Optimizing Perceptual Quality for Online Multimedia Systems with Fast-Paced Interactions

XU, Jingxi

A Thesis Submitted in Partial Fulfilment
of the Requirements for the Degree of
Doctor of Philosophy
in
Computer Science and Engineering

The Chinese University of Hong Kong
August 2017

Thesis Assessment Committee

Professor WONG Tien Tsin (Chair)

Professor WAH Wan Sang Benjamin (Thesis Supervisor)

Professor LYU Rung Tsong Michael (Committee Member)

Professor LAU W. H. Rynson (External Examiner)

Abstract of thesis entitled:

Optimizing Perceptual Quality for Online Multimedia Systems with Fast-Paced Interactions

Submitted by XU, Jingxi

for the degree of Doctor of Philosophy

at The Chinese University of Hong Kong in 2017

This research addresses the optimization of perceptual quality of online interactive multimedia applications with fast-paced interactions, including but not limited to voice-over-IP, videoconferencing, and multi-player online games. The main characteristic of these applications is that they need real-time performance and thus have limited time for the evaluation, calculation, and optimization of perceptual quality, where perceptual quality is the subjective satisfaction of user experience. It involves complex mapping from system controls to human perception that is difficult to model in a closed form. Rather, it is measured by offline pairwise subjective tests with considerably large overhead. To optimize the perceptual quality of applications interested in this thesis, we need an efficient methodology to collect sufficient human opinions regarding system controls, and generalize the opinions at run time per the running context and the network condition, and optimize the perceptual quality accordingly.

In this thesis, we propose a general framework for maintaining a stable network transmission for an interactive multimedia application. Based on a large scale measurement of nowadays Internet conditions, we develop a real-time algorithm in this layer for concealing network impairments based on run-time statistics.

Secondly, to model human opinions, we propose a probabilistic model called Just-Noticeable Difference (JND) surface, which is a function that maps the change of a control to the subjective awareness of the change. Along with a dominance property that describes the monotonicity of the data in the JND surface, we utilize an efficient algorithm for measuring the human opinions over the whole control space with only a small number of offline subjective tests.

Thirdly, to optimize perceptual quality at run time, we utilize another dominance property we have discovered to combine JND surfaces corresponding to a single independent control or multiple dependent controls.

Fourthly, we generalize the JND surfaces from the offline measured surfaces per the run-time network condition by transformations, and then combine them using probabilistic tools. The resulting combined JND surface is used in the objective function for the optimization of perceptual quality.

To verified the techniques and algorithms developed, we demonstrate their advantages with both proprietary VoIP systems including Skype and MSN, and an online game BZFlag.

摘要

本研究著眼於優化在線快節奏交互式多媒體系統的感知體驗，涉及的系統包括但不限於網絡電話、視頻聊天系統以及多人在線遊戲。這類系統的主要特徵在於，他們對實時性的高要求，導致系統僅有極為有限的時間進行評測、計算及優化用戶的感知體驗（亦被稱為主觀滿意度）。感知體驗涉及複雜的從系統控制到人類感知的映射，該映射難以用準確的形式解來建模。因此，感知體驗通常需要通過費時的離線用戶測試來獲取。為自動地最優化相關應用的感知體驗，我們需要一種高效的方法來收集足夠的針對系統控制的用戶意見，然後再將該意見在系統運行時根據上下文與網絡環境進行泛化。

本論文首先會提出一個泛用的框架，用以維持一個穩定的網絡環境，供交互式多媒體系統傳輸數據。我們根據收集到的大規模網絡數據，提出一個實時的算法，根據在線測試結果來解決網絡故障。

接著，我們提出一個名為最小可覺差曲面的概率模型，用以對用戶意見進行建模。該模型是一個將控制上的參數變動映射到主觀上的評價變動的函數。配合估優性質，我們提出一種高效的方法，用很少的離線用戶測試就能獲取整個控制空間上的用戶意見。

再者，我們提出利用另一估優性質來將與單個或多個控制相關的數個最小可覺差曲面合而為一，用以在線優化涉及多控制的感知體驗。

然後，我們提出根據網絡環境來變換及泛化離綫獲取的最小可覺差曲面，再用概率工具將它們合併。該合併曲面可作為在線優化感知體驗時的目標函數。

最後，我們用提出的方法來優化商業軟件 Skype 和 MSN，以及在線遊戲 BZFlag，以展示其優勢。

Acknowledgement

I would like to thank my thesis supervisor, Prof. Benjamin W. Wah, for his support during my thesis study. I would like to especially acknowledge his patience in guiding me to discover the most important problems in research, develop the theory with reasonable and justifiable assumptions, and overcome the difficulties by a divide-and-conquer methodology.

I would like to thank my committee members, Prof. Tien Tsin Wong for being the chair of the committee, Prof. Michael Rung Tsong Lyu for providing valuable advice during my whole study, and Prof. Moon Chuen Lee for being the previous committee member. I would like to especially thank Prof. Rynson W.H. Lau for being my external marker.

I would like to thank my entire family for their belief and support.

I would like to especially thank my friend Zhitang Cheng for his encouragement and help. I would also like to thank my colleagues Feihu Zhang, Furui Liu, Lili Liu, Weicong Liu, Mengyi Zhang, Kai Xing, Xiaotian Yu, and Yi Mei.

To my family for their love and support.

Contents

List of Figures

List of Tables

Abbreviations

JND	Just-noticeable Difference
ASQ	Audio Signal Quality
ATK	Attack
CE	Conversational Efficiency
CPR	Correctly-Received Packet Ratio
CS	Conversational Symmetry
DEF	Defend
EED	End-to-End Delay
FEC	Forward-Error Correction
HRD	Human Response Delay
ITU	International Telecommunication Union
LL	Local Lag
LPF	Local Perception Filter
MED	Mouth-to-Ear Delay
MMO	Multi-Player Online Games

MOS	Mean Opinion Score
MS	Mutual Silence
MTU	Maximum Transmission Unit
NAL	Network Abstraction Layer
QoE	Quality of Experience
QoS	Quality of Service
QP	Quantization Parameter
TFRC	TCP-Friendly Rate Control
UPR	Unreceived Packet Ratio
VoIP	Voice-over-IP

Chapter 1

Introduction

1.1 Online Multimedia Application Systems

1.1.1 Online Interactive Multimedia Systems

Online interactive multimedia applications are popular with improvements on the processing power of devices and the Internet bandwidth. These systems allow users to interact with remote users or devices as if they were in the same place, and provide convenient solutions for international cooperation.

Examples of these systems are voice-over-IP (VoIP), online games, remote virtual-reality interaction, remote surgery, remote cooperation tasks, etc. We use two representative multimedia systems, VoIP and online games, to illustrate our techniques developed for run-time applications in the thesis, as they are popular applications used by ordinary users rather than professional applications.

VoIP includes the popular online voice conversational applications as well as the videoconferencing over the Internet. Popular VoIP applications include Skype [84] and Google Talk [30]. Cisco [19] and Huawei [42] also have proprietary devices that can run VoIP without a computer. In a VoIP application, two or more people are talking as if they were face-to-face using the Internet.

Multi-player online games refer to computer games with multiple players who interact remotely in the real world over the Internet. Among them, fast-paced online games are increasingly popular with improvements in network bandwidth and reduced latency. Here, "fast-paced games" refers to games in which the reaction time required is near the limit of human reaction time (215 ms on average according to an online test [44]). While slow-paced online games can be trivially synchronized with a sufficiently long buffering time for late packets and error correction, fast-paced online games require a tight waiting time, which raise a challenging task for buffer allocation and online optimization. For this reason, we focus on those games with action durations ranging from 300 ms to 700 ms, and in particular scenarios with precise weapons which require accurate synchronization. Examples include shooting games with bullets or missiles [26, 64], fighting games with fast punching or kicking [24,103], and racing games with weapons shooting enemies [65]. In an online shooting game, players shoot targets of other parties with missiles in the same map.

Another example of online interactive multimedia systems is a remote robotic surgery system. In remote robotic surgery, a doctor can use a virtual-reality device to perform operations on a patient in another country. This system is more related to human-computer interaction so we do not focus on it in this thesis.

In this thesis, we study online interactive multimedia systems like those discussed above. These systems have some general characteristics. Firstly, more than one user are using the system simultaneously, and there are interactions among them. Secondly, the interactions should be presented to users as fast as possible to allow real-time cooperation. Thirdly, communication and collaborate data are transmitted over the Internet. Finally, users of these systems are sensitive to the latency in the interaction.

To elaborate these characteristics, consider a two-party VoIP or videoconferencing conversation. After a user speaks a talk segment, the data will transmit to the listener and play at the speaker in real time. If there is a latency of the transmission due to the delay in the Internet, the talk segment will be heard later by the listener. Because the listener should respond after hearing the talk segment, she will in turn speak later. This loop of delay will extend the overall duration for a conversation even with the same content, and make the conversation inefficient, which can degrade the quality of experience (QoE).

In the case of an online game, for example a shooting game, if there is latency in the Internet connection, players can find it difficult to beat other targets. This is because whether the other targets have been shot depends on how other players control their devices, say they can dodge the shot at the last second. The result of the dodging is then conveyed back to the attacker with a latency. This will lead to a blank period in which the bullet has shot the target but the target will not be destroyed until the dodging result has been received, and will hurt the QoE of the system.

Therefore, for these systems it is important to optimize the quality of experience (QoE) perceived by users, especially when the application runs on an Internet connection that is unreliable and has latency. There is a need of a general theory on the online optimization of QoE for the interactive multimedia systems over the Internet. In this thesis, we adopt a VoIP as well as an online shooting game as running examples to help elaborate our approaches, even though our approaches can be employed by other online interactive multimedia application systems.

1.1.2 Architecture

To clarify our study, it is necessary to depict the architecture of the systems we study, and point out the most important parts that we focus on in this thesis.

Overview

Figure 1.1 presents the overview of the system architecture. We relate our system to the Internet protocol suite (a.k.a TCP/IP 4-layer model) [29] as follows. We schedule the multimedia content and optimize its quality in the Application Layer of the protocol. We build a Network-Control Sub-Layer (Network-Control Layer for short) on top of the TCP and UDP in the Transport Layer. We make no changes in the Internet Layer or the Link Layer.

In detail, we control the presentation and schedule of multimedia contents in the *application layer*. In a VoIP, this can be the codec of audio and video, and the play-out scheduler of the conversation. In an online shooting game, this can be the graphics engine, the event presenter and the collision detector. Because this layer controls the direct interface with users, it is the most important layer that we study.

The application layer should run under existing network condi-

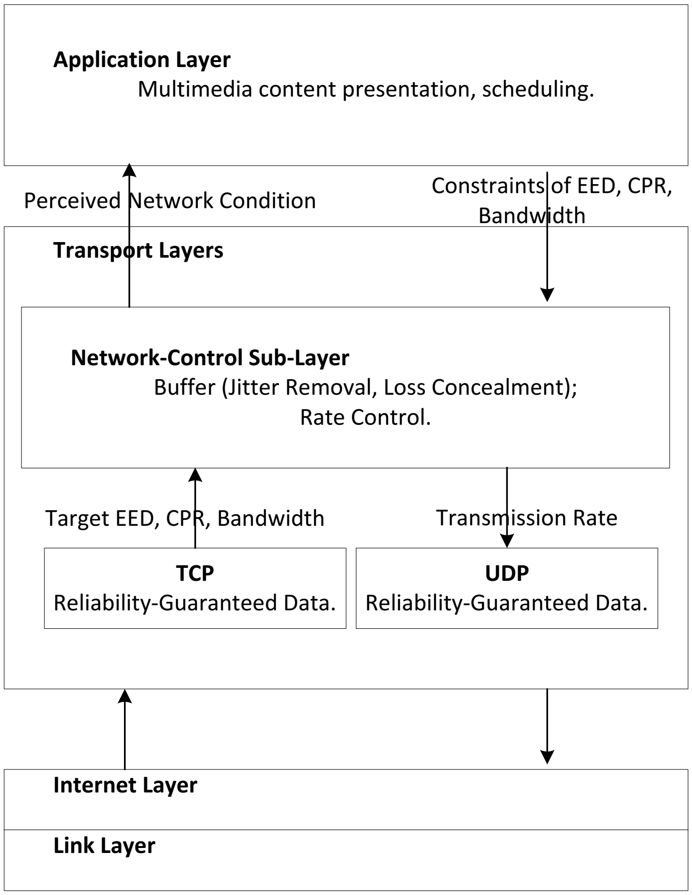

Application Layer
Multimedia content presentation, scheduling.
Perceived Network Condition
Constraints of EED, CPR, Bandwidth
Transport Layers
Network-Control Sub-Layer
Buffer (Jitter Removal, Loss Concealment);
Rate Control.
Target EED, CPR, Bandwidth
Transmission Rate
TCP
Reliability-Guaranteed Data.
UDP
Reliability-Guaranteed Data.
Internet Layer
Link Layer

Figure 1.1: Overview of the architecture of the online interactive multimedia system.

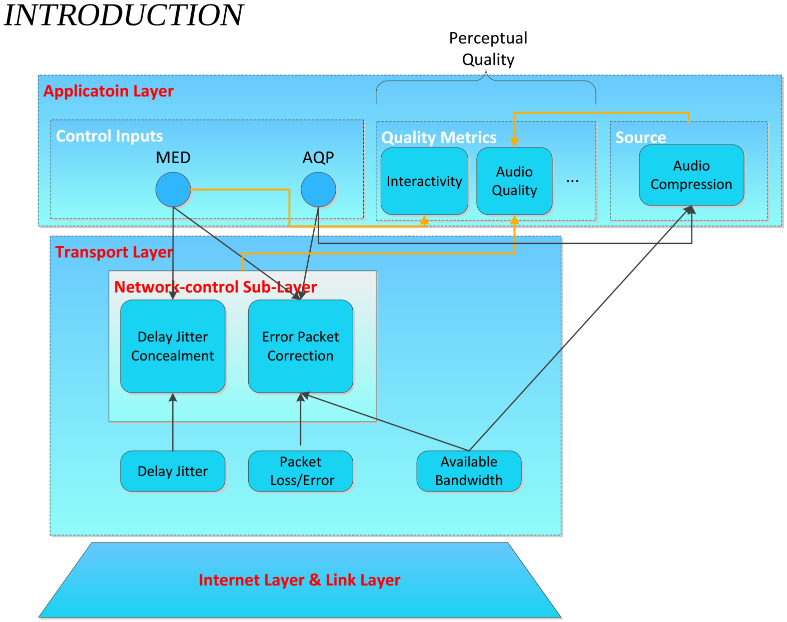


Figure 1.2: Architecture of VoIP systems.

tions. However, it does not directly interact with the Internet. With buffering and rate control strategies, a *network-control Layer* serves as an enhanced layer above the TCP/UDP protocol in the *Transport Layer* to provide network condition that can satisfy the requirements of the application layer. In VoIP and online game, this layer controls how the data are packetized and transmitted, and how the received packet are buffered and corrected. We transmit most real-time data of user inputs with UDP to assure the timing, and transmits system control data that require high reliability with TCP. Because TCP and UDP already satisfy our requirements, we do not further improve them in this thesis.

Finally, the bottom few layers are the *Internet Layer* and the *Link Layer* that we do not modify in this thesis.

VoIP

We depict the architecture of a VoIP system in Figure 1.2. It is shown that there are two important layers inside this application: the transport layer and the application layer.

VoIP/videoconferencing applications rely on an Internet connection to transmit the voice and video data. For this reason, its quality highly depends on the run-time network condition. Because the Internet condition is not reliable, we adopt a network-control layer to maintain a satisfactory channel quality.

In the application layer, these systems control the mouth-to-ear delay (MED) that affects both the conversation duration and the buffering period. They also control the audio quality parameter (AQP) and video quality parameter (VQP) that can determine the transmission bandwidth and the signal quality.

For simplicity of the demonstration, in this thesis we focus on the discussion of the voice-only VoIP. However, our approaches can also be adopted in videoconferencing, and discussion on videoconferencing will also be presented briefly in later chapters.

Online Games

Online games also include the layers in the Internet protocol suite. We depict their common architecture in Figure 1.3.

Rather than transmitting voice over the Internet, online games transmit real-time information of the events as well as the status of the objects. The network-control layer is used for maintaining the reliability of the transmission, including both the stable end-to-end delay (EED) as well as a correctly-received packet ratio (CPR). An

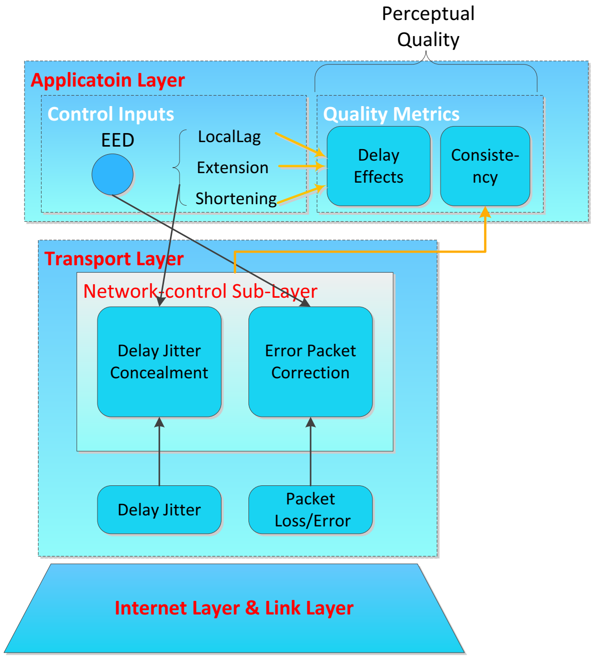


Figure 1.3: Architecture of fast-paced online games.

unstable EED can make the players feel the game “lag”, while a low CPR can produce glitches. Both of these effects are undesirable in online games. Further discussion of EED and CPR will appear in Section 1.1.4.

The application layer in online games controls the scheduling and the presentation of the game events, including the start and complete of every event. These controls are further constrained by the EED.

It is noticed that the architecture of an online game is similar to that of a VoIP. EED (equivalent to MED) is the major system control of both these systems.

Further discussion on the quantitative quality metrics and the perceptual quality will be left to Chapter 2.

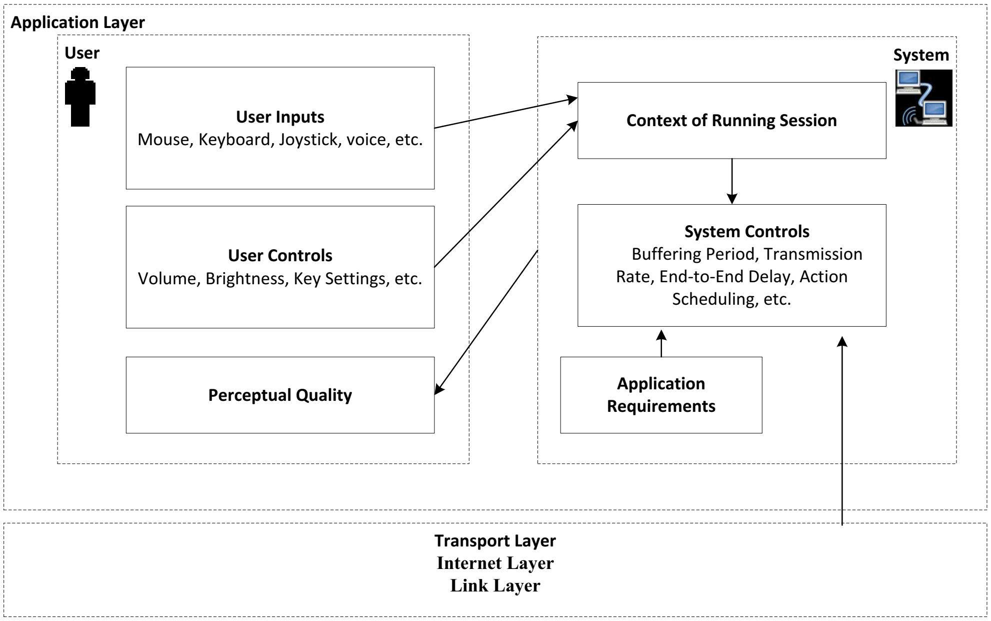

Application Layer
User
User Inputs
Mouse, Keyboard, Joystick, voice, etc.
User Controls
Volume, Brightness, Key Settings, etc.
Perceptual Quality
System
Context of Running Session
System Controls
Buffering Period, Transmission Rate, End-to-End Delay, Action Scheduling, etc.
Application Requirements
Transport Layer
Internet Layer
Link Layer

Figure 1.4: The input/output flow of the system in the perspective of controls.

1.1.3 Controls

In another perspective, Figure 1.4 depicts the flow of the feedback loop of the controls in online interactive multimedia application systems.

Inside the loop, the source of the change of states include user inputs, user controls, application requirements, and the network condition.

User inputs include users' interactions with the input hardware, while user controls include the basic settings in the hardware that can be easily adjusted by users at any time. The above two control sources are application-dependent inputs, which are under users' control but not under the system's control. On the other hand, the context of the running session is an abstraction of user controls and user inputs into quantitative quality metrics. These metrics serve as inputs to control the operation of the system and are under the system's control.

The application itself has some basic requirements that should be satisfied in run time no matter how the network condition changes. As the Internet behavior changes at run time, the network condition is also abstracted into quantitative metrics that are used as inputs to control the operations of the system.

All the inputs above are given to the system-control component that generates control signals to control the operation of the system, the system then output the results at the output interface of the multimedia system, which can then be perceived by users as perceptual quality.

We differentiate user input, user control, and system controls in these application systems. A *user input* is an operation by a human through an input hardware, while a *user control* is a setting of the hardware that can be changed by the user at any time.

As an example, user input in a VoIP system is the user voice in a conversation collected by a microphone. In an online game, user inputs can include the mouse press, the keyboard press, and the button press with the joystick.

Examples of user control include the volume of the microphone and the speaker in a VoIP system, as well as the brightness of the monitor for displaying the online game. Users can adjust these settings directly with the hardware without informing the application systems.

As aforementioned, these two types of controls are application-dependent inputs, which are under users' control but not under the system's control. Therefore, they are beyond the study of this thesis.

On the other hand, *system controls* are inner controlling parameters controlled by the application itself which can affect the way the system runs. These controls can be adjusted dynamically by the system at run time according to the running conditions (context, application requirements and network condition), therefore, are important to study for fast-paced interactive multimedia systems. We use the term system controls and system control inputs interchangeably in the later sections of this thesis to refer to this type of controls.

In Figure 1.5 we illustrate how an interactive system can be controlled by its system control, the end-to-end delay (EED). EED defines for how long an action from the local user is perceived by a remote user. This system control can affect the over all latency of the interaction, say a 100 ms EED can make the overall duration of the interaction longer than a 0 ms EED.

An example of the system control is the mouth-to-ear delay (MED) which controls the overall latency of a talk spurt from the mouth of a local user to the ear of a remote user. When the MED is higher than 400 ms, users are difficult to finish a talk without disturbing others during the speech.

Another example of the system control is the response time of the key press in a fighting game. A 200 ms response time can significantly slow the pace of the fighting in the game than a 0 ms response time.

Tuning these system controls will significantly affect the performance of the system. Therefore, in this thesis, we focus on how the tuning can be done to achieve the best QoE.

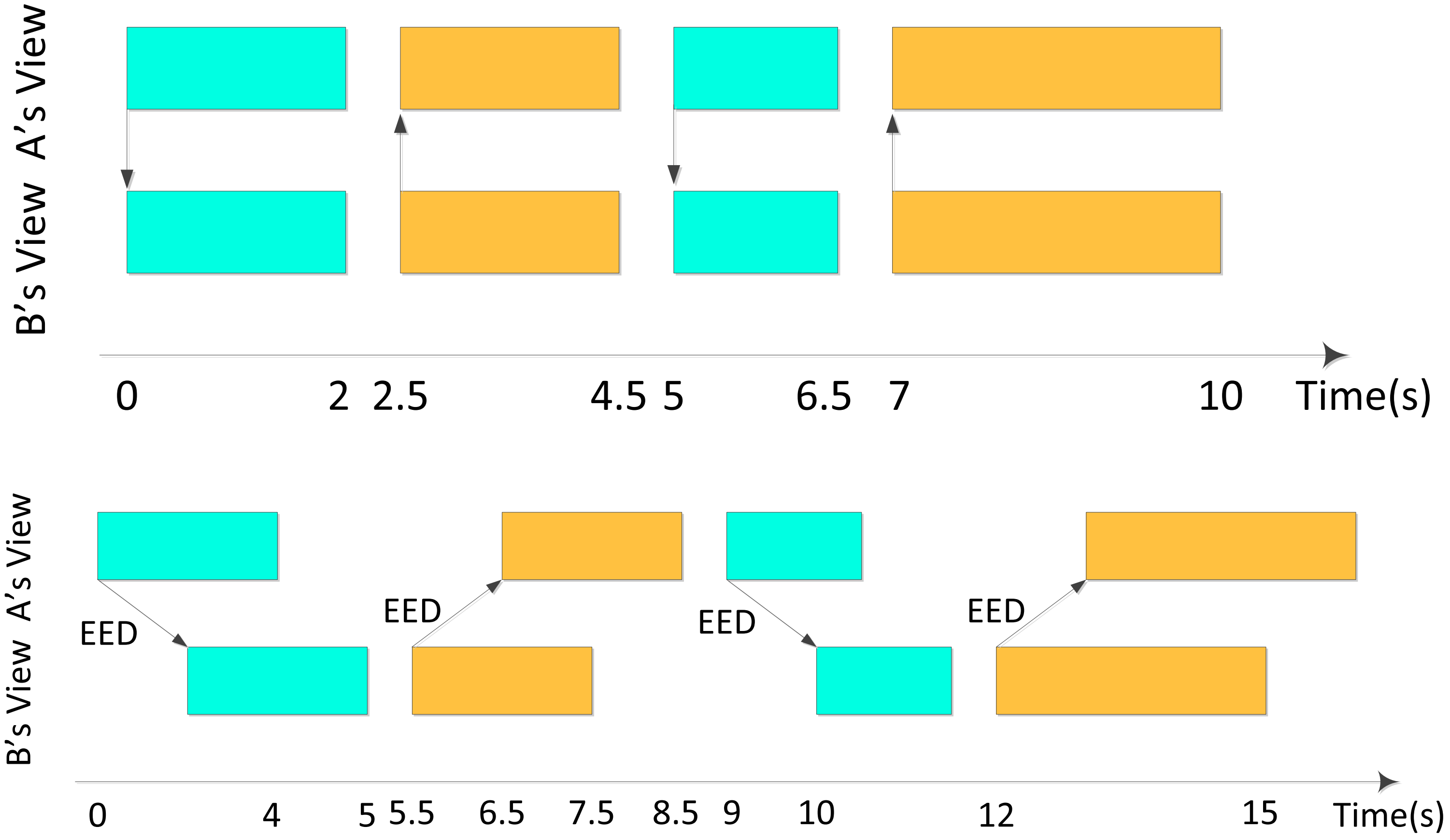


Figure 1.5: The interaction scenario in an interactive system when the EED is 0 ms and 1000 ms respectively. We use the 1000 ms delay emphasizing the effect of the latency to the conversation. Each bar in the figure marks the start and end of an action, and the arrow shows the direction of the transmission of the action. It is noticed that a long EED can significantly extend the overall duration of the interaction.

1.1.4 Application-Level Requirements

The application needs a stable network condition to run under. They have certain requirements on the network condition, which can be defined by a set of constraints of the system controls. For example, in a VoIP or a videoconferencing, the overall MED cannot be longer than 400 ms. Otherwise, the conversation can be confusing. In online games, it is directly required that the response latency should be less than 200 ms. Further, for the correct transmission of the game events, the loss rate should be less than 5%. These requirements on the network condition define a stable network condition in which the network control layer should provide to the application layer by using certain network controlling strategies.

Definition 1.1.1.

Perceived Network Condition is the network condition perceived by the users after the loss concealment strategy and jitter removal strategy have been performed in the network-control layer (these strategies are introduced in detail in Chapter 3).

While system controls can be of various types, they can be converted to constraints with common network metrics throughout any network applications. These network metrics include:

- *End-to-end delay (EED)*. It is the overall latency from the start of an event in the application layer of the local machine to the presentation of it in the application layer of the remote machine.

- *Correctly-received packet ratio (CPR)*. It is the percentage of packets received correctly in the order sent within EED.

- *Transmission bandwidth*. It measures the bandwidth required for transmit the data of the application.

We assume only the above network metrics are involved in the constraints. This assumption is justified by the fact that the latency, loss rate and bandwidth are common metrics in the Internet.

As examples, the MED in VoIP and the response time in online games and remote robotic surgery can be directly mapped to EED. The 5% loss rate in online games can be mapped to 95% CPR.

Even though the mapping can be simple, the translation from the network-layer condition (i.e. real Internet condition) to the required application-level user-perceived network condition is, however, non-trivial, because certain constraints would be determined under the run-time real Internet-layer conditions, which are non-stationary and involve multiple network metrics that can change frequently.

When the above system controls changes, we need some metrics to measure the effect of this change. For this purpose we utilize some *quantitative quality metrics*, which are measurements of the direct effect of the change caused by one or more system control(s).

Some quantitative quality metrics are so tightly connected to the system controls that they are system controls themselves. For example, in a VoIP application, the MED can be both a system control and a quantitative quality metric for measuring the end-to-end delay (EED), and therefore the interactivity of the system. In an online game and the remote robotic surgery, the response time can also be a quantitative quality metric of the delay effect.

With quantitative quality metrics, we can have a general impression of the performance of the system. However, they cannot precisely measure the QoE of the system. The reason is discussed in Section 1.2.

1.1.5 Network

Internet has non-stationary network behaviors, and the loss rate, delay, delay jitters (variance of delay), and bandwidth are difficult to predict in long term. However, to satisfy the application-level network requirement, the application should run under a relatively stable network condition. Fortunately, short term network behavior is relatively stationary, and we can predict them with network statistics collected in the past few sections, and then improve the network condition with a network-control layer.

With the constraints on the network metrics that are passed from the application layer, a network-control layer can employ a rate control strategy to maintain a relatively stable network behavior that can affect loss rate, delay, delay jitters, and bandwidth. We can also employ a buffering strategy to perform loss concealment and jitter removal under a given bandwidth, which can affect loss rate, delay, and delay jitter.

For example, in VoIP and online games, we can use a delay buffer to reduce the jitter in the Internet and provide a stable EED. We can also use forward-error correction (FEC) strategy to conceal packet losses in the transmission and to reduce the loss rate, in order to provide a high CPR. With the network-control layer, the application layer can focus on the scheduling and presentation of the multimedia contents, without looking into the low-level network environment.

As aforementioned, the network-control layer should satisfy the constraints of the network metrics which are required by the application layer under any run-time network condition, but long-term network behavior in the Internet is difficult to predict. This can lead to the following three problems in the implementation:

1. Short-term behavior will require additional overhead to collect real-time network information. Such information may be outdated or may not arrive in time. The collection also introduces additional overhead in the network.

2. Controls in the network-control layer based on short-term network behavior should adapt rapidly. However, the time is not sufficient to allow humans to be in the loop to make adjustments to the control.

3. The transmission rate of the application should adapt to the available bandwidth in the end-to-end connection in order to maintain a relatively stable network behavior. However, the available bandwidth changes from time to time and is non-stationary.

We focus on the solutions to these problems in Chapter 3.

Below the network control layer are other layers that connect to the Internet. These layers include the original transport layer which uses TCP and UDP protocol to transmit the packets of the multimedia application system. Because fast-paced multimedia systems have tight deadline for the presentation of the content, UDP is generally adopted to transmit media data and interaction data. TCP is used to transmit system control data that have a looser deadline and higher requirement of reliability.

Beside the TCP/UDP, other layers also include the Internet layer and the link layer. They are low-level network infrastructure, therefore, are beyond our study in this thesis.

In summary, the bottom layers in the Internet protocol suite provide the basic network condition for the multimedia system. It is non-stationary and changes from time to time. Therefore, traffic over this layer needs to be processed by the network control layer before used by the application layer.

1.2 Offline Subjective Test for Perceptual Quality

Even though a multimedia system can have multiple internal quantitative quality metrics, users only perceive the input and output at the interface. No matter how good the internal quantitative quality metrics are, users are not concerned about them. This is why the quality of how users operate the system at the interface is important. This quality is defined as follows.

Definition 1.2.1. *Perceptual quality is the overall quality perceived by a user at the interface of the system under the perceived network condition and the user inputs, user controls and multimedia requirements [77].*

It is also called quality of experience (QoE) in related works [37, 106], but perceptual quality focuses more on the evaluations by subjective tests while QoE concentrates on quantitative metrics.

As an example, in VoIP, the perceptual quality is the overall conversational experience with the system using the microphone as input and the speaker as output. The performance of the system can be affected by the network condition. For example, a poor Internet connection with losses and delays can make the conversation difficult to understand. Then we consider the perceptual quality of the VoIP poor. Similarly, in an online shooting game, the perceptual quality is the overall experience of the players who control the game using their keyboards and mice and see the output from the monitor. A poor network connection can worsen the playing experience because the target can be hard to shoot at.

The function of perceptual quality is unknown, because the list of inputs are unknown. Whether the quantitative quality metrics in hand can represent all the inputs is unclear. Even when the inputs are known, the function is undefined. We do not know how we can correctly pool the quantitative quality metrics and other inputs into an overall perceptual quality metric, because we lack a closed-form formula for integrating them together.

For example, in a VoIP application, the system controlled MED can affect both interactivity and signal quality, because a long MED can significantly extend the duration for finishing a conversation but allow more space for the buffer to smooth the late packets and improve the voice quality. However, we do not know whether the interactivity and signal quality can represent all the inputs of perceptual quality in VoIP. Further, even if they are all the inputs of the function of perceptual quality, we are not sure whether a short or long MED can result in better perceptual quality, because the function is undefined. Therefore, we should conduct a subjective test with respect to this system control and ask subjects to assess the output perceptual quality resulted by different MEDs. The case is similar in online games, where a long response time can worsen the control precise-ness but can improve the correctness of the transmission of the game events.

However, subjective tests are time-consuming and need to be conducted offline, where two methods are generally adopted for multimedia perceptual quality [6].

- *Absolute assessment*: This method ask subjects to directly grade the perceptual quality with a quantitative score given a single control scenario. These methods are intuitive, easy-to-conduct, and require less number of tests. However, a high level of expertise is required for this assessment because they only can hold a consistent standard and notice details that can lead to different quality. Furthermore, the same score does not mean the same perceptual quality. For example, two extremely poor perceptual quality can have the same low score, but the reason for the poor quality can be due to a long delay or poor signal quality. This can make the optimization based on the score difficult, because the same score can lead to totally different direction of the optimization.

- *Pairwise assessment*: This method instead ask subjects to compare two control scenarios and judge which has the better perceptual quality. It is easy to conduct even for non-experts, because they can compare details (see Figure 1.6). However, this method is expensive because it requires numerous comparisons.

Even though subjective tests can provide precise measurement results, they have several difficulties, the detailed survey of which will be presented in Chapter 2:

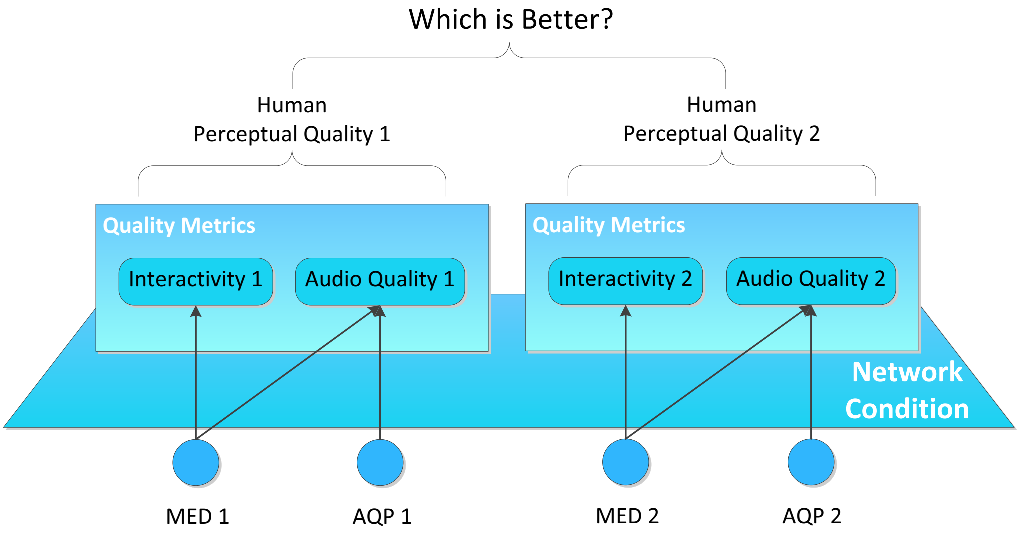


Figure 1.6: Pairwise comparison of two pairs of MED and AQP settings. Subjects are asked to determine which pair has better perceptual quality.

1. Although some quantitative quality metrics have certain relationship with respect to perceptual quality, the exact function is not well-defined or in a closed form, especially when trade-offs are involved.

2. Multiple quality metrics are involved in the system, but whether they are complete is unknown due to a lack of thorough understanding of human perception.

3. Subjective tests are expensive and are conducted offline. They cannot be done in real time operations. Further, each test only associates one control combination and the resulting quality. The total possible combinations on control and operating conditions are prohibitive.

In this thesis, we propose methods for addressing these difficulties. We study a method for measuring the perceptual quality regarding to the controls directly to capture all the trade-offs involved as well as all the quality factors that can be represented by quantitative quality metrics or not. We further propose a method for reducing the cost of subjective tests by reducing the number of tests but maintaining the precision. Further discussions are presented in Chapter 4 and Chapter 5.

1.3 Generalization

While offline subjective tests can collect precise assessment of the perceptual quality of a system, they cannot be performed online because the time they cost are beyond the delay threshold of real-time application. More specifically, real-time fast-paced interactive multimedia applications cannot involve run-time subjective tests for measuring perceptual quality. On the other hand, offline subjective tests cannot be unlimited. For instance, the network conditions in which we perform the subjective tests are a limited set and can only consist of some representative scenarios. All these problems make it difficult to optimize the perceptual quality at run time, which is the ultimate goal of our study.

The metric of perceptual quality is not well-defined because human perception on the combined effects of quantitative quality metrics is complex. Further, mapping from system controls to perceptual quality under the perceived network condition is unclear and may not have a closed-form. Moreover, network controls should change in response to the fast changes of the real network condition. As a result, the optimization with all the above factors can take longer time than the requirement of the real-time interactive application. For example, in a VoIP system, we need to know how we can tune the MED at run time under different network conditions to attain the best perceptual quality, according to our subjective test results conducted in offline tests. In online games, we need to carefully tune the response time of the game in order to achieve the best perceptual quality no matter how the network condition changes. The tuning is performed online but the guidance of it is based on offline subject tests.

For this reason, we need a method to generalize the offline subjective test results conducted in limited network conditions to online application that can run under any network condition. We adapt the formal definition of *generalization* in previous works [59, 77] as follows:

Definition 1.3.1. Generalization *is the ability to adapt to previously unseen situations using the data that have been collected.*

Generalization should be sufficiently fast to meet the requirement of real-time multimedia applications. We need to learn the mapping from application-level system controls to perceptual quality with the help of offline subjective tests, and then generalize the result to any online condition. The generalization should be sufficiently fast to allow run-time optimization

Generalization is difficult because offline subjective tests can only cover a small subset of network conditions due to the high cost of subjective tests, which cannot easily be applied to all online network conditions. Further, we should decide how the offline results are used according to run-time conditions, but this decision and the corresponding computation should be finished before the next real-time multimedia event, such as the display of a new video frame. The available duration for the computation is too short such that complex computation cannot be performed. We look into this difficulty in Chapter 5, and study a real-time algorithm for tackling this task.

1.4 Research Problem

With all the considerations above, we present the goal of this thesis as follows.

Overall Goal:

- In multimedia systems, we like to find the best combination of controls over the space of system controls in order to attain the best perceptual quality.
- This should be done in real time without incurring undue overhead in real time operations.

This goal suggests that our focus in optimization is the perceptual quality, and we need to bridge the gap between the offline measurement of it and real-time adjustment of the system controls.

To achieve these goals, we have the following difficulties.

First, to meet the first goal, we need a connection between the system controls and the perceptual quality. However, mapping between system controls represented in the form of quantitative quality metrics and the final user perception of the quality of the system is complex. The mapping cannot be easily determined by simple derivation, but relies on offline subjective tests. It is impossible to enumerate all the combinations of controls, because the space of controls and the combinations of run-time conditions is too large to be enumerated, especially when the dimension of control space is high. Machine learning algorithms are not useful for learning this mapping, because we do not have sufficient samples collected from subjective tests due to the high cost of each test. Furthermore, offline subjective tests can only cover a small subset of simulated network conditions. It is difficult to decide which network conditions are included in this subset.

To meet the second goal, we mainly face the difficulties for generalization. First, in a real-time system it is impossible to have online subjective tests at run time because of the long testing period. Second, it is difficult to reproduce run-time real network conditions in offline subjective tests with a perceived network condition, because run-time network behavior does not have well-defined simulation models.

Regarding these difficulties, we need to answer the following two problems.

Problem Statement:

1. **Offline**. How to find a complete mapping from the space of controls to perceptual quality with a reasonable number of offline subjective tests under a small subset of simulated network conditions?

2. **Online**. How to generalize the offline results with a small subset of simulated network conditions to online applications in a multimedia system under all possible real-time network condi-

tions and to determine the control to be used to achieve the best perceptual quality in real time?

Here, the first problem is related to the first point of the goal, and the second problem is related to the second point.

1.5 Approaches

We summarize the approaches for solving the problems and address the difficulties stated in last section.

We depict an overview of our detailed approach in Figure 1.7. For each layer of the application system, we propose the corresponding approaches for solving the two problems, which are divided into the offline learning of the mappings and the online operation of the system. In the stage of offline learning of the mappings from controls to perceptual quality, we schedule offline subjective tests with efficient method for performing subjective tests, then guide the application layer to run with the set of system controls determined by the subjective test scheduler. The application is running under a simulated network condition provided by the network-control layer. The result of the subjective tests are accumulated and stored in a compact structure that is then employed in the stage of online operation of the system. In the online stage, the system runs with the system controls determined at run time according to the network condition passed by the Internet layer and the system context. The system controls guide the application layer to schedule the multimedia contents and events, as well as tell the network-control layer to achieve the desired stable perceived network condition.

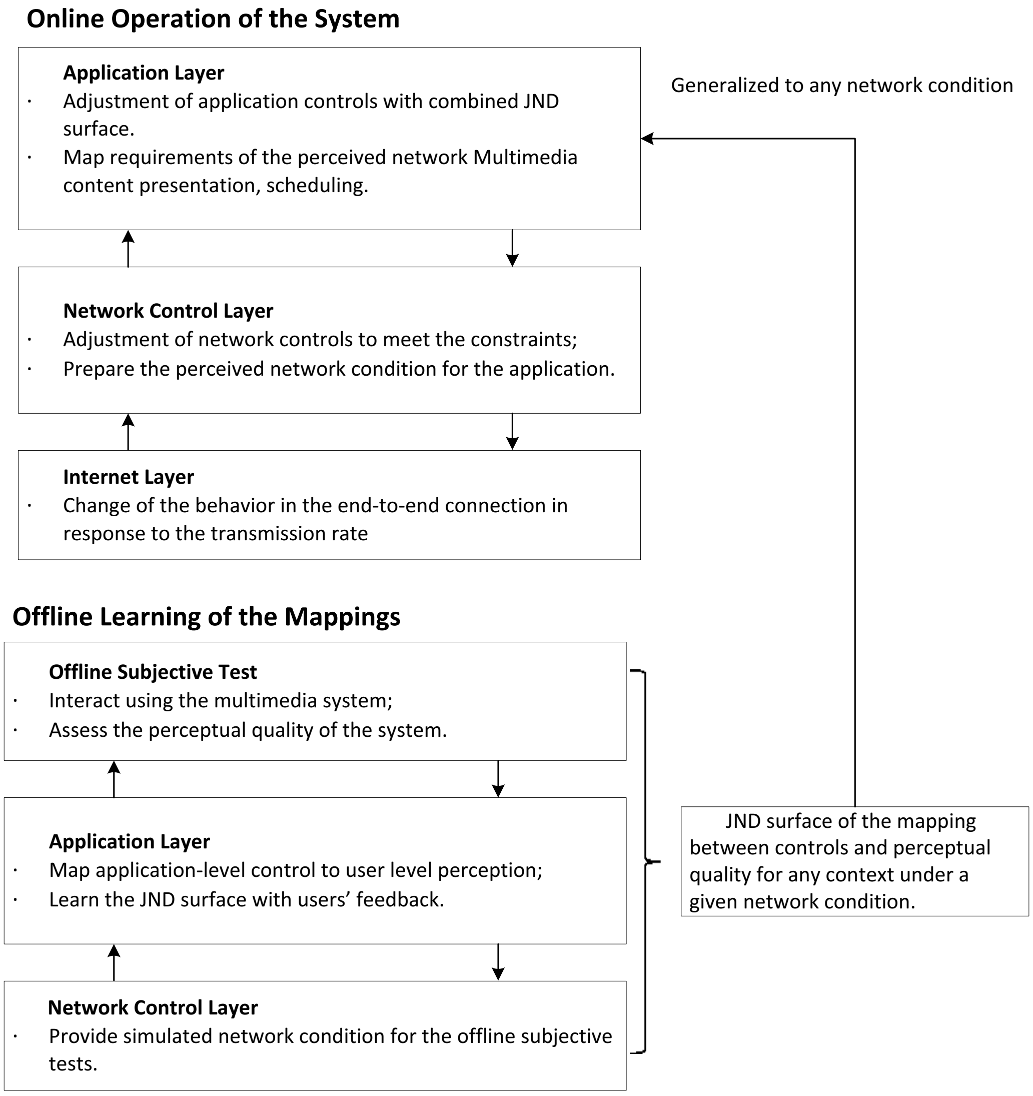

Online Operation of the System
Application Layer
Adjustment of application controls with combined JND surface.
Map requirements of the perceived network Multimedia content presentation, scheduling.
Generalized to any network condition
Network Control Layer
Adjustment of network controls to meet the constraints;
Prepare the perceived network condition for the application.
Internet Layer
Change of the behavior in the end-to-end connection in response to the transmission rate
Offline Learning of the Mappings
Offline Subjective Test
Interact using the multimedia system;
Assess the perceptual quality of the system.
Application Layer
Map application-level control to user level perception;
Learn the JND surface with users' feedback.
JND surface of the mapping between controls and perceptual quality for any context under a given network condition.
Network Control Layer
Provide simulated network condition for the offline subjective tests.

Figure 1.7: Overview of the approaches.

To have an overview of the approach proposed in this thesis, we list in Table 1.1 all the detailed tasks, as well as the sub-goals we achieve in each chapter using these approaches. As a demonstration of the advantage of the proposed approaches, we further evaluate our approaches in real systems, including a videoconferencing system in Chapter 6 and a fast-paced online game in Chapter 7.

Table 1.1: Summary of the approaches.

Chapter (Layer)	Goal	Approaches	
3 (Network Control Layer)	**Offline:** Provide the simulated network condition. **Online:** Find out the feasible network controls that can satisfy the requirement of the perceived network conditions.	1. Employ rate control strategies to ensure a relatively stable real network condition for the interactive multimedia system. 2. Employ loss concealment and jitter removal strategies to reduce loss rate and delay jitter of the real network condition to the required level.	
4 (User Layer and Application Layer)	**Offline:** Attain the mapping from control to perceptual quality in offline subjective test. **Online:** Adjust the application with the mapping from controls to perceptual quality.	Single Control	Multiple Controls
		1. Use the dominant property to measure the mapping from the control to perceptual quality; 2. Refine the result regarding the binomial error in the subjective test.	1. Decompose the problem into single -control problem; 2. Perform subjective test for each control using the single control algorithm that can handle binomial error; 3. Combine the mapping of individual control to that of multiple controls.
5 (Application Layer)	**Online:** Generalization in run-time real network conditions.	1. Find out the optimal controls of the application by combining the individual mappings from single control to perceptual quality together, and then search for the optimal operation point at run time. 2. Map the application-level requirement given by the optimization to constraints of the network metrics, and pass these constraints to the network-control layer.	

Our key innovations are as follows.

The first innovation is related to the offline subjective tests for discovering the mapping from controls to perceptual quality. We have found a new dominance property that allows the result of one subjective test to subsume the results of many other subjective tests. This means that those subjective tests whose results are subsumed do not have to be tested because their results would always be no better than the result of the dominating subjective test. This property will tremendously reduce the number of subjective tests used for finding the complete mapping from controls to perceptual quality. Because the subsumed cases need not be tested, the number of simulated network conditions can be strictly controlled to a reasonable number, while all run-time conditions have been represented without any loss of generalizability. These approaches will appear in Chapter 4.

The second innovation corresponds to the online operation of the system using the generalized results from offline subjective tests. With the dominance property, the complete mapping from controls to perceptual quality can be represented efficiently by a compact set of dominance relations. This compact representation allows the results obtained offline to be looked up efficiently at run time and be generalized to all run-time conditions. Because the offline combinations of controls are represented by a compact and small set of dominance relations, online search for the optimal combination of run-time controls can be done in real time. With the guidance of the generalized results from subjective tests, the required perceived network condition given by the optimization is then mapped to the network metrics and passed to the network-control layer. Loss concealment, jitter removal and rate control strategies are employed to transform the real Internet condition to the user-level network condition. These approaches appear in Chapter 3 and 5.

