that expresses how profitable is the corresponding allocation to a particular player. In the following sections, we will use the definition for \( u_k(s) \), shown below.

\[
u_k(s) = \theta_k qA_k(s) - qA_k(s)pA_k(s) \tag{5.9}
\]

We assume that player k has a valuation \( \theta_k \geq 0 \) for each unit of the resource he gets, so the total value of his allocation is \( \theta_k qA_k(s) \). His utility is thus defined as the difference between the value and the cost of the...
acquired quantity.

$$
\begin{pmatrix}
Q & p_1 \\
Q & p_2 \\
\ldots & \ldots \\
Q & p_4
\end{pmatrix}
$$

(5.10)

Finally we consider an auction game is defined by an allocation rule, a set of utility functions and an amount of available resource:

$$(Q, u_1(s), u_2(s), \ldots, u_k(s), A(s))$$

(5.11)

Now let us turn our attention to Vickerey Auctions. In the Vickerey Auction we assume that the resource is non-divisible. This means that $q_k = Q$ for all $k$. The bid profile thus has the following form:

Let us consider that players are ordered in such a way that :

$$0 \leq \theta_1 \leq \theta_2 \leq \ldots \leq \theta_K$$

(5.12)

According to the Second Price allocation rule, the player who bids at the highest price gets the resource, and he pays the second highest price. Now, let us consider each players utility $u_k(s)$. As we mentioned in the previous section the utility function for each player is defined as the difference between the actual value of the quantity he gets and its cost. Thus, for the Second Price auction game:

$$u_k = \theta_k - \max_{i \neq k} p_k, \text{ if } p_k \max_{i \neq k} p_l$$

(5.13)

or 0 otherwise. Now we will state and prove the following lemma:

**Incentive Compatibility: 1** For each player $k$, the strategy of bidding at his own valuation $p_k = \theta_k$ weakly dominates other strategies. i.e.

$$u_k(p_1, p_2, \ldots, \theta_k, \ldots, p_k) \geq u_k(p_1, p_2, \ldots, p_k, \ldots, p_K)$$

**Proof:** Let $r_k = \max_{i \neq k} p_k$. The proof for the lemma presented above derives directly from the graphical representations of $u_k$ as a function of $r_k$ as shown in figure 5.7.

For $r_k \leq \theta_k$ , we observe that $u_k$ decreases linearly, until it reaches zero, in situations 1 and 3, whereas, in
situation 2, it decreases rapidly to zero at a particular value \( p_k \) before the valuation \( \theta_k \). This happens because for \( r_k p_k \), the player is being allocated no quantity and therefore his utility is zero.

For \( r_k \geq \theta_k \), the utility is zero in situations 2 and 3 and negative for some values of \( r_k \), in situation 1. This happens because the player bids at prices higher than his valuation and therefore has to pay more than the value of the resource, in case he wins the auction game.

Comparing situations 1, 2, and 3 we observe that for every \( r_k \) the following property holds:

\[
u_k(p_1, p_2, \ldots, \theta_k, \ldots, p_K) \geq u_k(p_1, p_2, \ldots, p_K, \ldots, p_K)\]
Chapter 6

Audio in a Tele-Immersive System

Tele-Immersive systems represent the next generation of collaborative systems which allow the capture and representations of three dimensional models unlike the two dimensional capture and representation of immersive systems today.

At the sending site, multiple camera clusters are used to capture the 3D model (depth as well as color information) of real world objects in real time. The 3D models in the form of multiple video streams are streamed over the Internet and rendered on multiple display devices at the receiving site. In reality, both of the sites are capable of capturing and rendering the 3D models. Despite their potential, building 3D-based tele-immersive systems faces numerous challenges, due to the high-fidelity requirement on 3D reconstruction and the lack of existing hardware and software tools. Specifically,

1. The basic 3D camera cluster must be constructed by hand, due to the lack of existing off-the-shelf commercial products;

2. The setup of multiple camera clusters must be custom-designed, with the particular communication application (e.g., conferencing or distributed dancing choreography) in mind;

3. Multiple cameras must be accurately calibrated, in order to obtain high quality 3D models;

4. The high bandwidth 3D data must be streamed across the best-effort Internet in a timely fashion, in order to facilitate interactive communications.