1.6 Contributions

In this thesis, we focus on optimizing the perceptual quality in real-time interactive multimedia systems, and study methods for generalizing offline subjective test results to online controls of the system. By achieving this, we have the following major contributions in the thesis.

For offline subjective tests, we discover a dominance property that can significantly reduce the number of subjective tests to be conducted offline. By utilizing this property, we propose to represent the mappings from controls to perceptual quality by a JND surface, a data structure that can store the results of subjective tests in a compact way. We further develop a systematic methodology for generating this JND surface using a small number of subjective tests.

For the online operation of the system, the dominance property provides a compact representation of the space of all mappings from controls to perceptual quality, therefore, allowing the online search of mappings from controls to perceptual quality with real-time performance. With the dominance property and the JND surface, we propose an online algorithm for finding a suitable combination of controls for attaining good perceptual quality. We further propose a general network-control layer for fast-paced interactive multimedia systems which can provide an improved network condition for the stable running of these systems.

Our contributions can benefit the area of online interactive multimedia, which includes but not limited to online conversation, online games, and remote virtual reality interaction. With the proposed general approaches, numerous applications can have improved runtime perceptual quality in the error-prone Internet. Because our online generalization algorithm can complete within 5ms, which is much shorter than a video frame interval (16.7 ms), they can be adopted by various real-time multimedia applications without significant overhead. Furthermore, developers can also adopt our approaches to assess the expected user experience in the stage of development. Internet Service Providers (ISP) can perform provision for the network resource for users of online interactive multimedia with our approaches. These use cases show that the thesis can have considerable industrial value.

We summarize the problems we have and have not solved in Table 1.2.

Besides the major contributions, we also have innovations for improving multimedia systems, including:

1. We have evaluated the results in two real applications, the VoIP and the online multi-player shooting game BZFlag.

2. We have proposed strategies for improving the perceptual quality of VoIP and videoconferencing, using a prototype system built in our laboratory as well as some proprietary VoIP systems.

3. We have proposed strategies for maintaining consistency of fast-paced online games in such a way that the ordering of

Table 1.2: Summary of problems solved and not solved in this thesis.

Problem	Assumption	Major Results	Limitation	Section
How to find complete mapping from the space of control to perceptual quality with reasonable number of offline subjective tests?	1. Unbiased and synchronized tests. 2. Uniform level of expertise. 3. Awareness is continuous but not necessary smooth.	1. Dominance properties. 2. JND Surface model. 3. Efficient algorithm for performing subjective tests with low error. 4. Method for decomposing high-dimensional subjective opinions into individual surfaces.	Not suitable for multiple independent controls.	Chapter 4
How to generalize the offline results with a small subset of simulated network condition to online application in a multimedia system under all possible real-time network conditions?	1. Network one-way latency is lower than 400ms. 2. Network loss rate is lower than 5%. 3. Bandwidth is higher than the lowest bitrate of the video/audio codec.	1. A general network layer for maintaining a stable network condition. 2. Run-time transformation of JND surface to adapt the online network condition.	Not suitable for extremely poor network condition.	Chapter 3, Chapter 5.2
How to determine the control to be used to achieve the best perceptual quality in real time?	1. Dependent multiple controls. or 2. Single control with independent quantitative metrics.	1. An algorithm for finding the optimal control set in real time for multiple dependent controls. 2. An algorithm for finding the optimal control for a single control corresponding to multiple quantitative metrics.	Not suitable for multiple independent controls.	Chapter 4.5, Chapter 5.3

events in such games operating under network delays is the same as those operating without network delays.

1.7 Outline of the Thesis

In Chapter 2 we discuss related works in the areas studied. In Chapter 3 we propose a general network-control layer for fast-paced multimedia systems. In Chapter 4, we study the offline measurement of the mapping from controls to perceptual quality. We generalize it to online operation in Chapter 5. In Chapter 6 and 7 we evaluate the proposed methods in VoIP and online game respectively. In Chapter 8 we conclude the thesis and discuss the possible future work.

□ **End of chapter.**

Chapter 2

Background

In this chapter we discuss the necessary background for studying the problems in this thesis. The user experience of the interactions in online fast-paced interactive multimedia systems can be affected by the latency for transmitting interaction events. The experience can be improved by play-out scheduling algorithms, but there is a lack of run-time context-aware optimization for the scheduling. While the objective function of the optimization is the perceptual quality, it is not a well-model metric. Previous works tried to approximate it with the objective quantitative metrics, but these metrics either require time-consuming computation or only serve as a coarse model of the perceptual quality. Actually, a precise measurement of the perceptual quality involves offline subjective tests. How we can use the offline data to access the online perceptual quality in fast-paced interactive multimedia is the major concern in related works.

2.1 Interaction

Interaction is the process of multiple users performing a task with the application. It is the most important factor of online interactive multimedia systems. For this reason, the application layer is responsible for fine-tuning the perceptual quality on interaction.

In both VoIP and online games, we are interested in those scenarios which have fast-paced interactions; specifically when users' average response time to events in the system are no longer than 1000 ms. In these scenarios, there is insufficient time for humans to tune the system controls among the interactions, which makes the online generalization of offline subjective tests necessary.

The deficiencies in previous works about optimizing the perceptual quality on interaction are two-folded. On one hand, they lack a method for run-time perceptual quality based optimization of the presentation and scheduling of the multimedia contents. On the other hand, they do not consider the running context in the optimization. We first present basic definitions of the perceptual quality on interaction, then discuss previous works on optimizing this perceptual quality.

For most online applications, network latency can significantly affect the perceptual quality on interaction in fast-paced interactive applications. We have previously demonstrated how a 1000 ms EED for buffering packets transmitted in a connection with delays can affect the structure of a conversation in Figure 1.5.

We further demonstrate the undesirable effects in a VoIP. These include a longer overall conversation duration, a longer interval between talk segments, and asymmetric intervals [31, 56, 75, 76, 79]

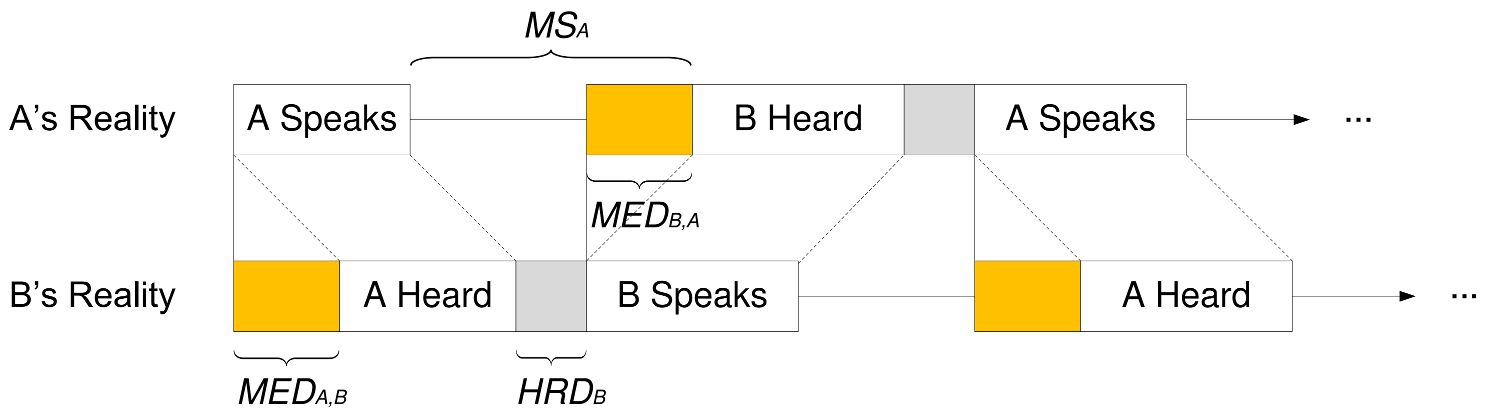


Figure 2.1: In video conferencing over the Internet, the delays introduced by network propagation and jitter buffers cause users at both sides to have different reality of the conversation. They need more time in waiting for the other party to respond [76].

(see Figure 2.1).

In contrast to audio streaming systems, interactivity is an important quality metric in VoIP and videoconferencing. Interactivity depends on delay, which changes the way the two parties relate to each other in a conversation. When delay in a session is long, each party would find more time waiting for the other to react and their perceptual experience would be degraded. Three metrics can measure interactivity: MED, conversational symmetry (CS) and conversational efficiency (CE) [31, 41, 76].

MED measures the delay before a speech segment is heard by the other side. It consists of the coding time, queuing time, network propagation latency, and playout delay in jitter buffers.

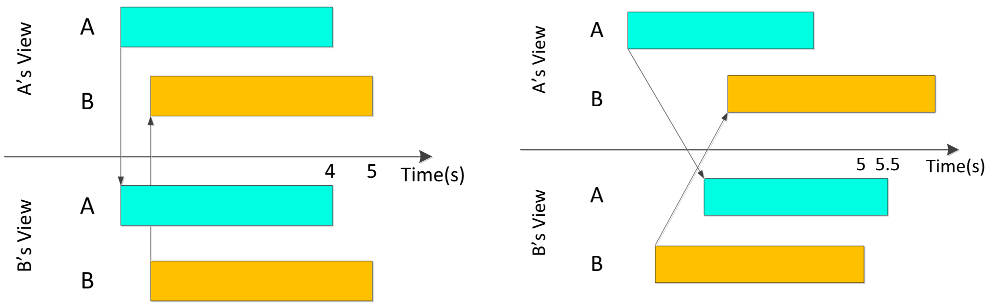
$$MED = t_{\text{encode}} + t_{\text{queue}} + t_{\text{propagation}} + t_{\text{decode}} + t_{\text{playout}}.$$

MED has a direct impact on the interactivity of a conversation. Figure 2.1 illustrates how MED changes the conversational scenario. MED changes the conversational structure as follows.

First, $MED_{A,B}$ will delay *A*'s speech in *B*'s reality. After listening to *A*'s speech, *B* will need a short human response delay HRD_B before replying to *A*. Because of $MED_{B,A}$, *B*'s speech will also be heard later in *A*'s reality. As a result, *A* will perceive that *B* is thinking unnaturally long, with a silence period of MS_A:

$$MS_A = MED_{A,B} + HRD_B + MED_{B,A}.$$

Similarly, with $MED_{B,A}$ and $MED_{A,B}$, *A* will discover the asymmetry in the response times; that is, *B*'s thinking time MS_B is longer than *A*'s thinking time HRD_A in *A*'s reality. We measure this asymmetry by CS, which is the ratio the longest MS over the shortest MS



(a) Reference Order

(b) Order with network Latencies

Figure 2.2: Physical network latencies can delay the completion of actions and cause the reordering of completions that are different when compared to the reference order.

in a user's reality:

$$CS = \frac{\max\ MS}{\min\ MS}.$$

In addition, both parties will find the conversation taking more time than a face-to-face conversation, whose effect can be measured by CE, a metric measuring the ratio of the time in a face-to-face conversation over the time in a video conferencing conversation:

$$CE = \frac{\text{speaking time} + \text{listening time} + \text{thinking time}}{\text{total time for a conversation}}.$$

All these quantitative quality metrics can be used to measure the interactivity in VoIP applications.

Similarly, in an online game, network latency can also lead to undesirable effects inside the game, which can lead to inconsistency in the game. Figure 2.2 illustrates how network latency can inverse the order of a shooting. In the figure, a bar represents the start and end of an attack action, for example, a bullet flying from the gun to the target. When there is no latency, player *A* will shoot the target earlier than *B*. However, with latency, *B* will find himself shooting the target earlier than *A*, even though *A* still finds her shot earlier. This is called the reordering problem and will be further discussed in Chapter 7.

Previous work has proposed methods for improving the perceptual quality on interaction. For VoIP and videoconfernecing, playout-scheduling algorithms have been proposed to arrange a conversation so that the conversation is less affected by the latency [10, 75, 76, 78, 79, 105]. Similarly, in online games, dead-reckoning algorithms [2,21,67,83,118], smooth correction algorithm [80], local lag algorithms [58,81,88,118], and local perception filters [82,85] have been proposed to conceal inconsistencies. All these algorithms are controlled by their system controls to adjust the behavior of scheduling at run time.

However, previous work cannot find the optimal system control(s) at run time when multiple quantitative quality metrics are involved. For example, in VoIP, interactivity can be degraded when MED is longer, but the signal quality will be improved. In online games, several delay-concealment algorithms can work together but are constrained by the overall EED; and how the time-slot is allocated to each algorithm involves trade-offs and needs to be guided by subjective tests. There is a lack of online optimization algorithm for settling these problems.

Furthermore, the setting of the system controls highly depends on the running context. For example, a fast conversation can be more sensitive to network latency because a longer interval between the talk segments can be easily noticed; a fast-paced online game can have more inconsistency than a slow-paced online game because in the latter case the actions have less probability to overlap. Therefore, we should also consider the run-time context of the application during the online optimization of perceptual quality on interaction.

In this thesis, our goal is to address the deficiencies of previous works. We propose online scheduling algorithms for online fast-paced interactive multimedia systems which consider both the network condition and the running context. Details are presented in Chapter 4 and 5.

2.2 Quantitative Metrics

Besides interaction, people are also interested in the qualities of audio and video in interactive multimedia applications. To measure these qualities of multimedia applications, previous work has studied using quantitative metrics to model the quality and has proposed algorithms for measuring it.

Unfortunately, these metrics rely on strong assumptions to work. They assume that there exists a perceptual model derived from domain knowledge [5, 22, 95], which is often difficult or expensive to attain due to the complexity of human perception. They also assume that the model can be represented in closed-form as a function of features of distortion [45, 97, 100]; however the inputs of the function is often incomplete.

A significant deficiency of these methods is that most of them only evaluate the output quality of the media content, while not considering the network condition during transmission and the system controls during run time. Without these considerations, there is a lack of the mapping from system controls to the final perceptual quality under run-time network condition. As a result, the tuning of the system controls to achieve better perceptual quality is difficult.

We review three categories of the quantitative metrics in the following sections.

Full-reference. Full-referenced quantitative metrics provide complete perceptual models that can measure the perceptual quality in any scenario of video or audio.

VQM [69] is a standardized method for measuring video quality. It first aligns the video spatially and temporally with the original video (called the *reference*), then calculate the loss of spatial information, the extent of the edge artifacts, the change of the distribution of the colors, the extent of the sharpened edge, the spatial-temporal interaction, and localized color impairments. After the calculation, it will poll these sub-metrics into an overall score. The score is normalized to 0 to 1 (with some extra cases larger than 1), the greater the better.

VSSIM [97], on the other hand, uses structural distortion as an estimate of perceived visual distortion for measuring video quality. The author extended the SSIM metric [96] for image quality to measure the video quality by considering the extent of the motion in the video frame. The algorithm also needs the reference video as it compares the distortion frame-by-frame.

For measuring audio quality, PESQ [6] is a widely used standardized algorithm. Similarly, the algorithm first align an audio clip with the reference audio, and then calculate the score regarding the distortions, errors, noise, and delay in the clip.

PEMO-Q [43] is a metric specific for measuring voice quality. It utilizes “internal representations” of the voice signal to improve the precision of the assessment. Still, it needs the high-quality reference audio for comparison.

In summary, these metrics require to compare the full recording of the video or voice content played at the receiver’s machine with the original content (i.e. the reference) at the sender. Therefore, they need an auxiliary channel for transmitting the output in the receiver back to the sender in order to calculate the quality. This can congest the network and increase round trip network delay. Further, these metrics do not consider the impact of the network condition to the quality of the media content; therefore, we cannot estimate how the network condition can affect the quality when tuning network controls.

Reduced-reference. Reduced-reference objective qualitative metrics transmit partial features of the output back to the sender for the comparison with the reference. The features are small, and can reduce the overhead in transmission.

Representative works of reduced-reference objective qualitative metrics include PSQM proposed by Lu et al. [57] for measuring visual distortions with features with which human perceive the video. This metric pay attention to certain area of visual signal due to the following factors: salient features in image/video, cues from domain knowledge, and association of other media. In this way, it can rely on partial part of the reference video in the assessment.

Anther partial metric was proposed by Winkler in his study [102]. This metric utilized many aspects of the visual system, including color perception, the multi-channel representation of the temporal and spatial mechanisms, contrast sensitivity, and the response properties of neurons in primary visual cortex. Similarly, it can use partial clip to assess the quality.

No-reference. To further reduce the overhead, no-reference objective qualitative metrics have been proposed. These methods do not require any reference for comparison; therefore, only the score of the quality is conveyed back to the sender, which significantly saves the transmission bandwidth.

To fulfill the goal of no reference, Keimel et al. [50] have extracted features directly from the NAL units in the H.264/AVC bitstream. From each slice in the NAL unit, it attains the following features: bits, average QP, motion vector length, and motion vector error. They then used partial least squares regression to estimate the quality with features without the reference.

Lu et al. [57] and Winkler et al. [102] also modify their methods to work without the reference, with a reduction in the precision of the estimation. This is done by removing the features which can only be gained from the reference video. By only using the blurriness and blockness features, the methods can estimate the quality using merely the distorted video clip.

Deficiencies. Even though the reduced-reference and no-reference objective qualitative metrics have reduced the overhead in the transmission of the recorded media, there are still deficiencies that need to be addressed. First, since the inputs of the model are features extracted from the output, there is a lack of mapping from control inputs to perceptual quality, which is unsuitable for guiding the run-time adjustment of system controls. Second, without considering the network condition, the metric cannot adapt to non-stationary and fast changes of the network environment. Finally, domain knowledge from experts are difficult to obtain. Such knowledge is very application-specific and is not generally available for a multimedia system.

With these considerations, in this thesis we do not adopt objective qualitative metrics to evaluate run-time perceptual quality. Rather, we directly discover the mapping from system controls to perceptual quality by offline subjective tests and then generalize the result to online network conditions. In this way, we can guide the run-time optimization of online multimedia systems with both precise estimate of the resulting perceptual quality and small overhead.

2.3 Subjective Measurement of Perceptual Quality

Subjective tests, although expensive to conduct, provide the most precise way to evaluate perceptual quality. Related works of subjective tests generally have two assumptions. First, when multiple subjects are asked to evaluate a system, their responses to the test are independent and identically distributed (IID). Subjects have equal expertise in evaluating the system. These assumptions allow researchers to use statistical tools to analyze the test results. Second, sufficient time for each test is necessary, and the tests are conducted offline in most cases. Test time can be on the order of minutes or longer according to the type and content of the subjective tests.

2.3.1 Types of Subjective Test

Subjective tests can be based on either absolute or pairwise assessments. We have briefly introduced them in Section 1.2. In this section we describe the details of these methods, which are necessary to understand why pairwise assessments are adopted and how they will relate to just-noticeable difference in the next section.

In absolute assessments, a standard method is to evaluate a mean opinion score (MOS), which is obtained by asking subjects to report their opinion using an absolute category rating and by taking an algebraic mean of their opinion when evaluating the same output [71, 87]. Using absolute assessments imposes a total order on perceptual quality, which may differ from reality. For instance, two scenarios in VoIP under different MEDs may lead to different interactivity and ASQ, but neither may be perceptually better than the other in subjective tests.

In contrast, in pairwise assessments, subjects are asked to do pairwise comparisons of two observed outputs under different control input settings and to choose the setting leading to the output with better perceptual quality [6, 33, 54, 87]. Specifically, subjects are asked a question on two scenarios under different system control assignments, one using the original setting (as *reference*), and the other using a modified setting (as *reference* + *modification*). The two scenarios are presented in a random order. Each subject can choose either alternative or answer undecided. When multiple subjects are asked to do pairwise comparisons, the number of responses from those preferring one over the other is statistical. It is measured by the *sample awareness* $\hat{\mathcal{A}}$, or the fraction of subjects who answer correctly to the question posed.

Pairwise assessments have higher complexity than absolute assessments because the Cartesian product of combinations of control-input settings must be tested. In practice, pairwise assessments have difficulty in handling more than one control input and more than two quality metrics. Heuristics must be employed to limit the search space.

For example, simplifying assumptions on the relation among subjects' opinion were made in a previous study [77] to limit the search space. This method assumed one control input, thereby greatly reduced the complexity. Further, it assumed a linear relation between subjects' noticeability on the difference; in this way the model of the comparison can be simplified.

For another example, the result of pairwise assessments of individual control were simply pooled together using a weighted average [38]. This oversimplifies the relation among the subjective opinions on multi-dimensional controls.

Note that while absolute assessments give a total order on perceptual quality, pairwise assessments only give a partial order. Only when all the alternatives are comparable can the two be equivalent.

Pairwise comparison-based subjective tests are widely used in subjective quality assessments because they do not rely on an absolute standard and allow non-experts to obtain reliable results for hard-to-differentiate differences between two stimuli. To study the precision of this method with non-experts, there were previous works on analyzing the error and convergence of such subjective tests. A simulation-based analysis showed the relation among stimuli, the number of subjects, and the RMS error [53]. The effect due to human error was also discussed. The result of pairwise comparison-based subjective tests was used to guide the optimization of scalable video coding [54]. In this application, the cost of subjective tests is high due to the combinational increase in the scenarios to be tested.

In this thesis, we assume pairwise assessments because even novice subjects without domain knowledge can easily detect the difference between two scenarios. We use pairwise comparison to refer to pairwise assessments in later text to emphasize that assessments are done by comparison.

2.3.2 Just-Noticeable Difference

Just-Noticeable Difference (JND) is a concept tightly connected to relative assessments. JND is the minimal change of an input (or multiple inputs) that can be perceived in output by humans. It has been employed to study human perception in numerous applications, including light intensity, brightness, loudness of sound, and various multimedia applications [27, 28].

JND is a statistical concept defined with respect to given $awareness$, which is defined as the fraction of sufficiently many human subjects that can correctly identify the output caused by the changed input, when the original input (ref) and the changed input ($ref + mod$) are presented one after another in a random order. With this definition, a 75% awareness level is generally used in psychophysics studies [27, 68, 115, 117]. Under this awareness level, JND is the amount of change of reference input ref in order to achieve 75% awareness: $JND(ref|awareness = 75\%)$. Obviously, awareness is the asymptotic form of sample awareness in pairwise comparisons. If we are interested in the awareness of a changed input $ref + mod$ from the reference ref, we can define JND as an awareness function: $p(ref, mod)$, where p is the fraction of humans perceiving the change from ref.

JND has attracted interests since Weber's proposal in the 1800's [27]. It states that, under given ability (awareness) to perceive a change ΔI in response to the intensity I of a stimulus, ΔI and I are linearly related:

$$\frac{\Delta I}{I} = k \text{ (for constant } k \text{ and awareness).} \tag{2.1}$$

The measurement of sensations was later studied by Fechner [27], who chose JND as the unit of sensation. By setting a zero point (fixed reference I_0), he extended Weber's Law into Fechner's Law and developed a differential equation that relates perception S (awareness) to I, leading to a logarithmic relation:

$$S \propto \log I \text{ (for given } I_0\text{).} \tag{2.2}$$

Stevens subsequently argued that Fechner's Law did not always hold. He proposed Steven's Power Law that modeled the relation as an exponential proportion [86]:

$$S \propto I^a \text{ (for stimulus-dependent exponent } a\text{).} \tag{2.3}$$

Thurstone further proposed more complex models on the output of comparative judgment for finding JND [91].

Other researchers have studied the relation between awareness and modification for a given reference (aka *psychometric function*). These models tend to be an S-shape curve, such as cumulative normal, logistic and Weibull models [28]. Recently, models from signal detection theory are also popular.

The above studies have led to analytic models relating JND, intensity, and awareness. Their advantage is that they do not require expensive subjective tests when applied, but they are not always true in applications with complex mapping between controls and perceptions.

With the models of psychometric functions, researchers have investigated methods for finding their parameters. The most popular method is called *Method of Constant Stimuli*. It asks subjects to compare the reference and the modification and to determine which one is brighter/stronger in intensity. After changing the modification several times using algorithms like PEST [90] or QUEST [99], it calculates the parameters from the results. The disadvantage of this class of methods is that they only provide the psychometric function for a given reference and requires Weber's law to generalize the results to other references.

Our proposed method in Chapter 4 is also a method of constant stimuli but relies on dominance property to sample the data, rather than on Weber's or Fechner's Laws to generalize the results under a given reference.

A wide range of multimedia applications have benefited from the use of JND. It is an effective quality assessment method because it ignores distortions and errors that are not noticeable [98]. Its usage can be further extended to applications of signal compression in run-time quality estimation when compressing images [92], video [49, 73], and audio [3]. JND can also help in determining control thresholds, such as guiding the insertion of watermarks in images [101]. It has been used in computer graphics to smooth out transitions in animation and to make artifacts unnoticeable [14, 94].

Table 2.1: Representative previous multimedia applications using JND.

Area	Multimedia Topic	Reference	Stimulus	Model
Signal Compression	Image	[92]	Luminance	Luminance masking factor is linear with medium to high background luminance
	Video	[49, 73]	Contrast	Exponential
	Audio	[3]	Loudness	Square Root
	3D Models	[14]	Levels-of-detail (LOD)	Locate JND where the HVS can just distinguish the difference between two LODs, assuming Weber's Law is satisfied
Watermark	Image	[101]	Luminance	Reuse the JND in image compression
Quality Assessment	Video	[98]	Intensity	The sensitivity fits to an exponential function
Virtual Reality	Haptic Data Reduction and Transmission	[36]	Velocity / Position	Multidimensional Weber's Law
	Head-Mounted Display Delay	[1]	Delay	Cumulative normal

It has been used to model thresholds in haptic-force feedback [4, 36] and head-mounted display delay [1] in virtual reality in order to ensure natural and comfortable feeling. There were recent applications on mobile video perception [113] and visual analytics [15]. The models used by the representative works are summarized in Table 2.1.

As shown in Table 2.1 The common deficiency in these applications is that they rely on the simplified linearity, logarithmic or similar assumptions stated earlier. Our proposed model in Chapter 4 overcomes this limitation and provides a more accurate JND surface that can be generated using a few subjective tests.

2.3.3 Subjective Measurements of Perceptual Quality in Multimedia Systems

In this section we list representative previous approaches for measuring perceptual quality in multimedia systems with subjective tests.

VoIP Two International Telecommunication Union (ITU) standards have been adopted for measuring the perceptual quality in VoIP.

ITU P.800 [46] prescribes listening tests for voice. As a standard, it presents in detail the setup and materials for subjective tests. It also compares degradation rating, pairwise rating, and the threshold method for rating. However, it does not provide an efficient test method for reducing the number of tests.

ITU P.910 [48] defines procedures for testing the non-interactive one-way quality in videoconferencing. Similarly, it presents the setup and contexts of the test sequences, and introduces three test methods, namely, absolute rating, pairwise rating, and degradation rating. Depending on the test method (conversational or listening, absolute or comparative), it can cost hours or days to finish all the tasks of the tests. Furthermore, to measure the interactivity for a full measurement of perceptual quality, it should be used along with ITU P.800 [46].

Online Games Zander et al. [116] and Chen et al. [11] tried to measure online-game player- sensitivity with quality of service (QoS). The former collected the running metrics of the game server, at the same time collected users' opinions using questionnaires, to find the relation between objective and subjective quality metrics. The latter collected the network latency and the loss rate of each connection, and measured the playing duration of the user using that connection, to find their correspondence. The authors found that user game-playing time is strongly related to the network QoS and is a potential indicator of user satisfaction.

Chen and El Zarki [13] studied perceptual effects of inconsistency in online games. It broke down the QoE into three factors, namely, the responsiveness, precision, and fairness. It then proposed the mapping function between them and QoE.

M. Claypool and K. Claypool [20] focused on the effects of network latency in online games. The tests in these methods generally require playing different scenarios of a game, therefore, requiring many hours for the entire suite of tests.

General Multimedia Systems Chen et al. [12] adopted method of limits in psychophysics to measure perceptual opinion. They repeatedly changed the system controls to test the point that subjects could not tolerate the quality (subjects indicating this by clicking their mice). Consistency over the test results was checked after the measurements to remove outliers. The system controls are then adopted as the minimal running requirement for such condition. However, generalization is discussed in this work by only testing all the control inputs one by one. The authors admitted that they need a more efficient way to perform the tests, considering the large number of combinations.

Wu et al. proposed a similar method [104] which used the method of comparison to measure the subjective QoE regarding system controls, without providing a generalization method for online adaption. They further proposed a general framework on QoE but did not provide a detailed algorithm for online generalization [106].

Deficiencies These approaches cannot perform the measurements at run time due to the long testing period. Furthermore, these offline tests examine only a small subset of combinations of control inputs and network conditions, and are generalized to run-time conditions using some heuristic methods. Because there are very limited combinations of conditions examined offline, the generalization to unknown run-time network conditions is problematic and difficult. We solve the first problem in Chapter 4 by an efficient subjective-test method that utilizes the dominance property, and the second problem in Chapter 5 by a real-time generalization algorithm.

2.4 Run-Time Operations Based on Perceptual Quality

Because an interactive multimedia system is running in real time, the operation need to be controlled at run-time to achieve the optimal perceptual quality under different conditions. To achieve this goal, previous works have studied the mapping from system controls to qualitative quality metrics as well as the generalization of offline subjective-test results.

2.4.1 Mapping from Controls to Quantitative Quality Metrics

These works aim at finding out the mapping from system controls to quantitative quality metrics rather than perceptual quality. They assume that network conditions are predictable and stationary, and quantitative quality metrics can be directly used to derive perceptual quality (which is not true according to Section 1.2).

There are ITU standards that aim to estimate the quantitative quality metrics according to the quality of traffic (the network bandwidth, network latency, and loss rate), as well as the coding parameters (the coding bitrate and the resolution). Examples of these works are ITU-T G.107 (a.k.a. E-model) [47] for VoIP and ITU G.1070 [35] for videoconferencing.

ITU-T G.107 [47] helps transmission planners have a precise assessment of user satisfactions of the end-to-end transmission performance. It uses many transmission-impairment metrics to calculate the overall satisfaction rating. It further adopts the room noise and the quantization distortion in the compression as factors in the rating.

ITU G.1070 [35] was built upon ITU-T G.107 and added the parameters for measuring video quality, including resolution, video frame loss rate, video frame rate, and video bit rate. Although they are high in precision, both G.107 and G.1070 rely on domain-knowledge and a large number of subjects to develop, which is time-consuming and expensive.

Egilmez et al. [25] proposed to optimize the scalable video streaming by a quantitative quality of metrics to guide the flow. Liang et al. [55] studied the optimization of quantitative quality metrics guided multi-cast in multi-player online games. Deficiencies of these methods include the fact that network condition in real systems is non-stationary and can only be predicted in a short future, as well as the fact that perceptual quality does not have a well-defined function of quantitative quality metrics.

Some systems collected user opinions offline as feedback to adjust quality. With the feedback, they tried to find a guide of the adjustment of perceptual quality at the end of a multimedia session. They assumed that coarse opinion about the overall experience rather than the precise evaluation is of interest.

An example of this type of method is that in Skype [84], after a session the system sometimes provides a window to ask for rating for the conversation. This cannot be used at run time when the user is talking; therefore, cannot optimize online performance.

Because only an overall coarse perceptual opinion is collected, these methods cannot guide the optimization of specific control of the system at run time, thereby cannot solve our second problem stated in Section 1.4.

2.4.2 Generalization for Online Optimization

To solve the same problem regarding the generalization problem aforementioned in Section 1.3, several past efforts have been contributed to the development of general methods for optimizing the perceptual quality of multimedia systems directly by tuning system controls. These methods explore a partial space of mappings from controls to perceptual quality in offline experiments and generalize the limited offline results to run-time situations using some heuristic methods (either statistical or non-statistical).

Huang et al. [39] used a heuristic weighted linear function to estimate the combined impact of the Internet variations and system adaptations in order to optimize the interactive multimedia with least flicker. Evolutionary algorithm was employed to find the most similar mapping to the current network condition. The corresponding control was then used in the system. Since the mapping was found only by similarity of scenario and metric of flicker, we do not know whether it can guide the system to achieve the optimal perceptual quality.

Sat and Wah [77] proposed a statistical method for searching offline the optimal control (only one control) of a VoIP system and then generalize them online using a learned SVM. Huang [40] proposed a modified version of Sat and Wah's method for a videoconferencing system. They have similar limitations: the mapping is generalized from several representative test cases by machine-learning algorithms. Because a machine-learning algorithm requires a large number of training samples to ensure the accuracy, the small number of test cases in these methods cannot ensure reliable generalization.

These methods have four deficiencies. First, network simulations via offline models is inadequate to model all possible run-time conditions because it is highly parametrized. Second, without proper network layer controls, it is unclear how the real Internet-layer conditions can be mapped to those simulated network layer conditions. Third, it is difficult to select the proper set of offline tests to properly cover the large space of controls and network conditions, or the large selected set is inadequate, given the cost of doing subjective tests offline. Finally, generalization of these methods is difficult, since the model to cover the space of controls and network conditions is unknown.

To address the first two deficiencies, we only handle the perceived network conditions during subjective tests but leave the adaption to different real network conditions to the network-control layer (Chapter 3). We address the third deficiency by an efficient subjective test method which utilizes the dominance property to select representative test sets (Chapter 4). We address the final deficiency by a real-time generalization algorithm that can cover the whole control space and adapt to any network conditions.

2.5 Summary

In this chapter we have reviewed necessary background for understanding the problem stated in this thesis. We have further investigated previous works on optimizing online interactive multimedia systems to see what have been done and what have not been solved. At the end of each section, we have listed the deficiencies of these previous works and described how they can be solved in later chapters. These can help understand the relation between previous works and our work.

□ **End of chapter.**

Chapter 3

Network-Control Layer

3.1 Overview of Strategies for Maintaining Stable Multimedia Session

Real-time multimedia session requires a stable network condition for the transmission of packets. We define stability as follows.

Definition 3.1.1. Stability. *The continuous playout of the multimedia content and events without sudden pause or significant quality degradation.*

The maintenance of stability require a low network latency and low average loss rate, because medium to high loss rate and delay can impact the reception of real-time multimedia data, degrade the experience of interactions, or even lead to a pause of the session. As connection through the Internet can have delays and loss patterns that change rapidly over the whole session, to allow any fast-paced interactive multimedia systems to run under a stable network condition, we need to have a specific layer to rapidly respond to the network environment and decide on which method to utilize before passing the packets to the application layer. We call this layer the network-control layer.

To simplify the discussion in this chapter, we have two assumptions regarding the multimedia applications we study:

Assumption 3.1.1. Constraints from the application level. *As have been mentioned in Chapter 1, the application layer passes requirements on loss rate, EED and minimal required bandwidth as hard constraints. For illustration in this chapter we assume the average loss rate to be lower than 5%, as most codeces can encounter problems when it is above this value, and games can have inconsistent events among the views of players. We further assume the EED to be less than 400 ms (a common minimal requirement of real-time interactive applications). The minimal bandwidth is set to be 100 Kbps as most real-time multimedia applications, especially videoconferencing applications, require at least this bandwidth.*

Assumption 3.1.2. Limitation of Loss concealment. *The built-in loss concealment strategy in videoconferencing and online games can tolerate at most 2 continuously lost packets. This assumption can be justified in that the video codec for videoconferencing using P frames with intra-blocks can conceal such losses, and audio codec as well as online games generally use three-way piggybacking that can recover at most two continuous lost packets.*

3.2 Problem Statement

There are three types of network impairments that can significantly affect the stability of online fast-paced interactive multimedia sys-

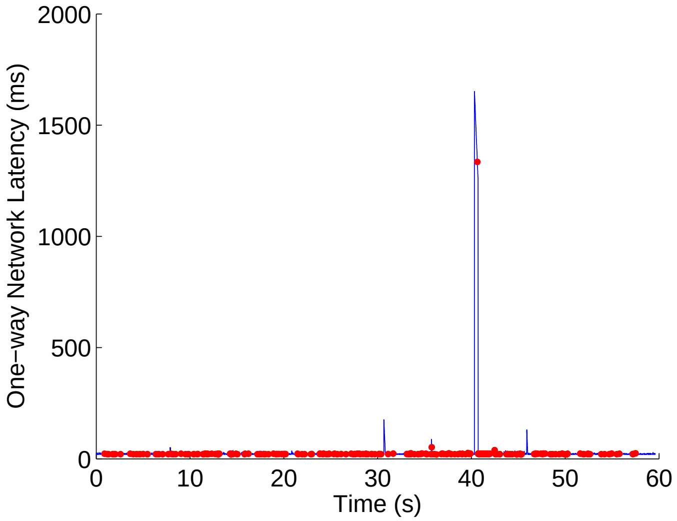


Figure 3.1: Illustration of a sudden increase in network latency and a high loss rate. Red points indicate lost packets.

tems.

First, network congestion can fill up the buffer, force packets to wait in the queue for a relatively long duration, leading to a large increase in latency, and significantly postpone the playout of real-time multimedia contents and interactive events. Figure 3.1 illustrates the case where network delay suddenly increases from a low average delay to a high latency. During the change of the delay, we have a long period that cannot receive any packets. This will unavoidably lead to a pause of the interaction.

Second, even though the network condition has been greatly improved nowadays, the connection can still have random losses due to transmission errors in wireless environment or small congestion in the link [110]. Most modern video and audio codecs have built-in loss concealment mechanisms that can tolerate small losses. However, they can only tolerate at most a 2% random loss rate. As we

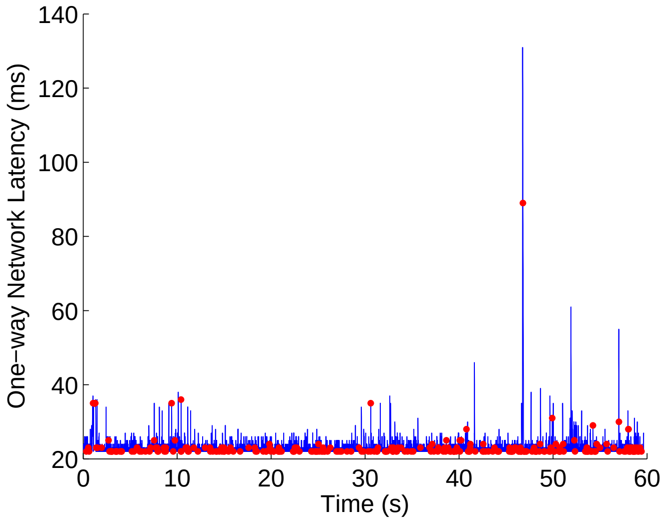


Figure 3.2: Illustrations of a trace with three groups of three-packet consecutive losses and other random losses with less than two consecutive losses.

have assumed our applications can run in a link with at most 5% random loss rate, we should bridge the gap between these two loss rates. It is worth mentioning that retransmission is not feasible for fast-paced interactive multimedia because the double transmission time can result in a significant pause in the multimedia session. Figure 3.1 depicts a link with more than 2% random loss throughout the session, which shows a long-term behavior that cannot be tolerated.

Third, interactive applications are not as robust to consecutive losses as to random losses. When multiple packets are not received within their time limit, correlation between video and audio frames, as well as game events, can be lost. Furthermore, some codecs place redundant data in the next few packets in order to recover the lost packet. When these packets are lost at the same time, we do not have other means to recover them. The artifacts caused cannot be concealed. Figure 3.2 shows a connection with groups of three-consecutive packets. It is noticed that more-than-two consecutive losses generally appear in a link already with many lost packets. However, there are only three groups of three-consecutive losses in this link. It indicates that the prediction of consecutive losses is impractical. Rather, we should reduce the random loss rate to avoid the occurrence of consecutive losses.

To further demonstrate the problems, we have conducted a large scale network experiments with a worldwide overlay network PlanetLab [18] to analyze the situation of network impairments in point-to-point links. For every two node of the test set, we synchronize their clocks using the NTP protocol [62]. We then transmit UDP packets with sequence number and time stamp inside for calculating the one-way network latency and the loss rate. The transmission rate are set from 100 Kbps to 1000 Kbps, equally separated at 8 transmission rates. For each transmission rate, we collect the network trace for 1 minute, stop for 10 seconds to avoid interference, then switch to a higher transmission rate, until we reach 1000 Kbps. We assume that in the same link the 1-minute long-term network condition will not change significantly; therefore, statistics on links collected in different rates can be treated as in continuous transmission.

Because the Internet is unlimitedly wide, it is not possible to enumerate all possible network patterns. Rather, we study the network behaviors with these representative network traces collected from a sufficiently large overlay network above the Internet, and try our best to extend its sampling period.

With the experiments, we have collected 11,935 valid traces in 2012/9/4 and 2012/9/5. According to our assumption regarding the loss rate (< 5%) and latency (< 400 ms), we filter pairs of nodes that cannot satisfy this requirement. Finally, we have 8,518 traces for analysis. Traces can provide the benefit for repeated experiments, which cannot be done by online experiments because the network condition tends to change after days.

Table 3.1 summaries the statistics of the traces. For each trace, we are interested in three types of statistics.

We calculate the average, minimal, and maximal one-way network latency, because they represent both the ordinary latency and the extremely high latency when a network congestion occurs. We do not calculate delay jitters because we have observed that nowadays network, the high jitter occurs only when there is network congestion. The highest value of the maximal latency is so high that it significantly increases the variance of the latency in the whole link, while it does not mean the link has a consistent delay jitter. Therefore, the maximal latency is more suitable for our analysis of network impairments that can disturb the transmission.

We also attain the average random network loss rate, for that it directly affects the quality of the received multimedia content. Note that we do not consider the many late arrived packets in a network congestion as random losses, as they are caused by different factors.

Further, we count the occurrences when less than or equal to two, as well as when more than two consecutive lost packets are observed. We further calculate the sum of them in each link. These statistics can provide useful information about the frequency of the consecutive-loss impairment. To sum up the statistics in each link, for all the statistics mentioned above, we get the maximum of them throughout all the traces, in order to observe the most extreme network condition that we need to handle.

Table 3.1: Statistics of the network traces collected on 2012/9/4 and 2012/9/4.

Type	Source	Destination	Max Avg Delay (ms)	Max Min Delay (ms)	Max Max Delay (ms)	Max Avg Loss	Max Occurrence of Consecutive Loss <= 2 Packets	Sum Occurrence of Consecutive Loss <= 2 Packets	Max Occurrence of Consecutive Loss > 2 Packets	Sum Occurrence of Consecutive Loss > 2 Packets
Intra-national	planetlab2.aut.ac.nz	planetlab1.cs.otago.ac.nz	127	15	4092	1.05%	97	3105	1	6
	planetlab1.cs.otago.ac.nz	planetlab2.aut.ac.nz	27	12	2132	0.91%	83	3112	1	9
Intra North America	metis.mcs.suffolk.edu	planetlab-2.cs.colostate.edu	75	24	3311	4.81%	501	52805	3	64
	pl1.rcc.uottawa.ca	node2.planetlab.albany.edu	104	13	3099	0.11%	13	153	1	1
	planetlab1.cs.unc.edu	planetlab1.uta.edu	35	24	2596	0.07%	0	0	1	1
	planetlab1.dtc.umn.edu	pl-node-1.csl.sri.com	130	115	2976	2.82%	391	532	3	6
	planetlab1.uta.edu	planetlab1.cs.unc.edu	56	27	907	0.04%	1	9	0	0
	planetlab6.csee.usf.edu	roam2.cs.ou.edu	24	18	2017	0.19%	29	756	1	1
	planetlab-2.calpoly-netlab.net	planetlab-1.elisa.cpsc.ucalgary.ca	49	26	2677	0.01%	1	1	0	0
	planetlab2.mnlab.cti.depaul.edu	roam2.cs.ou.edu	44	15	2154	0.22%	8	441	0	0
	planetlab2.netlab.uky.edu	planetslug4.cse.ucsc.edu	47	40	308	0.00%	0	0	0	0
	planetlab5.cs.cornell.edu	planetlab2.netlab.uky.edu	134	20	3235	0.15%	2	14	0	0
	planetslug4.cse.ucsc.edu	planetlab2.netlab.uky.edu	71	43	2957	0.07%	1	5	1	1
	roam2.cs.ou.edu	planetlab6.csee.usf.edu	42	29	2643	0.15%	2	112	1	17
Inter-national	metis.mcs.suffolk.edu	planetlab1.eurecom.fr	299	50	21509	5.26%	519	62156	3	83
	planetlab-01.ece.uprm.edu	pl2.zju.edu.cn	141	140	275	0.51%	48	3631	0	0
	planetlab1.ecs.vuw.ac.nz	planetlab-2.calpoly-netlab.net	188	76	3242	0.44%	25	48	4	7
	planetlab-1.elisa.cpsc.ucalgary.ca	planetlab-2.calpoly-netlab.net	36	17	2630	0.04%	3	9	0	0
	planetlab1.georgetown.edu	planetlab-1.cs.auckland.ac.nz	192	101	2958	0.06%	1	2	0	0
	planetlab-2.calpoly-netlab.net	planetlab1.ecs.vuw.ac.nz	103	74	3127	0.43%	21	38	3	3
	planetlab2.pop-mg.rnp.br	node2.planetlab.albany.edu	190	101	3391	0.58%	62	3922	0	0
	planetlab2.pop-pa.rnp.br	planetlab4.tamu.edu	153	150	781	0.58%	46	3558	0	0
	planetlab-2.ssvl.kth.se	pl02.comp.polyu.edu.hk	184	166	916	0.22%	27	111	1	2
	planetlab4.tamu.edu	planetlab2.pop-pa.rnp.br	189	149	2647	0.02%	1	2	0	0
	planetlab5.cs.cornell.edu	planetlab-node1.it-sudparis.eu	56	54	121	0.48%	33	220	2	6
	planetlabone.ccs.neu.edu	csplanetlab4.kaist.ac.kr	196	98	4888	0.18%	11	130	1	3
	planetslug4.cse.ucsc.edu	planetlab-node1.it-sudparis.eu	113	102	168	1.74%	227	5157	2	14

The results in Table 3.1 demonstrate our statements on the network impairments above.

First, the network latency generally remains at a low level, but can suddenly increase to as high as several seconds. This should lead to a disruptive pause in the session. Second, although the random loss rate in most links are low, there are several pairs of nodes which have a random loss rate higher than 2% and lower than 5%. We should reduce such loss rate to below 2% by loss concealment in order to maintain the smooth progress of a multimedia application. Third, we notice that most consecutive losses are corresponding to less than or equal to 2 packets. This supports that most multimedia applications have the ability to tolerate at most 2 consecutive lost packets. On the other hand, the more-than-three consecutive losses seldom occur, but they do exist. This adds difficulty to the prediction of their occurrence.

With these observations, our research problem is, how we can build a general framework for the stable transmission of fast-paced interactive multimedia data through the Internet. The non-trivial point of this problem is that we should decide in real-time which method we will adopt to settle the network impairments. Because fast-paced interactive multimedia applications have a tight playout deadline of multimedia content, if the method is not chosen sufficiently fast, network condition can change, and the network impairments cannot be handled correctly.

Our approach to solve the problem is to combine previous methods for handling the network impairments, but with a novel method that can dynamically decide which previous method we will use in real-time. The mechanism is based on efficient online tests for the availability of the assumption of the previous methods. Because we only use the necessary method when it can be used, we do not reduce the network bandwidth if reducing it does not help ease the network congestion. Further, we do not waste the valuable network bandwidth to transmit too much redundancy due to wrong estimation of the network condition. We will show that with the proposed method, the network impairments can be better solved, while at the same time, the throughput of the application can be increased.

In this chapter, we first discuss previous work for solving the network impairments in Section 3.3 and their limitations. We then propose the real-time method for deciding which previous method we should adopt to solve the impairments according to the current network condition in Section 3.4. Finally, we evaluate the proposed method in Section 3.5.

3.3 Methods for Solving Network Impairments

Our goal is to build a general framework for solving network impairments. Inside the framework, previous works are adopted as middle ware for settling specific problems. As we do not enforce the method for each problem, these previous works can be changed with other existing works. We review in detail representative methods used in this framework and their limitations, especially the assumptions in practical usage. We illustrate the condition where these methods not work. We leave our method for determining the usability of this middle ware in Section 3.4.

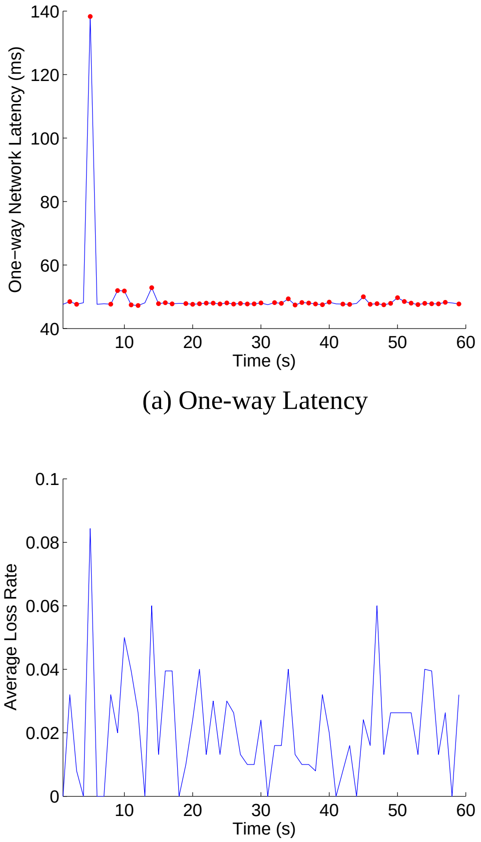
3.3.1 Congestion Avoidance

As aforementioned in Section 3.2, network congestion can incur high latency in the transmission, which will lead to a significant pause in the interaction and impact perceptual quality. Fortunately, this impairment can be settled by congestion avoidance algorithms which are generally referred to as rate control algorithms.

Congestion avoidance is a popular research topic for decades. A number of rate control algorithms have been proposed to achieve a proper transmission rate in the network, especially the shared Internet, to avoid high packet losses or delays when the network is congested [8, 9, 32, 66]. These algorithms generally provide an upper bound of the transmission rate, indicating that below this the transmission can be stable, and beyond this the transmission can be out of control.

These algorithms can dynamically adjust the transmission of the application, and avoid the congestion caused by the application itself, with a sacrifice of the transmission bandwidth. Figure 3.3 illustrates how a congestion avoidance algorithm TCP-Friendly Rate Control (TFRC) [32] reduces the transmission rate to avoid further congestion in the link, which reduces the network latency to the normal level, and maintains a loss rate that is below the upper bound (5%) prescribed in our assumption.

The goals of any rate control algorithm are to have a channel with a stable bandwidth that is TCP-friendly (in the presence of other TCP traffic) and to have a bandwidth that can satisfy the requirement of transmission bandwidth given by the application. The framework of the algorithm is presented in Algorithm 3.1. A rate control algo-



(a) One-way Latency

(b) Loss Rate

Figure 3.3: Illustrate of a congestion avoidance algorithm in a link. The high delay at the beginning of (a) being reduced in later transmission shows further congestion is avoided.

Algorithm 3.1 Pseudo code of a Rate Control Algorithm

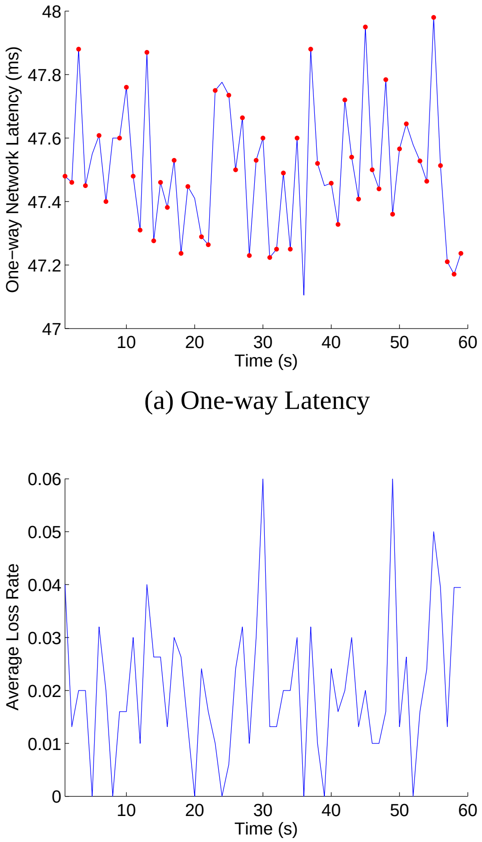
```
Initialize the average transmission rate $PktRate_{\text{init}}$.
while not end of the session do
    Receiver measures the loss event rate and sends back to sender.
    Sender uses the feedback packet to estimate RTT.
    With the loss event rate and RTT, sender uses the throughput equation to calculate the new transmission rate $PktRate_{\text{avg}}$.
    if $PktRate_{\text{avg}} > 2PktRate_{\text{rec}}$ then
        $PktRate_{\text{avg}} = 2PktRate_{\text{rec}}$;
    end if
    Sender transmits at the new rate $PktRate_{\text{avg}}$.
end while
```

rithm allows small variance of the transmission rate while keeping the average transmission rate at the calculated $PktRate_{\text{avg}}$.

The algorithm generally converges in less than 1 minute. While it may induce a very high transmission rate at the beginning, it will soon reduce to a reasonable rate. Therefore, the overhead of the algorithm can be omitted after the algorithm finishes its initialization.

When congestion avoidance algorithm works, it can provide an upper bound of the transmission rate, under which we can safely transmit the packets without worrying about the occurrence of high loss rates and delays. When it does not work, we have to use a conservative transmission rate (e.g. 100Kbps).

The deficiency of rate control algorithms is that, in some connections the congestion is not caused by our application itself, but remote applications which we cannot control. Under that condition, further reducing the transmission rate will be meaningless. Figure 3.4 illustrates a link on which, while TFRC is changing the transmission rate, the delay is not affected. It indicates that the congestion is caused not by queuing but by the router dropping packets due to the



(a) One-way Latency

(b) Loss Rate

Figure 3.4: Illustration of link where congestion avoidance algorithm does not affect the network latency. The delays in (a) remain at around 47 ms no matter how the transmission rate changes.

inability to process incoming packets.

3.3.2 Prediction-Based Redundant Packet Mechanisms

Random losses larger than 2% can degrade the quality of the multimedia content. Previous works have tried to solve this problem by sending redundant or parity data.

Forward-Error Correction (FEC) [72] utilizes parity to correct error bits during the transmission. In multimedia, the Reed-Solomon error correction version of FEC is adopted to recover lost packets. Specifically, when we send k more parity packets along with the original n packets, we can recover at most k lost packets out of the $n + k$ packets during the transmission. As we can only perform the recovery after we have received the n packets, we need to allocate a buffer for receiving all these packets. This may cause extra latency in the interaction, and the trade-offs will be discussed in Chapters 5 and 6.

Piggy-backing sends copies of the original packet along with the subsequent packets. In this way it can reduce the latency for waiting a number of packets before decoding them. However, because packets larger than MTU are forced to be divided into segments, only piggy-backing with small packets are meaningful. Still, the extra latency can delay the interaction.

While FEC can incur extra transmission rate for the parity packets, piggy-backing does not significantly affect the network condition in packet-switch network because the transmission is packet-BY-packet. This is also a factor in the consideration of choosing the suitable mechanisms for protection.

When prediction-based redundant packet mechanisms work, we can reduce the loss rate that the application layer will perceive to below 2%, thereby allowing them to present media with high quality and less interruption. When they do not work, we have to handle the lost packets with loss concealment methods.

The deficiency of prediction-based redundant packet mechanisms are that they may consume extra transmission bandwidth. FEC can consume transmission bandwidth, resulting in less bandwidth for transmitting the original data. Therefore, to transmit data efficiently, we should determine how much redundant data we should transmit in real-time. Piggy-backing only increases the bitrate but not the packet rate; therefore, it may help save the queuing time in some network condition. However, as aforementioned, it does not work when the packet itself is already large, which limits its usage in multimedia applications.

Both FEC and piggy-backing require precise prediction of the future network condition to efficiently utilize the bandwidth and the buffering time. The prediction is based on the stationary and correlation assumptions, which will be detailed discussed in Section 3.4. As an example of poor performance when the assumptions cannot be satisfied, Figure 3.4b illustrates a network condition in which the loss rate changes without predictable patterns. If we decide the redundancy according to a short window, the loss rate will change up and down, and the bandwidth will be wasted.

3.3.3 Loss Concealment

Consecutive losses drop the context of the multimedia contents, resulting in a pause in the content, which can hurt the perceptual quality. Many codecs and applications have built in mechanisms that can progress even with a few lost packets. Generally, at most 2 consecutive lost packets can be concealed, which meets our observation in the large-scaled network measurements in Section 3.2.

As a last resort, loss concealment is weak at handling high loss rate and high delay jitters, but strong at recovery of unpredictable losses. Because it is built-in function, using it will not incur extra usage of transmission bandwidth. Therefore, it can always be used as a stand by mechanism. Its deficiency is that it can handle at most 2 consecutive losses. We need prediction-based redundant packet mechanisms for reducing the number of consecutive losses to within 2 packets.

3.3.4 Framework for Combining Network-Impairment Settling Methods

We combine the above methods in our network-control layer as follows.

To reduce the instability of the transmission, our first strategy is to avoid the congestion in the link, because congestion can induce uncontrollably high delay and cause many late packets that cannot be recovered. Our second strategy is to recover the error due to only a few random losses or late packets with redundant packets being sent along with the ordinary packet if the link is sufficiently stationary and has certain correlation between the recent past and future be-

Table 3.2: Summary of previous methods used in the network-control layer

Problem	Method	Assumption	Deficiency
Sudden Increase of Network Delay	Congestion Avoidance	Congestion caused by local client	Waste bandwidth
Random Loss	Redundant Packet	Stationary and Correlative	Waste bandwidth
Consecutive Loss	Loss Concealment	Loss <= 2 consecutive packets	

havior. We also expect that redundant packets can help recover one packet when 3 consecutively lost packet occurs. These two types of strategies can be done shortly before the data is transmitted and are preferred.

Because the network behavior varies fast and can be unpredictable, we have to handle low level of losses that are left after the protection of the above strategies. In this case, we resort to loss concealment strategies to recover the lost data due to at most 2 consecutive lost packets.

We summarize the methods we use in the network-control layer in Table 3.2, along with the requirement for the methods to run and their deficiencies. The priority of the usage is from high to low in the table. As a sudden rise of network latency can interrupt the transmission, our first task is to avoid its occurrence with the rate control algorithm. We then tackle the remaining random losses by redundant packets if the assumptions for the prediction can be satisfied. Loss concealment is the last resort for the losses that still cannot be corrected.

3.4 Tests of the Usability of the Network-Impariment Settling Methods in the Framework

In this section we discuss how we determine at run time which network-impairment handling method is adopted in the network-control layer for fast-paced interactive multimedia. Again, our goal is to provide a general framework for existing strategies to fit in rather than reinventing them. Therefore, the tests are described in a general form.

3.4.1 Congestion Avoidance

To maintain a stable network behavior, we expect a rate control algorithm to provide a transmission rate which can maintain an average loss rate and delay satisfying the worst-case requirement given in Assumption 3.1.1. Our test is therefore stated as follows:

Test for Congestion Avoidance Algorithm. We test in a Ts window, with the suggested upper bound transmission rate $PktRate_{\max}$ by the congestion avoidance algorithm, whether the average loss rate p is consistently less than 5%, the EED is consistently less than 400 ms, and the actual throughput is consistently larger than 100 kbps.

Because the fast-paced interactive applications have a tight playout deadline, we do not have sufficient time for determining the size of the window T at run time. We therefore attain T by testing different T with the traces we have collected in Section 3.2. The simulation of the change of bandwidth is done by attaining the losses and delays from the trace with the nearest transmission rate (from one of the eight transmission rates) compared to the suggested transmis-

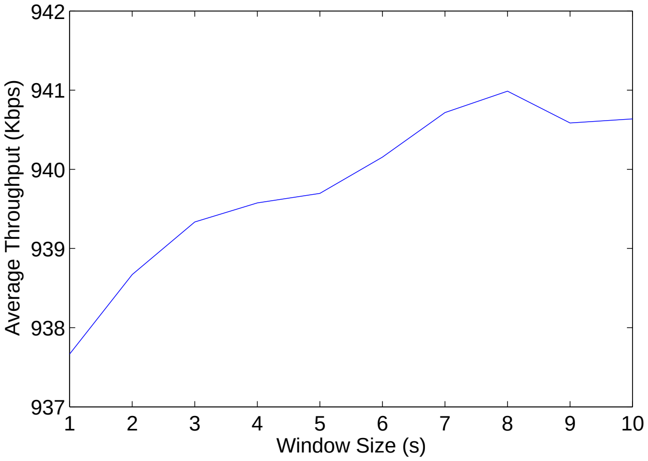


Figure 3.5: The average throughput achieved by the traces as a function of the window size of the congestion test.

sion rate by the rate control algorithm. We calculate the average throughput of the link in one-minute intervals with all the traces we have collected, and check which T can result in the highest average throughput. Figure 3.5 shows that when $T = 8$ s the throughput is highest; therefore, we use it as the window size at run time.

This test of the usability of the congestion avoidance algorithm is not a statistical test, because congestion avoidance algorithms change the transmission rate relatively slowly, resulting in a lack of samples. We consider the algorithm fails this test whenever one of the three metrics in Assumption 3.1.1 cannot satisfy the requirement. Under such a network link, we are aware that the congestion cannot be eliminated by reducing our transmission rate, so we have to transmit the packets at a fixed rate (1000 Kbps) for certain period (a $10s$ duration considering the update period of the network controls for avoiding flickers) until the restart of the congestion avoidance algorithm.

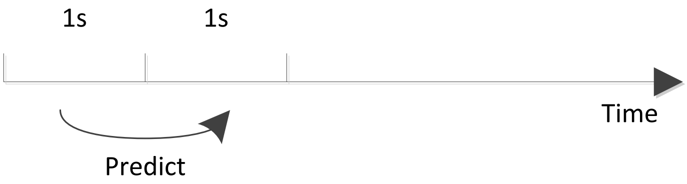


Figure 3.6: Predicting the metric in the next second according to the network behavior in the last second.

3.4.2 Stationarity and Correlation

After using the congestion avoidance algorithm, the network latency in the link becomes low and smooth. The remaining problem is the losses. It is known that the optimization of perceptual quality of multimedia applications requires real-time prediction of network loss rate for setting the redundancy level, because with the prediction we can allocate the optimal amount of redundant packets for maintaining the robustness of the transmission in face of losses, while not sacrificing the throughput of the ordinary data. Our goal is to predict the metrics of interest (average delay and loss rate) in the next second based on the data collected in the last second (Figure 3.6).

In a multimedia application, data is generated in logical units, like video frames, audio frames, and game operation sets. They are then packetized and transmitted to remote computers. Because each logical unit needs to be processed after the last packet of the unit is received, it is reasonable to transmit the packets of a unit under similar network settings to ensure they arrive at the remote machine at similar intervals (see Figure 3.7). Therefore, the update of the prediction is per unit-interval t_{unit}. Obviously, the interval is different from application to application.

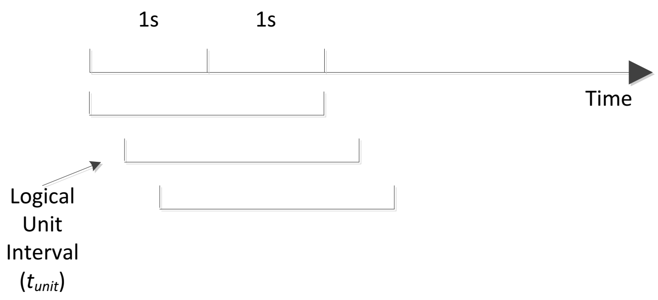


Figure 3.7: Updating the prediction per logical-unit interval (t_{unit}) in order to let packets of a unit arrive at the remote machine at a similar interval.

In summary, our problem is how to predict the network metrics in the next second based on the metrics in the last second per logical-unit interval (33.3 ms in videoconferencing or 16.7 ms in online games) at a given sending rate (100 Kbps to 1000 Kbps in videoconferencing or 100 Kbps in online games). Actually, the prediction requires the satisfaction of two requirements of the network loss rate, namely, stationarity and correlation.

Definition 3.4.1. Stationarity. *A stationary process is a stochastic process whose joint probability distribution does not change when shifted in time. [70]*

Definition 3.4.2. Correlation. *Correlation is the degree to which two or more quantities are linearly associated. [52]*

Stationarity is defined over a long period, indicating that the distribution is unchanged no matter when the sampling starts. For this reason, stationarity is defined over a long term. Correlation relies on stationarity, but it can measure short-term behavior on the average sense; therefore, it is suitable for predicting the average delay and loss rate for real-time multimedia applications. The prediction

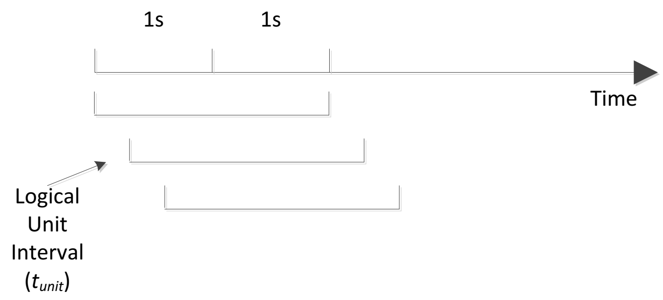


Figure 3.8: Before using correlation, it is necessary to test whether $\{X_{\text{loss}}\}$ is stationary over a given window.

of average behavior a short term is sufficient in most of the time, but still cannot handle some sudden lost or late packets. They will be handled by the loss concealment mechanism instead, which is discussed in the next section.

For testing stationarity, we need to collect samples in a meta-window T. We define $\{X\}$ to be the sequence of $x_t, x_{t+t_{\text{unit}}}, x_{t+2t_{\text{unit}}}, \ldots, x_{t+kt_{\text{unit}}}$. Note that $T - 1 = kt_{\text{unit}}$. (See Figure 3.8.)

We test the following hypothesis:

Hypothesis: $\{X_{\text{delay}}\}$ and $\{X_{\text{loss}}\}$ are respectively stationary over a Ts meta-window.

The test can be done by the Augmented Dickey-Fuller test [16]. It tests the null hypothesis of a unit root is present in a time-series sample against the alternative hypothesis that the time series is stationary. We test $\{X\}$ with a p-value equaled to 0.05, indicating that the observed data is stationary only if it can reject the null hypothesis with a high confidence.

There are trade-offs between the size of the meta window T and the quality of the test. With a larger window, the stationarity test is easier to fail, but we have more data for supporting the stationarity

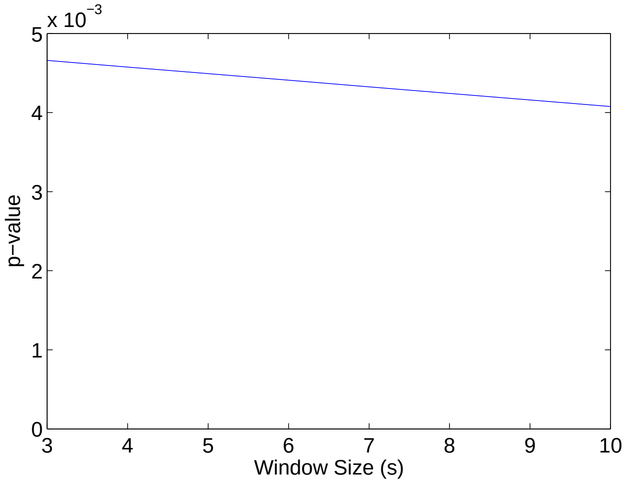


Figure 3.9: The p-value of the stationary test v.s. the window size.

and the correlation tests, and vice versa. Similar to the way we determine the windows size in Section 3.4.1, we test different settings of the window size with all the traces collect in Section 3.2. Figure 3.9 shows that when the window size increases, the probability that a stationary test can be passed slightly increases. This happens because when the window is too small, small variations of the value will be considered unstationary. The increase is slow because when more samples are included in the window, the probability of change will also increase, which cancels the benefit stated above. In short, stationarity is not significantly affected by the change of window size. As a result, we do not have a conclusion on the proper size of the window for testing stationarity. We will decide the proper size along with the correlation test.

After passing the stationarity test, we can further perform the test for the correlation of the average delay and loss rate to measure the quality of the correlation. This can be done by performing the autocorrelation analysis on the loss rate of sequence X.

Hypothesis: $\{X_{\text{loss}}\}$ respectively has correlation.

The test can be done by measuring the correlation coefficient R between X_{t-1} and X_t in the window T. $\|R\|$ equal to 1 indicates a perfect correlation. We consider $\|R\| \geq 0.8$ a sufficiently high correlation that can be used for the prediction. We then use autocorrelation analysis [52] to attain the function for the prediction, and transmit redundant packets to increase the robustness of the transmission per the predicted network behavior when the correlation exists. Otherwise, if the correlation dose not exist, we stop the dynamic adaption of redundancy and use a conservative redundancy level.

Similarly, we analyze the effect of the window size to the failure of the correlation test. Figure 3.10 shows that, when the window size is larger than 7 s, the correlation coefficient will be less than 0.8, the threshold for determining whether correlation exists in our algorithm. Therefore, we choose 7 s as the window size. Because correlation requires the stationary property to satisfy, the window size of the stationary test should be no less than 7 s. In short, we use 7 s as the window size for testing both stationarity and correlation.

It is worth mentioning that rejecting the null hypothesis of the above statistical tests does not ensure the correlation of the past and future network behavior, because the p-value indicates that there is still probability that the correlation is neither stationary nor having correlations (e.g. “95% not unstationary” does not mean “95% stationary”). Therefore, we use loss concealment to handle unexpected losses in the next section.

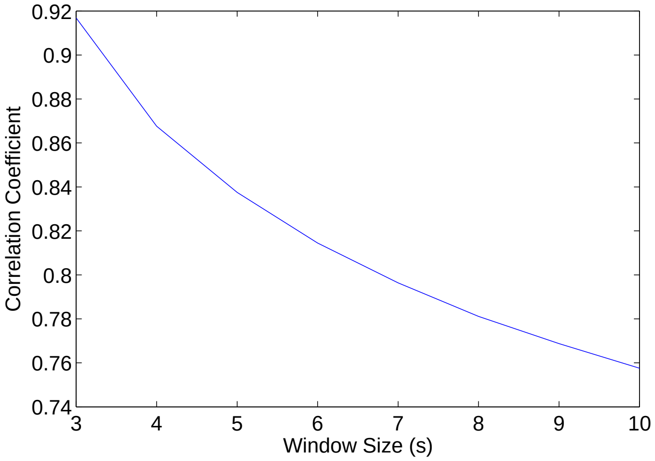


Figure 3.10: Correlation decreases as the window size increases.

3.4.3 Loss Concealment

Congestion avoidance and prediction-based redundant packets cannot handle all the network conditions. As aforementioned there are conditions in which they may fail the tests. Even when the tests are passed, emergent changes of network behavior can happen because statistical tests are based on probability. Fortunately, video codecs can tolerate small losses because they insert intra-blocks into P-frames that do not rely on previous frames. Audio packets and game packets generally have built-in piggy-backing mechanisms to produce extra backup in following packets in order to rapidly recover losses when only a few consecutive packets are lost. Therefore, when the loss rate is small, even when the congestion avoidance and the prediction-based redundant packets mechanisms do not work, the real-time system can still work as the last resort.

The test for whether loss concealment works or not depends on the tolerance of the consecutive losses by the codecs or the piggybacking method. Assumption 3.1.2 suggests that the test fails when the link has more than 2 continuously lost packet. Because loss concealment is the last resort, we cannot present the received data on a best-efforts basis.

In practice, such tests need not be performed, because the more-than-three consecutive loss situation seldom occurs after performing rate control. Another reason for this is that the loss concealment mechanism is usually built inside multimedia codecs. Even when we have found that loss concealment cannot be used, we cannot disable it in the codec.

3.4.4 Summary of the Tests

In this section we summarize the usage of the tests along with the network-impariment recovery methods.

Firstly, we test whether the congestion avoidance algorithm can provide a consistent upper bound of the transmission rate for maintaining a stable transmission (i.e. the loss rate and jitter are within the requirements). As the network behavior changes from time to time over the Internet, congestion avoidance is meaningful only when a consistent guidance can be given.

Secondly, we test whether the network behavior (in terms of the average loss rate and delay) is stationary. If it is, the link is considerably stable. Unfortunately, stationarity is meaningful for some long-term behavior. As the network behavior over the Internet changes rapidly, long-term behavior is not that important. Therefore, based on stationarity, we should further test the short-term correlation between the past and future network behavior. If the correlation is high, we can predict the behavior in a short time into the future with samples in the recent past. However, the correlation can only be measured on the average sense (over a long interval); otherwise, there are insufficient samples for the test. This means that the correlation tests still have some rapid change of network behavior that they cannot cover. This is why we need loss concealments.

Thirdly, because loss concealments are built-in and is the last resort for handling the remaining losses in the codecs, we always turn it on.

3.5 Experimental Results

In this section we demonstrate how our framework can ensure the stability of the transmissions under different network conditions.

We utilize the traces collected in Section 3.2 to perform simulations. The change of transmission rate is simulated with the same method stated in Section 3.4.1. Again, as we want our experiments to be repeatable, we perform neither congestion control nor prediction-based redundant packet mechanisms at run time. Instead, we collect the network traces under different transmission rate and reproduce the network behavior offline by combining the traces with the nearest transmission rate given by the congestion control algorithm. We assume that the background network traffic does not vary much during short term (less than 9 minutes) and the traces with different transmission rates can represent the true network condition as if we actually change the transmission rate at run time.

3.5.1 Evaluation of Network Conditions After using Network-Impairment Settling Methods

In this section, we evaluate the network condition after the rate control method and the redundant packet method are adopted regardless of whether the underlying requirements can be met. The implementation of the rate control is TFRC. The implementation of the redundant packet method is a Reed-Solomon FEC, where the prediction of the redundancy level is based on the statistics in the last second. For loss rates higher than 2%, a (4,1) Reed-Solomon FEC (1 parity packet for every 4 packets) is employed, as it can reduce a 5% loss rate to 2.26%, which is sufficient for reducing loss rates to below 2% in most cases, while at the same time does not overly waste the bandwidth. When the loss rate is less than 2%, the adaptive method does not use the FEC. To simply the discussion, we use a 400 ms buffer throughout the experiments. The methods are run no matter whether the requirements for their running them are satisfied. For the real-time method, we propose for deciding which method to use according to the run-time tests.

We calculate the statistics in Table 3.1 again after using the network-impairment settling methods, to check how the network impairments can be reduced. For loss concealments, we observe whether the number of three-packet consecutive losses is reduced. Finally, we calculate the throughput by comparing it to one without using these methods to see whether redundant packets can help increase the actual throughput or waste the bandwidth.

Table 3.3 shows that with the previous methods, the network impairments have been significantly reduced. Nearly all the network metrics have been improved. The maximal one-way latency is now controlled under 400 ms, and the loss rate is under 2%. The number of occurrences of the three-packet consecutive losses is also reduced. All these show that our network-control layer has chosen the proper methods for settling the network impairments. Most pairs of nodes have high transmission throughput near the upper bound of the transmission rate given by the application, but some connections still have relatively low throughput. Note that the occurrences of consecutive losses increase by a very small value in some connections, due to the fact that the change of transmission rate may incur some random combination of previously separated losses.

Figure 3.11 illustrates the network trace after processed by the network-control layer with both congestion avoidance and redundancy (the same link as that in Figure 3.3). We find that the loss rate is further reduced when compared to Figure 3.3b. With this low loss rate, the multimedia session can be stable. This method can be further improved by our test-based method, and we compare the performance in the next section.

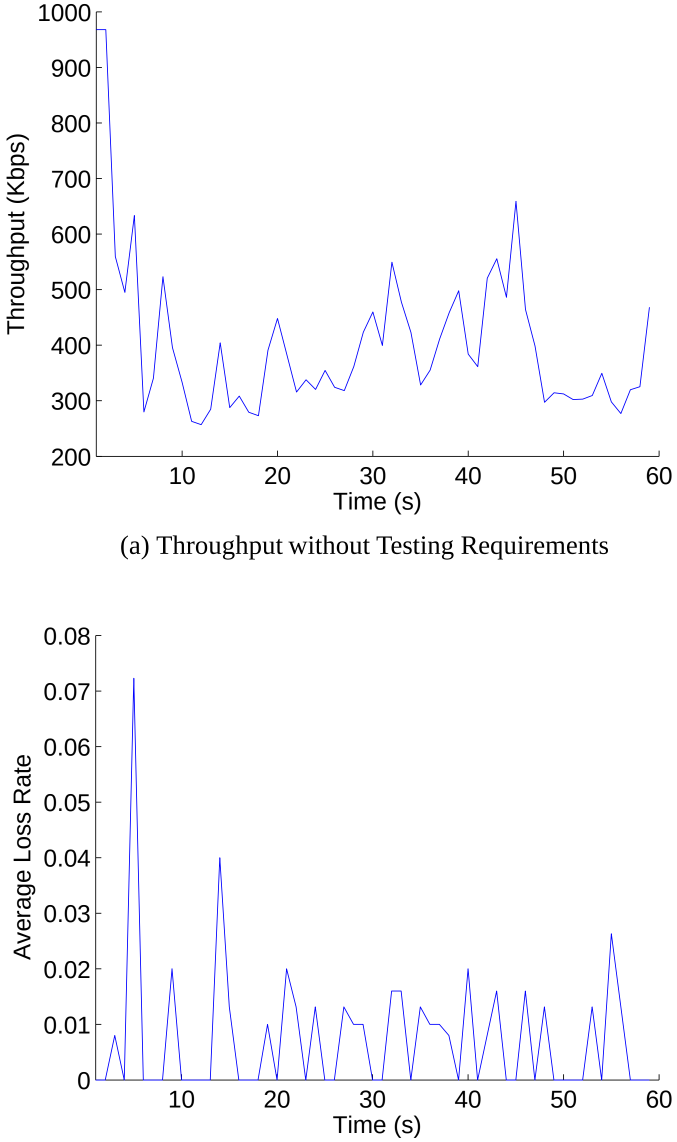
3.5.2 Evaluation of Network Conditions with the Test-Based Method

In this section, we evaluate the method proposed in Section 3.4, namely, to decide in real-time which network-impairment settlment method to use according to whether the run-time tests of the underlying requirements of the methods can be satisfied.

The proposed method does not use rate control to limit the transmission rate when the network condition does not react to the reduc-

Table 3.3: Statistics of the network condition when congestion avoidance algorithm and redundant packets are adopted, without checking whether the underlying requirements are met.

Type	Source	Destination	Max Avg Delay (ms)	Max Min Delay (ms)	Max Max Delay (ms)	Max Avg Loss	Max Occurrence of Consecutive Loss <= 2 Packets	Sum Occurrence of Consecutive Loss <= 2 Packets	Max Occurrence of Consecutive Loss > 2 Packets	Sum Occurrence of Consecutive Loss <= 2 Packets	Avg Throughput (Kbps)
Intra-national	planetlab2.aut.ac.nz	planetlab1.cs.otago.ac.nz	17	16	50	0.55%	79	721	1	2	993
	planetlab1.cs.otago.ac.nz	planetlab2.aut.ac.nz	13	12	51	0.51%	74	671	1	2	995
Intra-North America	metis.mcs.suffolk.edu	planetlab-2.cs.colostate.edu	54	24	1236	1.18%	136	4456	3	34	673
	pl1.rcc.uottawa.ca	node2.planetlab.albany.edu	24	13	692	0.09%	13	105	3	31	982
	planetlab1.cs.unc.edu	planetlab1.uta.edu	24	24	29	0.00%	0	0	0	0	1000
	planetlab1.dtc.umn.edu	pl-node-1.csl.sri.com	81	50	460	0.49%	46	62	2	8	985
	planetlab1.uta.edu	planetlab1.cs.unc.edu	27	27	27	0.00%	1	2	0	0	1000
	planetlab6.csee.usf.edu	roam2.cs.ou.edu	24	18	411	0.19%	27	198	1	1	999
	planetlab-2.calpoly-netlab.net	planetlab-1.elisa.cpsc.ucalgary.ca	36	25	544	0.04%	4	5	2	2	996
	planetlab2.mnlab.cti.depaul.edu	roam2.cs.ou.edu	16	15	69	0.06%	8	100	0	0	1000
	planetlab2.netlab.uky.edu	planetslug4.cse.ucsc.edu	40	40	42	0.00%	0	0	0	0	1000
	planetlab5.cs.cornell.edu	planetlab2.netlab.uky.edu	29	20	521	0.15%	12	23	6	10	986
	planetslug4.cse.ucsc.edu	planetlab2.netlab.uky.edu	68	43	690	0.07%	6	17	3	8	969
	roam2.cs.ou.edu	planetlab6.csee.usf.edu	29	29	29	0.00%	0	0	0	0	1000
Inter-national	metis.mcs.suffolk.edu	planetlab1.eurecom.fr	59	50	283	1.12%	59	1952	2	8	336
	planetlab-01.ece.uprm.edu	pl2.zju.edu.cn	140	135	210	0.34%	30	657	1	1	834
	planetlab1.ecs.vuw.ac.nz	planetlab-2.calpoly-netlab.net	107	76	1273	0.08%	6	59	3	28	961
	planetlab-1.elisa.cpsc.ucalgary.ca	planetlab-2.calpoly-netlab.net	32	16	612	0.07%	6	10	3	5	994
	planetlab1.georgetown.edu	planetlab-1.cs.auckland.ac.nz	106	101	627	0.04%	4	10	2	5	995
	planetlab-2.calpoly-netlab.net	planetlab1.ecs.vuw.ac.nz	86	75	779	0.02%	2	12	1	4	993
	planetlab2.pop-mg.rnp.br	node2.planetlab.albany.edu	114	101	1036	0.30%	28	723	3	4	817
	planetlab2.pop-pa.rnp.br	planetlab4.tamu.edu	151	150	173	0.26%	23	566	0	0	761
	planetlab-2.ssvl.kth.se	pl02.comp.polyu.edu.hk	166	166	184	0.11%	12	17	0	0	993
	planetlab4.tamu.edu	planetlab2.pop-pa.rnp.br	152	149	283	0.02%	2	2	1	1	998
	planetlab5.cs.cornell.edu	planetlab-node1.it-sudparis.eu	54	54	71	0.06%	9	46	1	2	998
	planetlabone.ccs.neu.edu	csplanetlab4.kaist.ac.kr	102	99	280	0.04%	5	16	1	2	994
	planetslug4.cse.ucsc.edu	planetlab-node1.it-sudparis.eu	112	102	138	0.75%	44	688	1	5	811



(a) Throughput without Testing Requirements

(b) Loss Rate

Figure 3.11: Illustration of the effect of the network-control layer using both TFRC and FEC for reducing the loss rate in the link tested in Figure 3.3. a) Throughput for later comparison with Figure 3.12a. b) The loss rate is maintained below 2% in most time.

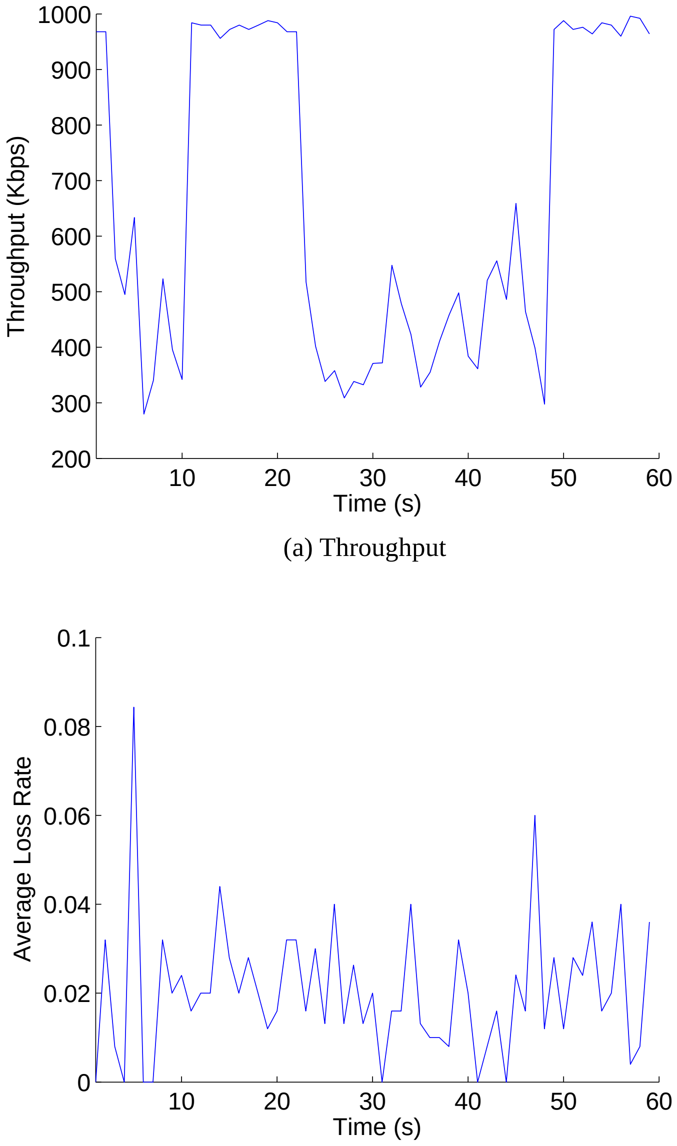
tion of the transmission rate (indicating a remote source of the congestion). Further, it does not use prediction to decide the redundancy level when the assumptions of the prediction does not hold. On the other hand, the transmission rate is kept at the maximal bandwidth if the rate control test fails, and the redundancy is kept at a (4,1) Reed-Solomon FEC if the stationarity and correlation tests fail.

Table 3.4 shows the results of the simulation with the same test set as that used in Tables 3.1 and 3.3. Our proposed method can achieve a more stable network condition than that in Table 3.3, where the maximal latency observed in the links are reduced from more than 1000 ms to around 800 ms. Furthermore, the number of occurrences of consecutive losses have been greatly reduced. Finally, although we do not achieve the maximal transmission rate where the available bandwidth is very high, we can increase the throughput when the available bandwidth is low, from 336 Kbps to 550 Kbps. All these results show that our proposed method has better understanding of the network condition. With the tests of the underlying requirements of the congestion avoidance algorithm and the prediction-based redundancy-packet algorithm, we use them only when they actually work, rather than using them all the time. The bandwidth is, therefore, used in the most necessary cases, and the transmissions are therefore more stable.

Figure 3.12 illustrates the network conditions in the same link as in Figure 3.11, but with our proposed network-impairment settlement methods. It demonstrates that the test-based method can provide a higher throughput while maintaining a stable network condition. According to our run-time tests of the congestion avoidance algorithm, we find that in certain period (10 s - 20 s, and 50 s - end)

Table 3.4: Statistics of the network condition when the congestion avoidance algorithm and redundant packets are adopted, while checking the underlying requirements before using the methods.

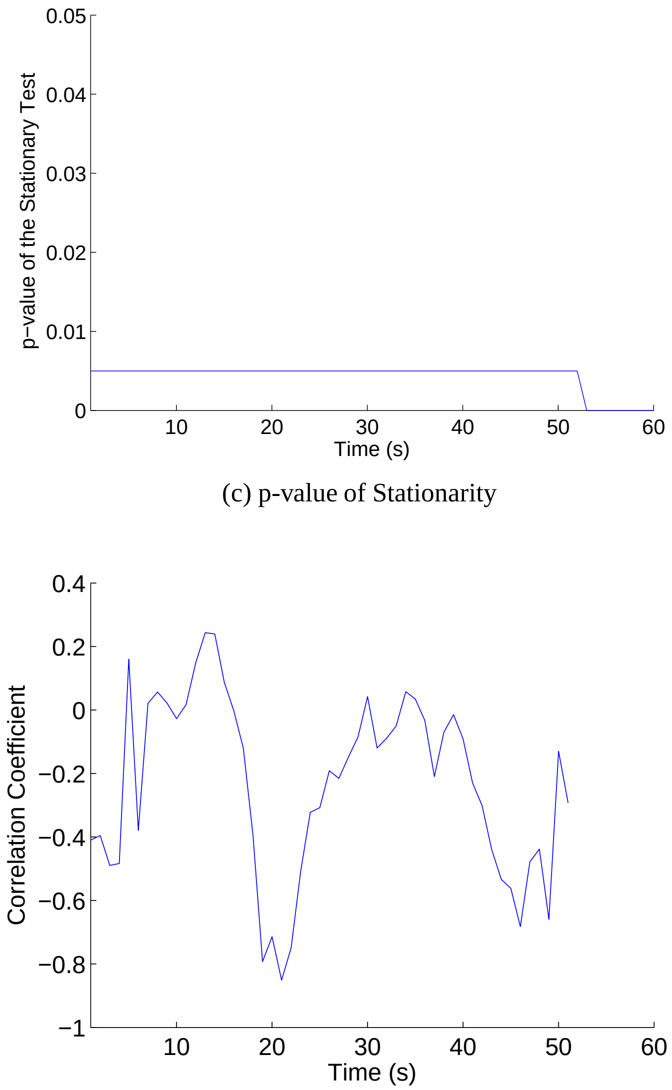
Type	Source	Destination	Max Avg Delay (ms)	Max Min Delay (ms)	Max Max Delay (ms)	Max Avg Loss	Max Occurrence of Consecutive Loss <= 2 Packets	Sum Occurrence of Consecutive Loss <= 2 Packets	Max Occurrence of Consecutive Loss > 2 Packets	Sum Occurrence of Consecutive Loss <= 2 Packets	Avg Throughput (Kbps)
Intra-national	planetlab2.aut.ac.nz	planetlab1.cs.otago.ac.nz	17	16	50	0.21%	30	220	1	2	869
	planetlab1.cs.otago.ac.nz	planetlab2.aut.ac.nz	13	12	51	0.26%	38	209	1	2	869
Intra-North America	metis.mcs.suffolk.edu	planetlab-2.cs.colostate.edu	49	24	1236	0.97%	123	2626	2	35	685
	pl1.rcc.uottawa.ca	node2.planetlab.albany.edu	24	13	692	0.07%	7	89	3	32	971
	planetlab1.cs.unc.edu	planetlab1.uta.edu	24	24	29	0.00%	0	0	0	0	1000
	planetlab1.dtc.umn.edu	pl-node-1.csl.sri.com	98	50	460	0.55%	62	80	3	9	986
	planetlab1.uta.edu	planetlab1.cs.unc.edu	27	27	27	0.01%	1	2	0	0	999
	planetlab6.csee.usf.edu	roam2.cs.ou.edu	24	18	411	0.06%	9	113	1	1	946
	planetlab-2.calpoly-netlab.net	planetlab-1.elisa.cpsc.ucalgary.ca	36	25	544	0.04%	4	5	2	2	995
	planetlab2.mnlab.cti.depaul.edu	roam2.cs.ou.edu	16	15	69	0.03%	4	77	0	0	966
	planetlab2.netlab.uky.edu	planetslug4.cse.ucsc.edu	40	40	42	0.00%	0	0	0	0	1000
	planetlab5.cs.cornell.edu	planetlab2.netlab.uky.edu	31	20	521	0.06%	6	21	3	9	987
	planetslug4.cse.ucsc.edu	planetlab2.netlab.uky.edu	68	43	690	0.06%	6	19	3	9	978
	roam2.cs.ou.edu	planetlab6.csee.usf.edu	29	29	29	0.00%	0	0	0	0	1000
Inter-national	metis.mcs.suffolk.edu	planetlab1.eurecom.fr	56	50	283	1.30%	166	2450	3	16	550
	planetlab-01.ece.uprm.edu	pl2.zju.edu.cn	140	135	210	0.08%	9	184	1	1	726
	planetlab1.ecs.vuw.ac.nz	planetlab-2.calpoly-netlab.net	102	76	797	0.10%	10	65	5	31	978
	planetlab-1.elisa.cpsc.ucalgary.ca	planetlab-2.calpoly-netlab.net	32	16	612	0.04%	4	8	2	4	996
	planetlab1.georgetown.edu	planetlab-1.cs.auckland.ac.nz	106	101	627	0.04%	4	10	2	5	994
	planetlab-2.calpoly-netlab.net	planetlab1.ecs.vuw.ac.nz	86	75	779	0.04%	4	14	2	5	994
	planetlab2.pop-mg.rnp.br	node2.planetlab.albany.edu	109	101	730	0.11%	11	211	1	2	709
	planetlab2.pop-pa.rnp.br	planetlab4.tamu.edu	151	150	173	0.09%	9	168	0	0	667
	planetlab-2.ssvl.kth.se	pl02.comp.polyu.edu.hk	166	166	184	0.04%	5	10	0	0	990
	planetlab4.tamu.edu	planetlab2.pop-pa.rnp.br	152	149	283	0.02%	2	2	1	1	998
	planetlab5.cs.cornell.edu	planetlab-node1.it-sudparis.eu	54	54	71	0.03%	5	34	1	1	986
	planetlabone.ccs.neu.edu	csplanetlab4.kaist.ac.kr	102	99	280	0.04%	5	16	1	2	995
	planetslug4.cse.ucsc.edu	planetlab-node1.it-sudparis.eu	112	102	138	0.27%	16	312	1	3	733



(a) Throughput

(b) Loss Rate

Figure 3.12: An illustration of the ability of the network-control layer in maintaining stability under complex conditions where the underlying requirements of congestion avoidance and prediction-based redundancy are not always satisfied. We test the underlying requirements before using these methods. These figures show that we achieve a higher throughput than the method (compare (a) with Figure 3.11a). We keep high throughput when the loss rate does not significantly increase when the transmission rate increases, as we know that the congestion is not caused by us per the test. We also always enable FEC, as (d) shows there is a weak correlation for prediction. This help us reduce the loss rate to below 2% in most time.



(c) p-value of Stationarity

(d) Correlation Coefficient

Figure 3.12: An illustration of the ability of the network-control layer in maintaining stability under complex conditions with tests of requirements. (cont.)

the algorithm does not work. In that case, we transmit the data at a fixed rate. The throughput shows that our decision is correct, as our throughput is higher than that with congestion avoidance algorithm always running. Figure 3.12b shows the network loss rate with our method. As depicted in Figures 3.12c and 3.12d, the loss rate in this link is stationary but has not correlation; therefore, we do not use the prediction-based redundancy method. We save the bandwidth to transmit more data in the link, and the higher throughput in our case proves that our decision is correct.

3.6 Summary

In this chapter we have presented a general framework in the network-control layer for providing a network environment that can satisfy the requirements of the application layer. We have discussed previous methods that can be combined in this framework and the test of the criteria for determining whether they can be used. Our simulation with a large data set collected from a world-wide overlay network proves the advantage of the proposed framework.

□ **End of chapter.**

Chapter 4

Offline Mapping of Controls to Perceptual Quality

We need a complete mapping from one or more system controls to the corresponding perceptual quality for tuning multimedia applications at run time, because perceptual quality cannot be easily measured in real time. As mentioned in Chapter 2, even attaining such mapping requires a time-consuming subjective test that cannot be performed at run time. Therefore, we need to collect the mapping beforehand through offline subjective tests and then generalize them later.

In this chapter, our goal is to find a complete mapping from one or more controls to perceptual quality with a finite and reasonable number of offline subjective tests, and to further use them to generalize to the complete mapping for all combinations of controls not tested under a given network condition. This goal can be further divided in two cases, one with a single system control, and the other with multiple system controls. They can be represented in a function with respectively a single parameter or multiple parameters:

f_{single} : *Control 1* $\rightarrow$ *Perceptual Quality under a given condition.*

f : *Control 1, Control 2, . . . , Control N* → *Perceptual Quality under a given network condition.*

Because the complete mapping is continuous and can have multiple dimensions, it is impossible to measure all the mappings over it. Instead, we need to sample representative points over the mapping (probably without a closed-form function) and then reconstruct it by generalizing the few samples to all the combinations of possible values of controls. Our *research problem* is, to schedule the offline subjective test so that we can collect only a few representative points of the mapping that is tested under expensive full tests while generalizing them to all the combinations of controls correctly .

To simplify the discussion, in this section, we assume the network condition is given. Without explicit specification, all the network condition in this chapter is the perceived network condition that is provided by the network-control layer (Chapter 3). We leave the study of generalizing the mapping to real network conditions with multiple network metrics in Chapter 5.

In this chapter, we first present the overview of the problem in Section 4.1. After that, we study the general property of JND surface in Sections 4.2 and 4.3. We then discuss the above two problems one by one in Sections 4.4 and 4.4.

4.1 Overview

Finding the representative points over the complete mapping to test is non-trivial. Without a strong assumption regarding the model of the mapping, it is not easy to estimate the error between the approximation constructed by the representative points and the real mapping. As a major innovation of this thesis, we have found a *dominance property* that can characterize mappings from system controls to perceptual quality. This property exists in the form of a *monotonicity relation* that allows the result of one subjective test to subsume the results of many other subjective tests. This means that those subjective tests whose results are subsumed do not have to be tested because their results would always be no better than the result of the dominating subjective test. The property will tremendously reduce the number of subjective tests used for finding the complete mapping from controls to perceptual quality.

By utilizing this dominance relation, we propose to represent the mappings from one control to perceptual quality by a JND surface, a graphical representation of a function without closed form for storing the results of subjective tests in a compact way. We further develop a systematic methodology for generating this JND surface using a few subjective tests and with prescribed approximation error.

When the dimension of the system control increases, we face the curse of dimensionality in the offline subjective test, as the number of tests will increase exponentially. With another novel *dominance property*, we can handle the mapping from multiple controls to perceptual quality. We call this property the human-focus property in which users will focus on the dominant change of the control in a fast-paced interactive system when multiple controls change simultaneously. This allows us to measure the combined perceptual quality by measuring the individual JND surfaces separately, which tremendously reduces the number of subjective tests. The result is a complete mapping from the space of one or more controls and given perceived network condition to perceptual quality.

In the following sections, we first define what a just-noticeable difference surface is and how it can help represent a complete mapping from system controls to perceptual quality. Then we present how we can efficiently attain the complete mapping from one system control to perceptual quality with offline subjective tests using the first dominance property (i.e. monotonicity). Finally, we present the way we attain the mapping from multiple system controls to perceptual quality with the second dominance property (i.e. human focus).

We summary the contents in each sections in Table 4.1.

4.2 Mapping from Controls to Perceptual Quality with JND

We have presented the relation between pairwise comparisons and JND in Chapter 2. According to the just-noticeable-difference (JND) concept in psychophysics, not every control-quality mapping will need to be tested ahead of time because humans can only discern sufficiently large changes in perceptual quality when control inputs are changed. To make the generation of such mappings tractable, previous work often uses some simplifying (though incorrect in general) linearity assumption between JND and a single reference control input.