The TEEVE application is modeled as a distributed multi-tier application (Figure 6.1). The first is the capturing tier that consists of 3D camera clusters with each cluster reconstructing one 3D video stream. The 3D streams are then forwarded over LAN to service nodes for compression. The second is the transmission tier that streams the compressed video over Internet2. The third is the displaying tier which decompresses and renders the received 3D-streams into an immersive video and forwards it over LAN to multiple displays which presents a pictorial view of tele-immersive environments with 3D camera clusters and displays. Each tier of the TEEVE framework deploys algorithm, service and protocol require clear, sound application data and timing models.
6.1 Design Space for Audio

Figure 6.1: Tele-Immersion System Architecture

Figure 6.2: Design Space for 3D Audio

Figure 6.2 gives an overview of the design space for Audio in a tele-immersive system. The design space can be partitioned into two channel and multi channel audio. This can also be thought of as 2D and 3D audio as two channel audio fails to provide a sense of depth and immersion.

Two-dimensional audio has either one or two channels of audio and is the most common format to capture and reproduce. There are two main application classes for this kind of audio. One application class is capture and reproduction of speech. For this one channel is usually considered sufficient. The other is the capture and reproduction of music.

The main problem of two-dimensional audio is the creation of a three dimensional enveloping effect for which more channels are required. This can be achieved in two ways. By discrete multi channel...
representation and wave field effect.

Discrete multi-channel surround attempts to create an enveloping effect by using multiple speakers and routes the right frequency to the appropriate speakers. However it suffers from the fact that the enveloping effect is restricted to a small area of the room. Wave field effect covered in Figure 6.3 recreates a sound envelope by an array of speakers. However capture of such audio is difficult.

For the purposes of the tele-immersive experiment we have concentrated on the capture of two-dimensional audio for many reasons:

1. The live capture and processing of 3D audio is difficult with the current equipment

2. The utility of such capture is not very useful in the experiments that we have conducted primarily collaboration and tele-immersive dancing.

3. Two dimensional audio system gives us a good baseline for future multichannel immersive systems.

However this could be a welcome addition to future systems.

6.2 Usage Scenarios

Some of the usage scenarios that audio in a tele-immersive system include but are not restricted to

1. Recording of entertainment events such as dancing or music events
2. Immersive collaboration for events like dancing.

3. Remote collaboration for applications like meetings.

In the first application scenario, there is no need for live processing of audio as audio can be post processed to generate the required number of audio channels. However, the second application scenario poses interesting challenges. Here we foresee two important applications: one is the synchronized playback of music to which collaborators can dance to. The other is the voice commands for enabling communication during performance. For the last application scenario, the most important audio input required is a audio conferencing system.

In our tele-immersive system we concentrate on the last two scenarios looking primarily at the playback of synchronized stored music files for dancing and a VoIP based audio conferencing system for remote collaboration.

For the playback of static audio files no streaming is required, but for the VoIP scenarios live streaming mechanisms must be enabled.

6.3 Implementation of Music Playback:

For music playback a simple audio player was developed in Java. The audio player plays back one frame of audio in a lock step synchronized fashion. The mp3 player is based on the JLayer framework which is an open source framework to playback MP3 files. A socket is created between the client and the server where both have the music file stored locally. The client sends commands to the server which receives the command and plays back one frame of audio. This proceeds while taking account of the round trip time of sending the message in order to keep both programs synchronized.

6.3.1 Implementation of VoIP system

Overview of VoIP

Voice over IP or IP telephony refers to the technologies and protocols involved in creating a telephony system which uses the internet protocol IP to transfer packets instead of the legacy telephony systems of the past. The obvious problems with such an approach is that the IP systems are best effort, with no guarantees of delivery. Using VoIP over an unreliable network or a network with excessive delay results in poor voice quality. In contrast, VoIP over a decongested, minimal-delay network can provide voice quality as good as (or even better than) that provided by traditional telephone networks. Hence a lot of effort has
been expanded in order to achieve some soft guarantees with respect to the quality of audio. VoIP also adds a number of protocols to make its usage as similar to the standard telecommunication system as possible. A detailed overview of VoIP is given in [32]. A small overview is given here. The VoIP protocol stack is shown in the figure 6.4

Fundamentally, IP telephony relies on the end-to-end paradigm for delivery of services. Signaling protocols are between the end systems involved in the call; network routers treat these signaling packets like any other data, ignoring any semantics implied by them. Note, however, that IP telephony can make use of signaling routers (which are effectively proxies) to assist in functions such as user location. In this case, these proxies can be used for routing only of initial signaling messages. Subsequent messages can be exchanged end-to-end. As a consequence of the end-to-end signaling paradigm, call state is as well end to end, as are instantiation of many telephony features. The Internet itself is both multi-service and service-independent. It provides packet-level transport, end-to-end, for whatever services are deployed at the end systems through higher layer protocols and software.