In optimizing the perceptual quality of real-time interactive multimedia systems, we need to find the function from controls to perceptual quality. Unfortunately, such a function may not have a closed-form representation because human perception cannot be easily mod-

Table 4.1: Summary of the problems studied in each section in Chapter 4.

Problem	Assumptions	Major Results	Limitations	Section
How to represent a complete mapping that can directly relate to the assessment of perceptual quality gained from the subjective tests?	1. Unbiased and synchronized tests. 2. Uniform level of expertise. 3. Awareness is continuous but not necessary smooth.	JND surface model.	Not suitable for non-IID responses.	Section 4.2
How to efficiently perform the subjective tests with only a few tests in general?		Dominance properties.		Section 4.3
How to efficiently perform the subjective tests with only a few tests for a single control?	Single control.	Efficient algorithm for performing subjective tests with low error.	Need other method for measuring mapping of multiple controls.	Section 4.4
How to efficiently perform the subjective tests with only a few tests for multiple control?	Multiple dependent controls.	Method for decomposing high-dimensional subjective opinions into individual surfaces.	Cannot handle multiple independent controls.	Section 4.5

eled. We need to develop a way to represent a complete mapping that can directly relate to the assessment of perceptual quality gained from subjective tests.

In this section, we relax previous work's assumptions and present a probabilistic function with multiple arguments and its graphical representation, the JND surface, that relates system controls to JND as well as awareness (probability that subjects can notice the difference). The theory leads to efficient algorithms for generating offline the complete mapping between system control values and the corresponding perceptual quality using a few subjective tests.

A JND surface includes subjective opinions (in the form of pairwise comparisons) on all pairs of multi-dimensional controls in the control space under the same network condition. The mapping is shown below:

JND Surface $\leftrightarrow$ f : *Control 1, Control 2,* $\ldots$*, Control N* $\rightarrow$ *Perceptual Quality under a given Network Condition.*

With a JND surface, system controls can be mapped to awareness, and awareness can be converted to perceptual quality. In this way, a multi-dimensional JND surface acts as a bridge between the controls and the perceptual quality under a given network condition.

4.2.1 Formal Definitions

To clarify the discussions, we provide the basic definitions for the concepts that are used in this chapter [107–109].

Definition 4.2.1. Pairwise Subjective Tests of Perceptual Quality*: In a subjective test, multiple subjects are asked to answer a question posed with respect to two scenarios under different* control assignments *(settings of the control parameters), one using the original setting (as reference), the other using a modified setting (as reference + modification). The scenarios using these settings are presented in a random order. Each subject can choose one of the alternatives, or answer undecided.*

The reference and modification are defined follows:

Definition 4.2.2. Reference ($\vec{r}$**)** *is the vector of original setting of controls.*

Definition 4.2.3. Modification ($\vec{m}$**)** *is the vector of changes corresponding to the respective controls in* $\vec{r}$*.*

To allow meaningful collection of the statistics from subjective tests, we make similar assumptions as in previous works:

Assumption 4.2.1. Unbiased and synchronized tests. *Under given* $\vec{r}$ *and* $\vec{m}$*, each subject gives one assessment. All subjects then complete their assessments before proceeding to another set of tests with different* r *and* m *.*

Asking each subject to carry out one subjective assessment avoids any bias in repeated tests. Synchronizing all the tests allows the result of one set of tests to guide the selection of the next $\vec{r}$ and $\vec{m}$ for testing.

Assumption 4.2.2. Uniform level of expertise. *All subjects have the same level of expertise, and their ability to perceive differences between two sets of* $\vec{r}$ *and* $\vec{m}$ *is independent and identically distributed (IID).*

This assumption allows responses from multiple subjects to be evaluated statistically. As a result, $\hat{A}$ will approach p as more subjects are involved.

According to the way the pairwise comparison is conducted, we have two types of statistics for the results:

Definition 4.2.4. Sample awareness $\hat{A}$ *is the fraction of subjects that can perceive the changes during pairwise subjective tests. Each pairwise subjective test presents the outputs of the system using the reference and the modified system controls in a* ***random*** *order. An undecided subject is assumed to contribute 0.5 to the sample awareness.*

Definition 4.2.5. Sample noticeability $\hat{A}_{notice}$ *is the fraction of subjects that can perceive the changes during a pairwise subjective test. The pairwise subjective test presents the outputs of the system using the reference and the modified system controls in a* ***fixed*** *order. An undecided subject is assumed to contribute 0.5 to the sample noticeability.*

The difference between the two statistics is that the former is calculated from tests in which subjects need to guess the order of the alternatives, while the latter is calculated from tests in which subjects have been told the order. With many subjects, we can achieve the asymptotic values of these two statistics:

Definition 4.2.6. Awareness p *is the asymptotic value of* $\hat{A}$ *when the number of subjects is large.*

Regarding awareness, we further address its difficulty in measurement with the following assumption:

Assumption 4.2.3. Non-smoothness. *The awareness over the surface is continuous but not necessary smooth.*

Definition 4.2.7. Noticeability P_{notice} *is the asymptotic value of* $\hat{A}_{notice}$ *when the number of subjects is large.*

Note that awareness p by itself is larger than the probability of subjects who can *actually* notice a change, as there are subjects who give the correct answer by random guesses. Without random guesses, there are $P_{\text{notice}} = 2p - 1$ (defined as the *probability of noticeability*) of those subjects who can actually notice the change. Assuming the responses of subjects to be independent and identically distributed, $P_{\text{notice}}N$ of the N subjects can actually notice the change, while $(1-P_{\text{notice}})N$ answer by random guesses. Hence, $P_{\text{notice}}N \times 1 + (1-P_{\text{notice}})N \times 0.5 = pN \Rightarrow P_{\text{notice}} = 2p - 1$ [108]. When the modified setting is the same as the original ($M = 0$), 0% of the subjects can notice the change and everyone responds by random guesses; hence, $p = 50\%$ ($\Rightarrow P_{\text{notice}} = 0$). In psychophysics, $p = 75\%$ ($\Rightarrow P_{\text{notice}} = 0.5$) is generally used as a level of noticeable change. In short, p and P_{notice} are equivalent representations of noticeability.

Now we can define the JND surface as follows:

Definition 4.2.8. JND surface $p(\vec{r}, \vec{m})$ *is a function* p *that maps reference* $\vec{r} \in R^n$ *and modification* $\vec{m} \in R^n$ *to awareness* $p \in [0.5, 1]$.

In practice, interesting regions in a JND surface are specified beforehand, and $\vec{r}$ and $\vec{m}$ are first normalized to $[0, 1]^n$. According to Axiom 4.3.1, when $\vec{m}$ is given, larger $\vec{r}$ will result in a smaller p; when $\vec{r}$ is given, larger $\vec{m}$ will result in larger p.

A JND surface is stored as an $K_1 \times K_2$ 2-dimensional array, where K_1 and K_2 are the discretization levels of the x and y axis respectively. The discretization levels are set according to the practical control of the application. For example, in an online game, 1 frame is the smallest unit of the video presented to the players. In a fast-paced game with 60 fps video, action duration of less than 1000 ms can be modeled by K_1 and K_2 equal to 60 frame. Storing this data can be done in less than 200 KB memory, which is insignificant in modern desktop computers.

A JND surface provides a complete mapping from multi-dimensional controls and given network condition to perceptual quality. All the possible multi-dimensional controls in the control space are spanned in the axes of references, which consist of the vector of controls $\vec{r}$. The perceptual quality is represented in the form of results of pairwise comparisons. The color in the JND surface shows subjective awareness p of the changes of controls from the reference vector $\vec{r}$ to the modified control vector $\vec{r}+\vec{m}$ under the same network condition, where $\vec{m}$ is the vector of modification. When the modification is too large, the change is completely noticeable, and subjects will always choose the better one. Therefore, $\vec{m}$ only needs to cover those modifications that are below the complete noticeable threshold.

4.2.2 Illustration

Single control

To illustrate how we map a single system control to perceptual quality with a JND surface, consider a fast-pace two-player multimedia game studied in Section 2 [107]. In this game, A can hit B from ei-

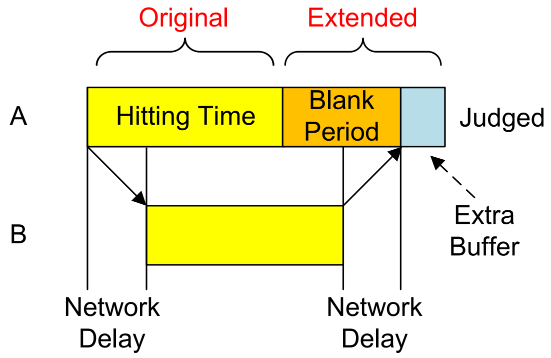


Figure 4.1: The hitting time in Player A's view should be extended sufficiently in order to cover network delay. This extension may be noticed by player A.

ther the upper or the lower side, and B can defend against the attack by moving a paddle up or down.

The system control in this game is the delay between making an action by one user and seeing the response by the other. Due to network delays, A will see a *blank period* when A's hitting time is the same as that of B's; that is, at the end of A's hitting time, B's response has not arrived in A's view because A's action will need to travel to B and B's response to travel to A.

To allow A's action to be synchronized with B's response, Figure 4.1 illustrates the use of a local perception filter [85] to extend A's hitting time by an amount to cover the round-trip network delay. Additional jitter buffer delays may also be included to smooth delay variations.

Note that a longer hitting time can provide more room for receiving a response, although the extension may be noticeable if it is too long. Our goal is to determine the maximum extension for motions with different hitting times such that the modification is barely noticeable.

Figure 4.2 shows the test results in the graphical representation

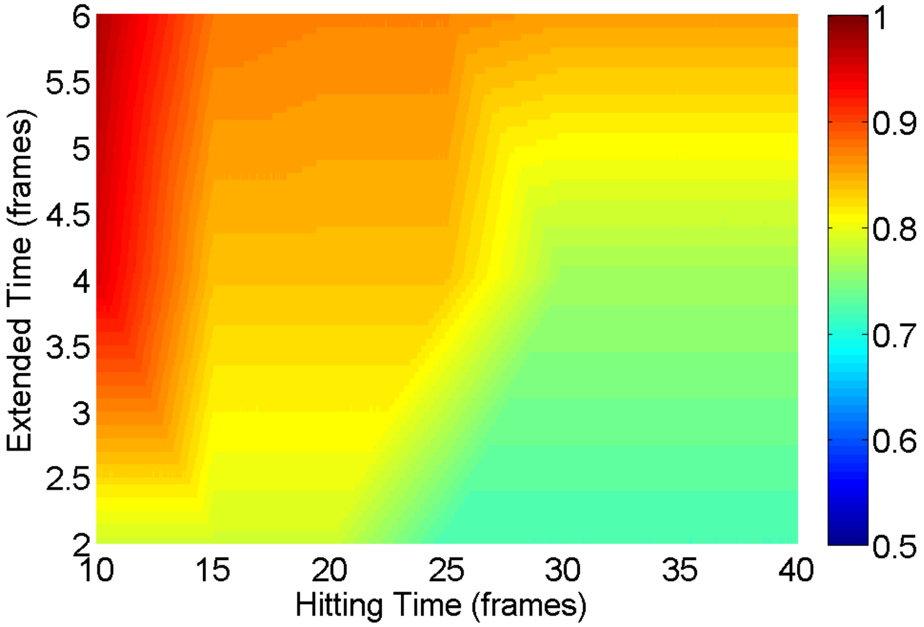


Figure 4.2: A JND surface for a 2-player game that shows the relation among the probability of being noticed (awareness, denoted by the color surface), the modification (JND in the y-axis), and the control assignment (x-axis). Since the game runs in 60 fps, both the original hitting and the extended times in the figure are in unit of $\frac{1}{60}$ sec.

of *JND surface* (in Section 4.2.1), where each point shows the measured awareness or probability that A's extended hitting time can be detected by 14 subjects. It shows that a longer hitting time is less likely to be noticed, and that a motion with a larger modification is more likely to be noticed.

The graphical representation allows us to determine the best system control (buffering time) when given the hitting time of a certain motion. For example, when the hitting time is 25 frame-times and the target awareness is less than 75%, the extension should be no more than 3 frame-times.

As another example, Figure 4.3 illustrates the use of a JND sur-

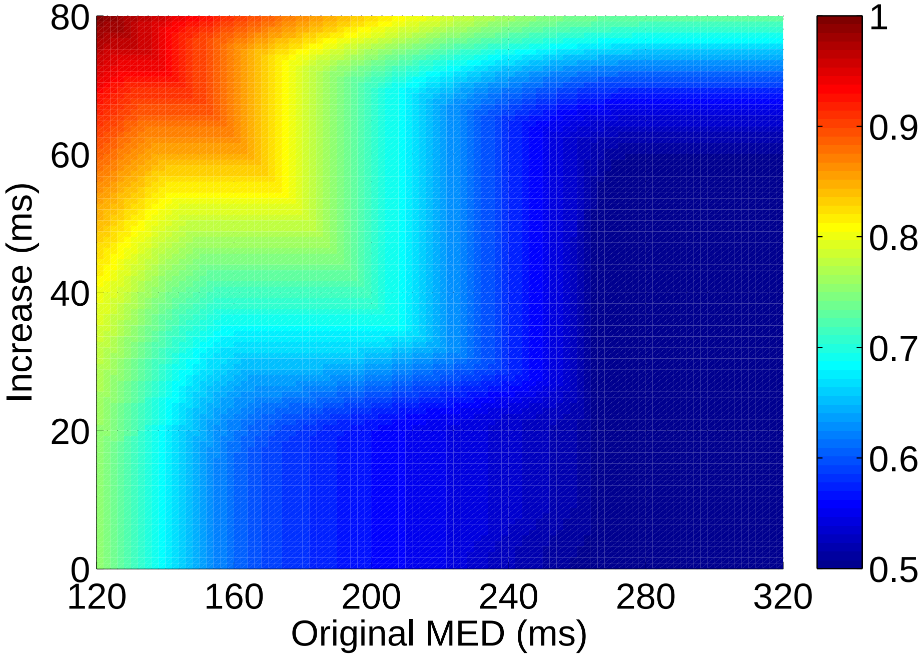


Figure 4.3: A JND surface for a voice-over-IP application with a lossy connection. Awareness is improved when MED is increased. Note that awareness is continuous but non-smooth (shown in the step pattern) due to the discrete buffering time for receiving parity packets.

face in a voice-over-IP system. The system is controlled by a parameter called mouth-to-ear delay (MED), which includes network latency, buffering time, and device latency. By increasing MED under the same network and device condition, the buffering time becomes longer, thereby allowing more parity packets to be received. This can improve the signal quality in a connection with packet losses.

Multiple controls

Following the example of the fast-pace two-player multimedia game [107], we illustrate how multiple system controls are mapped to perceptual quality in a JND surface. In the previous single-control case, we only consider how extending the hitting time can be noticed by users. Actually, besides extending the hitting time, we can also shorten the hitting time, or even delaying the start of an action, to cover the blank period. Therefore we have three system controls that can be changed simultaneously.

Figure 4.5 presents the JND surface with these settings. Rather than showing a surface with higher dimension which is difficult to understand, we show the JND surfaces with one control changing and the other controls fixed. It is noticed that the surfaces still follow monotonicity.

4.3 Dominance Properties

Unlike analytic models studied in psychophysics, our study is based on axioms and assumptions presented in this section that lead to useful properties and algorithms for finding JND surfaces. Their

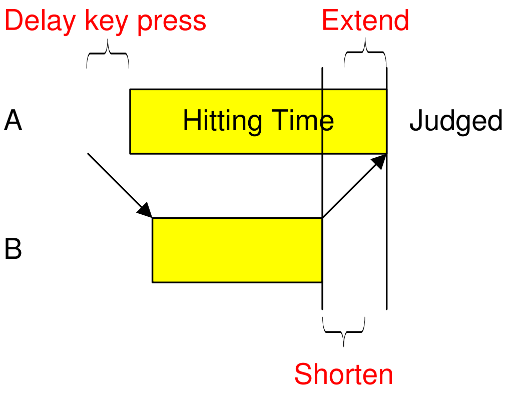
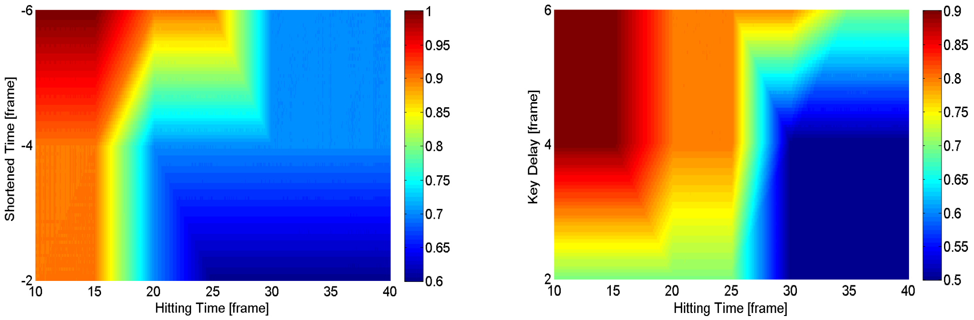


Figure 4.4: The blank period can be concealed by 3 methods simultaneously: extending/shortening the hitting time, and delaying the start of an action. Each method is controlled by one system control that prescribe its extent. In total, we have 3 system controls.



(a) Changing the m_{shorten} dimension

(b) Changing the m_{delay} dimension

Figure 4.5: The JND surface of the multiple-control case. Because showing a surface higher than 3-D is difficult, we instead show the surface when only one of the three dimensions is changed and the other two are fixed. a) Only the hitting time is shorten. b) Only the start of an action is delayed.

advantage is that they are generally applicable to many multimedia applications.

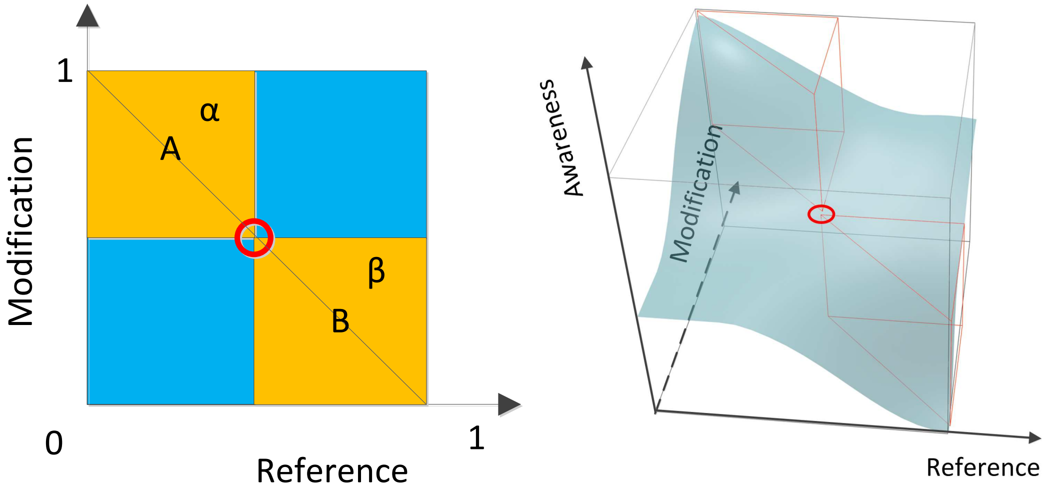
We first present two novel properties that can trim the search space of representative test points and tremendously reduce the number of subjective tests.

4.3.1 Monotonicity

The first dominance property we have found is a relation that allows us to prune the search space when scheduling subjective tests. The axiom is obtained from the following two intuitions. First, A larger modification to a control can result in more significant changes. Second, when the modification is small with respect to the control, the change is less significant. Based on these understandings, we have the axiom of monotonicity:

Axiom 4.3.1. ***Dominance Property 1: Monotonicity.*** *(a) $p(r, m)$ is monotonically non-increasing with respect to reference r, as a given modification m is less noticeable under a larger r. (b) $p(r, m)$ is monotonically non-decreasing with respect to m, as a larger m is more noticeable under a given r. (c) For multi-dimensional modifications, $p(r_1, r_2, \ldots, r_n,\ m_1, m_2, \ldots, m_n)$ is monotonically non-decreasing when all $m_i, i = 1, \ldots, m$, are non-decreasing.*

With this axiom, we can derive theorems that can describe how one subjective test result can dominate results of some other subjective tests, what results of subjective tests cannot be dominated, and how we can utilize these properties to prune the JND surface into halves and conduct the subjective tests in a divide-and-conquer way.



(a) Top view

(b) Side view

Figure 4.6: (a) By observing the JND surface from the top, a test point (in red) divides the surface into Regions A and B. The awareness of A (*resp.*, B) is larger (*resp.*, smaller) than that of the point. (b) In the side view of the surface, the two outlined red prisms correspond to the yellow regions marked by α and β in (a).

Important lemmas can be derived from this axiom. We use the figure of a JND surface to help illustrate these lemmas. Figure 4.6 shows the top and the side views of a JND surface. The awareness of a test point (in red) is known after a subjective test. For simplicity, the axes are all normalized into the $[0, 1]$ range.

Because depicting points in 3D is not easy, we depict them in the top view. For example, points to the top-left of the red point in Figure 4.6a refer to points with smaller reference and larger modification (Region A). Similarly, points to the bottom-right of the red point are in Region B. Because a JND surface is a function of reference and modification, a reference in the top view can uniquely locate a point in the surface.

The concept of top-left and bottom-right can be generalized to the case with multi-dimensional controls. We refer to points with smaller references and larger modifications than the red point as the top-left points, and points with larger references and small modifications as the bottom-right points.

Definition 4.3.1. *Point B is to the* **top-left** *of point A if and only $r_{i,B} < r_{i,A}$ for $i = 1 \ldots N$ and $m_{i,B} > m_{i,A}$ for $i = 1 \ldots N$. Point B is to the* **bottom-right** *of point A if and only $r_{i,B} > r_{i,A}$ for $i = 1 \ldots N$ and $m_{i,B} < m_{i,A}$ for $i = 1 \ldots N$.*

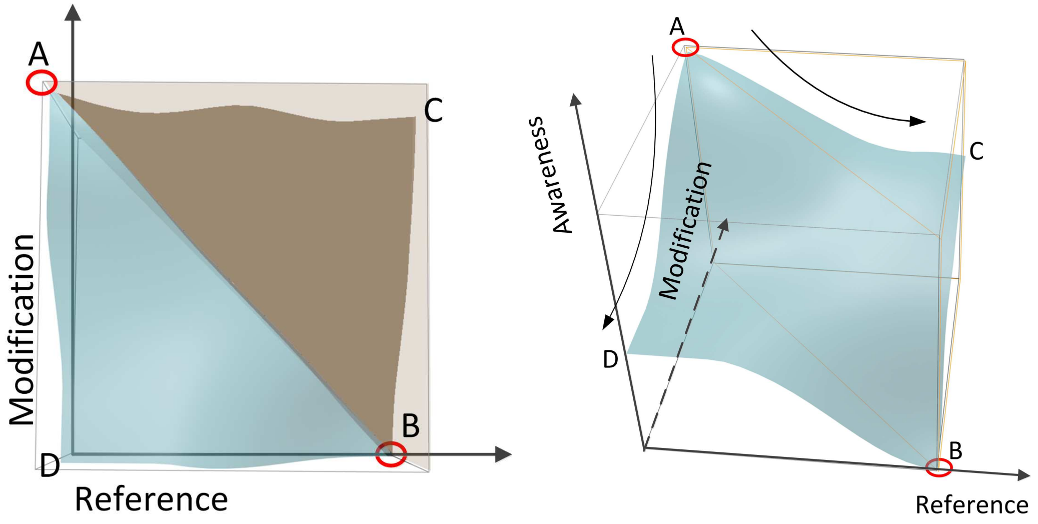
Now we present the lemmas as follows.

Lemma 4.3.1. *The awareness of a point indicates the lower (*resp.*, upper) bound of the awareness of points to the top-left (*resp.*, bottom-right) of this point.*

Proof. From the monotonicity property in Axiom 4.3.1, the awareness of points to the top-left (like Region A in Figure 4.6a) of a test point is larger than its own awareness, and the awareness of points to the bottom-right (like Region B) is smaller. □

Lemma 4.3.2. *A pair of top-left and bottom-right test points prescribe the bounding right triangular prism of all possible JND surfaces that pass through these points.*

Proof. Figure 4.7 shows the top and side views of a bounding right triangular prism. Referring to Figure 4.7a, the brown region is right-triangular due to the directions of monotonicity. Due to monotonicity, all points in the brown right triangular prism should have awareness between the awareness of A and B. Figure 4.7b further shows that half of the JND surface should be inside the corresponding right



(a) Top view

(b) Side view

Figure 4.7: Awareness of A and B defines the range of awareness of all points in the brown prism.

triangular prism. For convenience, we call C the right-angle corner point. Note that with multi-dimensional controls, a top-left point has references always smaller than a bottom-right point, and has modifications always larger than a bottom-right point. □

From Lemmas 4.3.1 and 4.3.2, the height of the top-left and bottom-right triangular prisms in Figure 4.6b shows the range of awareness of any JND surface that passes through these two prisms. The following lemma shows the property of the other two corner points.

Lemma 4.3.3. *The top-right and bottom-left test points (C and D in Figure 4.7b) cannot bound the awareness of JND surfaces passing through the right triangular prism.*

Proof. The lemma is true because monotonicity does not apply in the top-right to bottom-left direction. □

These lemmas provide a method to trim the test space and reduce the number of offline subjective tests. We know that every two test points, one at the top-left and the other at the top-right, can indicate the range of the awareness within the rectangle region bounded by them. Therefore, the awareness inside the region can be estimated without further measurement. Intuitively, as long as we properly arrange the test points over the JND surface, the unknown region can be minimized; therefore, the real JND surface can be reconstructed precisely. This result is formally proved in Section 4.4.2.

4.3.2 Human Focus

The second dominance property we have discovered implies a basic property about what human will focus on in fast-paced interactive multimedia applications.

When multiple system controls are applied together, their effects are perceived by humans as a whole. Some previous studies proposed to combine their effects on JND by taking the maximum JND. They argued that humans tended to notice the most significant change when there were multiple changes [17]. This is not always true in practice as p of a large change in one control may not be larger than p of a smaller change in another control. Note that the chance of perceiving a change is based on p, not on the magnitude of the change. Some other methods used square-root to calculate the combined JND [93]. An integration was also proposed when an explicit JND function was given [23].

All the previous methods considered the combined changes in determining their overall effect. Our analysis, however, shows that

p is a true indication on whether a subject can (probabilistically and actually) detect a change. Hence, a better approach to determine the overall effect is to utilize p. By assuming that subjects will notice the change with the maximum p when there are multiple changes, we propose to decompose the evaluation of a multi-dimensional p into the evaluation of individual p's, each corresponding to one control assignment.

Axiom 4.3.2. ***Dominance Property 2: Human Focus.*** *In fast-paced interactions, human will focus on the change with the highest p when multiple dependent system controls change simultaneously: $p_{comb} = \max_{i=1...N}\{p_i\}$.*

This axiom is true because when changes happen in very short periods for human to identify them one by one, human have to only grasp the most significant change. In an extreme case, when the multiple changes are not very different from each other, then what humans will perceive will be just a composite of all the changes but not the change with the highest p. However, in this case, the axiom still applies because the change with the highest p will still be very close to the composite effect of all the changes.

4.4 Mapping from Single Control to Perceptual Quality

We firstly discuss a simplified case in which there is only one control that needs to be mapped to perceptual quality. Our goal in this section is to find a complete mapping from one control to perceptual quality with a finite and reasonable number of offline subjective tests under a given perceived network condition.

4.4.1 Approximating a JND Surface

A JND surface without error in measurement cannot be obtained, because there are infinite pairs of system controls. The best we can do is to approximate the JND surface by the least approximation error.

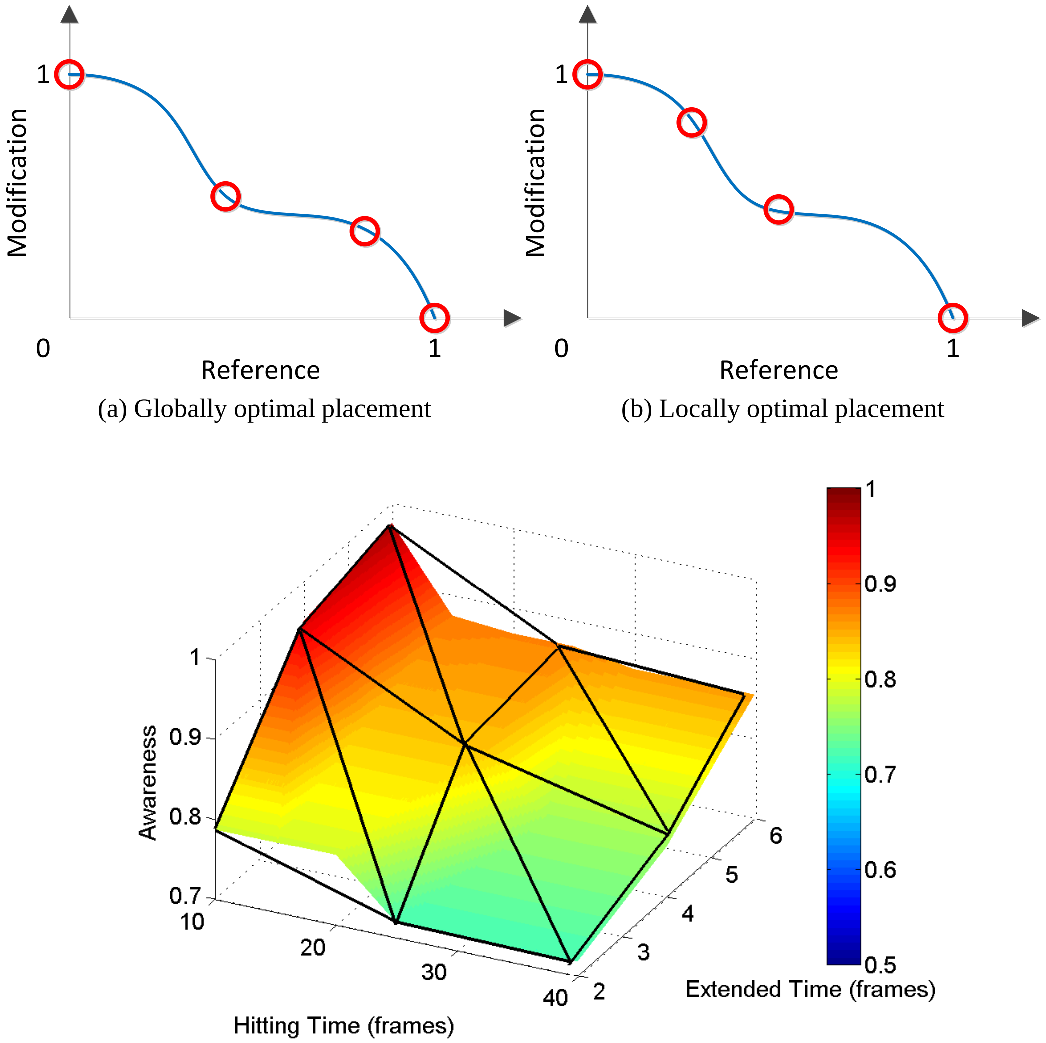
To estimate a JND surface with given ranges of r and m, we like to schedule a sequence of subjective tests in order to minimize the error of the estimated surface.

Consider Figure 4.8a in which test points are selected to approximate the curve using piecewise linear interpolations. By defining the error as the area between the original and the piecewise curves, the problem becomes the selection of the test points to minimize the error. Since we have no prior knowledge of the curve, we can use the result of one test to determine the next test point.

In the 3-D case, the surface in Figure 4.8c can be approximated by triangular linear approximation, where vertices are test points. The reason why it is not approximated by a global surface function is because the surface may be non-smooth (Assumption 4.2.3). Our goal is to place the test points in order to minimize the volume between the original surface and the linear approximation.

Next, we define an approximation mesh of a JND surface and our objective function.

Definition 4.4.1. Approximation mesh. *An approximation mesh of a JND surface is a triangular mesh whose vertices are test points*



(a) Globally optimal placement

(b) Locally optimal placement

(c) Triangular linear approximation of a 3-D surface

Figure 4.8: The best placement of test points depends on the information available on the curve.

corresponding to results obtained by subjective tests.

Let $A = p(r, m)$ *be a JND surface and* $f(r, m)$ *be the triangular approximation of* $p(r, m)$*:*

$$\begin{gathered} f_{\mathcal{P}}(r, m) = \ell(p(r_{A}, m_{A}), p(r_{B}, m_{B}), p(r_{C}, m_{C})) \\ \forall\, Tri(A, B, C) \text{ where } A, B \text{ and } C \in \mathcal{P}. \end{gathered} \tag{4.1}$$

Here ℓ *is a linear function that interpolates a plane in a triangular region* Tri *defined by* A, B *and* C *that belong to the set of test points* $\mathcal{P}$*. The set of all* $f_{\mathcal{P}}(r, m)$ *is the approximation mesh of the original JND surface.*

Definition 4.4.2. Absolute error $\mathcal{E}$ *is the volume between the JND surface and the approximation mesh. Given* $f_{\mathcal{P}}(r, m)$ *in (4.1),* $\mathcal{E}$ *is determined by* $\mathcal{P}$ *as follows:*

$$\mathcal{E} = \int_{0}^{1} \int_{0}^{1} |\, p(r, m) - f_{\mathcal{P}}(r, m) \,|\, dr\, dm \ \ \forall\, r, m. \tag{4.2}$$

To minimize $\mathcal{E}$*, we need to optimally place the test points* $P_1, P_2, \ldots,$ *and* P_k*, where* k *is the number of tests performed by each subject.*

In practice, only a finite number of points in a JND surface can be sampled. After sampling the awareness at A, B, and C, there are infinitely many JND surfaces passing through them that satisfy monotonicity in Axiom 4.3.1. Figure 4.9 shows that the absolute error cannot be uniquely specified because the original nonlinear surface passing through A, B, and C is not unique. (Figure 4.9 shows 3 of these surfaces.) Further, Point C is not fixed because it can be anywhere along the height of the prism and still satisfies monotonicity. For this reason, we define the average absolute error as follows.

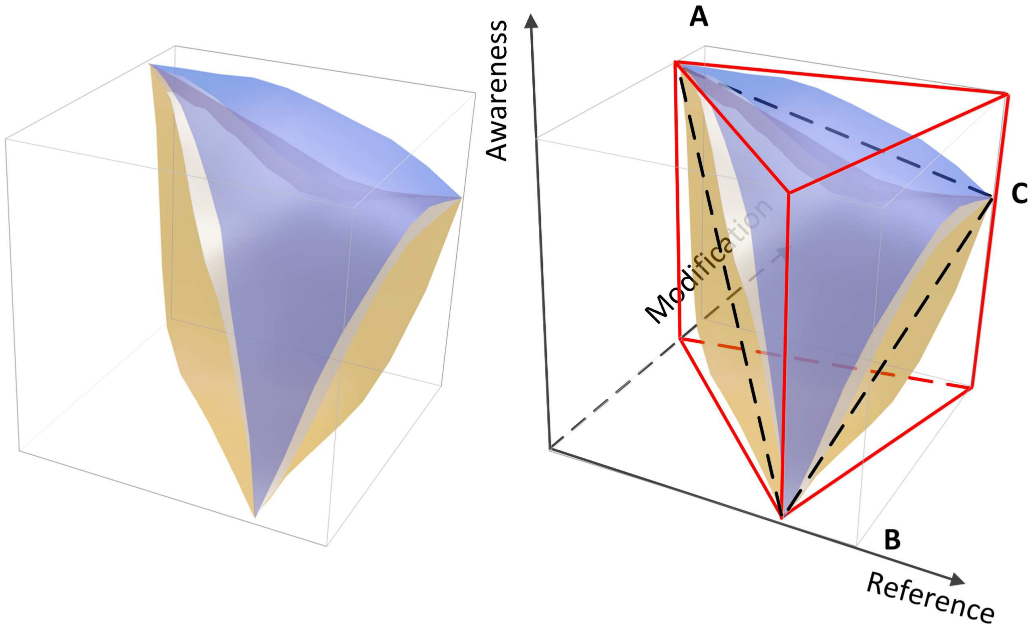


Figure 4.9: Three possible JND surfaces (blue, white, and yellow) passing through A, B, and C whose approximation mesh is denoted by black dashed lines. The awareness of these surfaces is bounded by the height of the triangular prism (outlined in red).

Definition 4.4.3. Average absolute error *inside a prism is the average of the absolute errors between all the approximation meshes and the corresponding JND surfaces that pass through the highest point A and the lowest point B and that satisfy monotonicity in Axiom 4.3.1.*

Since the 3-D JND surface can be decomposed into triangular prisms, the average absolute error of the surface is the sum of the average absolute errors of the component prisms.

4.4.2 Minimizing the Average Absolute Error

Based on the axiom and assumptions stated earlier, we aim to approximate a JND surface using a triangular mesh generated from a number of test points that are properly placed in order to minimize the sum of the average absolute errors.

Because we do not assume any shape of the JND surface, we will not be able to calculate the average absolute error in closed form. To this end, we use the volume of the bounding prism as an approximation of the average absolute error. Also because of the non-smoothness of the JND surface (Assumption 4.2.3), the surface is approximated by a triangular mesh instead of by a global surface function.

Since awareness of a point is unknown before a subjective test is performed and the model to predict future states is very complex, our approach is to determine the test points sequentially after the current state has been known. For example, the best placements of test points in Figure 4.8a are at places where the curve changes direction rapidly. Without this information, one can only determine the best point to test next based on information obtained in the current test, such as those in Figure 4.8b.

Note that formulating a model to predict future states is difficult because parameters in perceptual modeling are not quantified. Further, the monotonicity property in Axiom 4.3.1 is too weak to support a precise model.

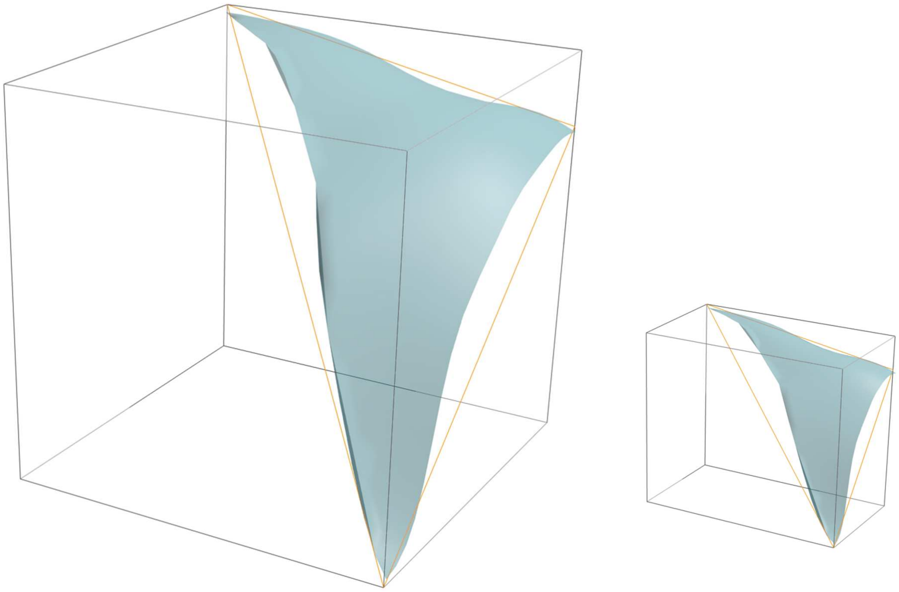
In the next several sections, we estimate the average absolute error, based on the monotonicity property of a JND surface. We then present the method for sequentially selecting test points based on the average absolute errors found.

With the axiom and lemmas described in Section 4.3.1, we show in the following corollary that the best triangulation of the JND surface is when every triangle is a right-angle triangle when *projected* to the reference-modification plane:

Corollary 4.4.1. *The triangulation of a JND surface, when projected onto the reference-modification plane, should be right-angle triangles.*

Proof. With Lemma 4.3.2, it is known that points A and B prescribe the upper and lower bounds of the awareness of points. Because the directions of monotonicity are from left to right and from top to bottom in the projection, the boundary of the projection should be along these directions. Therefore, each projected region should be a right-angle triangle. □

Theorem 4.4.1. *The average absolute error inside a bounding right-angle triangular prism is proportional to the volume of this prism.*



(a) Original

(b) Resized

Figure 4.10: A JND surface and its approximation mesh inside (a) the original right-angle triangular prism; (b) the resized right-angle triangular prism. The transformation from (a) to (b) can be done by sacling the axes.

Proof. Since the average absolute error in a right-angle triangular prism is not easy to calculate, we approach the proof by comparing the errors of two right-angle triangular prisms. Referring to Figure 4.10, when we resize such a prism, we only change its size in the X, Y and Z directions. For every surface and approximation mesh inside this prism, we can map them to the resized prism by applying the same transformation. That is, when we map a surface and its approximation mesh from the original right-angle triangular prism to a new right-angle triangular prism, we scale reference r by α, modification m by β, and awareness p by γ. Because this transformation only changes the size of the X, Y and Z axis, the ratio between the absolute error (namely, the volume between the surface and the approximation mesh) and the volume of prism does not change.

To simplify the calculation and without loss of generality, we prove the property in the entire cube. Let $p(r, m)$ and $f(r, m)$ be the functions of a JND surface and its approximation mesh. The absolute error after transformation is:

$$\begin{aligned}
\mathcal{E}_{\text{new}} &= \int_0^\beta \int_0^\alpha \left| \gamma p\left(\frac{r}{\alpha}, \frac{m}{\beta}\right) - \gamma f_{\mathcal{P}}\left(\frac{r}{\alpha}, \frac{m}{\beta}\right) \right| dr\, dm \\
&= \alpha\beta\gamma \int_0^1 \int_0^1 \left| p\left(\frac{r}{\alpha}, \frac{m}{\beta}\right) - f_{\mathcal{P}}\left(\frac{r}{\alpha}, \frac{m}{\beta}\right) \right| d\frac{r}{\alpha}\, d\frac{m}{\beta} \\
&= \alpha\beta\gamma \int_0^1 \int_0^1 \mid p(r,m) - f_{\mathcal{P}}(r,m) \mid dr\, dm \\
&= \alpha\beta\gamma\, \mathcal{E}_{\text{original}}. \qquad (4.3)
\end{aligned}$$

The ratio to the transformed cube is:

$$\frac{\mathcal{E}_{\text{new}}}{\alpha\beta\gamma} = \alpha\beta\gamma\,\frac{\mathcal{E}_{\text{original}}}{\alpha\beta\gamma} = \frac{\mathcal{E}_{\text{original}}}{1}.$$

Since the JND surface is not unique in the cube, we consider the average absolute error when the JND surface and the corresponding approximation mesh can vary.

Let $p(r, m, \theta)$ and $f(r, m, \mu)$ be the function of the JND surface and its approximation under (unknown) parameter set θ and μ. Because θ and μ are independent of r and m, the average absolute error is:

$$\begin{aligned}
\tilde{\mathcal{E}}_{\text{new}} &= \int\int\int_0^\beta\int_0^\alpha \left|\gamma p\left(\frac{r}{\alpha}, \frac{m}{\beta}, \theta\right) - \gamma f_{\mathcal{P}}\left(\frac{r}{\alpha}, \frac{m}{\beta}, \mu\right)\right| \\
&\quad Pr(\theta, \mu)\, dr\, dm\, d\mu\, d\theta \\
&= \alpha\beta\gamma \int\int\int_0^1\int_0^1 |p(r, m) - f_{\mathcal{P}}(r, m)| \\
&\quad Pr(\theta, \mu)\, dr\, dm\, d\mu\, d\theta \\
&= \alpha\beta\gamma\, \tilde{\mathcal{E}}_{\text{original}}.
\end{aligned}$$

Therefore

$$\frac{\tilde{\mathcal{E}}_{\text{new}}}{\alpha\beta\gamma} = \alpha\beta\gamma\,\frac{\tilde{\mathcal{E}}_{\text{original}}}{\alpha\beta\gamma} = \frac{\tilde{\mathcal{E}}_{\text{original}}}{1}.$$

□

4.4.3 Selection of the Next Test Point

Based on the average absolute error estimated in Theorem 4.4.1, we discuss in this section our strategy for choosing a suitable point in the JND surface to test in the current stage that can best reduce the average absolute error. We show that the center point of a rectangular region (top view) is the best point to test in that region.

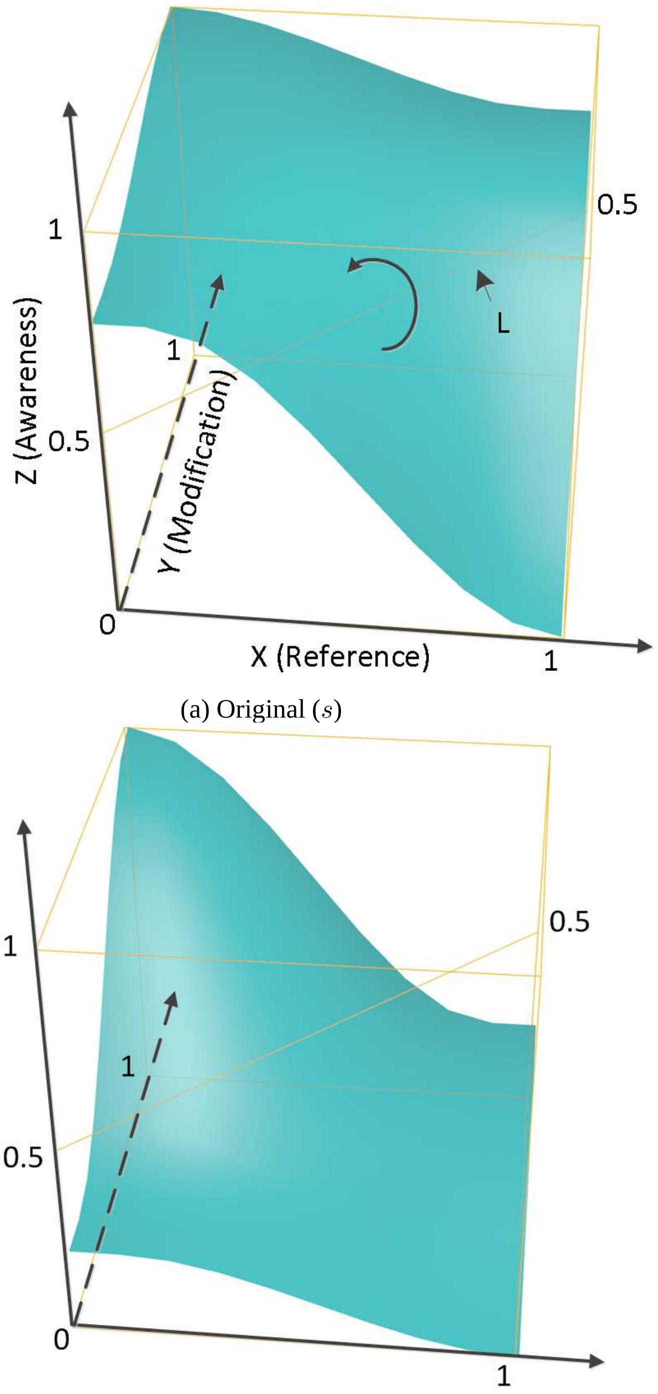
Lemmas 4.3.2 and 4.3.3 have shown that a top-left and a bottom-right test point can bound a rectangular region. Inside the region, there are infinitely many possible JND surfaces that pass through the two corner points while satisfying the monotonicity property in Axiom 4.3.1. Let $\mathcal{S}$ be the collection of these JND surfaces.

Definition 4.4.4. *$\mathcal{S}$ is the collection of JND surfaces in a rectangular region (bounded by top-left point A and bottom-right point B) that pass through both A and B while satisfying Axiom 4.3.1.*

Next, we show that the surfaces in $\mathcal{S}$ appears in pairs, and each pair have the same error with respect to the center-point of the normalized rectangular region.

Theorem 4.4.2. Symmetry. *In the normalized cube of length 1 bounded by A and B in Figure 4.11, for any JND surface $s \in \mathcal{S}$, there exists exactly one $s' \in \mathcal{S}$ that is axially symmetric to s;* i.e., *s can be rotated 180° around Line L passing through $(0, 0, 0.5)$ and $(1, 1, 0.5)$ to get to s'.*

Proof. For any (x, y, z) in the original surface s, the transformation to get to (x', y', z') can be done [7] by first transforming the whole space (with the surface and L), while keeping the X, Y and Z axes unchanged, so that L is moved to the Z axis. We then rotate the space by 180° about the Z axis. Finally, we transform the whole space so that L goes back to the original place.



(a) Original (s)

(b) Rotated 180° (s')

Figure 4.11: Any JND surface in the bounding cube has exactly one symmetric JND surface around Line L passing through $(0, 0, 0.5)$ and $(1, 1, 0.5)$ (yellow line in the middle of the cube) when rotated 180°.

1. Transform the space using transformation matrix T_p while keeping the orientation of Line L in such a way that L passes through the origin and lies on the X-Y plane.

2. Next, rotate using T_{XZ} the above space around the Z axis, and let L lie on the X-Z plane. After the rotation, L lies along the X axis.

3. Rotate using T_Z the space about the Y axis and let L lie along the Z axis.

4. Rotate by $180°$ using R_Z about the Z axis.

5. Apply the inverse transformations of Steps 3, 2, and 1, respectively. That is, rotate using T_Z^{-1} the space about the Y axis and let L lie along the X axis (the state before applying Step 3), and so on.

To perform the above transformations, the transformation matrix is:

$$\begin{aligned} T_{\text{comb}} &= T_P^{-1} T_{XZ}^{-1} T_Z^{-1} R_Z T_Z T_{XZ} T_P \\ &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \end{aligned} \tag{4.4}$$

$$(x', y', z', 1)' = T_{\text{comb}}(x, y, z, 1)' = (y, x, 1 - z, 1)' \tag{4.5}$$

Firstly, it is clear that for any $0 \leq x, y, z \leq 1$, we have $0 \leq x', y', z' \leq 1$; *i.e.*, the symmetric point is still inside the bounding cube. It is also easy to show that each point in s has exactly one corresponding point in s'.

Secondly, the symmetric JND surface s' still satisfies the monotonicity property in Axiom 4.3.1 due to (4.5):

$$\begin{aligned} x_1^{'} < x_2^{'} \text{ and } y_1^{'} > y_2^{'} &\Rightarrow y_1 < y_2 \text{ and } x_1 > x_2 \\ &\Rightarrow z_1 < z_2 \quad \text{(Axiom 4.3.1)} \\ &\Rightarrow 1 - z_1 > 1 - z_2 \\ &\Rightarrow z_1^{'} > z_2^{'}. \end{aligned}$$

□

Figure 4.11 illustrates the theorem. It is clear that after the rotation, the new surface still passes through the two corner points and satisfies monotonicity.

With Theorem 4.4.2, the center-point of the cube (from the top view) has the same error to the JND surfaces in pairs. To best approximate a JND surface without any knowledge on the shape or the distribution of the points, choosing the center-point of the region to test is the best choice; otherwise, the test point can be biased towards one of the surfaces that exist in pairs.

Based on Theorem 4.4.1, the best test point in each stage is the center point inside the region with the largest volume.

Theorem 4.4.2 also applies to regions subdivided from a larger region. This is true because the continuity stated in Assumption 4.2.3 can be satisfied even when the neighboring regions are considered separately. For any JND surface in S_1 of a given region (see Definition 4.4.4), the boundary curve of this surface is monotonic. Because S_2 of the neighboring region contains all the monotonic surfaces, it should also contain the surface with this boundary curve.

Algorithm 4.1 summarizes the procedure where subjective tests always report an accurate awareness.

Algorithm 4.1 Finding surface without sampling uncertainties

```
Require: $p(ref, mod)$: fraction of subjects who correctly identify the modified
    control input $ref + mod$; $\delta$: required error threshold;
Ensure: JND surface $p(ref, mod)$;
Measure $p(0,1)$ and $p(1,0)$; add them to $P_{\text{tested}}$;
while $\max l^2|p_i - p_j| > \delta$, where $l$ is the length of the diagonal of the square
    region, $i, j \in P_{\text{tested}}$ and no mid-point in between was tested do;
    Perform subjective tests to measure $p_m$, the mid-point of $p_i$ and $p_j$;
    Add $p_m$ to $P_{\text{tested}}$;
end while
Interpolate $p(ref, mod)$ with $P_{\text{tested}}$;
```

4.4.4 Uncertainties due to Limited Subjective Tests

Because each subjective test is a sampling process, the sampled awareness is a random variable. In this section we discuss strategies for calculating the volume (used in selecting the prism inside which the mid-point will be tested) and for ensuring that monotonicity always holds. Note that, with a lack of model for JND surfaces, it will be difficult to develop a comprehensive statistical model on the uncertainties of subjective tests.

The distribution of sampled awareness is as follows.

Theorem 4.4.3. Binomial distribution. *In a subjective test with n subjects, if the probability that a subject can discover the better alternative is p, then the number of subjects finding that one alternative is better than the other follows a binomial distribution:*

$$n\hat{p} \sim B(n, p).$$

Proof. The sum of IID random variables (Assumption 4.2.1 and 4.2.2) with Bernoulli distribution follows a Binomial distribution. □

In the selection of the best point to test, we need to compare the volume of each region and to find the one with the largest value. As awareness follows a binomial distribution and may be dependent due to monotonicity in Axiom 4.3.1, we calculate the joint probability of awareness in order to get the expectation of the height of the prisms.

After pairwise comparisons, we perform the test in the region that dominates other regions in volume. This region will most likely reduce the average absolute error by the maximum amount.

To reduce the computational complexity, we heuristically assume the independence of awareness p. In this case, we calculate the volume of the prisms independently and choose the center point of the largest prism to test. The expectation of the volume is as follows.

$$p^*_{A,B} = l^2 \int\limits_0^1 \int\limits_0^{p_A} (p_A - p_B) Pr(p_A|\hat{p}_A) Pr(p_B|\hat{p}_B)\, dp_B\, dp_A, \qquad (4.6)$$

where p is the real awareness, $\hat{p}$ is the sampled awareness in subjective tests; p_A (*resp.*, p_B) is the upper (*resp.*, lower) bound of the awareness in this prism; and l is the length of the diagonal of the square region.

After conducting the subjective tests, we need to adjust the awareness found (each a random variable) to ensure that they satisfy monotonicity required in Axiom 4.3.1. Our heuristic approach is to calculate the average of all JND surfaces that satisfy Axiom 4.3.1 by integrating over all possible values of awareness P that satisfy monotonicity:

$$p(r, m) = \int f_P(r, m) Pr(P|\hat{P}) dP, \tag{4.7}$$

where f is the region-wise linear approximation based on the set of test point vector P (referring to Definition 4.4.1) that satisfies monotonicity. The joint probability of dependent random variables in vector P is calculated by discretizing the value of awareness. After this adjustment, $p(r, m)$ is the final desired output.

Algorithm 4.2 summarizes the procedure for the case where limited subjective tests give rise to sampling uncertainties that follow the binomial distribution.

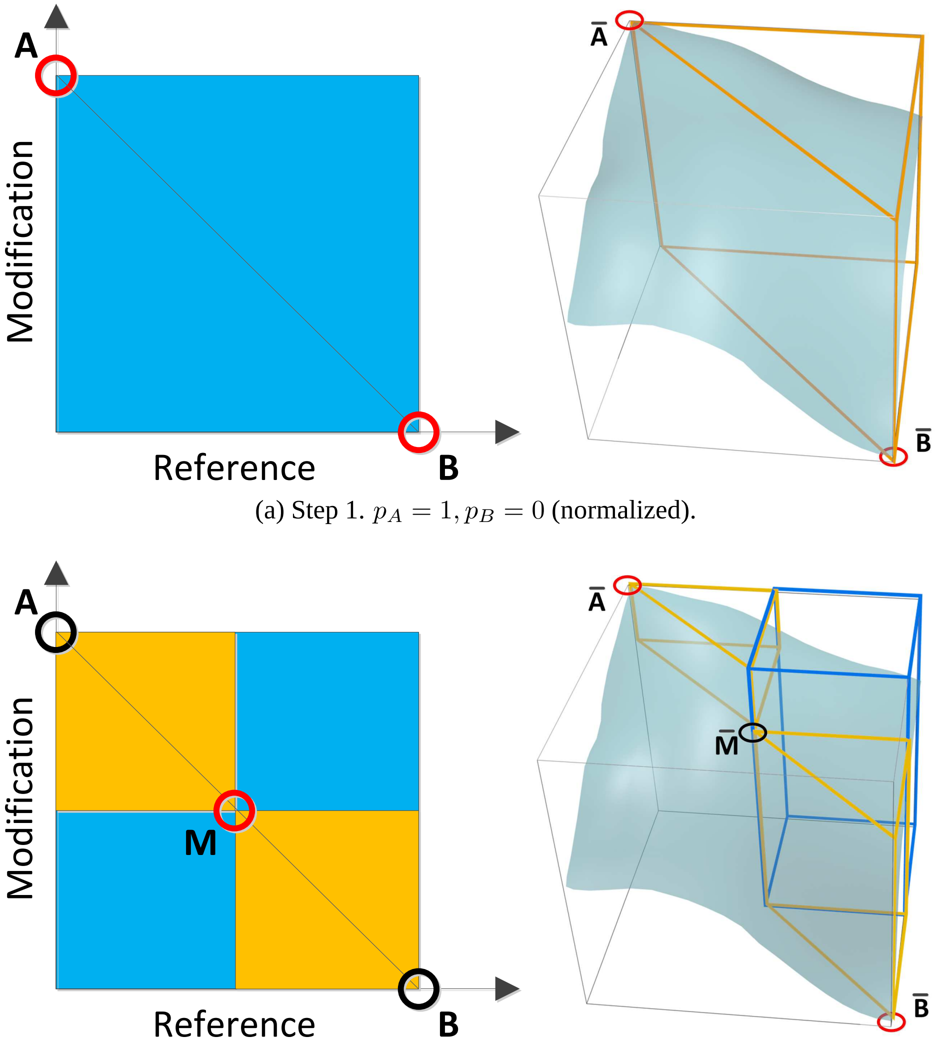
Algorithm 4.2 Finding surface with sampling uncertainties

```
Require: $\hat{p}(ref, mod)$: fraction of subjects who correctly identify the modified
    control input $ref + mod$; $\delta$: required error threshold.
Ensure: JND surface $p(ref, mod)$;
Measure $\hat{p}(0, 1)$ and $\hat{p}(1, 0)$; add them to $P_{\text{tested}}$;
while $\max p^*_{i,j} > \delta$ where $i, j \in P_{\text{tested}}$, $p^*$ is defined in (4.6), and no mid-point
    in between was tested do;
    Perform subjective tests to measure $\hat{p}_m$, the mid-point of $\hat{p}_i$ and $\hat{p}_j$;
    Add $\hat{p}_m$ to $P_{\text{tested}}$;
end while
Fix $\forall \hat{p} \in P_{\text{tested}}$ that do not satisfy monotonicity using (4.7);
Interpolate $p(ref, mod)$ with the fixed $P_{\text{tested}}$;
```

4.4.5 Illustration of Algorithm

Figure 4.12 illustrates a sequence of subjective tests performed. For simplicity, we assume no binomial errors in subjective tests.

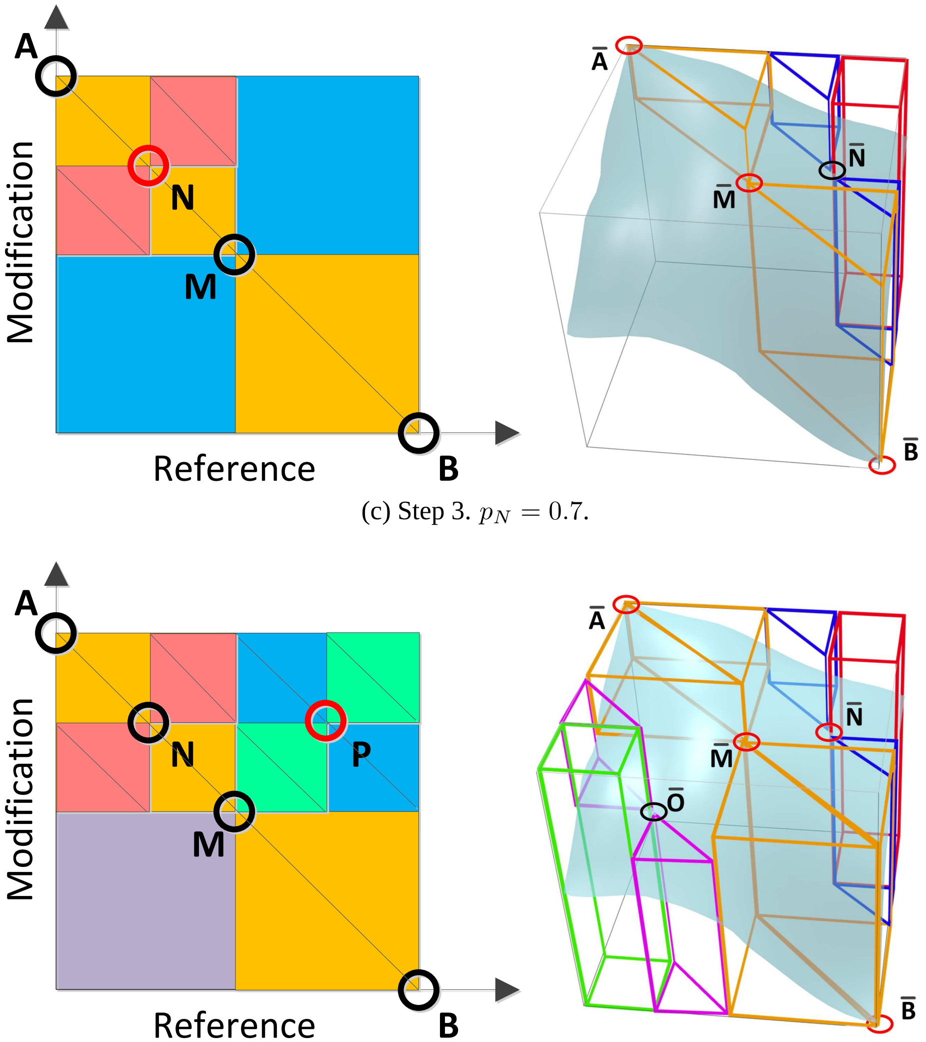
In Figure 4.12a, we start testing A and B. The bounding prism is shown in yellow. In the side view, we hide the prism at the opposite



(a) Step 1. $p_A = 1, p_B = 0$ (normalized).

(b) Step 2. $p_M = 0.7$.

Figure 4.12: An illustration of a sequence of subjective tests and the generation of the approximation mesh. The left panels show the test points (say A), and the right panels, how the test results (say awareness $\bar{A}$) reduce the average absolute error (or volume). The blue half-transparent surface is the real JND surface we want to approximate; the red circles are measured test points; and the black point is to be tested in this step. For clarity, some nearby points are not connected because several prisms as well as points attached to them are hidden.



(c) Step 3. $p_N = 0.7$.

(d) Step 4. $p_O = 0.6$.

Figure 4.12: An illustration of a sequence of subjective tests and the generation of the approximation mesh. (cont.)

side of the diagonal for clarity.

In Figure 4.12b, we test the mid-point M at $a = 0.5, b = 0.5$, as there is only one prism (the opposite one is symmetric) and it has the largest volume. The test result is shown in the right panel, where the awareness of M is shown as its height. The two yellow prisms are shrunk after M has been added. Note the height of the blue prisms. Because M does not bound the awareness of points to its top-right, the awareness of points in the blue prisms (merged into a blue cuboid) is still bounded by the awareness of A and B.

In Figure 4.12c, as the blue prism/cuboid has the largest volume (the same with that of the symmetric one at opposite side of the diagonal) among all prisms, we test its mid-point N. Similarly, the prisms are shrunk and two new red prisms are added. For clarity, another two red prisms in the opposite side are hidden.

In Figure 4.12d, we test the other blue prism mentioned above because it now has the largest volume.

After testing the points illustrated above, we have prisms that prescribe the ranges of awareness throughout the surface. That is, our algorithm has decomposed the surface on a region-by-region basis, and each piece of the surface is bounded by the corresponding prism.

Finally, we interpolate points not tested in order to approximate the entire surface. We simply use linear approximation because we have no priori knowledge on using more complex interpolation methods (see Figure 4.13). The green lines are the edges of the triangular meshes defined by these points. For clarity, we hide some prisms as well as the corresponding green lines.

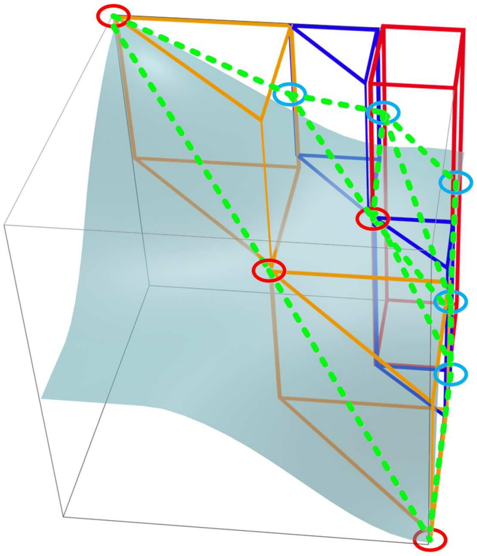


Figure 4.13: An illustration of generating an approximation mesh. Red circles are test points measured. Blue circles are generated by linear interpolations. For clarify, the approximation of the other half of the surface is omitted.

4.4.6 Experimental Results

In this section we present experimental results using synthetic data and a real application. The goals of these experiments are to demonstrate that our proposed greedy algorithm can attain the desired accuracy with a limited number of subjective tests and to show that the JND surface found can be used in a real application to improve perceptual quality.

Synthetic JND surfaces

We generate synthetic JND surfaces with $N = 6 \times 6$ grid points evenly placed on the X-Y plane. Their awareness is randomly generated by a uniform distribution but satisfies the monotonicity requirement in Axiom 4.3.1. We then use cubic interpolations to expand them into a smooth surface with 101×101 grid points. We generate a new JND surface for each application of Algorithm 4.1 and 4.2.

Algorithms tested

We evaluate Algorithm 4.1 and 4.2 by selecting $k \ll N$ test points in a step-by-step fashion using the information collected. After selecting the points, the surface is interpolated linearly to expand the awareness to a 101×101 discrete-point set, which is compared with the synthetic surface. We measure quality by the average approximation error.

Next, we find the upper-bound performance by approximating the synthetic surface using $k \ll N$ points as best as possible. This is done by a brute-force method that tries all possible ways of choosing k among N points and finding the placement that minimizes the average approximation error.

We also compare the performance against a peer method [74]. Because the method only sampled patterns with 3, 5, 9, 13, and more test points, considering the complexity, we only tested 5 points in our experiments.

Experimental results without binomial sampling uncertainties

We assume that the exact awareness is returned when testing a point on the surface. The purpose of the experiment is to evaluate the approximation quality of the algorithm without sampling uncertainties.

Figure 4.14 shows the results. When 5 points were generated, the error between the benchmark and our method is small, whereas the

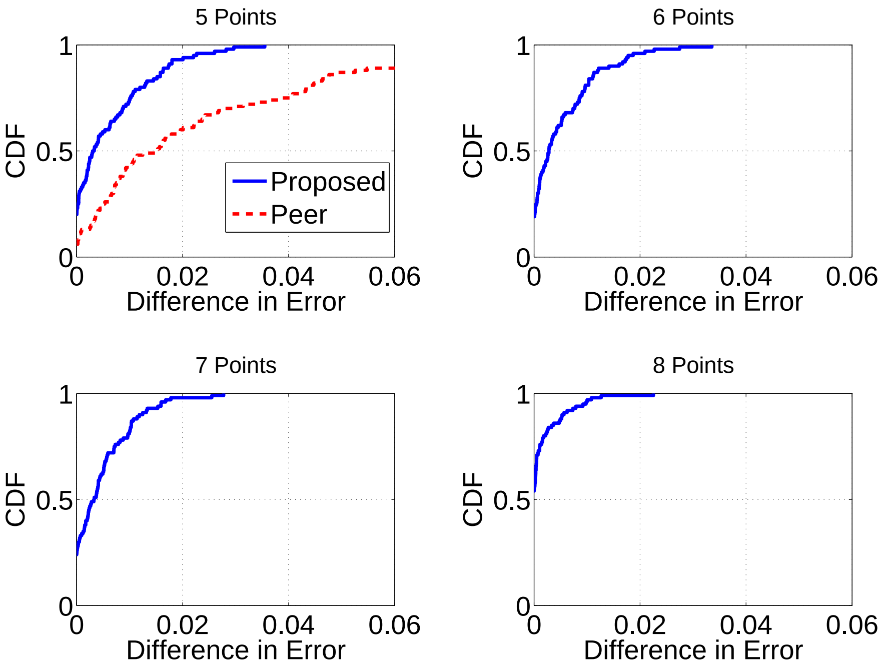
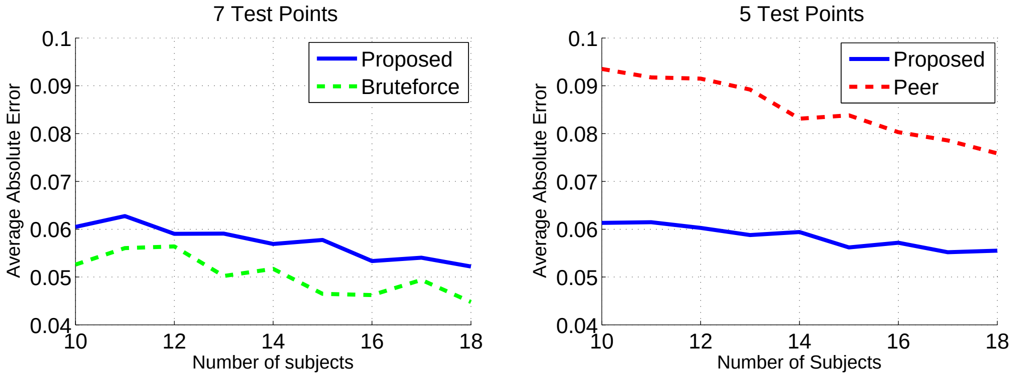


Figure 4.14: Empirical CDFs showing the difference in approximation errors between the benchmark and our proposed method, as well as between the benchmark and a peer method [74] with 5 points, with no binomial uncertainties in subjective tests (averaged over 300 different synthetic JND surfaces).

peer method performed much worse. When 7 points were generated, 90% of the surface has error smaller than 0.015, which is adequate as the range of awareness is between 0.5 and 1.0. The results show that our method is competitive with respect to the brute-force method.

Experimental results with binomial sampling uncertainties

This experiment is used to evaluate the performance when the number of tests is finite. In this case, a random error is introduced in the sample awareness, which follows the binomial distribution $B(n, p)$. For the benchmark method, we use the brute-force method to find the optimal placement. Because this tries all possible combinations,



(a) Proposed vs. Benchmark

(b) Proposed vs. Peer

Figure 4.15: Comparison of approximation errors when the results of subjective tests are with binomial errors (averaged over 300 different synthetic JND surfaces). The errors of our proposed algorithm in (a) decrease faster than those in (b) because more test points are used.

it will have smaller errors when compared to the greedy method.

Figure 4.15a shows the average difference between the benchmark and our greedy method. The result shows two properties. 1) Our method has comparable precision in approximating the JND surface even though each point can only be sampled once, whereas the benchmark method can re-sample each point multiple times and find the one with the smallest error. 2) The difference in average approximation error is generally reduced as the number of subjects is increased, although there are some fluctuations due to binomial sampling uncertainties.

Figure 4.15b compares the performance between our method and the peer method [74]. It shows that our method outperforms the peer method, independent of the number of subjects.

In short, these experimental results show that our proposed method for approximating JND surfaces using limited subjective tests is cost-effective and reliable.

4.4.7 Summary

We have presented a general framework for mapping one system control to perceptual quality according to human perception. A theory for measuring JND surfaces with only a few subjective tests has been developed. The JND surface can then be used to fit any run-time condition, as it contains the subjective awareness on changes under the whole range of control values. The effectiveness of the method has been evaluated by Monte-Carlo simulations.

4.5 Mapping Multiple Controls to Perceptual Quality

Although our results have been presented with respect to one system control, they can be extended to multiple system controls. A trivial approach is to measure a multi-dimensional JND surface with all the controls. However, considering the exponential complexity, it is generally not feasible to perform extensive subjective tests for such a surface.

To avoid the exponential complexity, one way is to exploit the independence of controls. If the quality metrics regarding one system control are independent, then we can separately measure the corresponding JND surfaces and combine them probabilistically. As JND surfaces are usually used to guide a quality optimization, their optimal combination (such as the one resulting in the least awareness) can be searched within the constraints of the controls according to the running condition of the underlying multimedia application. In this way, the dimension of the JND surface can be reduced to a manageable size. This method can be used for developing efficient online algorithms for optimizing multimedia applications. When there are many independent controls, because of the high complexity of the combination, we do not have efficient online algorithm.

In practical multimedia applications, system controls can be dependent, because human can perceive multiple media sources as a whole, and these sources are controlled by the system controls simultaneously. The second dominance property we have found, namely the human-focus property, allow us to decompose the JND surface of multiple system controls into individual ones which can then be measured by the method for a single control. By utilizing this property in the optimization, we can greatly reduce the number of offline subjective tests. In this section, we focus on this more general case. We provide a closed-form solution for the optimization with a simple constraint. When the constraint is complex, we need to search all the JND surfaces in exponential complexity, which cannot be done at run-time without heuristic approaches.

Table 4.2 summarizes the type of optimizations in real-time in online fast-paced interactive multimedia and our solutions for solving them in this thesis.

In this section we present the optimization method for multiple dependent controls first, because they are a more general form when considering that controls tend to be dependent in multimedia applications. We further demonstrate how we can optimize the perceptual quality even when the constraints are complex in Section 7.3.3 as a supplement to the discussion in this section. Because it uses applica-

Table 4.2: Type of optimization and solution

Type	Sub-Type	Solution	Section
Independent Controls	Single Control with Simple/Complex Constraint	Combining JND surfaces	5.3
	Multiple Controls	No efficient algorithm	
Dependent Controls	Single/Multiple Control(s) with a Simple Constraint	Equating their noticeability	4.5.3
	Single/Multiple Control(s) with a Complex Constraint	Depending on certain constraints for limiting search space	4.5.3 7.3.3

tion dependent constraints for limiting the search space, we present the content along with the application. We leave the discussion of the special case for single independent control will be discussed later in Section 5.3, as it is useful when a run-time application can only provides limited computational time. We do not solve the case with multiple independent controls because we lack efficient methods for reducing the search space.

In this section, we use the concept of noticeability defined in Section 4.2.1 as a mechanisim for evaluating the combined effects.

4.5.1 Reducing the Complexity by Concentrating upon Optimization

JND surface is measured for optimizing perceptual quality of the multimedia applications. To optimize a multimedia system with multiple controls, it is not necessary to attain a complete high-dimensional JND surface.

Actually, in most cases, our goal in the optimization is to minimize $P_{\text{notice}}^{\textbf{COMB}}(ref, m)$ under given ref, leading to the least noticeable artifact in the multimedia application, thereby resulting in the best perceptual quality.

$$\mathcal{P} = \min \ P_{\text{notice}}^{\textbf{COMB}}(ref, m). \tag{4.8}$$

Sometimes the goal of the optimization is to maximize perceptual quality. In that case, the optimization can still be formulated into an equivalent problem of minimizing the artifacts that can degrade perceptual quality. Therefore, in this section we only discuss the optimization problem above.

Our problem is how to to measure $P_{\text{notice}}^{\textbf{COMB}}(ref, m)$, the JND surface of the combined perceptual quality. As there are multiple dependent system controls, the high-dimensional JND surface will be expensive to measure by subjective tests. Further, the controls in m may have dependent effects. Since we do not know their dependence, we define $P_{\text{notice}}^{\textbf{COMB}}(ref, m)$ using function f without a closed form.

$$\begin{aligned}
& P_{\text{notice-f}}^{\textbf{COMB}}(ref, m) && (4.9)\\
= \ & f(P_{\text{notice}}^{\text{metric 1}'}(ref, m_1), \ P_{\text{notice}}^{\text{metric 2}'}(ref, m_1)), \\
& \ldots, \ P_{\text{notice}}^{\text{metric n}'}(ref, m_n)(ref, m_n)) && (4.10)\\
\text{where} \quad & P_{\text{notice}}^{\text{metric } i'}(ref, m_i) \\
= \ & P_{\text{notice}}^{\text{metric } i}(ref, m_i \,|\, m_j = 0, j \neq i), i = 1 \ldots n. && (4.11)
\end{aligned}$$

Here, $P_{\text{notice}}^{\text{metric i}'}(ref, m_i)$ is the quantitative metric corresponding to the control m_i. The quantitative metric is carefully chosen so that it represents an undesired effect, or an artifact. The higher m_i is, the more degradation it will result in the overall perceptual quality.

As an illustration, following the example in Section 4.2.2, we have JND surfaces on users' perceptions of the delay effects based on, respectively, delaying, extending, and shortening individual actions. In each surface, $P_{\text{notice}}$is expressed as $P_{\text{notice}}(ref, m)$, where ref is the duration of an action, and m is the network latency. We like to find $P^{\mathbf{COMB}}_{\text{notice}}(ref, m)$, the JND surface of the combined strategy. As $m = (m^{\mathbf{LL}}, m^{\mathbf{LPF1}}, m^{\mathbf{LPF2}})$, the 5-D JND surface will be expensive to measure by subjective tests. Further, the controls in m may have dependent effects. Since we do not know their dependence, we define $P^{\mathbf{COMB}}_{\text{notice}}(ref, m)$ using function f without a closed form.

$$\begin{aligned} & P^{\mathbf{COMB}}_{\text{notice-f}}(ref, m) \\ = \; & f(P'_{\text{notice}}(ref, m^{\mathbf{LL}}),\ P'_{\text{notice}}(ref, m^{\mathbf{LPF1}}),\ P'_{\text{notice}}(ref, m^{\mathbf{LPF2}})) \end{aligned}$$

$$\text{where} \quad \begin{cases} P'_{\text{notice}}(ref, m^{\mathbf{LL}}) = P_{\text{notice}}(ref, m^{\mathbf{LL}} \mid m^{\mathbf{LPF1}} = 0, m^{\mathbf{LPF2}} = 0) \\ P'_{\text{notice}}(ref, m^{\mathbf{LPF1}}) = P_{\text{notice}}(ref, m^{\mathbf{LPF1}} \mid m^{\mathbf{LL}} = 0, m^{\mathbf{LPF2}} = 0) \\ P'_{\text{notice}}(ref, m^{\mathbf{LPF2}}) = P_{\text{notice}}(ref, m^{\mathbf{LPF2}} \mid m^{\mathbf{LL}} = 0, m^{\mathbf{LPF1}} = 0). \end{cases}$$

The problem is how we can attain $P^{\mathbf{COMB}}_{\text{notice-f}}(ref, m)$ with the individual JND surfaces of each strategy in hand.

4.5.2 Theorems Derived from the Dominance Properties

The following two lemmas can be derived directly from Axiom 4.3.1.

Lemma 4.5.1.

$$\begin{aligned} \textit{Let} \quad & P_{\textit{max}} = \max_{i=1\ldots n} \{P^{\textit{metric } i'}_{\textit{notice}}(ref, m_i)\} \\ \textit{then} \quad & \hat{m}_i \geq m_i \textit{ for } P^{\textit{metric } i'}_{\textit{notice}}(ref, \hat{m}_i) = P_{\textit{max}} \end{aligned} \qquad (4.12)$$

Lemma 4.5.2. *Using the above definitions, we have*

$$P^{\textit{COMB}}_{\textit{notice-f}}(ref, m_1, m_2, \ldots, m_n) \qquad (4.13)$$

$$\leq \; P^{\textit{COMB}}_{\textit{notice-f}}(ref, \hat{m}_1, \hat{m}_2, \ldots, \hat{m}_n). \qquad (4.14)$$

The following assumption is a more specific form of the human-focus dominance property. It is based on the observation that in fast-paced multimedia systems, subjects will only notice the dominant delay effect of the dependent controls but not those due to individual controls when compared to the reference. This is further discussed in Chapter 7.

Property 4.5.1. *Given $\hat{m} = (\hat{m}_1, \hat{m}_2, \ldots, \hat{m}_n)$, then $P^{\textbf{\textit{COMB}}}_{\textit{notice-f}}(ref, \hat{m})$ is equal to the maximum of the three individual noticeabilities when they are equal.*

$$P^{\textbf{\textit{COMB}}}_{\textit{notice-f}}(ref, \hat{m}) \;=\; \max_{i=1\ldots n} \{P^{\textit{metric } i'}_{\textit{notice}}(ref, \hat{m}_i)\}. \qquad (4.15)$$

Corollary 4.5.1. $P^{\textbf{\textit{COMB}}}_{\textit{notice-f}}(ref, \hat{m}) = P_{\textit{max}}$.

The proof is straightforward by applying Property 4.5.1 and Lemma 4.5.1.

Corollary 4.5.2.

$$P^{\textbf{\textit{COMB}}}_{\textit{notice-f}}(ref, m) \;\leq\; \max \{P^{\textit{metric } i'}_{\textit{notice}}(ref, m_i)\}. \qquad (4.16)$$

Proof. This can be proved by combining Lemma 4.5.2 and Corollary 4.5.1. □

Without knowing the closed form of function f in (4.10), the best we can do is to minimize the upper bound of f. According to Corollary 4.5.2, we have

$$\overline{\mathcal{P}} \;=\; \min\; \max \{P^{\text{metric } i'}_{\text{notice}}(ref, m_i)\}. \qquad (4.17)$$

4.5.3 Solution of the Optimization

Case 1: Simple Constraint

In this section we study the optimization when the constraint is $\mathbf{k}'\mathbf{m} \leq c$, where $\mathbf{k}$ is a vector of positive weights ($k_i > 0 \ \forall 1 \leq i \leq n$), $\mathbf{m} = m_1, m_2, \ldots, m_n$ is a vector of the control parameters, and c is a constant. We prove that the optimal solution of the optimization appears when the individual control parameters result in the same noticeabilities of the degradations.

The optimization problem is as follows.

$$\overline{\mathcal{P}} \ = \ \min\ \max\{P_{\text{notice}}^{\text{metric}\ i'}(ref, m_i)\}. \tag{4.18}$$

$$\text{subject to } \mathbf{k}'\mathbf{m} \leq c, \mathbf{k} > 0, c > 0. \tag{4.19}$$

The following theorem proves the optimal solution to (4.18)-(4.19).

Theorem 4.5.1. *The optimal solution to (4.18)-(4.19) is* $(\overline{m}_1, \overline{m}_2, \ldots, \overline{m}_n)$*, where*

$$\begin{aligned} P_{\textit{notice}}^{\textit{metric}\ 1'}(ref, \overline{m}_1) \ &= \ P_{\textit{notice}}^{\textit{metric}\ 2'}(ref, \overline{m}_2) \\ &= \ \ldots \ = \ P_{\textit{notice}}^{\textit{metric}\ n'}(ref, \overline{m}_n). \end{aligned} \tag{4.20}$$

Proof. The proof is by contradiction. If (4.20) is false, then without loss of generality, we assume that $P_{\text{notice}}^{\text{metric}\ 1'}(ref, \overline{m}_1)$ is the largest among all $P_{\text{notice}}^{\text{metric}\ i'}(ref, \overline{m}_i)$, $1 \leq i \leq n$.

Based on Axiom 4.3.1, we know that $P_{\text{notice}}^{\text{metric}\ i'}(ref, \overline{m}_i), i = 1 \ldots n$ are, respectively, monotonically non-decreasing with increasing $\overline{m}_i, i = 1 \ldots n$, To get the optimal solution, $P_{\text{notice}}^{\text{metric}\ 1'}(ref, \overline{m}_1)$ should be reduced as much as possible. However, as $P_{\text{notice}}^{\text{metric}\ 1'}(ref, \overline{m}_1)$ is reduced, $P_{\text{notice}}^{\text{metric}\ j'}(ref, \overline{m}_j), 1 < in$ will be increased due to (4.19). These lead to larger $P_{\text{notice}}^{\text{metric}\ j'}(ref, m_j), 1 < i \leq n$.

As we assume that $P_{\text{notice}}^{\text{metric }1'}(ref, \overline{m}_1)$ is the largest among all $P_{\text{notice}}^{\text{metric }i'}(ref, \overline{m}_1)$, the optimal solution of (4.18)-(4.19) is when $\overline{\mathcal{P}} = P_{\text{notice}}^{\text{metric }1'}(ref, \overline{m}_1) = P_{\text{notice}}^{\text{metric }2'}(ref, \overline{m}_2) + \delta_2 = P_{\text{notice}}^{\text{metric }3'}(ref, \overline{m}_3) + \delta_3 = \cdots = P_{\text{notice}}^{\text{metric }n'}(ref, \overline{m}_n) + \delta_n$. However, we can always find $\delta_i > \delta_i^{'} > 0, 1 < i \leq n$ such that $P_{\text{notice}}^{\text{metric }1'}(ref, \overline{m}_1) > P_{\text{notice}}^{\text{metric }1'}(ref, \overline{\overline{m}}_1) = P_{\text{notice}}^{\text{metric }i'}(ref, \overline{m}_i) + \delta_i^{'}, 1 < i \leq n$. Therefore $P_{\text{notice}}^{\text{metric }1'}(ref, \overline{m}_1)$ is not optimal. Contradiction! □

Case 2: Complex Constraints

We discuss the strategies in searching the optimal solution when the constraints do not follow the simple form in Section 4.5.3.

When the optimization is complex to be solved in a closed form and many variables and constraints as well as different ref's exist, it is hard to find an optimal solution at run time. Fortunately, several observations can help simplify the problem.

Firstly, the complex problem can be divided into several subproblems, each of which can be solved by the optimization in Section 4.5.3.

Secondly, several constraints have already limited the range of the solution, so the search of the optimal solution can be performed quickly.

Thirdly, in multi-user systems, not all users have interactions simultaneously. Therefore, the search space of the optimization can be greatly reduced when we only focus on the users that have possible interactions in a short future.

As these methods are highly application-dependent, we demonstrate them in detail with a fast-paced online shooting game in Chapter 7.

4.5.4 Summary

In this section, we discuss how multiple system controls can be mapped to perceptual quality by a JND surface with multiple dimensions. We presents how the problem can be simplified when the JND surface is used for the optimization of perceptual quality and prove theorems derived from the dominance properties that can be used to simplify the optimization. The theory can be used in later chapters for optimizing real applications.

4.6 Conclusion

In this chapter we have discussed how the two dominance properties we have found can reduce the number of offline subjective tests for attaining the mapping from either single of multiple system control(s) to perceptual quality. The efficiency and precission of the proposed theory are either proved by numerical simulations or by mathematical proofs. With the help of JND surfaces, the mapping can be stored in a compact form which can be used in later chapters for the generalization under different run-time conditions. We have further summarized the results we have attained in Table 4.1.

□ **End of chapter.**

Chapter 5

Online Generalization of the Subjective Opinions

In last chapter, we have presented how subjective opinions are collected in a set of offline tests using a JND surface. Because offline subjective tests are performed in a given network condition, the results can only be reused when the network condition is the same as that in the tests, or when the change of the condition does not affect the quality with which we are consulting the subjects. Further, online multimedia application requires fast adaption to the change of network condition. We need a sufficiently fast algorithm for optimizing the multimedia application using the JND surface. To solve these problems, in this chapter we study the theory on how we can generalize the subjective opinions we have collected in last chapter (i.e. the JND surface) to any network condition that is provided by the network-control layer. We further study a fast algorithm for a single control in optimizing the online fast-paced interactive multimedia applications when the quantitative quality metrics can be considered independent. For simplicity, quantitative quality metrics are called quality metrics for short in this chapter.

5.1 Problems and Approaches

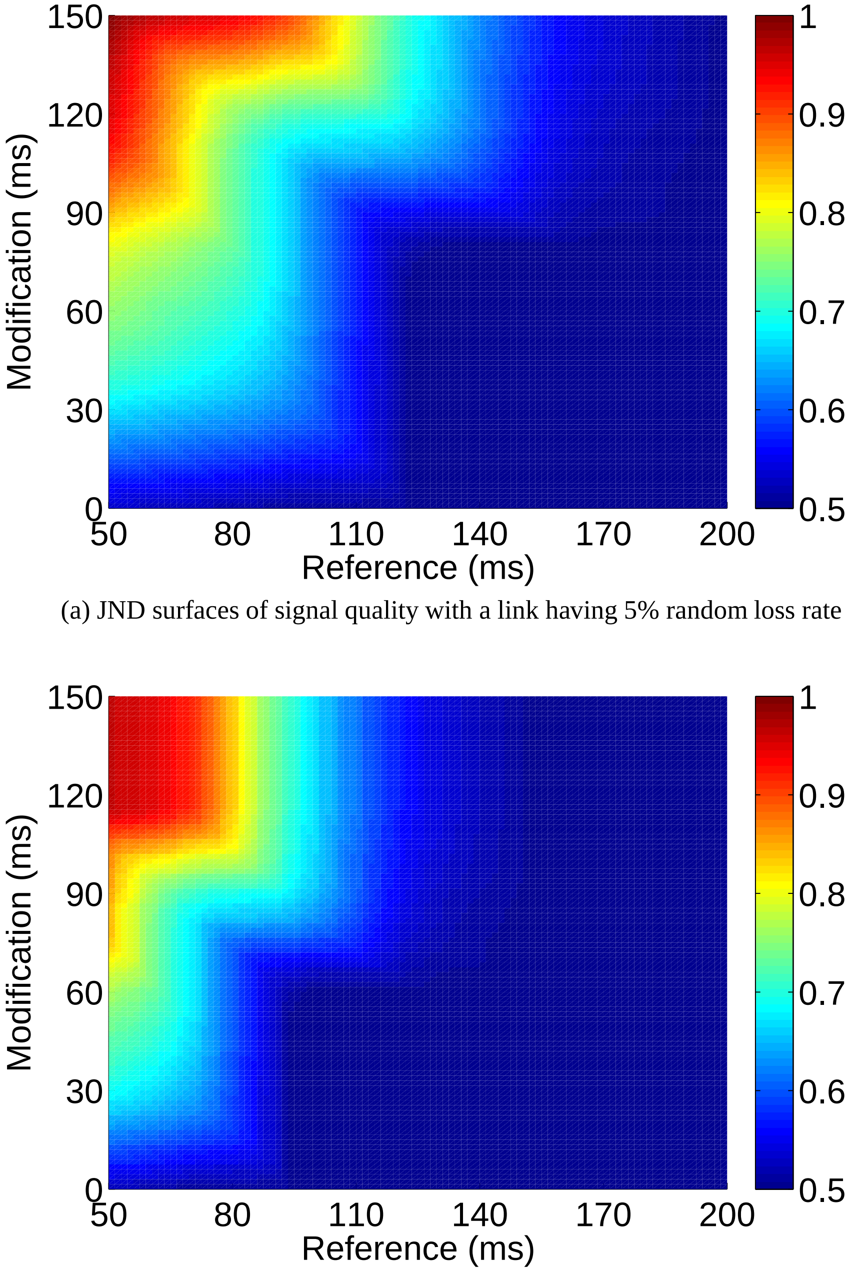
Our research problem is two-folded.

5.1.1 Problem 1: Inconsistence of the JND Surface under Different Network Conditions

When the network condition becomes worse, the packet loss rate, the one-way network latency, and the delay jitters will increase accordingly. Given the same system controls, the application will perform worse with such worse network condition. For example, the signal quality of the audio and video can degrade. The online game may have more significant latency. For this reason, the JND surface we have collected in offline subjective tests under a certain simulated network condition cannot be directly reused under various online network conditions.

Figure 5.1a illustrates the JND surface of the audio signal quality in a link with 2% random losses. Comparing it with Figure 5.1b which is collected in a link with 5% random losses, all the corresponding points in the latter JND surface have higher awareness of the degradation, which shows that more subjects can notice the degradation of the audio signal quality in the case with more random losses.

Our problem is, given an offline JND surface like Figure 5.1a, how we can generalize it to any network condition at any time, so that we can have an online JND surface like Figure 5.1b.



(a) JND surfaces of signal quality with a link having 5% random loss rate

(b) JND surfaces of signal quality with a link having 2% random loss rate

Figure 5.1: Synthetic JND surfaces (both axes normalized to [0,1]) about the audio signal quality when the random loss rate is a) 5% and b) 2%. We are interested in how the former can be transformed to the latter one.

5.1.2 Problem 2: High Complexity Search in Online Optimization

In Chapter 4 we have studied the optimization of the perceptual quality by minimizing the noticeability where degradation for multiple dependent controls. Recall Table 4.2, when the constraint of the optimization is in a simple form in Section 4.5.3 Case 1. We have an efficient method to find the solution. However, when the constraint is complex, which is not uncommon in multimedia applications, we cannot guarantee real-time performance. This is because the objective function is a $\min\max$ function that involves multiple JND surfaces, all of which cannot be represented in closed form. It means that we have to search over the many JND surfaces with an exponential complexity ($O(k^n)$, where k is the resolution of the JND surface through the Y-axis, and n is the number of the quantitative quality metrics) to find the optimal solution instead of the polynomial complexity with a simple constraint. When the number of system controls is large, the search time can significantly increase the latency of the system, which is not desirable in fast-paced interactive systems.

Our problem is, how we can accelerate the optimization so that it can run efficiently in fast-paced interactive multimedia applications.

5.1.3 Approaches

To solve the first problem, our approach is to separate the dependence of network conditions from the JND surface. We handle the complex network condition with the network-control layer presented in Chapter 3. After the processes in that layer, the perceived network condition in the application layer has only small random losses. In that case, we can use the loss rate as the anchor to transform the JND surface collected in offline measurement to fit the near perceived network condition. This is possible because humans have the same awareness on signal quality when facing the same small random losses in a given similar context (say in a videoconferencing scenario with very few motions). This problem is discussed and solved in Section 5.2.

To solve the second problem, in Section 5.3 we assume a simplified but practical running condition in which multiple critical quantitative quality metrics are corresponding to a *single* system control.

Assumption 5.1.1. Single Control. *We assume that a single control corresponds to multiple JND surface, each is for an independent quantitative metric for the efficient algorithm discussed in Section 5.3.*

With the relation between the JND surface and the quantitative quality, we can handle trade-offs with probabilistic methods to generate a single combined JND surface that can represent human preference when these quantitative quality metrics are involved. Finally, we search for the optimal control in this JND surface. In this way we can simplify the search of many JND surfaces into the search of only a few combined JND surfaces, and the complexity can be reduced to $O(mk^{(n/m)})$, where m is the number of system controls.

5.2 Generalization of JND Surface in Online Network Condition

In this section, we study how we can generalize the offline measured JND surface according to an online network condition.

5.2.1 Separation of Network Condition and the JND Surface

As bandwidth has increased significantly in recent years, nowadays network has relatively stable operating condition. We have shown in Chapter 3 that the perceived network condition, i.e. the network condition after processing in the network-control layer, has only small random losses, and without significant delay jitters and consecutive losses.

Therefore, it is reasonable to follow the Internet Protocol Suite and process all network traffic in the lower transport layers. In this way, the generalization discussed in this chapter are only limited to the application layer. In other words, all the network condition in this chapter is a "filtered" version that has more stable network behavior. We have called this network condition the perceived network condition in Chapter 3.

With this strategy, we can simply transform the JND surface we have measured in offline subjective tests under a simulated link with losses to attain the desired JND surface at run time.

For example, in offline subjective tests we have measured the JND surface based on a simulated link with 5% random loss rate. We study how we can transform this JND surface to the one based on an online link with 2% random loss rate.

5.2.2 Transformation of the JND Surface under Small Random Losses

As shown in Figure 5.1, subjective opinions regarding the change of system controls can be affected by the underlying network condition. Because the application layer always communicates with the network-control layer to adjust the system controls, it is necessary for the application layer to react to the change of network condition and use a new JND surface that can adapt to the online network condition.

The network-control layer has helped remove most network impairments, and the remaining factors that can affect the subjective opinions are the change of network latency and the loss rate.

When the average network delay increases, we can simply increase the starting EED in the JND surface without changing the surface itself. The mapping is as follows:

$$P_{\text{notice,new}}(r, m) = P_{\text{notice,old}}(r + d, m),$$

where d is the change in the network latency.

We can modify the JND surface in this fashion because network latency and network loss rate are generally independently change, expect under a severe network congestion, in which case interactive multimedia application will already be suspended. We do not need to modify the interactivity surface because the new network condition should not change human sensitivity on interactivity. In short, we can do online modification of the offline-measured JND surface.

After the transformation on network latency, we consider the transformation on network loss. When the average network loss rate changes, we can transform the JND surface based on the original loss rate and the modified loss rate. Recall the definition of a JND surface in Definition 4.2.8. The input of the function is the referenced control and the modification, and the output is the awareness based on the subjective tests performed. As long as the reference and the modification are the same, the subjective test has already been done, and the function value remains unchanged. In this way, we use the following transformation:

$$\begin{aligned} P_{\text{notice,new}}(r, m) &= P_{\text{notice,old}}(f1(r, p_{\text{loss,old}}, p_{\text{loss,new}}), \\ &\quad f2(m, p_{\text{loss,old}}, p_{\text{loss,new}})), \end{aligned} \tag{5.1}$$

where $f1$ (resp. $f2$) is a function that calculates the corresponding r (resp. m) in the original surface according to the old and new network loss rates and the error correction algorithm. Note that $f1$ and $f2$ are valid as long as the original loss rate is larger than the new loss rate. It means that as long as we have measured the original JND surface in a network condition with the largest network loss rate r_{max} assumed (5% according to the assumption in Table 1.2 for the network-control layer), the JND surface for other network loss rate can be attained by transformation.