Protocols such as RSVP are used to reserve resources. These protocols are application independent, and reservations may take place before or after actual flow of data begins. When used after the flow of data begins, the data will be treated as best effort. As a result, IP telephony can be used without per-call resource reservation in networks with sufficient capacity. Functions, such as payment and carrier selection, are more readily handled by the protocols, such as RSVP and RTSP, which carry the addresses. RTP or the Real-time Transport Protocol forms a major part of the VoIP protocol suite and is used for data transfer. Some of the features of RTP for data transfer are:

Figure 6.4: VOIP Protocol Stack

![Figure 6.4: VOIP Protocol Stack](image-url)
1. Sequencing

2. Intra-media synchronization

3. Inter-media synchronization

4. Payload identification

5. Frame indication

6. Multicast-friendly

7. Media independent

8. Mixers and translators

9. QoS feedback

10. Loose Session Control

11. Encryption

**RTCP: Control and Management**

The Real Time Control Protocol, RTCP, is the companion control protocol for RTP. Media senders (sources) and receivers (sinks) periodically send RTCP packets to the same multicast group (but different ports) as is used to distribute RTP packets. Each RTCP packet contains a number of elements, usually a sender report (SR) or receiver report followed by source descriptions (SDES). Each serves a different function: Sender Reports (SR) are generated by users who are also sending media (RTP sources). They describe the amount of data sent so far, as well as correlating the RTP sampling timestamp and absolute (all clock time to allow synchronization between different media. Receiver Reports (RR) are sent by RTP session participants which are receiving media (RTP sinks). Each such report contains one block for each RTP source in the group. Each block describes the instantaneous and cumulative loss rate and jitter from that source. The block also indicates the last timestamp and delay since receiving a sender report, allowing sources to estimate their distance to sinks. Source Descriptor (SDES) packets are used for session control. These packets are sent periodically to control the channel.

**Resource Reservation**

Given the importance of telephony services, a significant fraction of the Internet bandwidth is consumed by voice and video, that is, RTP-based protocols. Due to its tight delay constraints, IP tel streams are also
likely candidates for guaranteed QOS. Unfortunately, existing proposals such as RSVP are rather complex, largely due to features. There are also proposals of simplified, sender-based resource reservation protocols which can be found (albeit not using RTCP) in the Scalable Reservation Protocol (SRP). It has to be noted that resource reservation is not supported over the current internet. The OpenH323 library however adds a differentiated service request in the TOS field of the IP header of the packet.

6.3.2 Overview of PWLIB

PWLib is a moderately large class library that has its genesis many years ago as a method to product applications to run on both Microsoft Windows and Unix X-Windows systems. It also was to have a Macintosh port but this never eventuated.

Since then the system has grown to having good application to areas other than mere Windows GUI portability. Classes for I/O portability, multi-threading portability, aid in producing unix daemons and NT services portably and all sorts of internet protocols were added over the years. All this over and above basic ”container” classes such as arrays, linear lists, sorted lists (RB Tree) and dictionaries (hash tables).

6.4 Overview of OpenH323

OpenH323 is a project committed to the development of an Open Source H.323 protocol stack that is available for use by both private and commercial users. It is a full fledged implementation of the VoIP protocol stack. It is built on top on PWLIB which is a portable library for application development. PWLIB applications work well on windows or Linux, Thus the OpenH323 code which is built above PWLIB works on both windows and a variety of POSIX operating systems.

Since Open H323 is a library implementation, applications can be developed on top of it for a variety of purposes. We use the OpenPhone implementation which is a windows GUI application which supports all the functionality of OpenH323 library.

6.4.1 OpenH323 Call Diagram

The Call Graph in figure 6.5 is a reproduction of [1], It Describes the call stack for OpenH323. The highlighted portions refer to the call stack which is relevant to our experiments and is used for synchronization as described in the next chapter.
Figure 6.5: Call Graph of OpenH323, The shaded area represents the classes of interest for modification

Figure 6.6: Class Diagram of OpenH323 protocol
6.5 Experiments with Voice

A screen shot of the OpenPhone application windows is shown in 6.7. Two kinds of experiments were performed with the application. In one set of tests the qualitative aspects of the audio were studied. In this set of experiments audio is captured and transmitted from the site at the University of Illinois to
the site at the University of California at Berkeley where it is recorded. The quality of audio is compared
to the original over an extended period of time. From the qualitative evaluation it is discovered that the
quality of audio is perfectly fine for voice applications but suffers greatly for music or vocal applications
where a part of the frequency spectrum is cut off.