Lemma 5.2.1. $f1$ *(resp.* $f2$*) are monotonically non-decreasing regarding* r *(resp.* m*).*

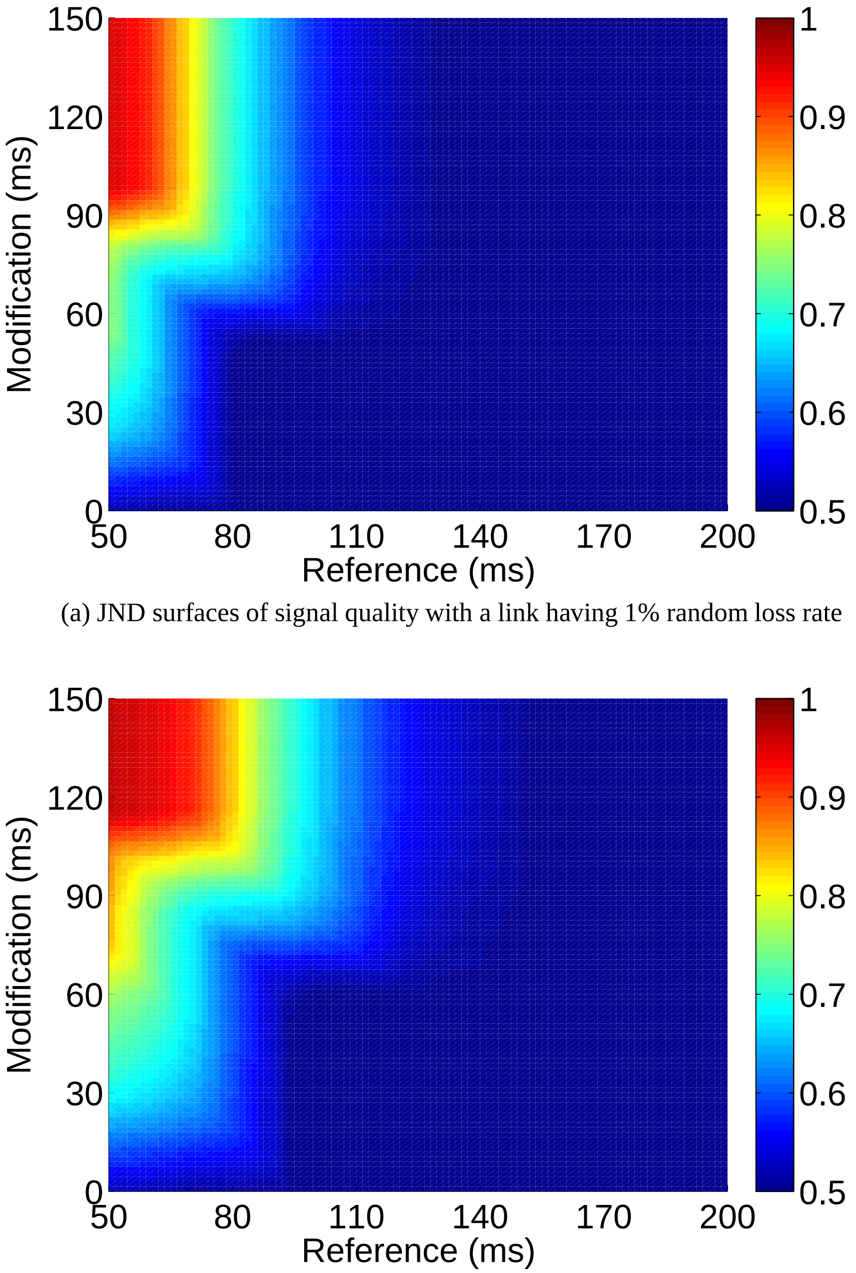
Proof. This is true because a larger r or m indicate a higher loss rate for error correction, which also needs more error correction in the new setting. □

Theorem 5.2.1. *A JND surface measured in the largest possible network loss rate is sufficient for the transformation to any JND surface* *under another loss rate.*

Proof. Let $r_{\text{max,new}}$ (resp. $r_{\text{max,old}}$) be the maximal control under the loss rate p_{loss} (resp. 5%). For any network loss rate $p_{\text{loss}} \leq 5\%$, because we need less resource for handling the loss, $f1(r_{\text{max,new}}, 5\%, p_{\text{loss}})$ $< r_{\text{max,old}}$. Further, according to Lemma 5.2.1, $f1(r, 5\%, p_{\text{loss}})$ $< f1(r_{\text{max,new}}, 5\%, p_{\text{loss}})$. Therefore, $f1(r, 5\%, p_{\text{loss}}) < r_{\text{max,old}}$, meaning that the value is in the range of the JND surface measured at 5% loss rate. A similar proof applies to $f2$. □

5% is set per our assumption on the application layer. If the codec in application layer can perform good with a high loss rate, then the value can be even higher.

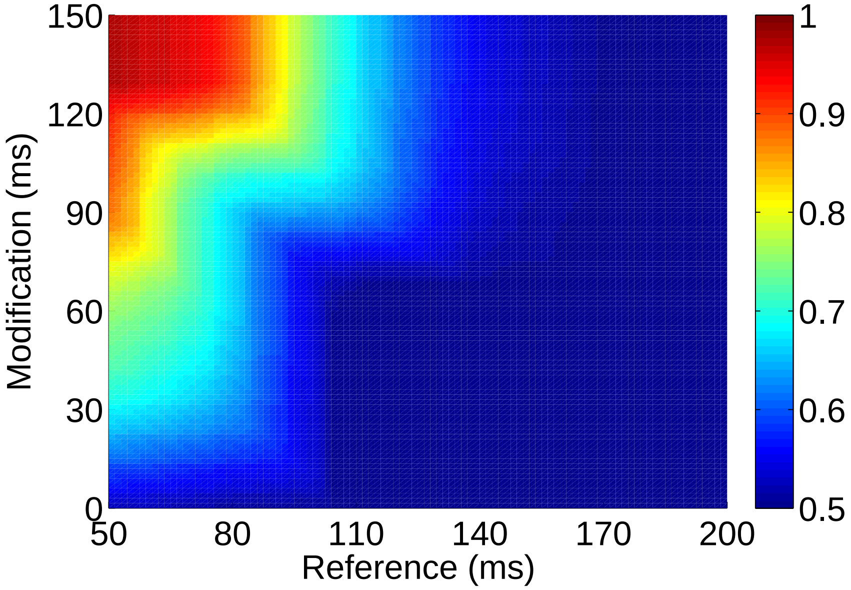
Figure 5.2 presents the JND surfaces when the network has a 1%, 2%, 3%, and 4% loss rates, respectively. All of them are generated from the offline JND surface shown in Figure 5.1b with the proposed method in this section. The time for generating each of them in an ordinary Desktop PC with an Intel Core 2 Duo E8400 CPU is 7.1 ms in average. This clearly shows the efficiency of our algorithm. It is worth mentioning that the left part of the 1%-loss-rate surface has more red point at the middle than those in the 2%-loss-rate surface. This is because with a low loss rate, a small change in the loss rate after error correction is significant. It also shows that the transformation of the JND surfaces cannot be done by a simple point-to-point calculation, because the mapping is based on the loss rate instead of the EED.



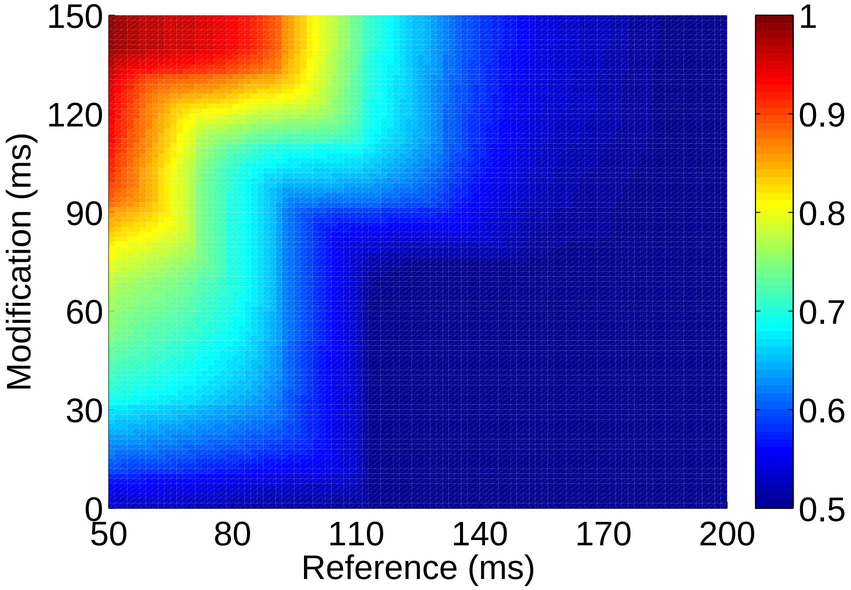
(a) JND surfaces of signal quality with a link having 1% random loss rate

(b) JND surfaces of signal quality with a link having 2% random loss rate

Figure 5.2: JND surfaces (both axes normalized to [0,1]) on the audio signal quality when the random loss rate is 1% - 4% in simulated links. The network latency is 50 ms with no delay jitters. The packet interval is set to 50 ms. The error correction strategy is 4-way piggy-backing.



(c) JND surfaces of signal quality with a link having 3% random loss rate



(d) JND surfaces of signal quality with a link having 4% random loss rate

Figure 5.2: JND surfaces on the audio signal quality. (cont.)

5.3 Efficient Optimization Algorithm with JND Surfaces of a Single Control

In this section, we propose an efficient algorithm for optimizing the perceptual quality with independent JND surfaces corresponding to a single control, as stated in Assumption 5.1.1.

To understand why this simplification is useful in real applications, we revisit the controls in multimedia systems. Online fast-paced interactive multimedia systems have some critical system controls. For example, we have studied the benefits of a longer buffer for previous chapters in reducing the loss rate and delay jitters in transmission. The buffering time is part of the overall EED which is a critical system control. Another critical system control is the transmission bandwidth, which has been managed by the network-control layer. When we try to improve the efficiency in online optimization of perceptual quality of the system, we need to focus on the critical system controls rather than individual quantitative quality metrics. In other words, we can group the JND surfaces corresponding to quantitative metrics of the same system control into a combined JND surface, instead of treating them separately. This can greatly reduce the computational complexity in searching for the optimal system controls. In this case, we only search a few combined JND surfaces, each corresponding to one system control, instead of searching all the JND surfaces simultaneously.

To understand the idea of the combined JND surface, we first study how quantitative qualities are represented in individual JND surfaces. We then show how trade-offs in the optimization are represented in the form of human awareness of positive and negative changes. We finally propose a method for tackling the trade-offs in a combined JND surface and find the optimal system control in that surface.

5.3.1 Quality Metric, JND Surface, and Trade-offs

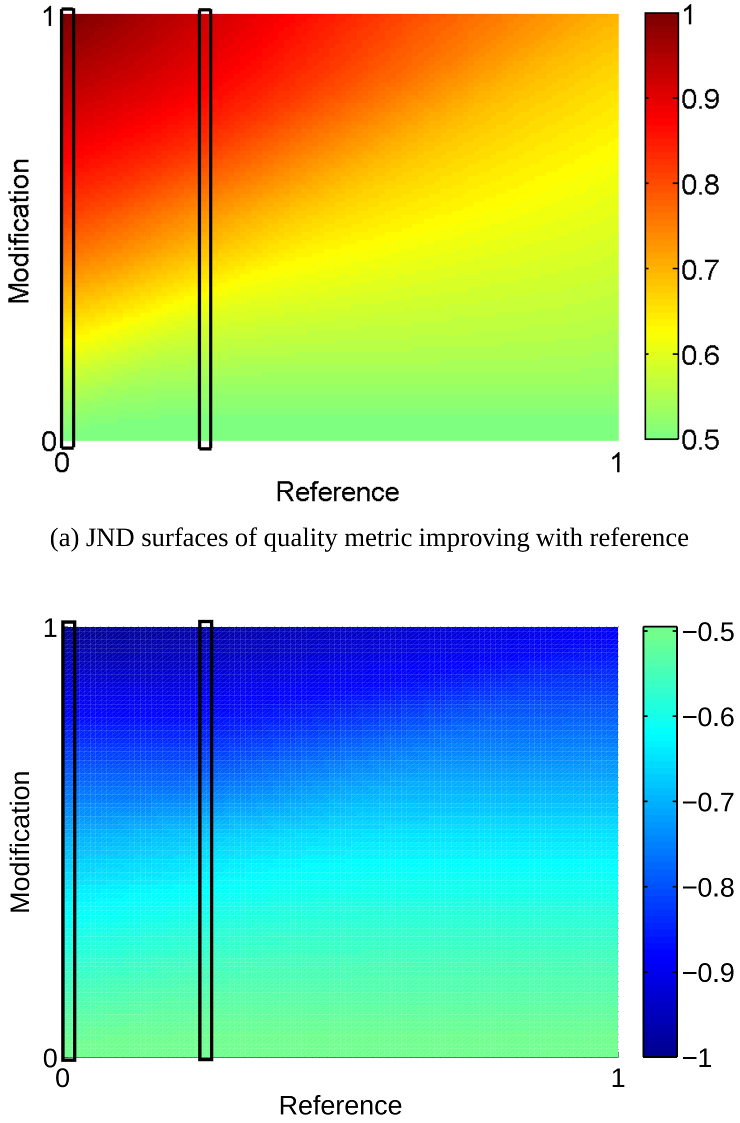
To understand the trade-offs among quantitative qualities in JND surfaces, we first refer to the relation between awareness and the corresponding perceptual quality.

With one system control, a JND surface plots $p(ref, mod)$, where p is the awareness that measures the probability of subjects who can identify the output due to the modified reference $ref + mod$ from that of ref when the outputs are presented in a random order.

With a simple extension, a JND surface can represent the modification when it lead to improvement or degradation.

Definition 5.3.1. *Let p be the awareness of a pairwise comparison of two outputs of a multimedia system, one due to a reference input (ref) and other due to a modification of the reference ($ref + mod$). Then $p > 0.5$ (*resp. *$p < 0.5$) indicates that $ref + mod$ has better (*resp. *worse) relative perceptual quality than ref, whereas $p = 0.5$ indicates the same perceptual quality.*

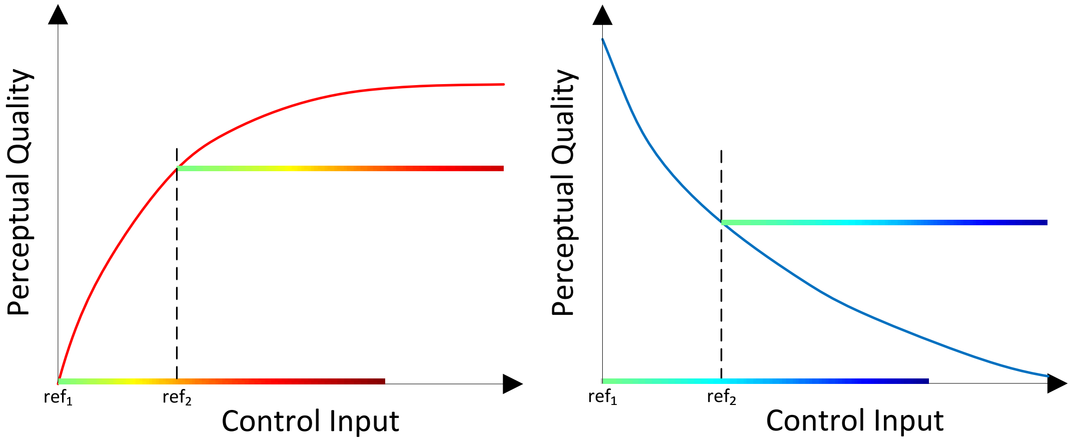
Figure 5.3 illustrates two JND surfaces when the quality metric either improves or degrades monotonically after the control input is changed with respect to ref. We show the ideal JND surfaces, the upper ones with respect to a quality improved with an increased mod, and the lower ones with respect to a quality degraded with an increased mod. For example, when the system control is EED, the former one can be the signal quality and the latter one can be



(a) JND surfaces of quality metric improving with reference

(b) JND surfaces of quality metric degrading with reference

Figure 5.3: Synthetic JND surfaces (both axes normalized to [0,1]) when the quality metric is improving or degrading with respect to the control input. The bar shows the increased absolute awareness of the change with a larger modification, either showing the fraction of subjects who can correctly identify the output with better (the top figure) or poorer (the bottom figure) quality caused by an increased mod. An absolute value of awareness indicates the fraction, whereas a negative value indicates a degradation in perceptual quality.



(a) Per. quality for Fig. 5.3a Left.

(b) Per. quality for Fig. 5.3b Left.

Figure 5.4: Relation between awareness and relative perceptual quality. The horizontal color bars (obtained from Figure 5.3) show how the absolute value of awareness increases as the modification is increased. a) The awareness of the improved quality becomes larger as the control input increases with respect to a fixed reference ref_1. The increase is slower when the fixed reference is ref_2. b) The awareness of the degraded quality becomes larger as the modification is increased. Note that the y-axis only shows the relative magnitude of perceptual quality.

the interactivity in VoIP or videoconferencing. In online games, the former one can be the precision of the states in the game, and the latter can be the interactivity.

Next, we study the trade-offs among quality metrics. When ref is given, we illustrate in Figure 5.4 the relative perceptual quality in Figure 5.3. Consider two references ref_1 and ref_2. Starting from ref_1 in Figure 5.4a, the absolute value of awareness increases monotonically (because perceptual quality is better) when the modified control input $ref_1 + mod$ is larger. This same applies to ref_2, but the increase is slower, as perceptual quality increases with a slower trend. Figure 5.4b shows a similar behavior, where negative awareness indicates a degraded quality after the change.

The above observation illustrates an important relation between awareness and relative perceptual quality, namely, when compared to the same reference, a higher awareness indicates better perceptual quality. This is stated formally as follows.

Lemma 5.3.1. *Let $Q(x)$ be the perceptual quality of a simplex quality metric controlled by input x. With given reference ref,*

$$\begin{aligned} & p(ref, mod_1) \ > \ p(ref, mod_2) \\ & \quad \Rightarrow \ Q(ref + mod_1) \ > \ Q(ref + mod_2) \qquad (5.2) \\ & Q(ref + mod_1) \ > \ Q(ref + mod_2) \\ & \quad \Rightarrow \ p(ref, mod_1) \geq p(ref, mod_2). \qquad (5.3) \end{aligned}$$

With Lemma 5.3.1, the (relative) perceptual quality of a quality metric can be fully evaluated by the corresponding JND surface.

Figure 5.4 clearly illustrates the trade-offs among quality metrics. It shows that two quality metrics change in opposite directions when mod increases (note that the number of metrics is not limited to 2). With a larger mod, we can have better quality in Figure 5.4a, whileas we have a poorer quality in Figure 5.4b. How we can find the optimal setting of the system control, i.e. $ref + mod_{\text{opt}}$, is a problem that needs to be solved in the optimization. As an example, in VoIP or videoconferencing, only when EED is increased to the deadline of an additional parity packet can ASQ be improved. Obviously, there are trade-offs between short and long EEDs. While a short EED can provide good interactivity (discussed in Chapter 2), a long EED can allow the receipt of more parity packets, resulting in a better ASQ. As another example, in online games, a longer buffering time can provide more precise update of the states in the game, but the delay can worsen a player's experience of the operation.

5.3.2 Combining JND Surfaces Regarding a Single Control

We have discussed the algorithm for searching the optimal dependent system controls in Section 4.5. However, it is not an efficient online algorithm. In this section, we present an efficient method for obtaining the optimal system control. This method is based on an assumption that quality metrics relating to the same system control are independent in human perception.

Assumption 5.3.1. ***Independence***. *The quality metrics corresponding to the same system control are independent in human perception.*

This assumption can be justified by that humans handle different sensations mainly by different parts in the brain. While overlap can exist, they are not significant and can be ignored in the simplification.

We define the combined awareness to measure the result of a pairwise comparison when evaluating the relative perceptual quality Q of an application (see Definition 5.3.1).

Definition 5.3.2. *The* ***combined awareness*** *of multiple independent JND surfaces is the fraction of sufficiently many subjects who can notice the output caused by a changed input to have better perceptual quality than the output caused by the original input.*

Recall that awareness of a quality metric represents the fraction of subjects who can correctly identify the output caused by a modified reference from the original when they are shown in a random order. Although this is a probabilistic concept, the combined awareness of multiple independent simplex quality metrics is not simply a product of their awareness because awareness is not independent even when the corresponding metrics are independent. For example, when there is no modification, the awareness of ASQ and interactivity is 50%, but the combined awareness is still 50%, not $1-(1-0.5)(1-0.5)=0.75$.

To allow awareness of multiple independent metrics to be interpreted probabilistically, we convert it into *noticeability* previously defined in Chapter 4. To simplify our definition, we use in this section the absolute value of p as awareness.

Definition 5.3.3. *Noticeability $\mu(ref, mod)$ is the fraction of N subjects who can correctly perceive the change after ref is changed by mod when N is sufficiently large.*

For the same subjective test, awareness p and noticeability μ are related to each other as follows:

$$\mu = 2p - 1. \tag{5.4}$$

Before we can calculate the combined awareness, we need the following result to relate the perceptual quality of multiple simplex metrics and the combined relative perceptual quality. For simplicity, we only present the result for cases with two simplex quality metrics. A general case with more than two quality metrics can be similarly derived.

Axiom 5.3.1. *Let Q_1 and Q_2 (*resp. *$Q_1^{'}$ and $Q_2^{'}$) be the relative perceptual quality of two simplex quality metrics with respect to the original (*resp. *modified) control inputs. Further, let Q_{comb} and*

$Q^{'}_{comb}$ be the corresponding combined relative perceptual quality. Then,

a) $Q^{'}_1 > Q_1$ *and* $Q^{'}_2 \geq Q_2 \Rightarrow Q^{'}_{comb} > Q_{comb}$

b) $Q^{'}_1 \leq Q_1$ *and* $Q^{'}_2 < Q_2 \Rightarrow Q^{'}_{comb} < Q_{comb}$

c) $Q^{'}_1 = Q_1$ *and* $Q^{'}_2 = Q_2 \Rightarrow Q^{'}_{comb} = Q_{comb}$

d) $Q^{'}_1 > Q_1$ *and* $Q^{'}_2 < Q_2 \Rightarrow Q^{'}_{comb}\ ?\ Q_{comb}$

e) $Q^{'}_1 < Q_1$ *and* $Q^{'}_2 > Q_2 \Rightarrow Q^{'}_{comb}\ ?\ Q_{comb}$.

The first two conditions in the axiom can be explained as follows. When we present two alternative outputs for a system with two simplex quality metrics, a subject will always be able to identify the alternative with better perceptual quality if the perceptual quality of one quality metric is improved while the other is not degraded.

The third condition corresponds to the case when subjects cannot identify a change in perceptual quality for both quality metrics. As a result, subjects will respond by a random guess.

The last two conditions correspond to cases where one metric is improved and the other is degraded. Depending on the amount of modification with respect to the reference, it is possible that subjects may notice better, the same, or worse overall perceptual quality between the outputs corresponding to the reference and the modified reference. The outcome will not be known until actual subjective tests are performed.

For the last two cases, we assume that the probability for them to happen is low for the majority of references and modifications (to be verified at the end of this section). Under this assumption, we simplify their combined awareness to be 0.5; that is, subjects will respond by random guesses. This simplification may generate some small errors in awareness. However, these errors are only large in extreme regions in the combined JND surface but not significant near the regions of interest.

This method can be easily extended to the case with multiple quality metrics, by simply using a voting scheme to determine whether the improvement is more significant than the degradation.

To compute P_{comb}, the combined awareness of two simplex quality metrics when the control input changes from ref to $ref + mod$, let μ_1 (*resp.* μ_2) be the fraction of subjects who notice an improvement (*resp.* degradation) for the two quality metrics. According to the independence assumption, the fraction of subjects is:

$$\begin{cases} \mu_1(1-\mu_2) & \text{who prefer modification} \\ \mu_2(1-\mu_1) & \text{who do not prefer modification} \\ (1-\mu_1)(1-\mu_2) & \text{who find no difference} \\ \mu_1\mu_2 & \text{who find improvement in the} \\ & \text{first and degradation in second} \end{cases} \tag{5.5}$$

The combined awareness is then calculated as:

$$\begin{aligned} P_{\text{comb}} &= \mu_1(1-\mu_2) \times 1 + \mu_2(1-\mu_1) \times 0 \\ &\quad +(1-\mu_1)(1-\mu_2) \times 0.5 + \mu_1\mu_2 \times 0.5 \\ &= \frac{1+\mu_1-\mu_2}{2} \\ &= 0.5 + p_1 - p_2. \end{aligned} \tag{5.6}$$

According to Definition 5.3.2, $P_{\text{comb}} = 1$ are for subjects who choose the modified input (corresponding to the first case in (5.5)); $P_{\text{comb}} =$

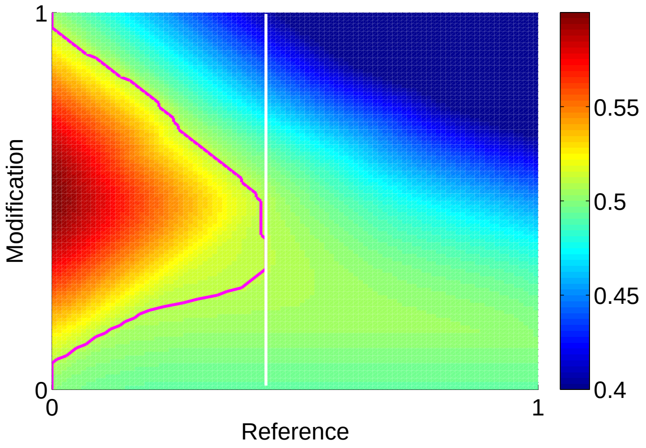


Figure 5.5: Simplified combined JND surface of two synthetic surfaces in Figure 5.3. Awareness $p > 50\%$ indicates subjects prefer an increased reference, whereas $p < 50\%$ indicates subjects prefer the original reference. The pink curve bounds the region where awareness is larger than 50%. The solid line indicates the local optimum in the reference.

0 are for subjects who choose the original input (corresponding to the second case in (5.5)); and $P_{\text{comb}} = 0.5$ are for subjects who make a random guess (corresponding to the last two cases in (5.5)).

Figure 5.5 depicts the resulting JND surface derived using (5.6) when combining the two synthetic JND surfaces in the left panels of Figure 5.3.

Note that in computing P_{comb}, the contribution of those subjects who find improvement in one metric but degradation in the other in (5.6) is $0.5\mu_1\mu_2$. Figure 5.6 illustrates the value of this term for every point in the combined JND surface in Figure 5.5. The result shows that the amount is small throughout the bottom and the middle parts of the surface, which are the regions of interest and contain the local maxima in relative perceptual quality.

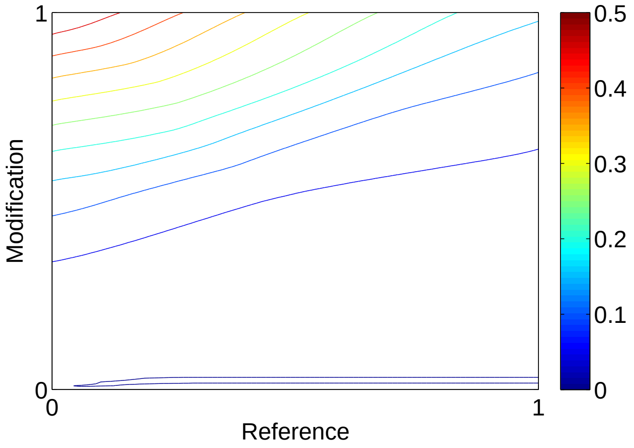


Figure 5.6: The contour shows the value of $0.5\mu_1\mu_2$ in (5.6). The values at the bottom and middle parts are small. These regions are of interest and contain the local maxima in relative perceptual quality. Its small value means that this term will not significantly affect the search result.

5.3.3 Best Operating Points Using The Combined Surface

Given the JND surface of the combined awareness, we can derive the resulting relative perceptual quality. The result will allow us to find the best control input that gives the best relative perceptual quality. The following corollary is used to search for the local maxima in perceptual quality.

Corollary 5.3.1. *For any given δ,*

- *$Q(x)$ is the local maximum in $[x, x+\delta]$ if $P(x, y) \leq 0.5$ for all $0 \leq y \leq \delta$;*
- *$Q(x)$ is not the local maximum in $[x, x+\delta]$ if $P(x, y) > 0.5$ for any $0 \leq y \leq \delta$.*

Proof. Based on Definition 5.3.1, the first part follows from the fact that $P(x, y) \leq 0.5$ indicates $Q(x) \geq Q(x+y)$. Similarly,

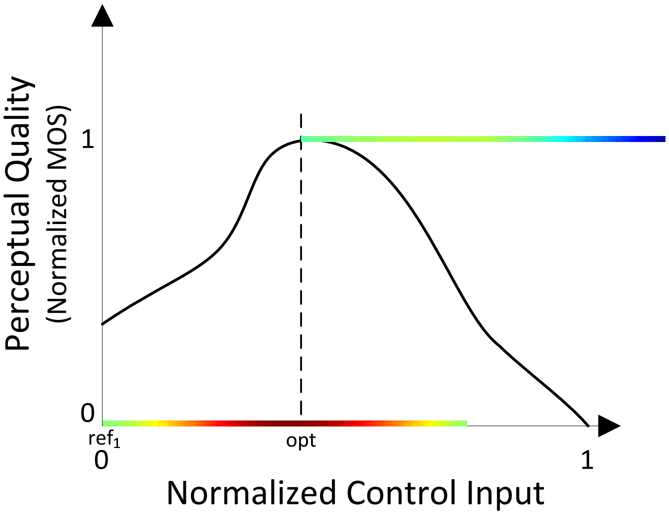


Figure 5.7: The resulting overall relative perceptual quality derived from the combined JND surface in Figure 5.5. The surface indicates a better change with $P > 0.5$, a poorer change with $P < 0.5$, and no change with $P = 0.5$. The awareness bar starting from ref_1 shows the awareness increasing until the local maximum opt is reached. The awareness bar starting from opt shows the awareness decreasing from 0.5 as the reference increases.

the second part follows from the fact that $P(x, y) > 0.5$ indicates $Q(x) < Q(x + y)$. □

In the corollary we only define the local maximum from x to $x + \delta$ because in a JND surface we can only measure awareness when the control input is increased. However, if we only check for optimality on one side, it is possible there exists $x - \delta \leq x^{'} < x$ such that $Q(x^{'}) > Q(x) > Q(x + y)$ where $0 \leq y \leq \delta$. To assure that the result is also the maximum on the other side of the region, we look for the first ref in the combined JND surface that satisfies $P(ref, y) \leq 0.5$ for all $0 \leq y \leq \delta$. Then there should not exist $Q(ref - \delta) > Q(ref)$; otherwise, $ref - \delta$ can be found to satisfy the condition, considering that δ is sufficiently small.

In practice, we like to find the local maximum within the largest possible range, and the largest δ we can identify in the JND surface is the range of the increase $y_{\max}$. Therefore, we discard any ref that does not satisfy $P(ref, y) \leq 0.5$, where $0 \leq y \leq y_{\max}$, until we find the first ref that satisfies the condition. Figure 5.5 illustrates the region where we have discarded the ref (bounded by the pink curve), as well as the first ref that satisfies the condition (indicated by the solid white line). Figure 5.7 further shows the resulting relative perceptual quality derived from the combined JND surface.

As there may be multiple local maxima in perceptual quality depending on δ, we can reduce δ from $y_{\max}$ to find other ref if necessary.

5.3.4 Complexity

The combination of JND surfaces require only a computation for each point in the final combined surface; therefore, the computational complexity is $O(k^2)$, where k is the resolution of each JND surface through the X and Y axes.

The search of the optimal solution in the combined JND surface also requires a one-pass search over the combined JND surface; therefore, the complexity is also $O(k^2)$.

When there are multiple system controls, we can classify all n JND surfaces by system controls, and combine the JND surfaces corresponding to the same system control. If all the JND surfaces corresponding to the same system control are independent, we can at best achieve $O(mk^{(n/m)})$, where m is the number of system controls. This is significantly faster than the algorithm in Section 4.5 which has a $O(k^n)$ complexity. If some of them are dependent, then they can still be used in the algorithm proposed in Section 4.5 to find the optimal control along with the combined JND surfaces obtained by the algorithm in this chapter.

5.4 Summary

In this chapter, we have proposed efficient online algorithms for generalizing a JND surface to any online network condition as well as accelerating the algorithm for the optimization. The analysis of the computational complexity clearly shows that our algorithms can meet the requirement of fast-paced interactive multimedia applications.

As have been shown in Table 4.2, by now we have provided solutions for single independent control with both simple and complex constraints, as well as multiple dependent controls with simple constraints. We also have application-dependent constraints for reducing the search space when solving the optimization problem under dependent controls with complex constraints, which will be discussed in Section 7.3.3. We do not have efficient solutions for multiple independent controls due to the high complexity in the optimization.

□ **End of chapter.**

Chapter 6

Evaluations in Videoconferencing

In this chapter, we demonstrate how we can improve the perceptual quality of videoconferencing systems using the method proposed in this thesis. We present how we can optimize existing proprietary videoconferencing systems without the source code using our network-control layer and our JND based method. Next, we replace the audio codec of that system with an open-source voice-only codec and to present how our method tackles trade-offs and find the optimal system control.

6.1 Improving Proprietary Videoconferencing Systems with Our Proposed Methods

Video conferencing systems are popular nowadays for social as well as business communications. Free systems like Skype (v5.10.0.116 in this study) and Windows Live Messenger (v15.4.3555.308) have attracted many users, but their quality may not be consistent under different network conditions. Commercial systems, in contrast, have more consistent quality but have high initial investments and some have high operating costs.

To achieve good QoE in a video conferencing system under given network and conversational conditions, it is important to operate the system at an *operating point* with a set of properly chosen parameters. Although finding the best point is difficult, it is possible for subjects to compare in a relative sense the perceptual quality of two operating points and to identify whether one is better than or indistinguishable from another. Such a comparison entails trade-offs among the objective metrics, where some metrics may lead to perceptual improvement while others may cause degradations.

In comparing operating points of existing systems, we have found that many of them are sub-optimal. Some overly emphasize interactivity without sufficient attention to signal quality. In some cases, the EED is not sufficient to cover the network delay as well as the buffering time to smooth delay jitters and to recover lost packets. Without proper trade-offs between signal quality and interactivity, the overall QoE will be low. Our experimental results have shown that signal quality can be significantly improved if EED is slightly extended. To address this trade-off, it is important to study the extent of extending EED so that the degraded interactivity will not be perceived.

We address this question by using JND. In the context of interactive video conferencing under the same network condition, JND defines a range of EEDs from the original EED (operating point) within which humans cannot perceive any difference in interactivity (in a statistical sense) between the original and the new operating points. For those EEDs in the JND, we are interested in the maximum EED. By increasing the original EED to this maximum EED, we can improve the quality of the system through additional loss concealment mechanisms, without incurring perceptible changes in interactivity.

6.1.1 Quantitative Metrics

For evaluating one-way quantitative video quality, we adopt a standardized metric called Video Quality Metric (VQM) [69]. By pooling various factors in a linear fashion into an overall metric, VQM demonstrates a higher correlation to the subjective mean opinion score (MOS) than traditional signal quality metrics like PSNR. VQM generally maps MOS into $[0.0, 1.0]$ range, with a smaller value representing better subjective quality and $VQM = 0$ implying a lossless quality. On the other hand, for evaluating one-way quantitative audio quality, we adopt an International Telecommunication Union (ITU) standard called Perceptual Evaluation of Speech Quality (PESQ) [6] that uses a human-speech model to capture the various factors affecting perceptual quality. Again, it has a high correlation to perceptual MOS. The range of PESQ is $[-0.5, 4.5]$, and the larger the better. Note that both VQM and PESQ can only be used in offline tests because they require the original sequence as a reference, in addition to their high computational complexities.

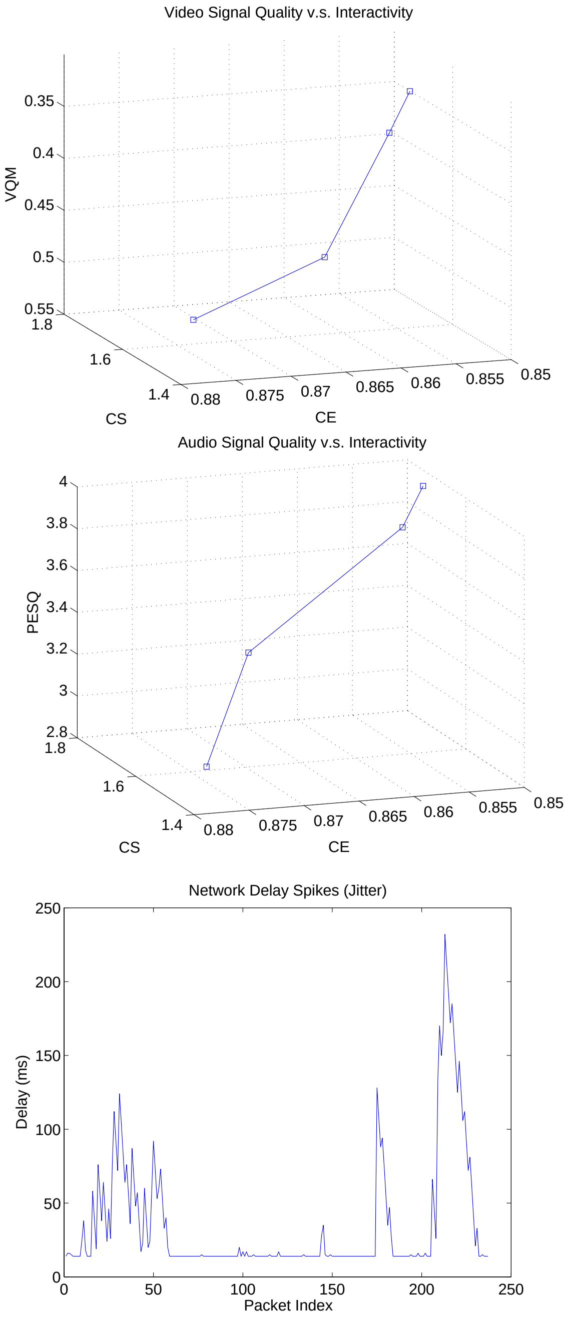
As mentioned in Chapter 2, three metrics can be used to measure interactivity: EED, conversational symmetry (CS) and conversational efficiency (CE) [76]. In a face-to-face conversation, (EED, CS, CE) is $(0, 1.0, 1.0)$. Small EED and CS as well as large CE indicate good interactivity. When EED is increased, CS becomes larger and CE, smaller, which reflect a degradation in interactivity.

In short, we use five quantitative metrics to measure QoE: VQM, PESQ, EED, CS, and CE. We do not consider QoS (quality-of-service) metrics like loss rate, delay, jitter and throughput because their effects have been considered in the above metrics. Moreover, some of them may not be available in proprietary systems like Skype (and thus cannot be used to measure quality). For this reason, we do not use QoS metrics like the E-model [47] and G.1070 [89].

Given the five metrics presented above, trade-offs must be made among them in order to arrive at the best QoE. For example, a larger jitter buffer (which is a part of the overall EED) can mitigate the late arrivals of packets due to network jitters and provide more time for receiving redundant data in recovering lost packets. However, it will cause a longer waiting time for users to receive replies in a conversation. On the other hand, a shorter jitter buffer reduces the chance that a late packet can be received and a lost packet be recovered, and thus degrades the signal quality [76].

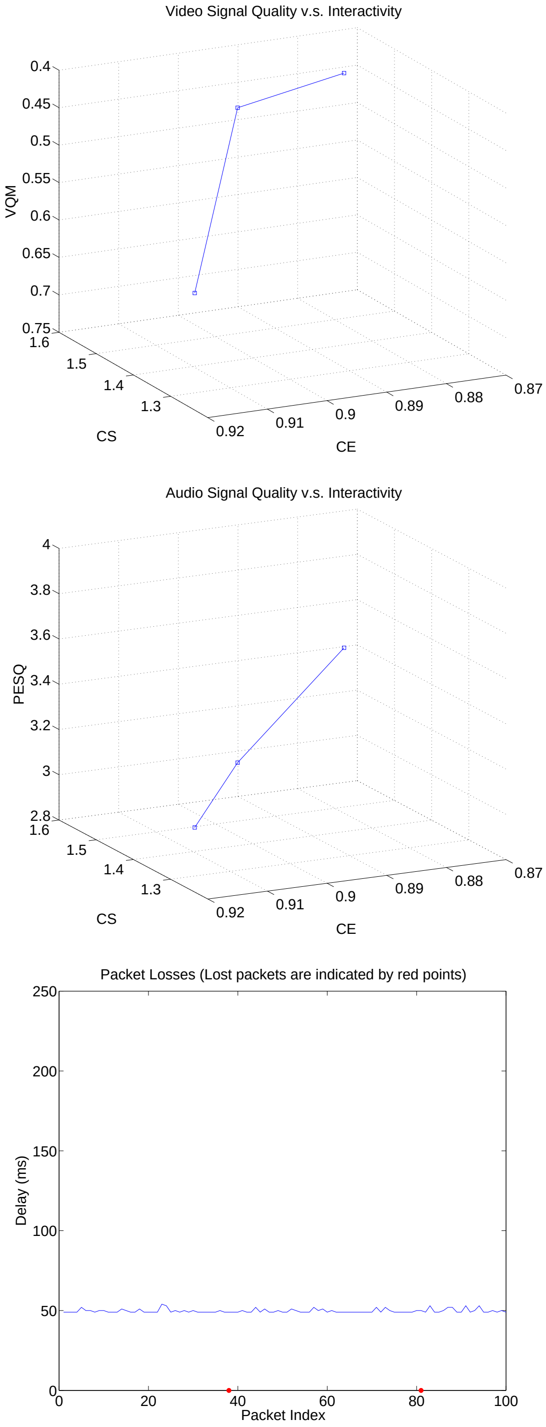
Figure 6.1 shows the relationship between VQM (*respectively* PESQ) versus CS and CE in Skype [84] and Windows Live Messenger (MSN for short) [61] under error-prone connections. The performance was measured by a testbed *RealTalk* (details in Section 6.1.2) we have developed for improving the performance of proprietary systems. By increasing EED, we can use the additional buffering delay to better protect the multimedia data, leading to better one-way video and audio qualities. At the same time, as mentioned above, the interactivity of the conversation may suffer.

The results in Figure 6.1 show that EED affects not only VQM and PESQ but CS and CE as well. Further, the default operating points (EEDs) of both Skype and MSN under these network condi-



a) Skype (under a jittery network condition shown in the third subfigure)

Figure 6.1: Video and audio signal qualities versus interactivity in Skype and MSN. The default operating points of these systems are at the bottom left (with MED of 255 ms for Skype and 176 ms for MSN). By increasing MED to the top-right operating point (with MED of 310 ms for Skype and 257 ms for MSN), VQM and PESQ are both improved while CS and CE are both degraded.



b) MSN (under a lossy network condition shown in the third subfigure)

Figure 6.1: Video and audio signal qualities versus interactivity in Skype and MSN. (cont.)

tions are sub-optimal, leading to poor video and audio qualities. Our results further show that their quality can be improved by increasing EED. An interesting question is, therefore, to what extent can EED be increased, without incurring significant degradation on interactivity that can be perceived by users? We answer this question by using the concept of JND.

6.1.2 Methodology

In this section, we present the loss-concealment methods made possible by the extended EED to within JND. We then demonstrate the approach for improving the QoE of existing systems.

Loss Concealments by Extending EED

In a jittery network, packets may arrive late under large jitters. In this case, the playout buffer will underflow, leading to freezes and degraded QoE. To smooth network jitters, a longer playout buffer can be used to store more packets and to smooth delay jitters.

On the other hand, in a lossy network, packets may be dropped and media data lost, again leading to freezes and low QoE. To recover those lost packets in real time, FEC can be used to add redundancy in transmission and to allow lost packets to be reconstructed. This approach will require additional bandwidth for sending redundant data and a longer playout buffer for receiving all the packets in an FEC block before carrying out recovery.

The size of the playout buffer (and EED) can be increased by JND without being perceptible. With this extra time, we have more room to smooth out jitters and to recover lost packets. With EED increased by JND, the probability for a packet arriving late becomes

$$p_{\text{late}} = 1 - CDF(EED + JND(EED)),$$

where CDF is the cumulative distribution function of network delays. The probability that a packet is lost becomes

$$p_{\text{lost}} = \sum_{l=0}^{N_{\text{S}}-1} Pr(\text{only } l \text{ packets are received}),$$

where N_{S} is the number of source packets in an FEC block. The minimal buffering time for packets in an FEC block is

$$t_{\text{buffer}} = (N_{\text{S}} + N_{\text{R}} - 1)t_{\text{interval}},$$

where N_{R} is the number of redundant packets in an FEC block, and t_{interval} is the packet transmission period. With the additional JND, the number of redundant packets that can be used in FEC is

$$t_{\text{buffer}} \leq EED + JND(EED) - t_{\text{delay}},$$

where t_{delay} is a random variable of packet delay.

Here, $JND(EED)$ is obtained from a JND surface with the awareness set to 75%, which is generally considered the threshold by which human will not notice the difference.

Design of a the Network-Control Layer

In this section, we present the design of a network-control layer in the form of a packet interceptor [111] to capture and modify the packet traffic in existing video conferencing systems. Our approach has two advantages. First, it can enhance an existing system and

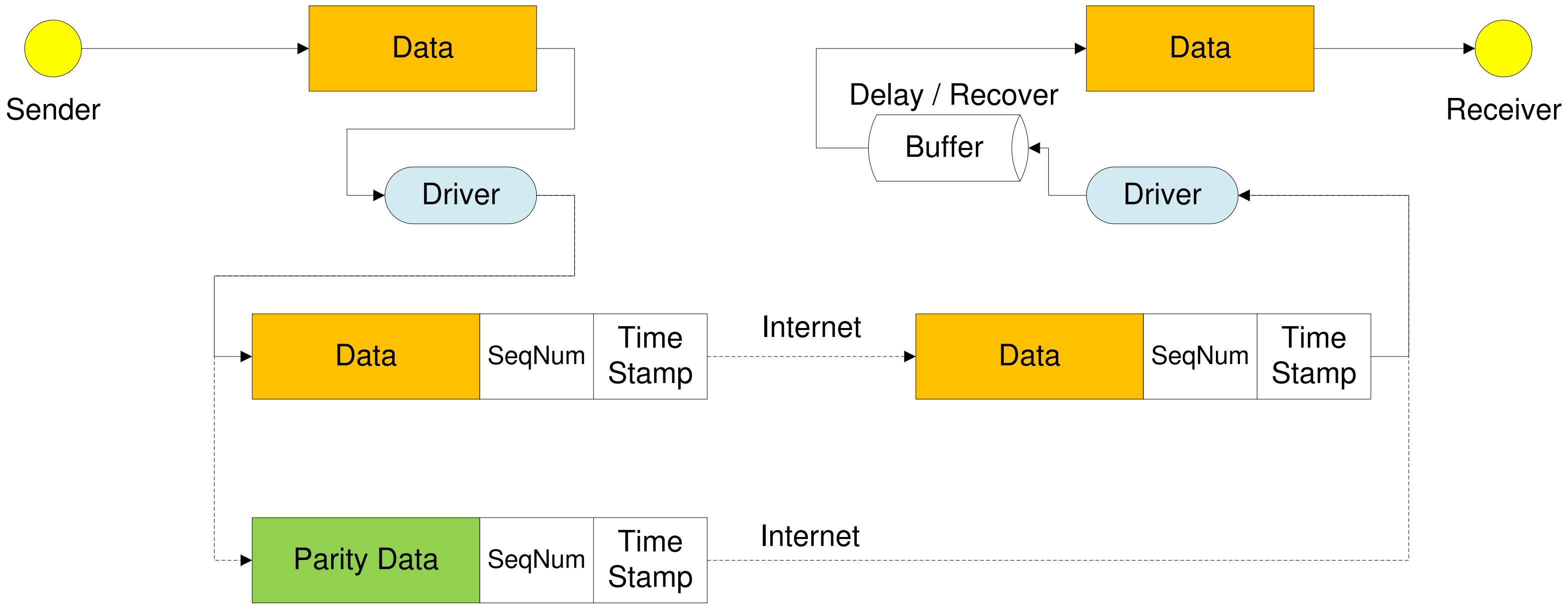


Figure 6.2: The interceptor deployed in Windows to intercept, modify and inject UDP traffic for proprietary systems.

allows its performance before and after to be fairly compared. Second, it can be readily accepted by users of existing systems because it improves their perceptual quality without changing to a new system.

We have picked Skype and MSN as our targets for improvement. By implementing the scheme in Windows kernel mode, it allows their traffic to be modified outside of their black-box designs. To do this in real time, we have developed a kernel driver using the Windows Filtering Platform [60]. This is part of the Windows Driver Kit in Windows 7 that provides ways to intercept, modify, and inject traffic in various layers.

In our implementation, we only intercept UDP traffic of the video conferencing system. As depicted in Figure 6.2, we add a sequence number and a time stamp to each packet before it is sent. In the receiver, the driver buffers the packets and releases each to the video conferencing system after reaching certain delay from the time each was sent. The additional data will not exceed the maximum transmission unit (MTU) of 1500 bytes, as the size of the largest packet produced by Skype is 1406 bytes and that by MSN is 1078 bytes.

For FEC protection, we send an extra packet with parity data for every several original packets. The parity packet is coded by Reed-Solomon code that allows a lost packet to be recovered if there is only one packet lost in the FEC block. After detecting a discontinuity in the sequence number, we recover the lost packet and send it to the video conferencing system. By succinctly choosing the size of the FEC block, the additional bandwidth incurred is small, while offering protection to the original audiovisual data.

The rate control is performed by the TFRC algorithm built in the proprietary videoconferencing systems. While this algorithm cannot be easily modified by the outer traffic filter, it already satisfies our requirements in the network-control layer, so we just keep it unmodified.

RealTalk: A Testbed for Evaluating QoE in Proprietary Systems

Figure 6.3 shows the architecture of *RealTalk*, a testbed for evaluating proprietary video conferencing systems. The testbed consists of two Windows 7 machines serving as the video conferencing clients. The clocks of these machines are synchronized by Net Time Protocol (NTP) to ensure accurate measurements of conversational delays. An additional Linux machine serving as a network emulator is connected to the two clients, where Trace Control for Netem (TCN) [51], a trace-based network emulator, was installed to emulate different network conditions using the traces collected.

To generate input video frames in real time under various frame sizes and rates, we have developed a virtual camera program with Microsoft DirectShow. Audio is injected by Virtual Audio Cable [63], a software for redirecting audio from one source to another.

In our testbed, each client behaves like a human that speaks and replies using a pre-recorded audiovisual sequence. After detecting the end of the other party's speech segment (using special markers inserted into the audiovisual stream to indicate the start and end times of each speech segment), it waits for HRD before playing the next segment. This approach allows our testbed to simulate conversations with different delays from a single audiovisual source.

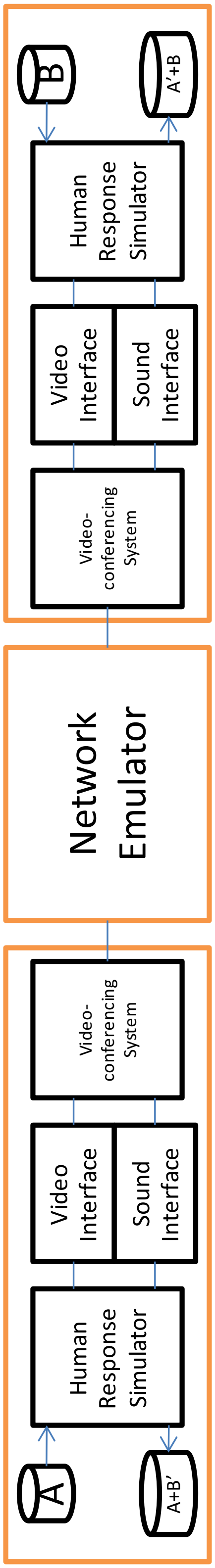


Figure 6.3: The architecture of *RealTalk*, an evaluation testbed for proprietary video conferencing systems.

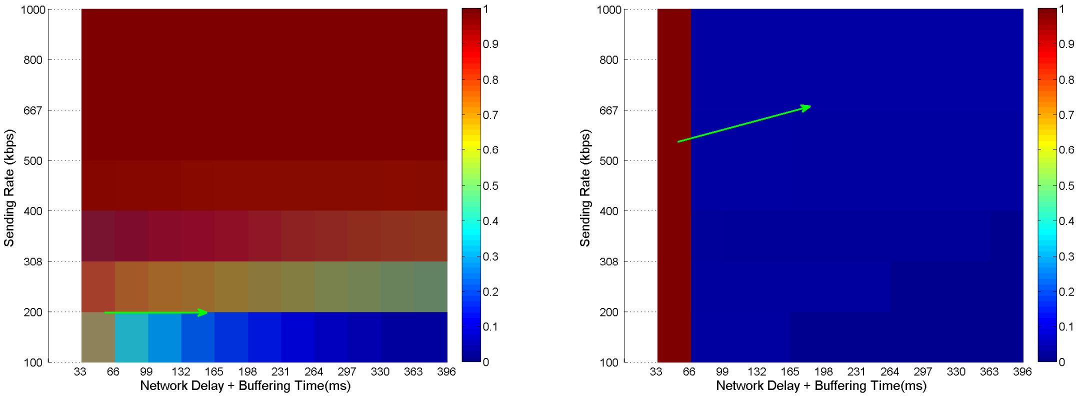
Table 6.1: QoE results of our proposed scheme with Skype v5.10.0.116 and Windows Live Messenger (MSN) v15.4.3555.308 in 2012.

System	Interceptor	VQM	PESQ	EED (ms)	CS	CE	% of Subjects Preferring Scheme
Skype	Off	0.54	3.77	239	1.54	0.88	0%
	On	0.36	3.36	251	1.57	0.88	100%
MSN	Off	0.76	3.08	276	1.63	0.87	0%
	On	0.41	3.72	363	1.82	0.83	100%

Experimental Results

We have found that Skype performs poorly under a jittery connection. Based on the 100-ms JND found by the JND surface, our network-control layer adds a 100-ms buffer to Skype to smooth its jitters. To ensure the same sending rate, we enforce the maximum bandwidth in the TCN network emulator. (Otherwise, the sender client in Skype will increase its sending rate after the jitters have been removed by our interceptor.) One way to limit the bandwidth without the TCN is to include another interceptor at the sender client.

Table 6.1 shows significant improvements in video quality with the use of our interceptor, where VQM has improved by 33%. Audio quality is decreased slightly by 11%, without being perceived in subjective tests. The degraded audio quality may be a result of the higher video quality. It is surprising to find that the overall MED increases by only 12 ms with 100 ms buffers inserted by our interceptor. This nonlinear change in MED may be caused by some time-consuming loss-concealment functions in Skype, which are bypassed when Skype finds a lower UPR. Figure 6.4a further



(a) Skype under a jittery connection

(b) MSN under a lossy connection

Figure 6.4: The change in UPR by using our traffic interceptor, where the green arrow indicates the change in the operating points. In MSN, the packet sending rate is increased by 20% in order to implement FEC, although the increased sending rate does not change the average loss rate and average jitter.

illustrates the network condition used for testing Skype. The new operating point has significantly lower UPR, leading to better signal quality and a non-perceptible change in interactivity. Note that the reduced UPR is a result of the extended MED for smoothing delay jitters or performing FEC. Without the interceptor, there is no time to implement these loss-concealment schemes.

For MSN, we found that it performed poorly under lossy conditions. With a lossy connection and a high turn frequency, the JND is 138 ms (also obtained in with a JND surface). We thus add a 138-ms buffer in our interceptor to conceal those lost packets using FEC. Table 6.1 shows that, under the same network setting, video quality is improved by 46% and the audio quality is improved by 21%. The final MED is increased by 87 ms, which is within the target JND. For a similar reason as that in Skype, the better signal quality is due to the significant reduction in UPR (Figure 6.4b).

The last column of Table 6.1 also shows the results of the subjective tests conducted to determine the quality of Skype and MSN before and after deploying the interceptor. A total of 8 subjects were invited to perform the tests, and all found that the quality to be better with the interceptor included. Another experiment on an older version of Google Video Conferencing v3.2.4.8431 in Gmail [30] also showed the improvement of using our proposed interceptor in lossy networks. However, Google changed its design in 2012 from a peer-to-peer architecture to a server-client architecture, and it is not possible to use *RealTalk* to replay traces with this new architecture.

Our results clearly show that our scheme can improve the QoE of existing video conferencing systems, without incurring perceptible degradations on interactivity.

6.2 Optimizing the Voice Module

6.2.1 Setup

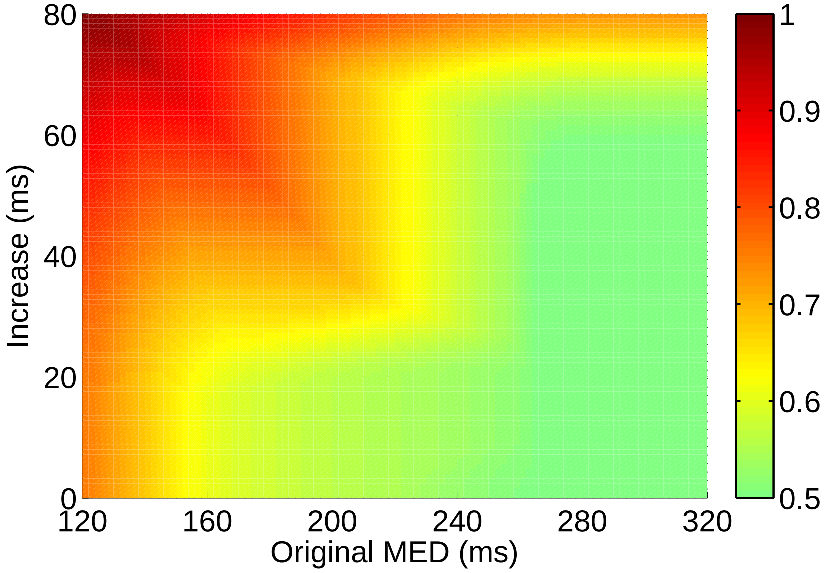
While the proprietary videoconferencing systems are too complex to modify because they are not open-source, the audio codec in the system can be easily replaced by an open-source codec. Because the transmission bandwidth of the audio is considerably low when compared to the transmission bandwidth of the video codec, muting the voice in the system and transmit our own voice traffic does not incur many changes in the traffic behavior of the proprietary videoconferencing systems. This provides us a way to verify the quality of our online optimization algorithm.

We have implemented such a codec using the ITU recommended audio codec G.722.2 with the highest voice quality. As the bitrate is low even with this setting, we do not care about the congestion control. The transmission of the packets with the voice data is simulated with network traces collected from PlanetLab, and the network-control layer is used for buffering the packets and recovering lost packets.

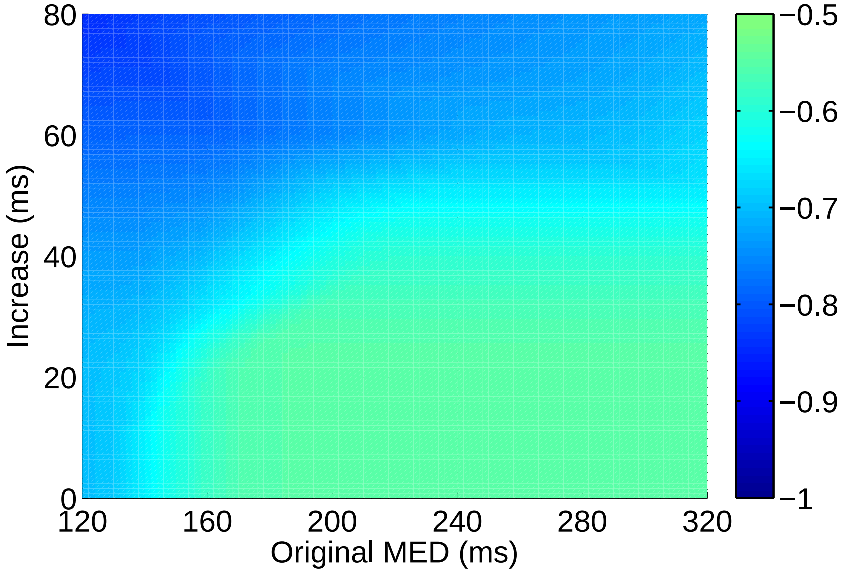
6.2.2 JND Surfaces

Figure 6.5 shows the JND surfaces of the signal quality and the interactivity respectively with our voice codec. The step pattern in Figure 5.3a is due to the discrete time interval for receiving parity packets in an error-prone network. Only when EED is increased to the deadline of an additional parity packet can ASQ be improved.

With the JND surfaces Figure 6.5, we use the algorithm for combining independent quality metrics in Chapter 5 to generate the corresponding combined JND surface in the left panel in Figure 6.6a. The independence of the audio signal quality and the interactivity can be justified by that a slow conversation and a fast conversation does not have significant impact on voice quality, and vice versa. The local maxima is identified by the right solid lines in the left panel, which correspond to the best EED for achieving high perceptual quality. The left solid line illustrates another local maximum we can find when δ is reduced. The right panel in Figure 6.6a illustrates the relative perceptual quality derived from the JND surface.

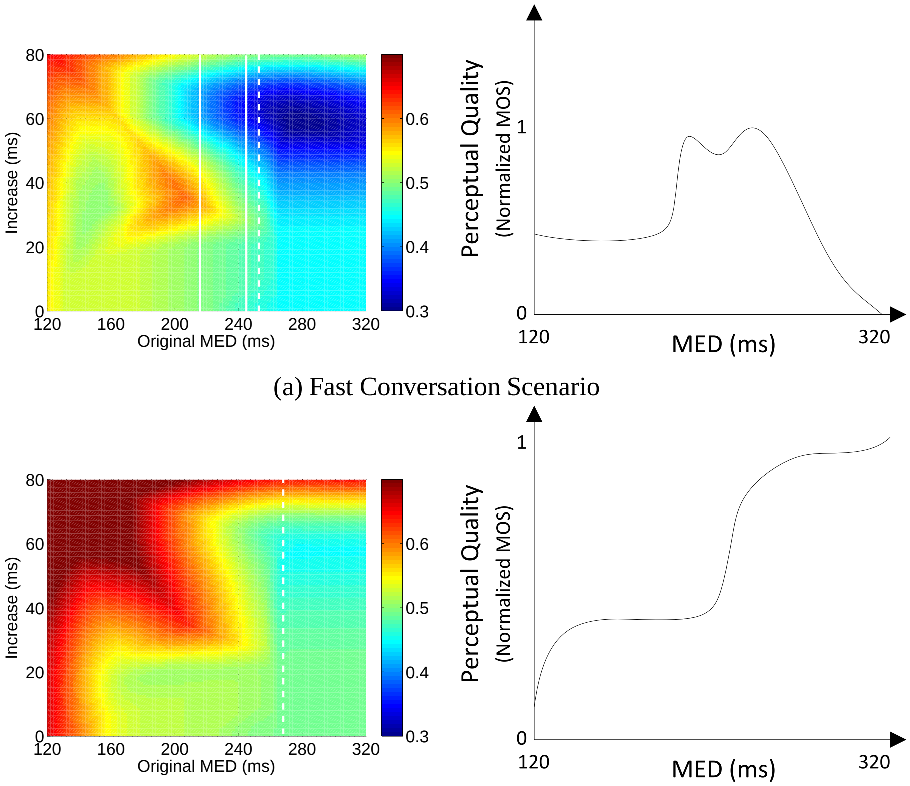


(a) JND surfaces of quality metric improved as the system control increases



(b) JND surfaces of quality metric degraded as the system control increases

Figure 6.5: JND surfaces in VoIP in an error-prone network showing the fraction of subjects who can correctly identify the output with better ASQ (*resp.* poorer interactivity) caused by an increased EED. An absolute value of awareness indicates the fraction, whereas a negative value indicates a degradation in perceptual quality.



(a) Fast Conversation Scenario

(b) Slow Conversation Scenario

Figure 6.6: The combined JND surface and the corresponding relative perceptual quality for the VoIP application using the surfaces on ASQ and interactivity in Figure 6.5. The optimal controls found by our method is shown in the solid white lines, whereas that found by the previous method [77] in the dashed white line. a) The surface in a fast-conversation scenario shows the optimal EED $ref = 240$ and the second optimal EED $ref = 210$. The first optimal EED is better because $P(210, 30) > 0.5$. The dashed line indicates the optimal EED $ref = 245$ found by the previous method [77], which has a similar performance. b) The surface in a slow-conversation scenario shows a much larger optimal EED. Even when $P(320, 80) > 0.9$, $ref = 400$ is found to be a better EED than $ref = 320$. In this case, we use the maximum possible EED $ref = 400$ as the optimal control. (The solid line $ref = 400$ is outside the range of the x-axis.) The optimal EED found by the previous method is $ref = 270$, which has poorer perceptual quality than ours: $P(270, 80) > 0.9$ and $P(270, 130) = 1.0$.

6.2.3 Comparison with Related Method

As aforementioned in Chapter 2, the method by Sat and Wah [77] is most relevant to our method for optimizing multimedia systems that do not have well-defined objective metrics. We thus compare our method with this previous method.

Solution quality Figure 6.6 shows the optimal controls found by our method and the peer method. The solid white lines in Figure 6.6a show the two best EEDs found by our method, whereas the dashed line shows the best EED found by the peer method. The resulting optimal EEDs are very similar with comparable performance. Figure 6.6b shows our optimal EED is 400ms which is the maximum possible of the system, and the optimal EED of the peer method is 270ms. The awareness shows that when the original EED is 270ms and the increase is larger than 80ms, nearly 100% of the subjects will prefer the increase, which shows that our EED provides better perceptual quality than the peer's.

Cost of subjective tests As discussed in Section 6.2.4, because we have decomposed the measure of awareness on interactivity and signal quality, we do not need to perform subjective tests for finding the optimal MED when network latency changes. In comparison, the peer method needs new subjective tests for each condition of network latency. Depending on the discretization of latency, the subjective tests will need to be repeated several times. Moreover, when the conversational scenario changes, the peer method has to redo the subjective tests for each network latency. As an example, if network latency is discretized to 3 levels and the conversation has 4 scenarios, the peer method will need to perform subjective tests in 3×4 cases which are combinatorially increasing, while our method only needs to measure $1 + 4$ cases which are linearly increasing. Obviously our method will need fewer subjective tests when the combination of running conditions is larger.

6.2.4 Generalization of the Results in VoIP

Besides the generalization discussed in Chapter 5, another generalization is needed when interactivity changes. In VoIP, the conversational behavior may change when a different topic is discussed. A business conversation may have a fast turn frequency, whereas a casual conversation may instead be slow. These two conversational conditions will lead to different JND surfaces on interactivity. Figure 6.6a (*resp.* Figure 6.6b) illustrates the JND surface based on the interactivity surface measured for a fast (*resp.* slow) conversation and the common ASQ surface in Figure 6.5a. We find that the two surfaces are not very different, considering the distribution of the improved (red) and degraded (blue) regions. Hence, interpolations can be made between the two JND surfaces for interactivities in between. All these computations can be done at run time.

6.3 Summary

In this chapter we have demonstrated how we can optimize a VoIP system using JND surfaces. We further study the method for improving existing proprietary videoconferencing systems using JND.

We have attained better signal quality of audio and video when compared to existing popular videoconferencing software including Skype v5.10.0.116 and Windows Live Messenger v15.4.3555.308. We have further used our optimization algorithm to improve the audio quality, with far less subjective tests required than a previous method.

□ **End of chapter.**

Chapter 7

Evaluations in Multi-Player Online Games

In this chapter we use our proposed algorithm [112] in Section 4.5 for handling dependent quality metrics to reduce the delay effects while maintaining the consistency in a delay-sensitive multi-player online games BZFlag.

7.1 Background

Multi-player online games refer to computer games with multiple players who interact remotely in the real world over the Internet. Among them, *fast-paced online games* are increasingly popular with improvements in network bandwidth and reduced latency. Here, "fast-paced games" refers to games in which the reaction time required is near the limit of human reaction time (215 ms on average according to an online test [44]). We are interested in those games with action durations ranging from 300 ms to 700 ms, and in particular scenarios with precise weapons. Examples include shooting games with bullets or missiles, fighting games with fast punching or kicking, and racing games with weapons shooting enemies.

In these games, players stay in a common game world in *virtual space*, even though they may be separated in *physical space* in the real world. In virtual space, players perceive *virtual time*. Note that virtual time is not necessarily the same as *physical time* because it can be affected by network and buffering latencies.

An *action* is a translation or a movement of a virtual object triggered by a player that has some effects in virtual space. As an action will not have any effect before it completes, the order of actions is defined by their *completion order*.

We define the *reference order* of actions to be the order of completions in virtual space without network latency (*i.e.,* in a *virtual perfect world*). Figure 7.1(a) illustrates the reference order in a two-player game in which two players are shooting at the same target. A shaded box shows the start, duration, and completion of an action that represents the shooting of a bullet in a player's view in virtual space. When there is no network latency, the messages of A's and B's actions are immediately sent to the other player's virtual space. In both spaces, A's action terminates at t_1, and B's action terminates at t_2, where $t_2 > t_1$ defines the reference order. The example can be generalized to more than two players, and the orders perceived by all players in their virtual spaces are the same. In this case, virtual time is aligned to physical time.

In contrast, when there is network latency, the *actual order* of completions of actions in virtual space can be different from the reference order. Figure 7.1(b) shows that the messages of A's and B's actions are delayed by network latency $t_4 - t_3$ (assuming the same two-way latencies). In B's view, B's action still terminates at

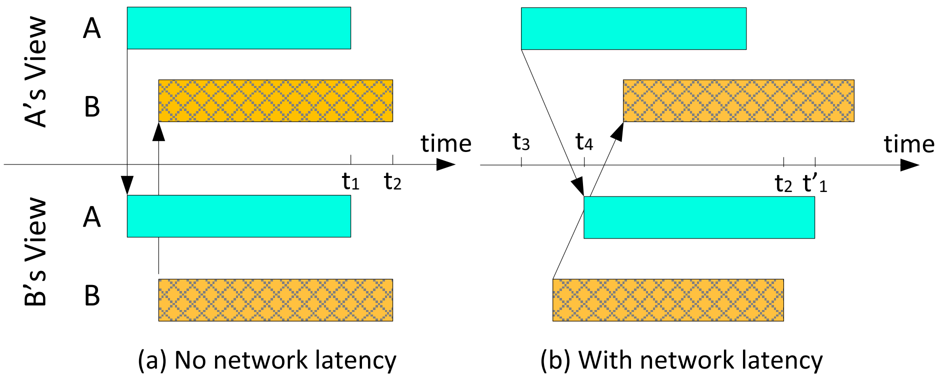


Figure 7.1: Physical network latencies can delay the completion of actions and cause the reordering of completions that are different when compared to the reference order.

t_2, but A's action is now delayed by $t_4 - t_3$ and terminates at $t_1^{'} = t_1 + t_4 - t_3$. Note that $t_1^{'} > t_2$ implies a reversed order of completions when compared to the reference order; that is, in A's view, A's action terminates earlier than B's action, whereas in B's view, A's action terminates later. We call this phenomenon the *reordering problem*.

Reordering is directly related to consistency and correctness in the continuous domain [58]. Consistency requires the state (action order in our context) at time t be the same in any two players' virtual spaces if both have completed all operations supposed to be executed before t. Correctness further requires the state to be the same as the virtual perfect site (reference order in our context).

In this chapter, we define *strong consistency* to mean identical actual and reference orders (thus satisfying consistency and correctness defined in [58]), and that the interval between any two completion times is unchanged when compared to the reference order. The latter requirement is important for the following reasons.

1. If the intervals of completion times of multiple actions are longer

than those in the reference, then the delays in the completion times will accumulate. A later action may, therefore, terminate significantly later, and its effect can be perceived.

2. If an interval is shorter than the reference, then the deadline of the corresponding action is moved up, and there may be difficulty to process the action.

Our definition also applies to the traditional well-studied multiplayer online games (MMOs), where consistency of states over clients is important. Related work has been reviewed in a recent survey paper [114]. Many techniques have been developed for maintaining consistency. Dead-reckoning is useful for predicting an unreceived future state if the corresponding play pattern is relatively simple [83]. Rollback moves virtual time backwards and replays the actions in the correct order once reordering is detected. It has been widely adopted because it is easy to implement. Recent approaches have focused on smooth correction techniques that repair inconsistent states during a game. However, subjective studies have found that such corrections are noticeable and annoying. In general, important factors that can affect players' detection of corrections [80] include a player's locus of attention, and the smoothness and duration of a correction.

The above approaches, however, can lead to perceptible artifacts in fast-paced games, like fighting, racing and shooting games, as soon as an inconsistency occurs and before it is fixed [88]. These games have higher requirements on keeping their states consistent than MMOs because the durations of their actions are very short and comparable to the network latency among clients. Players thus need

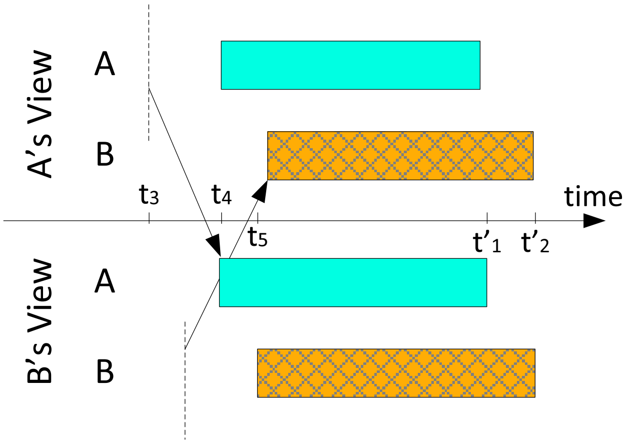


Figure 7.2: In delaying an action by the longest network latency, traditional local-lag algorithms solve the reordering problem by compensating the virtual delay caused by the network latency. However the artifact may be perceived by players, as a player has to wait for $t_4 - t_3$ before her action is carried out.

to pay high attention to all fast-paced actions. For example, in a fighting game, a defender should keep changing the way she guards when defending the constantly changing attack patterns. Such a high attention level as well as the short durations of actions can render the artifacts of rollback or correction perceptible [80].

A number of techniques have been developed to solve the above problem. The local-lag method [58] maintains consistency by delaying an action in virtual space slightly after a player initiates the action. Figure 7.2 illustrates the local-lag method. The longest network latency ($t_4 - t_3$) is first estimated. In A's view, A's action is delayed by $t_4 - t_3$ before it is transmitted to B. B's action in A's view is then forced to be delayed by $\delta_{B,B} = t_4 - t_3$ to t_5 in order to assure all actions that should start before this action in A to have enough time to inform B. The completion times of A's and B's actions are, respectively, $t_1^{'}$ and $t_2^{'}$ ($t_1^{'} < t_2^{'}$), which are the same as the reference order in Figure 7.1(a). Although this can avoid inconsistencies when compared to rollback and smooth correction techniques, it can lead to noticeably sluggish response.

To demonstrate that the local-lag strategy can produce noticeable delay effects, we modified the code of an open-source online tank-battle game BZFlag [64] to implement the strategy. The game has rapid actions as well as fast interactions.

To better represent fast-paced games, we modified the base speed of a bullet in BZFlag from the original setting of 100 units/s to 300 units/s. Using bullets of different durations, we hired 14 students to perform subjective tests and asked each to choose which setting had slower response: one in a reference network without latency, and another in a network with latency but with the local-lag strategy. If they could correctly identify the second setting, then the local-lag strategy was unable to conceal the delay effects.

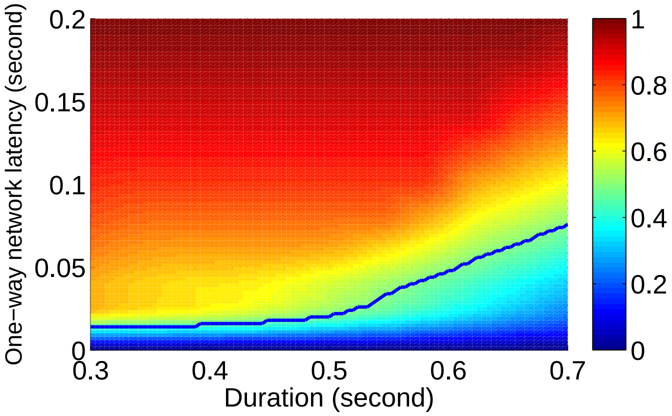


Figure 7.3: $P_{\text{notice}}$when using the local-lag strategy to conceal the effects of network latency in BZFlag. The x-axis shows the original action duration common in all virtual spaces; the y-axis is the one-way network latency. The dark blue curve shows the contour P_{notice}=50%.

Figure 7.3 displays the result using a JND surface measured with the algorithm proposed in Chapter 4. It shows that the local-lag strategy cannot conceal the delay effects because 50% of the subjects can identify the scenario with a short 25-ms one-way network latency, even when the action duration is 0.5 second (that covers about one third of the distance in the game map and is 20 times longer than the network latency).

Other methods like local perception filters [85], instead, modify the durations of actions in order to maintain consistency, but they can produce noticeable change of durations, especially in fast-paced actions. In this chapter, we call these *delay effects* because they cause players to feel “sluggishness” or “having undue slowness” in their games. These were also called “glitches” in some previous papers.

The *blank-period problem* occurs when the response of the target is unpredictable at the time the action from the attacker is completed. We illustrate it using a simple scenario with one attacker and one defender. There is no reordering here as there is only one action from the attacker.

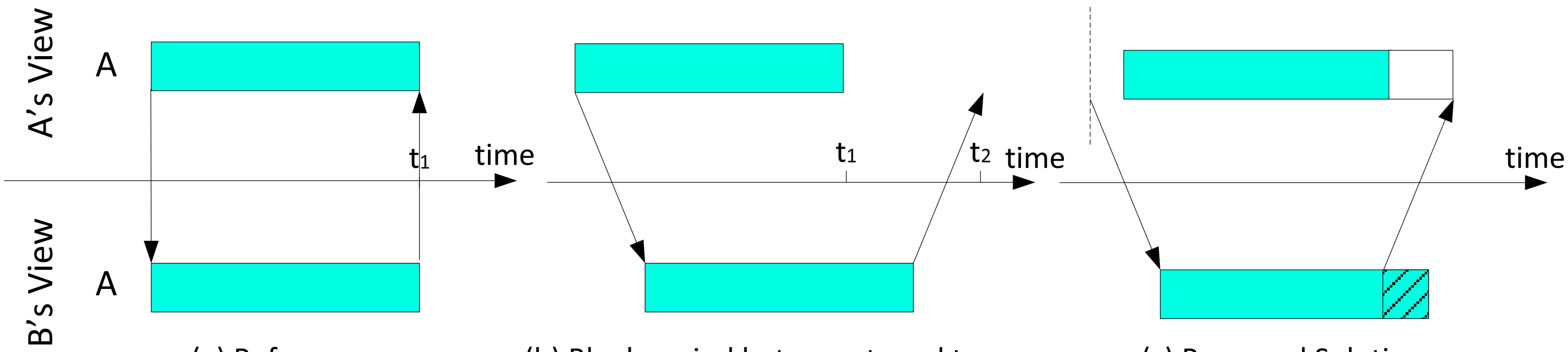
In the reference case with no network latency, an attack action terminates when the target’s response is received. Figure 7.4(a) shows that A’s action terminates at t_1 in both A’s and B’s view. In contrast, when there is latency, the action in the attacker’s view terminates before the target’s response has been received. Without knowing the *unpredictable* response, there is a blank period in which the target’s response cannot be displayed in the attacker’s view. Figure 7.4(b) shows that the action in A’s view terminates at t_1, whereas the target’s response is available at $t_2 = t_1 + RTT$. An inconsistency may occur if a random outcome is displayed in A’s view in the blank period. To ensure consistency, the action in the attacker's view should not terminate until the target's response has been received.

Besides solving the reordering problem, traditional methods can also be adopted to solve the blank-period problem. The local-lag [58] and the local perception-filter methods [85] can reschedule actions in order to let them complete after the response has been received. However, as mentioned above, these methods may lead to significant delay effects that are perceived by players.

Our goal in this chapter is to develop methods that can significantly reduce the perception of delay effects when strong consistency is maintained in fast-paced online game. Our approach is to determine at run time the best setting of control parameters that can conceal the delay effects. The possible combinations of control parameters to use are found by offline experiments evaluated by the JND surface.

Our study is based on five assumptions that are general enough to cover many fast-paced interactive online games. They are, however, not essential for designing strategies in slow-paced games because these games have sufficient slacks for performing rollbacks and other smooth corrections without being noticed by players.

Assumption 7.1.1. *We assume that network latencies in the near future are similar to those of the recent past (typically in the last few seconds). This assumption allows a priori setting of the size of delay buffers. Many previous studies [34,85,88] rely on this implicit assumption, which has also been verified in Chapter 6 and can be test at run time (See Chapter 3). We further assume the network loss is sufficient small after the buffering and error concealment in*



(a) Reference (b) Blank period between t_1 and t_2 (c) Proposed Solution

Figure 7.4: An illustration of the blank-period problem and its proposed solution [107].

the network-control layer, as online games generally runs in a better network condition than which videoconferencing runs in.

Assumption 7.1.2. *We develop our methods with respect to one weapon. The methods developed can be generalized well to games with multiple weapons because changing weapons is a relatively slow action, and there is sufficient time to notify other players when a player changes her weapon.*

Assumption 7.1.3. *We assume that weapons are precise, that each player can attack one target at a time, and that multiple attackers can attack the same or multiple targets at the same time. We do not consider imprecise weapons (such as a boom or a field magic) that attack multiple targets at the same time. In this case, players do not care about the order of the attacks as well as the consistency of their completion times.*

Assumption 7.1.4. *An action is realized only when it completes, which is common for precise weapons. For instance, the effect of shooting a bullet is realized when the bullet arrives at the target. Without this, it is not possible to conceal delay effects by changing the timing or duration of an action.*

Assumption 7.1.5. *An attack action (such as punching or kicking in fighting games or shooting a bullet in shooting games) is much faster than the movements of avatars of players in virtual space. This means that the duration of an action does not vary much with respect to the movements of avatars. The assumption is reasonable as the reaction times in fast-paced games are near the limit of human reaction times, and such a short duration can only relate to an attack* *action, rather than the movement of an avatar. Example action durations studied are 300-700 ms, although slower-paced games with longer action durations can also benefit from our algorithms.*

Problem Statement. Based on these five assumptions, we study the reordering of action completions in fast-paced multi-player online games running in a network with latency. We study it under two cases, one in which the target's response is predictable and another in which it is not. In the first case, an attacker does not have to wait for the target's response before proceeding to her next action. On the other hand, in the second case, the outcome to an action is unknown until the target's response has been received (i.e. the *blank period*). This means that an attacker has to wait for the target's response before proceeding to her next action. Under each of these two cases, we study two related sub-problems.

In the first subproblem, we develop an analytic method for achieving strong consistency at run time.

Figure 7.5 illustrates the approach that corrects an inconsistent order by combining the local-lag strategy (by delaying the start of a player's action) and the local-perception filters (by extending the duration of her action and by shortening the duration of the other player's action). Because all these are small adjustments, the delay effect due to each will be less noticeable than that caused by each adjustment when applied in isolation.

In the second subproblem, we develop a polynomial-time algorithm for finding the optimal multi-dimensional P_{notice} of delay effects caused by multiple controls that are perceived as a whole. Our approach decomposes the evaluation of the multi-dimensional P_{notice}

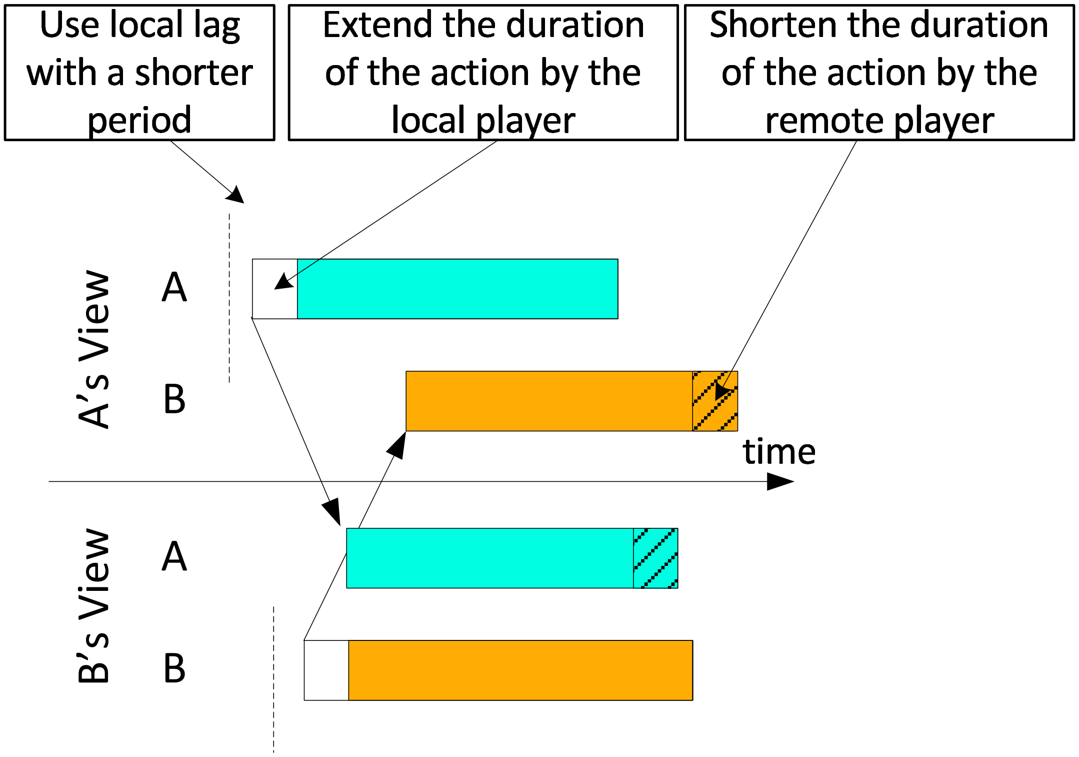

Use local lag with a shorter period
Extend the duration of the action by the local player
Shorten the duration of the action by the remote player
A's View
A
B
time
B's View
A
B

Figure 7.5: An illustration of our proposed approach for solving the reordering problem.

into multiple simpler subproblems of evaluating P_{notice} of each control. With the results in Chapter 4.5, we show that, when the target's response is predictable, optimality occurs when P_{notice} of all its component controls are equal. This reduces the complexity of finding the optimal P_{notice} from exponential to linear. If the response is unpredictable, the optimal P_{notice} can still be found in polynomial time.

Our solution extends existing methods [107, 111] for solving the blank-period problem described above when the target's response is unknown ahead of time. Our previous approach reschedules the actions in order for an attacker to have consistent information from a defender, without taking into account the reordering of action completions. Our current approach allows strong consistency to be enforced, both for targets with predictable as well as unpredictable responses.

This chapter is divided into four sections. Sections 7.2 and 7.3 present our solutions for solving the reordering problem when a target's response is predictable as well as unpredictable. For simplicity, we use a shooting game as a running example but evaluate the algorithms using the online game BZFlag [64]. Finally, a summary is drawn in Section 7.4.

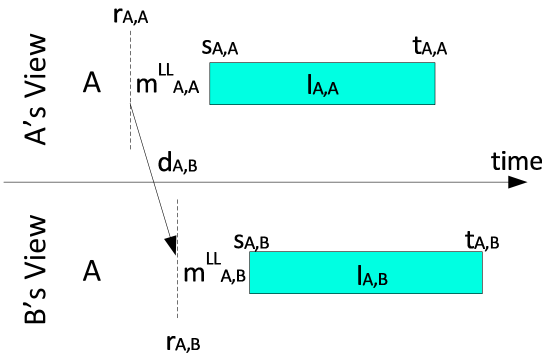
7.2 Solving the Reordering Problem for Targets with Predictable Response

In this and the next sections we solve the reordering problem in fast-paced multi-player games. By extending the local player's action while shortening the corresponding remote player's action, we show that delay effects can be made less aware while strong consistency can be maintained. We divide our discussion into two scenarios. In this section, we assume that the target's response is predictable, which means that the attacker does not have to wait for the target's response before starting her next action. In Section 7.3 we assume that the target's action is unpredictable. To maintain strong consistency without rollback or correction, the attacker needs to wait for the target's response before initiating her next action.

Figure 7.6 summarizes the symbols defined on actions. Let $s_{A,B}$, $r_{A,B}$, and $t_{A,B}$ be, respectively the starting time, time it is ready, and completion time of A's action in B's virtual space. In B's virtual space, $r_{A,B} = r_{A,A} + d_{A,B}$, where $d_{A,B}$ is the network latency between A and B. Note that $s_{A,B}$ and $r_{A,B}$ may not be the same.

We aim to design control strategies to improve $P_{\text{notice}}(R, M)$. We identify the strategy used to control M by its superscript and omit it when obvious. For example, $P_{\text{notice}}(R, M^{\textbf{LL}})$ refers to the case when $M^{\textbf{LL}}$ is controlled by the local-lag strategy.

Note that P_{notice} represents the probability of noticeability of a strategy used to control M and is common for all players using this strategy. A player using the strategy will have her P_{notice} represented by a JND surface, which is obtained by instantiating $P_{\text{notice}}(R, M)$ to $P_{\text{notice}}(ref, m)$, where ref and m correspond to the reference and modification used by the player. We further use the second superscript to distinguish the role of a player in whose view the modification is made (by default the role is attacker **ATK**). The subscripts A, B are similar to those used in actions. For example, $P_{\text{notice}}(ref_{A,B}^{\textbf{ATK}}, m_{A,B}^{\textbf{LPF1},\textbf{ATK}})$ stands for P_{notice} when the action duration of attacker A in B's view has changed from $ref^{\textbf{ATK}}$ to



Symbol	Definition
$r_{A,B}$	time when A's action is ready in B's virtual space
$m_{A,B}^{\text{strategy}}$	modification of time in A's action when shown in B's virtual space using the strategy defined
$s_{A,B}$	actual starting time of A's action and shown in B's virtual space
$l_{A,B}$	duration of A's action when shown in B's virtual space
$t_{A,B}$	completion time of A's action when shown in B's virtual space
$d_{A,B}$	physical network latency from A to B
LL	local-lag strategy
LPF1	strategy that extends the duration of action
LPF2	strategy that shortens the duration of action
ATK	Attacker
DEF	Defender

Figure 7.6: Symbols defined on the action performed by player A in A's and B's virtual spaces, respectively.

$ref^{\textbf{ATK}} + m^{\textbf{LPF1},\textbf{ATK}}$. We omit the subscripts when the player and the view are obvious.

Since the defenders studied in this section have predictable responses, all actions are by default initiated by attackers, and it is not necessary to indicate their role. In Section 7.3, the role of a player is to be explicitly indicated (whether she is an attacker or a defender) when a target's action is not predictable.

7.2.1 Maintaining Strong Consistency by the Local-Lag Strategy

In the reordering problem illustrated in Figure 7.1, network latency causes the actual order to be inconsistent with the reference order. We prove in this section that the local-lag method [58] can always maintain strong consistency. Let $m^{\textbf{LL}}_{i,j}$ be the local lag before an action is started. We have

$$s_{i,j} = r_{i,j} + m^{\textbf{LL}}_{i,j}. \tag{7.1}$$

We first define synchronization delay before proving the theorem.

Definition 7.2.1. *The* ***synchronization delay*** $\Delta t_{i,j}$ *of* i*'s action in* j*'s virtual space is the delay between the time when the action is initiated by* i *in her view to the time when this action is started in* j*'s view:*

$$\begin{aligned} \Delta t_{i,j} &= s_{i,j} - r_{i,i} \\ &= r_{i,j} + m^{\boldsymbol{LL}}_{i,j} - r_{i,i} \\ &= r_{i,i} + d_{i,j} + m^{\boldsymbol{LL}}_{i,j} - r_{i,i} \\ &= d_{i,j} + m^{\boldsymbol{LL}}_{i,j}. \end{aligned} \tag{7.2}$$

Theorem 7.2.1. *The local-lag strategy can maintain a strongly consistent order with respect to the reference when the synchronization delay of i's action in j's virtual space is $D_{\max} = \max_{x,y} d_{x,y}$, the maximum one-way latency between any two clients. That is, strong consistency is maintained by the local-lag strategy when*

$$m^{\boldsymbol{LL}}_{i,j} = \Delta t_{i,j} - d_{i,j} = D_{\max} - d_{i,j}. \tag{7.3}$$

Proof. According to Definition 7.2.1, the maximum $\Delta t_{i,j}$ is governed by $D_{\max}$. That is, in the worst case, the local-lag strategy will need to conceal $\Delta t_{i,j} = D_{\max}$. To ensure strong consistency in all views, we have i's action terminate in j's virtual space at

$$t_{i,j} \ = \ t_{i,i} + \Delta t_{i,j} \ = \ t_{i,i} + D_{\max}. \tag{7.4}$$

When $\Delta t_{i,j} = D_{\max}$, the delay between the completions of p's and q's actions is

$$t_{p,j} - t_{q,j} = t_{p,p} + \Delta t_{p,j} - (t_{q,q} + \Delta t_{q,j}) = t_{p,p} - t_{q,q}. \tag{7.5}$$

This proves that the delay is strongly consistent with that of the reference. □

The long synchronization delay specified in Theorem 7.2.1 may lead to noticeable delay effects. To address this issue, we next propose a better strategy that can maintain strong consistency while incurring less noticeable delay effects.

7.2.2 Proposed Strategy for Maintaining Strong Consistency

We first show a necessary and sufficient condition for maintaining strong consistency. This condition can be calculated at run time

Algorithm 7.1 Proposed Strategy for Solving the Reordering Problem

```
Require: Offline-measured JND surfaces instantiated from the strategies based
    on local lag $P_{\text{notice}}(R, M^{\mathbf{LL}})$ and local perception filters $P_{\text{notice}}(R, M^{\mathbf{LPF1}})$ and
    $P_{\text{notice}}(R, M^{\mathbf{LPF2}})$;
Ensure: The extent of modification due to local lag ($m^{\mathbf{LL}}$) and local perception
    filters ($m^{\mathbf{LPF1}}$ and $m^{\mathbf{LPF2}}$);
Estimate the duration of actions using (7.7);
Estimate network latency;
if action is carried out by a local player then
    Search for the optimal $m^{\mathbf{LL}}$ and $m^{\mathbf{LPF1}}$ using (7.16)-(7.17);
else
    Search for the optimal $m^{\mathbf{LPF2}}$ using (7.16)-(7.17);
end if
Modify the action using the $m^{\mathbf{LL}}$, $m^{\mathbf{LPF1}}$ and $m^{\mathbf{LPF2}}$ found.
```

based on information estimated from the recent past. Using this condition, we show that the duration of actions can be adjusted in order to maintain strong consistency.

Algorithm 7.1 presents our proposed strategy, which has been illustrated in Figure 7.5.

Necessary and Sufficient Condition for Maintaining Strong Consistency

To ensure that our proposed strategy can maintain a strongly consistent order with respect to the reference, we first evaluate the completion time of the action initiated by i to j.

$$t_{i,j} \;=\; s_{i,j} + l_{i,j} = r_{i,i} + d_{i,j} + m^{\mathbf{LL}}_{i,j} + l_{i,j}. \tag{7.6}$$

Note that j does not know $r_{i,i}$ and $t_{i,i}$ $(i \neq j)$ before receiving the message of that action. However, $l_{i,j}$ is predictable, as the duration of an action can be estimated by the speed of the shot and the virtual distance between i and j in the recent past and does not vary much with respect to the movements of players (Assumption 7.1.5 in Section 7.1). Let $dist_{i,j}(t-1)$ and $dist_{i,j}(t)$ be, respectively, the virtual distances between i and j in the previous and the current time windows.

$$l_{i,j} = dist_{i,j}(t)/v_{i,j} \approx dist_{i,j}(t-1)/v_{i,j}. \tag{7.7}$$

The necessary and sufficient condition for maintaining strong consistency is as follows.

Theorem 7.2.2. *Strong consistency in targets with predictable responses can be maintained by the local-lag strategy if and only if*

$$\begin{aligned} & d_{p,j} + m^{\boldsymbol{LL}}_{p,j} + l_{p,j} - d_{q,j} - m^{\boldsymbol{LL}}_{q,j} - l_{q,j} && (7.8)\\ = \; & m^{\boldsymbol{LL}}_{p,p} + l_{p,p} - m^{\boldsymbol{LL}}_{q,q} - l_{q,q}. && (7.9) \end{aligned}$$

Proof. According to (7.5), to maintain strong consistency for players p and q,

$$t_{p,j} - t_{q,j} = t_{p,p} - t_{q,q}. \tag{7.10}$$

By expanding both sides of (7.10), we have

$$\begin{aligned} & t_{p,j} - t_{q,j} \\ = \; & r_{p,p} + d_{p,j} + m^{\mathbf{LL}}_{p,j} + l_{p,j} \\ & -(r_{q,q} + d_{q,j} + m^{\mathbf{LL}}_{q,j} + l_{q,j}) && (7.11)\\ & t_{p,p} - t_{q,q} \\ = \; & r_{p,p} + m^{\mathbf{LL}}_{p,p} + l_{p,p} - (r_{q,q} + m^{\mathbf{LL}}_{q,q} + l_{q,q}). && (7.12) \end{aligned}$$

The theorem is proved by substituting (7.11)-(7.12) into (7.10). □

By substituting (7.3) into (7.8), strong consistency can be enforced by the local-lag strategy before the message of i's action is received by j. This is possible because $d_{i,j}$ is predictable (Assumption 7.1.1) and (7.8) does not have any unknown terms involving r.

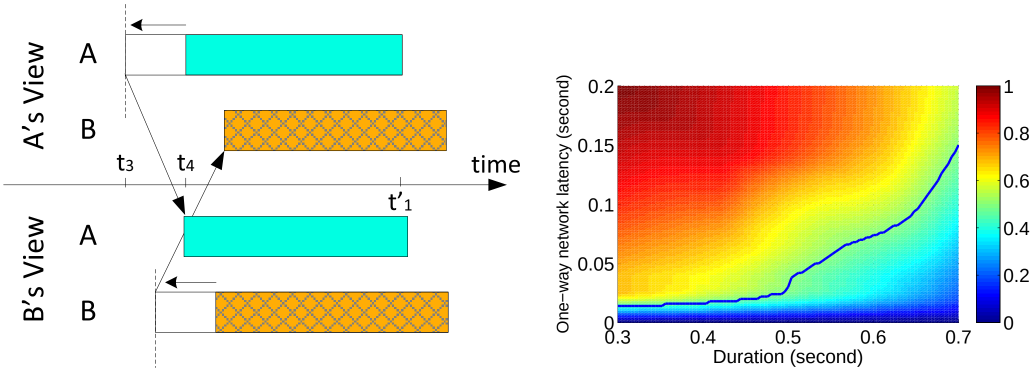
An important observation of (7.8) is that both $m_{i,j}^{\mathbf{LL}}$ and $l_{i,j}$ can be changed without violating strong consistency, as long as (7.8) is satisfied. In the rest of this section, we present methods to do several small changes on the durations of actions in order to make the overall effect less noticeable. We prove the correctness of the proposed strategy in two steps. First, we prove that the reordering problem can be solved by extending or by shortening the duration of action(s). This provides the basis for proving the correctness of the combined strategy in the second step. We further show that P_{notice} of the combined strategy can be significantly reduced.