The quantitative evaluation involved the monitoring of the packet loss rates during audio sessions where a
large number of audio files are streamed. The number of packets lost remains low. However in our initial
experiments the packets which are late caused problems as these packets are as destructive as packets lost
in terms of the final audio quality. This however can easily be corrected by increasing the size of the buffer
used. Thus in our setup the jitter buffer is increased to 1000 ms. After increasing the jitter buffer. The
screen shot shown in Figure 6.8 shows the loss rate of the packets which is zero over a long session.

Figure 6.9: Scatter Plot showing the arrival of Audio Frames and Video Frames

The setup is also evaluated by sending audio and tele-immersive video together and measuring the effect of
one on the other. The loss rate is again observed not to differ much from when the audio is transmitted
independently.
The final tests involved the transfer of audio and video in a lock step fashion and studying the arrival time of audio and video. The graph shown in Figure 6.9, compares the time of arrival of audio and the corresponding video frame. This graph is constructed after correcting for the fact that the audio and video are sending at different frame rates. The graph also compares the arrival rate with the line $y=x$. Points on the line $y=x$ arrive at the same time. From the graph it is seen that the audio and video arrive approximately at the same time and this fact is used in the next section for synchronization.
Chapter 7

Synchronization in a TeleImmersive Environment

7.1 Introduction

The addition of audio to a tele-immersive system greatly adds to its usability. However audio without sync with the video can cause irritation to the final user of the video. Ralf Steinmetz in his seminal work [34] studied the human perception of synchronization. The results presented in the paper help guide our work. However, some of the results presented are not directly applicable to our work because:

1. The frame rate is considerably lower.
2. Lip synchronization has tight synchronization constraints. However in the low resolution of the TI system, the Lip is not readily observable by the user.
3. Most of the applicable application areas have a higher emphasis on action than conferencing.

Hence the synchronization constraints of the system are considerably lower than a generic video conferencing application which concentrates mainly on a headshot of the user. The rest of this chapter proceeds as follows. First the arrival rate of the audio and video is studied, followed by a description of the simple scheme used.

7.2 Arrival rate of audio vs. video

The audio and video sub systems in our tele-immersive environment are implemented as separate systems. Another source of problem in such a system is the fact that while video is based on a TCP based system, the audio system is implemented over UDP. Thus we transmit audio and video from the site at the University of Illinois to the University of California at Berkeley, and measure the arrival time of the audio and video packets. The audio and video are synchronized at the start and then there is no further synchronization between them. Since a video frame is not a single frame but in fact is composite and consists of many composite video frames. Two statistics are studied, namely, the arrival time of the audio
frame when we compare to the median video frame, and the arrival time of the audio frame when compared to the last frame of the video. The graphs shown in Figure 7.1 and Figure 7.2 are constructed after accounting for the disparity in the frame rate as the audio is transmitting at 30 frames per second and the video is transmitting at 7 frames per second.

![Graph showing average arrival time of video frames](image)

Figure 7.1: Plot showing the arrival of audio frames when compared to the arrival of the median macro - video frames

The graph is very encouraging as it shows that:

1. There is a high degree of synchronization between the arrival rate of the audio and the video.

2. There is no difference in synchronizing between the last video frame and the media video frame.

3. The audio and video do not drift out of synchronization although the systems are diverse.

Using this information a simple synchronization scheme is developed which is presented in the next section.
Figure 7.2: Plot showing the arrival of audio frames when compared to the arrival of the last macro-video frame
7.3 Simple Synchronization

We experiment with a simple receiver driven synchronization. In this scheme a message is sent from the audio which is the master scheme to the video for synchronization. This is done after taking into account the disparity between the audio and the video frame rate. The video is buffered to that point and on receiving the message, the video is extracted from the buffer and played back. This simple scheme needed to be modified as the playback times for the audio and video are different. The video takes more time to render. Hence the audio is delayed to come in line with the video.

7.4 Qualitative Evaluation

It is difficult to measure the synchronization parameters quantitatively. Hence a qualitative study is made. The audio and video are captured and the video is played back to measure if there is any noticeable lag. After evaluation from a number of human subjects, it is found that the synchronization between the video and the audio is good with no noticeable lag.
References