Maintaining Strong Consistency by Extending the Durations of Actions

Figure 7.7(a) illustrates the strategy. Starting from the local-lag setting in Figure 7.2, instead of delaying the start of a local action, we extend the action in each player's view in order to cover the empty period before it starts, while keeping the completion time unchanged.

Figure 7.7(b) shows the JND surface after conducting subjective tests to measure subjects' sensitivity on detecting delay effects.

When using this strategy, the optimal extension is exactly the one-way latency from the player who started the action to the receiver. A longer period will lead to a larger change and make more players notice the delay effects, whereas a shorter extension is infeasible due to the definition of strong consistency.



(a) Extending durations

(b) Corresponding JND surface

Figure 7.7: Strong consistency can be maintained by extending the duration of actions. (a) Starting from the state in Figure 7.2 where actions are adjusted by the local-lag strategy, we extend the duration of the local actions (shown as unshaded boxes, each with an arrow). In A's view, A's action now starts at t_3 like that in the reference, but still ends at $t_1^{'}$. We readjust the duration similarly in B's view. Strong consistency is maintained because the completion times are not changed from the local-lag strategy. (b) P_{notice} due to extending the durations of actions in BZFlag is lower than that of the local-lag method (cf Figure 7.3 whose axes are defined in the same way). The curve shows the contour P_{notice}=50%.

The following corollary states the correctness on maintaining strong consistency.

Corollary 7.2.1. *Starting from the state where actions are adjusted by the local-lag strategy, extending the starting time of local actions will not change the strong consistency of completions with respect to the reference.*

Proof. Starting from the state where actions are adjusted by the local-lag strategy, we know from (7.8) in Theorem 7.2.1 that the order is strongly consistent. We fix the completion time of a local action but extend its duration while at the same time reducing its local lag. These amount to modifying the left hand side of (7.8):

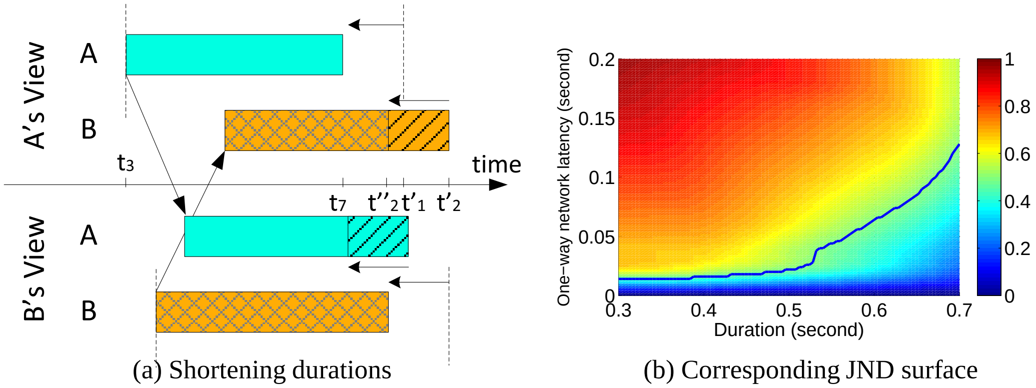
$$\begin{aligned} & m_{p,p}^{\mathbf{LL}} - m_{p,p}^{\mathbf{LPF1}} + l_{p,p} + m_{p,p}^{\mathbf{LPF1}} + m_{q,q}^{\mathbf{LL}} - m_{q,q}^{\mathbf{LPF1}} - l_{q,q} + m_{q,q}^{\mathbf{LPF1}} \\ = \; & m_{p,p}^{\mathbf{LL}} + l_{p,p} - m_{q,q}^{\mathbf{LL}} - l_{q,q}, \end{aligned}$$

where $m_{p,p}^{\mathbf{LPF1}}$ and $m_{q,q}^{\mathbf{LPF1}}$ are the extent by which the durations are extended. As the condition in (7.8) is unchanged, strong consistency remains to be satisfied. □

Maintaining Strong Consistency by Shortening the Durations of Actions

Figure 7.8(a) illustrates the strategy. Starting from the state in Figure 7.2 where actions are adjusted by the local-lag strategy, we start the local action in each player's view earlier, but terminates the other player's action earlier while keeping its starting time unchanged.

Figure 7.8(b) shows the JND surface after conducting subjective tests to measure subjects' sensitivity on detecting delay effects. Similar to before, the optimal period to shorten the durations of actions



(a) Shortening durations

(b) Corresponding JND surface

Figure 7.8: Strong consistency can be maintained by shortening the durations of actions. (a) Starting from the state in Figure 7.2 whose actions are controlled by the local-lag strategy, we shorten the durations of remote actions (shown as shaded boxes). In A's view, B's action now completes at t''_2instead of t'_2, without changing its starting time. Because the duration of B's action can be estimated in A's view (Assumption 7.1.5), A can also start its action earlier and allows it to complete at t_7 as in the reference. Hence, the order is still strongly consistent. (b) P_{notice} due to shortening the durations of actions in BZFlag is lower than that of the local-lag method (cf Figure 7.3 whose axes are defined in the same way). The curve shows the P_{notice}=50% contour.

is exactly the one-way network latency from the player who started the action to the receiver.

Similar to the strategy that extends the duration of actions, the optimal period to shorten the duration of actions is exactly the one-way network latency from the player who started the action to the receiver.

The following corollary states the correctness on maintaining strong consistency.

Corollary 7.2.2. *Starting from the state where actions are adjusted by the local-lag strategy, shortening the remote action and starting the local action earlier will not change the strong consistency of completions with respect to the reference.*

Proof. Unlike the proof in the last corollary, we need to modify the left side of (7.8). This modification depends on the value of p.

(a) When $p = j$, we move forward the local action by reducing its local lag $m^{\mathbf{LL}}_{j,j}$, while shortening the duration of the remote action $l_{q,j}$.

$$\begin{aligned} d_{j,j} + (m^{\mathbf{LL}}_{j,j} - m^{\mathbf{LPF2}}_{j,j}) \; &+ \; l_{j,j} - d_{q,j} - (l_{q,j} - m^{\mathbf{LPF2}}_{j,j}) - m^{\mathbf{LL}}_{q,j} \\ &= \; d_{j,j} + m^{\mathbf{LL}}_{j,j} + l_{j,j} - d_{q,j} - l_{q,j} - m^{\mathbf{LL}}_{q,j}. \end{aligned}$$

(b) When $p \neq j$, for any two remote actions, because their durations are shortened by the same extent, strong consistency is unchanged.

$$\begin{aligned} d_{p,j} + m^{\mathbf{LL}}_{p,j} + (l_{p,j} \; &- \; m^{\mathbf{LPF2}}_{j,j}) - d_{q,j} - (l_{q,j} - m^{\mathbf{LPF2}}_{j,j}) - m^{\mathbf{LL}}_{q,j} \\ &= \; d_{p,j} + m^{\mathbf{LL}}_{p,j} + l_{p,j} - d_{q,j} - l_{q,j} - m^{\mathbf{LL}}_{q,j}. \end{aligned}$$

Both of these conditions still maintains the left hand side of (7.8) unchanged, which ensures strong consistency according to Theorem 7.2.2. □

Maintaining Strong Consistency in the Combined Strategy

The three strategies (local-lag, extending and shortening an action) can be combined to solve the reordering problem. Figure 7.9 shows the corresponding adjustments. Based on the state in which actions are adjusted by the local-lag strategy, we can extend a player's action, or start it earlier while shortening the other player's action. By comparing this figure with Figure 7.1(a), we find identical orders of action completions as well as intervals between action completions in each view, which shows the correctness of the proposed approach.

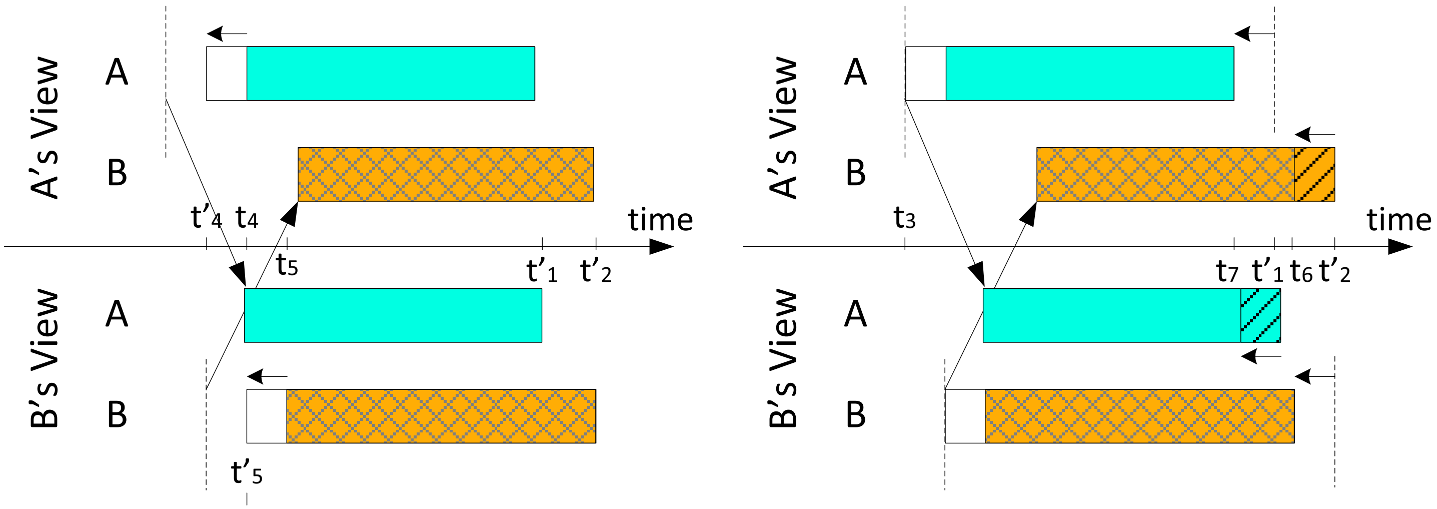
According to (7.3), the maximum synchronization delay to be concealed by the local-lag strategy alone is $m_{i,i}^{\mathbf{LL}} = D_{\max} = \max_{x,y} d_{x,y}$. After adjusting the durations of actions, the new local-lag delay in i's virtual space becomes significantly shorter,

$$m_{i,i}^{\mathbf{LL}} = D_{\max} - m_{i,i}^{\mathbf{LPF1}} - m_{j,i}^{\mathbf{LPF2}}. \tag{7.13}$$

This means that the combined approach leads to a smaller change in each of $m_{i,j}^{\mathbf{LL}}$, $m_{p,p}^{\mathbf{LPF1}}$, $m_{q,q}^{\mathbf{LPF2}}$, resulting less noticeable delay effects due to each change.

Theorem 7.2.3. *The order of completions in the combined strategy is strongly consistent with respect to the reference if and only if (7.13) is satisfied.*

Proof. The proof is done by combining the proofs of Corollaries 7.2.1 and 7.2.2. □



(a) Extending the local actions

(b) Shortening the remote actions

Figure 7.9: In every virtual space, we extend the local actions and start them earlier, while shortening the remote actions in order to reduce their synchronization delay. (a) Starting from the state in Figure 7.2, we extend the durations of local actions (shown as unshaded boxes). In A's view, A's action now starts at t_4'. When this is small compared to the original duration $t_1' - t_4$, it is less likely to be noticed. Similarly, B's local action is extended. (b) Starting from the state in (a), we shorten the remote actions. In A's view, we move B's completion time from t_2' to t_6. Because the duration of B's action can be estimated in A's view, A can also start her action earlier by the same amount (from t_1' to t_7). Hence, the order is still strongly consistent. Similarly, in B's view, we can do the same modification.

Similarly to the last two strategies, the optimal setting in the combined strategy should exactly fill $D_{\max}$. Any deviation will lead to undesirable delay effects.

7.2.3 Optimizing the Combined Strategy

As shown in (7.13), the combined strategy entails the control of the durations and the delays of actions while satisfying strong consistency.

Our goal in the optimization is to minimize $P_{\text{notice}}^{\textbf{COMB}}(ref, m)$ under given ref.

$$\begin{aligned} \mathcal{P} &= \min\ P_{\text{notice}}^{\textbf{COMB}}(ref, m) && (7.14) \\ \text{subject to } m &= m^{\textbf{LL}} + m^{\textbf{LPF1}} + m^{\textbf{LPF2}} \\ &= D_{\max}. && (7.15) \end{aligned}$$

Without knowing the closed form of function f in (4.12), the best we can do is to minimize the upper bound of f. According to the discussion in Section 4.5, we have

$$\begin{aligned} \overline{\mathcal{P}} &= \min\ \max\{P'_{\text{notice}}(ref, m^{\textbf{LL}}), P'_{\text{notice}}(ref, m^{\textbf{LPF1}}), \\ & \quad P'_{\text{notice}}(ref, m^{\textbf{LPF2}})\} && (7.16) \\ & \quad \text{subject to } m = m^{\textbf{LL}} + m^{\textbf{LPF1}} + m^{\textbf{LPF2}} = D_{\max}. && (7.17) \end{aligned}$$

Theorem 4.5.1 allows us to find the optimal solution to the upper bound of $P_{\text{notice-f}}^{\textbf{COMB}}$ at any (ref, m) when given the three JND surfaces $P'_{\text{notice}}(ref, m^{\textbf{LL}})$, $P'_{\text{notice}}(ref, m^{\textbf{LPF1}})$, and $P'_{\text{notice}}(ref, m^{\textbf{LPF2}})$. We search over the three surfaces to find m that satisfies (7.17). In each surface, for a given ref, we first build a bidirectional graph that connects each m to the corresponding P'_{notice} (discretized to k_3 levels).

The complexity is $O(K_1K_2)$, where K_1 and K_2 are, respectively, the discretization levels of the JND surface along the ref and m axes. Then we enumerate the K_3 levels of P'_{notice}, starting from the highest value, and find the first P'_{notice} that satisfies (7.17). At the same time, we use the graph to find the corresponding m. The complexity is $O(K_3)$.

In short, the complexity is linear with respect to K_3, the level of discretization in P'_{notice}. In our implementation, $K_3 = 101$ is sufficiently high, which corresponds to a 4-ms interval in the X axis and 2 ms in the Y axis. The search can be done within 1 ms by a desktop computer with Intel Core 2 Duo E8400 3 GHz CPU. As we use at most 4 JND surfaces, each with around 40 KB, all the surfaces can be stored in main memory. Hence, the search can be done in real time using the offline collected JND surfaces.

7.2.4 Experimental Evaluations on BZFlag

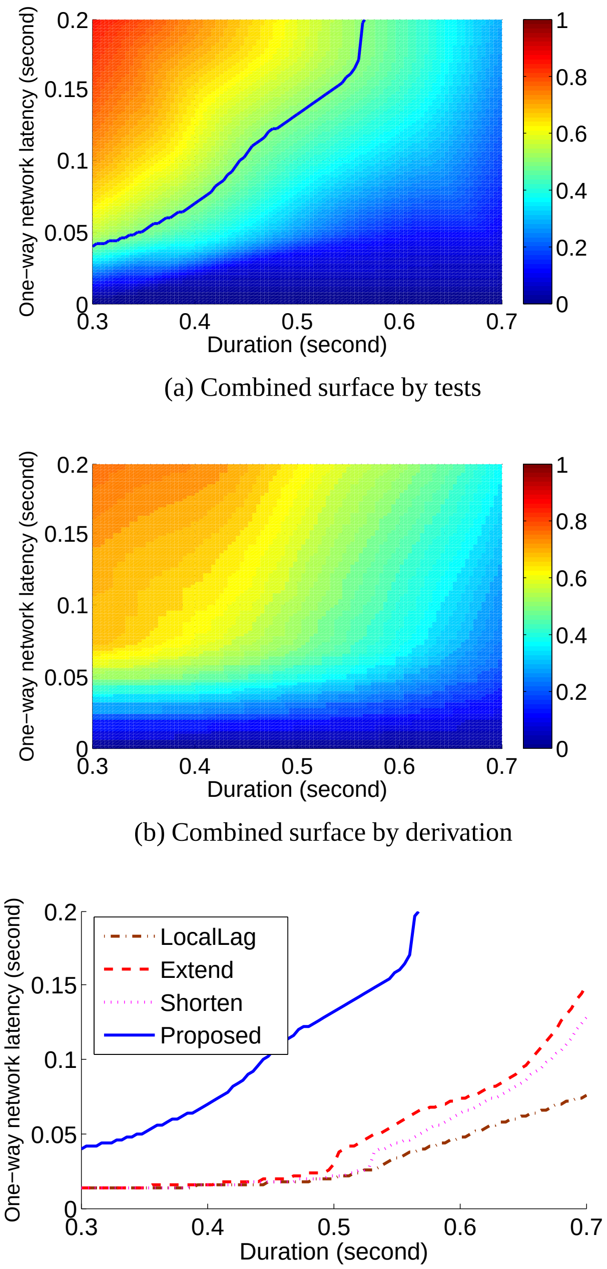
We present experimental results on evaluating the combined strategy on BZFlag [64], based on the three JND surfaces found by subjective tests. We measure the combined surface using the optimal setting in (7.17) for each ref. As the combined surface still follows Axiom 1, we conduct subjective tests to sample some critical points and approximate the surface using our algorithm in Chapter 4.

The results shown in Figure 7.10(a) are significantly better than those of methods using the local lag and local-perception filters (Figures 7.3, Figure 7.7(b), and 7.8(b)).

To demonstrate Property 4.5.1, we first show in Figure 7.10(a) the combined JND surface obtained by subjective tests based on the left-hand side of (4.15). As f is unknown, we directly measure $P^{\textbf{COMB}}_{\text{notice-f}}(ref, m)$ using subjective tests. Next, we show in Figure 7.10(b) the combined JND surface derived using the setting in (4.20), which was obtained by minimizing the upper bound of f in (7.17) and taken from the right-hand side of (4.20). Property 4.5.1 is demonstrated, as the two figures are very similar, with small differences at the corners that are reasonable in subjective tests with limited subjects.

To further illustrate the improvement of the combined strategy, we compare in Figure 7.10(c) the network latency when $P_{\text{notice}}=$ 50%. The graph shows that the combined strategy can maintain strong consistency while concealing delay effects, even with much larger network latency. Alternatively, when using the same latency as the method with extended durations, the combined strategy can use the extra latency in delay buffers to smooth network jitters and provide better loss concealment, resulting in greater stability of the game in real time.

In short, our results clearly show the merit of the combined strategy for solving the reordering problem when compared to the previous methods. The combined strategy leads to lower P_{notice} on delay effects under a given latency, while providing better loss concealment at the same level of P_{notice}.



(a) Combined surface by tests

(b) Combined surface by derivation

(c) Latency at P_{notice}=50%

Figure 7.10: Performance of the combined strategy for solving the reordering problem on targets with predictable responses. (a) JND surface of the combined strategy found by subjective tests. The x-axis shows the duration of the action, and the y-axis, the one-way network latency. (b) JND surface of the combined strategy found by using (4.15). Its similarity to a) illustrates Property 4.5.1. (c) One-way latency in running the game, when players can correctly notice a difference with respect to the reference at P_{notice}=50%.

7.3 Solving the Reordering and the Blank-Period Problems Together

In this section, we consider cases in which defenders' responses are not known a priori to attackers. As illustrated in Figure 7.4 earlier, there will be a blank period in which an attacker does not know the outcome of her action until the defender's response has been received. To avoid inconsistent outcomes or rollbacks, the attacker will need to wait for the defender's response before proceeding.

The blank-period problem described here may occur in conjunction with the reordering problem when there are multiple attackers. In this section, we present methods for concealing such delay effects. We first show the solution to the blank-period problem with one attacker and one defender. We then combine this solution with that in Section 7.2 for solving the reordering and the blank-period problems for multiple attackers.

7.3.1 Necessary and Sufficient Condition for Concealing the Blank Period

Referring to the blank-period problem stated in Figure 7.4, the action in attacker i's view would terminate only after knowing defender j's action. Based on the three control strategies in Section 7.2.3, let $m_{i,j}^{\mathbf{LL}}$, $m_{i,j}^{\mathbf{LPF1}}$, and $m_{i,j}^{\mathbf{LPF2}}$ be, respectively, the extent i's action is delayed, extended, and shortened in j's virtual space. Because a player may serve a dual role as an attacker as well as a defender, we add superscripts **ATK** and **DEF** to m to identify her role. The necessary and sufficient condition is stated as follows.

Theorem 7.3.1. *The necessary and sufficient condition for attacker i to conceal the blank period when waiting for defender j's response is*

$$m_{i,i}^{\textbf{\textit{LL}},\textbf{\textit{ATK}}} + m_{i,i}^{\textbf{\textit{LPF1}},\textbf{\textit{ATK}}} + m_{i,j}^{\textbf{\textit{LPF2}},\textbf{\textit{DEF}}} = d_{i,j} + d_{j,i}. \tag{7.18}$$

Proof. Referring to Figure 7.4, i would have received j's response when

$$t_{i,i} \;=\; t_{i,j} + d_{j,i}. \tag{7.19}$$

By expanding both sides, we have

$$t_{i,i} = r_{i,i} + m_{i,i}^{\textbf{LL},\textbf{ATK}} + l_{i,i} + m_{i,i}^{\textbf{LPF1},\textbf{ATK}}$$
$$\text{and } t_{i,j} + d_{j,i} = r_{i,i} + d_{i,j} + l_{i,i} - m_{i,j}^{\textbf{LPF2},\textbf{DEF}} + d_{j,i}.$$

Eq. (7.18) follows after simplifying and rearranging the terms. □

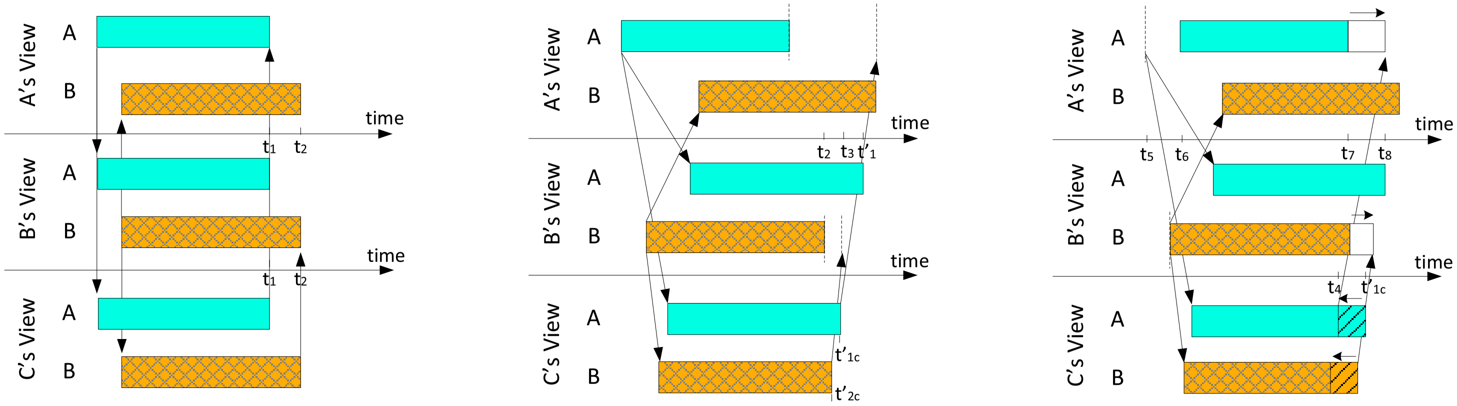
Similar to Theorem 7.2.2, (7.18) does not have any unpredictable terms involving r. Hence, it can be enforced by both players before their actions are carried out.

7.3.2 The Blank-Period and The Reordering Problems With Multiple Attackers

The blank-period and the reordering problems can happen together when there are multiple attackers instead of one attacker.

Figure 7.11(a) illustrates the reference case under no network latency. Here, the orders of completions are the same in all virtual spaces; that is, $t_2 > t_1$.

Figure 7.11(b) illustrates the case under network latency. The blank-period problem happens in B's view between t_2 and t_3 during



(a) Reference with no latency

(b) Case with network latency

(c) Previous solution

Figure 7.11: An illustration of the blank-period and the reordering problems under network latency when A and B shoot C, who is trying to defend against the attack. (a) Reference under no latency. (b) Case under latency. (c) A basic method [107] for solving the blank-period problem may lead to reordering.

which B does not know the result of the shot from A to C. It also happens in A's view in which the result of A's shot to C is not known until the message from C is received at A. On the other hand, the reordering problem happens in B's view in which the order of A's and B's completion times ($t_1^{'} > t_2$) is inconsistent with the reference ($t_1 < t_2$). This also happens in C's view ($t_{1c}^{'} > t_{2c}^{'}$). The example can be extended to a scenario with more than two attackers when the new attackers and their associated actions are added between B's and C's views.

To solve the blank-period problem, a basic approach is to delay the start of the attacker's action and extend its duration, while shorten the defender's action. These changes will allow the local action to complete only after receiving the defender's response.

Figure 7.11(c) illustrates the case in which the above approach may inadvertently reorder the completions of actions when there are multiple attackers. Consider A's action. In her view, we delay the start of her action from t_5 to t_6 and extend its duration from t_7 to t_8. We also shorten the duration of her action in C's view from $t_{1c}^{'}$ to t_4. Similar steps can, respectively, be applied to B's action in B's and C's views. Reordering occurs in B's and C's views when compared to the reference in Figure 7.11(a). The figure can also be extended to a scenario with more than two attackers.

7.3.3 Proposed Strategy

In this section we combine the strategy described above for solving the blank-period problem [107] and the strategy in Section 7.2.2 for solving the reordering problem in order to solve both problems together. Algorithm 7.2 shows the pseudo code. We prove its correctness in Section 7.3.3 and discuss its optimization in Section 7.3.3. Note that each player will instantiate her surfaces from the set of common offline-measured surfaces and may operate under a different operating point in the game.

Necessary and Sufficient Conditions for Solving the Blank-Period and Reordering Problems

The following theorem proves the correctness of Algorithm 7.2 under the general case when there are multiple attackers and multiple defenders.

Theorem 7.3.2. *Let $D_{\max}$ be the maximum network latency between any two players. Algorithm 7.2 is correct and solves the reordering and the blank-period problems together for multiple attackers and defenders if and only the following conditions are satisfied for all pairs of attackers i and j and defender k.*

(a) In each virtual space, the order of completion times is strongly consistent with the reference, both in attackers i's and j's views and defender k's view.

$$\begin{aligned} m_{i,i}^{\textbf{\textit{LL}},\textbf{\textit{ATK}}} + m_{i,i}^{\textbf{\textit{LPF1}},\textbf{\textit{ATK}}} + m_{j,i}^{\textbf{\textit{LPF2}},\textbf{\textit{ATK}}} &= D_{\max}, \qquad (7.20) \\ m_{j,j}^{\textbf{\textit{LL}},\textbf{\textit{ATK}}} + m_{j,j}^{\textbf{\textit{LPF1}},\textbf{\textit{ATK}}} + m_{i,j}^{\textbf{\textit{LPF2}},\textbf{\textit{ATK}}} &= D_{\max}, \qquad (7.21) \\ m_{i,k}^{\textbf{\textit{LPF2}},\textbf{\textit{DEF}}} - m_{j,k}^{\textbf{\textit{LPF2}},\textbf{\textit{DEF}}} &= d_{i,k} - d_{j,k}. (7.22) \end{aligned}$$

(b) For each attacker, her local action terminates after receiving defender k's response.

$$\begin{aligned} m_{i,i}^{\textbf{\textit{LL}},\textbf{\textit{ATK}}} + m_{i,i}^{\textbf{\textit{LPF1}},\textbf{\textit{ATK}}} + m_{i,k}^{\textbf{\textit{LPF2}},\textbf{\textit{DEF}}} &= d_{i,k} + d_{k,i} \ (7.23) \\ m_{j,j}^{\textbf{\textit{LL}},\textbf{\textit{ATK}}} + m_{j,j}^{\textbf{\textit{LPF1}},\textbf{\textit{ATK}}} + m_{j,k}^{\textbf{\textit{LPF2}},\textbf{\textit{DEF}}} &= d_{j,k} + d_{k,j}. (7.24) \end{aligned}$$

Proof. We prove the two parts separately.

(a) *Solving the reordering problem.* There are two steps in proving this part.

Firstly, strong consistency of completion times in all attackers' views is ensured because (7.20) and (7.21) are the same as (7.13) that has been proved in Theorem 7.2.3.

Secondly, we prove the strong consistency of completion time in defender k's view. For any actions from i and j in defender k's view, we shorten them by $m_{i,k}^{\textbf{LPF2}}$ and $m_{j,k}^{\textbf{LPF2}}$, respectively. By (7.22), the completion order of these actions are enforced as follows:

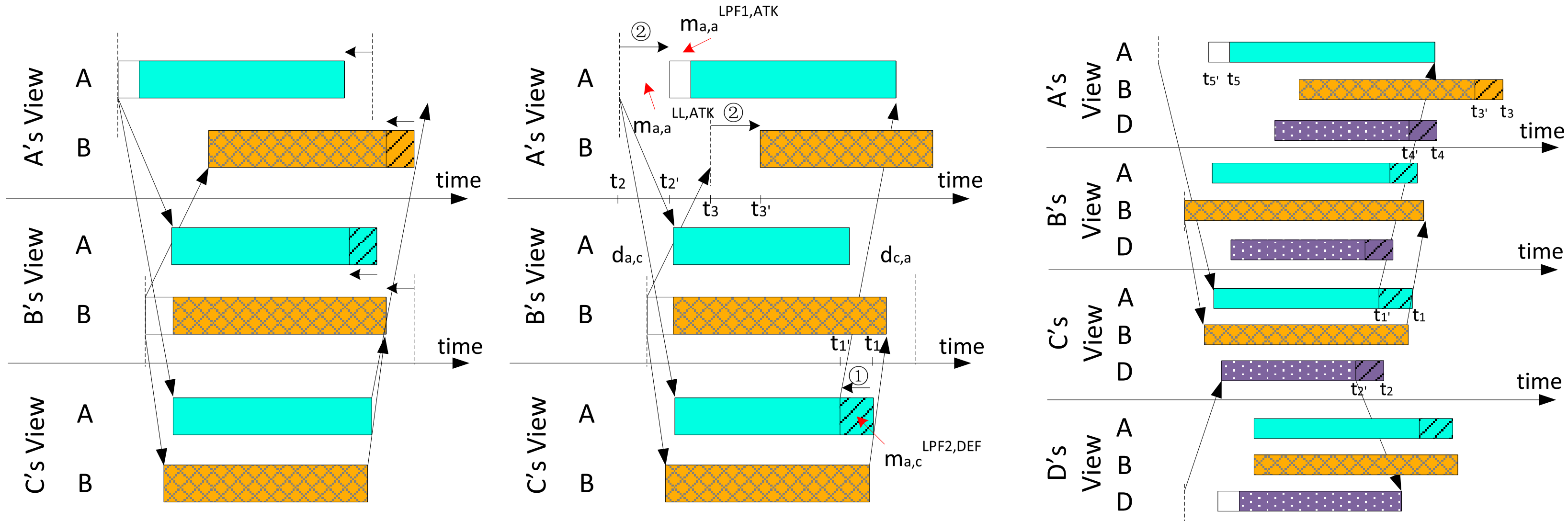
$$\begin{aligned} & t_{i,k} - t_{j,k} \\ = \quad & t_{i,i} + d_{i,k} - m_{i,k}^{\textbf{LPF2},\textbf{DEF}} - (t_{j,j} + d_{j,k} - m_{j,k}^{\textbf{LPF2},\textbf{DEF}}) \\ = \quad & t_{i,i} - t_{j,j}. \end{aligned} \tag{7.25}$$

(b) *Solving the blank-period problem.* Eq's (7.23) and (7.24) are exactly the necessary and sufficient condition proved in Theorem 7.3.1 for concealing the blank period.

The general case of multiple attackers and multiple defenders follows by combining the two parts and by considering any two attackers and any one defender. □

Figure 7.12(a) illustrates Steps 7-8 of Algorithm 7.2 that apply (7.20) and (7.23) for Attacker A (*resp.* (7.21) and (7.24) for B) to solve the reordering problem. Comparing it to Figure 7.9, it is clear that the modifications to the actions of both attackers are similar. It also illustrates the application of (7.23)-(7.24) to partially solve the blank-period problem.

Figure 7.12(b) illustrates Step 10 (①) and Step 12 (②) of Algorithm 7.2.



(a) Steps 7-8 of Algorithm 2

(b) Steps 10 and 12 of Algorithm 2

(c) Scenario with 3 attackers A, B, and D shooting C

Figure 7.12: An illustration of Algorithm 7.2 in solving the example in Figure 7.11 when A and B shoot C, who is trying to defend against the attack. (a) Steps 7-8. (b) Step 10 (① in the figure) and Step 12 (② in the figure). (c) Extension to a scenario with 3 attackers when a new attacker D joins the game.

In Step 10, we shorten the duration of A's action in defender C's view so that the difference between A's and B's adjustments in C satisfies (7.22). As shown in C's view, A's action that originally completes at t_1 now completes earlier at $t_1^{'}$. In contrast, B's action need not be shortened in C's view because $m_{B,C}^{\textbf{LPF2},\textbf{DEF}} = 0$ already satisfies (7.24).

In Step 12, we apply (7.23) and (7.24) to delay the start of actions in attackers A's and B's views in order to let their actions complete when the response from defender C is received. As shown in A's view, to make A's action complete when C's response is received, we delay its start from t_2 to $t_2^{'}$. To maintain the correct order of the completions (which has been assured by Steps 7-8), we further delay the start of B's action in A's view from t_3 to $t_3^{'}$ so that $t_3^{'} - t_3 = t_2^{'} - t_2 = m_{A,A}^{\textbf{LL},\textbf{ATK}}$. This is basically the application of (7.20) and (7.21) in solving the reordering problem, as Figure 7.12(b) is based on Figure 7.12(a). In contrast, as the latency between B and C is short, and extending B's duration by Step 7 is sufficient to cover the blank period, C's response already arrives at B on time, and the start of B's action needs not be delayed in B's view.

To show that the solution is correct, we compare the reference order in Figure 7.11(a) with the order in Figure 7.12(b). It is clear that strong consistency is maintained. Further, C's responses arrive at A and B exactly when their actions complete.

Figure 7.12(c) shows another example of applying Algorithm 7.2 on three attackers A, B and D shooting defender C. The corresponding reference order under no latency is similar to the order in Figure 7.11(a), but with D's action starting and completing slightly earlier than A's. For simplicity, we only show the messages between the attackers and the defender but not among the attackers.

By comparing Figures 7.12(c) and 7.12(a), we need to schedule D's action in A's and B's views, as well as A's and B's actions in D's view. These can be done by Steps 7-8 of Algorithm 7.2. By comparing Figures 7.12(c) and 7.12(b), we apply Step 10 to ensure that strong consistency is satisfied in C's view (Figure 7.11(a)). Note that D's action completes slightly earlier than A's. Finally, to solve the blank-period problem, Step 12 schedules the completion times of all attackers to the point when C's response is received.

Optimizing the Proposed Strategy

In this section we present the optimization for achieving the minimum $P^{\textbf{COMB}}_{\text{notice-f}}$ on delay effects when using Algorithm 7.2. The optimization has several differences with respect to that in (7.16)-(7.17). (a) We consider $N^{\textbf{ATK}}$ attackers and $N^{\textbf{DEF}}$ defenders. (b) We address the general case in which players may have different ref. (c) We shorten the durations of all defenders' actions in order to solve the reordering problem in their views, while delaying the starting times of all attackers' actions in order to solve the blank-period problem in the attackers' views.

Since each player cannot see the game in another player's view, we could optimize the combined noticeability in each view separately. However, this may result in some views having highly noticeable delay effects, while others have less noticeable delay effects. To maintain fairness in a multi-player game [107], we minimize the noticeability in all the views simultaneously.

Using notation similar to that of (4.12) in Section 7.2.3, we aim to minimize the upper bound of the combined noticeability in attackers i's and j's views.

$$\begin{aligned} & P^{\textbf{COMB}}_{\text{notice-f}}(ref, m^{\textbf{ATK}}_{i,i}) \\ \leq\ & \max\{P'_{\text{notice}}(ref_i, m^{\textbf{LL},\textbf{ATK}}_{i,i}),\ P'_{\text{notice}}(ref_i, m^{\textbf{LPF1},\textbf{ATK}}_{i,i}), \\ & P'_{\text{notice}}(ref_j, m^{\textbf{LPF2},\textbf{ATK}}_{j,i})\}. \end{aligned} \tag{7.26}$$

In defender k's view, we employ the **LPF2** strategy with no combined noticeability,

$$\begin{aligned} & P_{\text{notice}}(ref, m^{\textbf{DEF}}_{x,k}) \\ =\ & P'_{\text{notice}}(ref_x, m^{\textbf{LPF2},\textbf{DEF}}_{x,k}) \text{ where } x = i, j. \end{aligned} \tag{7.27}$$

The overall optimization is now stated as follows.

$$\begin{aligned} \overline{\mathcal{P}} =\ & \min \max_{\substack{1\leq i,j\leq N^{\textbf{ATK}}, i\neq j, \\ 1\leq k\leq N^{\textbf{DEF}}}} \max\{P^{\textbf{COMB}}_{\text{notice}}(ref, m^{\textbf{ATK}}_{i,i}), \\ & P_{\text{notice}}(ref, m^{\textbf{DEF}}_{i,k}), P_{\text{notice}}(ref, m^{\textbf{DEF}}_{j,k})\} \\ =\ & \min \max_{\substack{1\leq i,j\leq N^{\textbf{ATK}}, i\neq j, \\ 1\leq k\leq N^{\textbf{DEF}}}} \max\{P'_{\text{notice}}(ref_i, m^{\textbf{LL},\textbf{ATK}}_{i,i}), \\ & P'_{\text{notice}}(ref_i, m^{\textbf{LPF1},\textbf{ATK}}_{i,i}), \qquad (7.28) \\ & P'_{\text{notice}}(ref_j, m^{\textbf{LPF2},\textbf{ATK}}_{j,i}),\ P'_{\text{notice}}(ref_i, m^{\textbf{LPF2},\textbf{DEF}}_{i,k}), \\ & P'_{\text{notice}}(ref_j, m^{\textbf{LPF2},\textbf{DEF}}_{j,k})\} \end{aligned}$$

subject to

$$m^{\textbf{LL},\textbf{ATK}}_{i,i} + m^{\textbf{LPF1},\textbf{ATK}}_{i,i} + m^{\textbf{LPF2},\textbf{ATK}}_{j,i} = D_{\max} \tag{7.29}$$

$$m^{\textbf{LL},\textbf{ATK}}_{j,j} + m^{\textbf{LPF1},\textbf{ATK}}_{j,j} + m^{\textbf{LPF2},\textbf{ATK}}_{i,j} = D_{\max} \tag{7.30}$$

$$m^{\textbf{LPF2},\textbf{DEF}}_{i,k} - m^{\textbf{LPF2},\textbf{DEF}}_{j,k} = d_{i,k} - d_{j,k} \tag{7.31}$$

$$m^{\textbf{LL},\textbf{ATK}}_{i,i} + m^{\textbf{LPF1},\textbf{ATK}}_{i,i} + m^{\textbf{LPF2},\textbf{DEF}}_{i,k} = d_{i,k} + d_{k,i} \tag{7.32}$$

$$m^{\textbf{LL},\textbf{ATK}}_{j,j} + m^{\textbf{LPF1},\textbf{ATK}}_{j,j} + m^{\textbf{LPF2},\textbf{DEF}}_{j,k} = d_{j,k} + d_{k,j}, \tag{7.33}$$

where (7.29)-(7.33) are the same as (7.20)-(7.24) in Theorem 7.3.2.

Comparing the objective function in (7.28) with that in (7.16), we have one more $\max$ operator for aggregating the P'_{notice} of modifications to all the attackers' actions in every view. This is used to maintain fairness (as mentioned earlier) across all the players by bounding the maximum P_{notice}.

We allow different ref's in (7.28), as actions of different durations can appear together. There are also two new terms, $P'_{\text{notice}}(ref_i, m_{i,k}^{\textbf{LPF2},\textbf{DEF}})$ and $P'_{\text{notice}}(ref_j, m_{j,k}^{\textbf{LPF2},\textbf{DEF}})$, that represent the shortened durations of actions by attackers i and j in defender k's view. They are used to limit the delay effects in the defenders' views and to avoid some attackers' actions being too fast, making them difficult for defenders to guard against.

Comparing (7.29)-(7.33) with (7.17), there are new constraints for addressing the reordering as well the blank-period problems.

The above optimization is complex to solve in a closed form. The many variables and constraints as well as different ref's make it hard to find an optimal solution at run time. Fortunately, several observations can help simplify the problem.

Firstly, from (7.31), once $m_{i',k}^{\textbf{LPF2},\textbf{DEF}}$ is determined, any $m_{i,k}^{\textbf{LPF2},\textbf{DEF}}$, $1 \leq i \leq N^{\textbf{ATK}}$, $i \neq i'$, can be determined uniquely.

Secondly, by combining (7.29) and (7.32), we can directly get $m_{j,i}^{\textbf{LPF2},\textbf{ATK}}$ once $m_{i,k}^{\textbf{LPF2},\textbf{DEF}}$ is found in the last step. This also applies when we combine (7.30) and (7.33).

Thirdly, from (7.32), we know that $m_{i,i}^{\textbf{LL},\textbf{ATK}} + m_{i,i}^{\textbf{LPF1},\textbf{ATK}}$ is determined once $m_{i,k}^{\textbf{LPF2},\textbf{DEF}}$ is found. As shown in the proof of Theorem 4.5.1, the optimal solution is attained when $P'_{\text{notice}}(ref_i, m_{i,i}^{\textbf{LL},\textbf{ATK}}) = P'_{\text{notice}}(ref_i, m_{i,i}^{\textbf{LPF1},\textbf{ATK}})$, which allows the optimal $m_{i,i}^{\textbf{LL},\textbf{ATK}}$ and $m_{i,i}^{\textbf{LPF1},\textbf{ATK}}$

to be found directly. This also applies to (7.33).

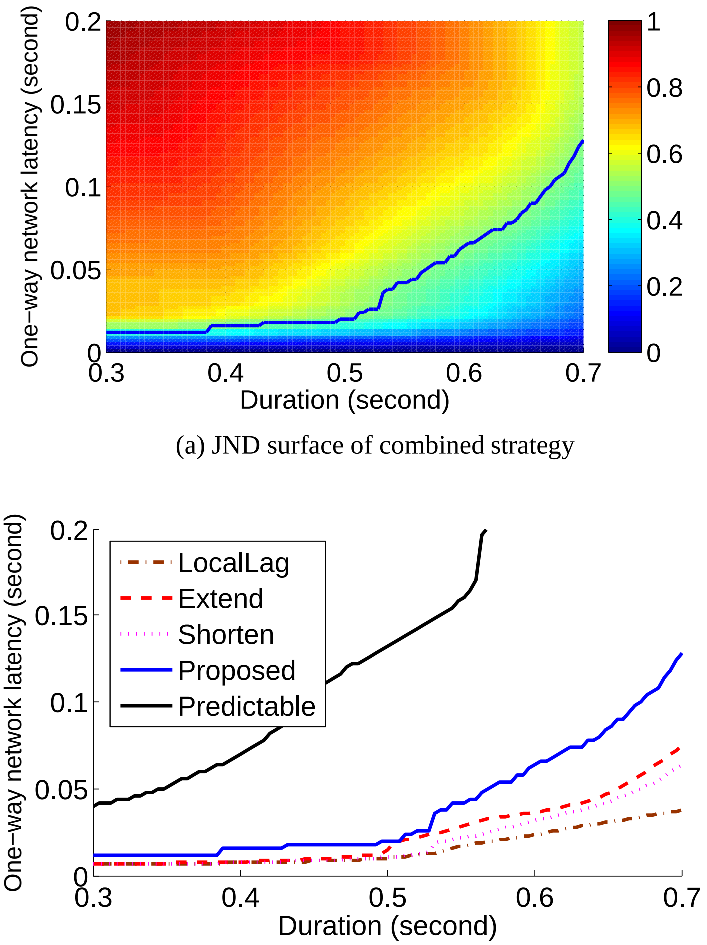
With these observations, the optimal solution to (7.28)-(7.33) can be found by enumerating the value of a single control variable $m_{i',k}^{\textbf{LPF2},\textbf{DEF}}$. The computational complexity is, thus, $O((N^{\textbf{ATK}})^2 N^{\textbf{DEF}} k^2)$, where k is the discretization level of $m_{i',k}^{\textbf{LPF2},\textbf{DEF}}$. We use k^2 instead of k, because one more loop is needed for finding $m_{i,i}^{\textbf{LL},\textbf{ATK}}$ and $m_{i,i}^{\textbf{LPF1},\textbf{ATK}}$ that satisfy $P'_{\text{notice}}(ref_i, m_{i,i}^{\textbf{LL},\textbf{ATK}}) = P'_{\text{notice}}(ref_i, m_{i,i}^{\textbf{LPF1},\textbf{ATK}})$.

As the overall complexity is low, we can search for the optimal control values at run-time. This can be finished within 5 ms by a computer with an Intel Core 2 Duo E8300 3 GHz CPU. The size of each JND surface is 40 KB, which is sufficiently small.

Experimental Evaluations on BZFlag

Similar to the results in Section 7.2.4, we present in this section the evaluation of Algorithm 7.2 on BZFlag [64]. To simplify the illustration, we use two attackers and one defender in the following experiments. We assume identical ref's for all the players, which allow the common ref to be shown in the x-axis of a single JND surface. We further assume the players to have the same instantiated JND surfaces in Algorithm 7.1 when using the strategies in isolation based on the local-lag method and the local perception filters.

Figure 7.13 shows the resulting JND surface after applying Algorithm 7.2, as well as the tolerable one-way latencies with P_{notice}=50%. The results show that the tolerable one-way latencies are much smaller than the corresponding latencies in Figure 7.10(c). These degradations are also reflected in the higher P_{notice} values in Figure 7.13(a) when compared to those in Figure 7.10(a). For example, when the



(a) JND surface of combined strategy

(b) Latency at P_{notice}=50%

Figure 7.13: Performance of the combined strategy with 2 attackers and 1 defender having identical reference durations for solving the reordering and the blank-period problems with unpredictable defender's actions. (a) JND surface of the combined strategy in running BZFlag. The x-axis shows the action duration, and the y-axis, the one-way latency. (b) One-way latency when players can notice a change with respect to the reference at P_{notice}=50%. The degradations between the scenario with predicted defender's actions (labeled "Predictable" and shown in Figure 7.10(a)) and the current result (labeled "Proposed") are caused by the time to handle the blank-period problem.

reference duration is 0.5 sec, the strategy based on extended durations allows more than 40 ms tolerable one-way latency in Figure 7.10(c), but it only allows around 25 ms here. The degradations between the results of Algorithm 7.1 (labeled "Predictable") and those of Algorithm 7.2 (labeled "Proposed") in Figure 7.13(a) are attributed to the additional time to handle the blank-period problem. Algorithm 7.2, however, leads to much better tolerable latencies when compared to the other strategies. This means that the game can run in a network with higher latency while providing comparable playing experience.

Note that Figure 7.13(a) is similar to Figure 7.8(b). This similarity can be explained by (7.29)-(7.32), *i.e.*, $m_{i,k}^{\textbf{LPF2},\textbf{DEF}} - m_{j,i}^{\textbf{LPF2},\textbf{ATK}} = d_{i,k} + d_{k,i} - D_{\max}$. As we assume $d_{i,k} = d_{k,i} = D_{\max}$, the optimal solution appears at $m_{j,i}^{\textbf{LPF2},\textbf{ATK}} = 0$ and $m_{i,k}^{\textbf{LPF2},\textbf{DEF}} = D_{\max}$ when we minimize the delay effect. That is, the strategy with shortened durations at the defender's view causes the dominant delay effect in the combined strategy.

In summary, Algorithm 7.2 leads to less noticeable delay effects while maintaining strong consistency as well as concealing blank periods. Further, the controls found allow the system to operate with the same P_{notice} but with higher latency.

7.4 Summary

In this chapter, we have developed a novel method for ensuring strong consistency on the completion times of actions, while minimizing noticeable delay effects due to network latencies, in fast multi-player online games running on IP networks. We have proposed a new approach for minimizing delay effects on user perception. The success of our approach is based on optimizing multiple controls together, each causing less noticeable delay effects than when applying the corresponding control in isolation. Finally, we have evaluated our approach by conducting subjective tests using a popular open-source online shooting game BZFlag and have shown significant performance improvements over previous strategies.

□ **End of chapter.**

Algorithm 7.2 Strategy for Solving the Blank-Period and the Reordering Problems

```
Require: Offline-measured JND surfaces based on local lag ($P'_{\text{notice}}(R, M^{\textbf{LL}})$)
    and local perception filters ($P'_{\text{notice}}(R, M^{\textbf{LPF1}})$ and $P'_{\text{notice}}(R, M^{\textbf{LPF2}})$); JND
    surfaces for each player instantiated from the offline-measured JND surfaces;
    $N^{\textbf{ATK}}$ attackers; $N^{\textbf{DEF}}$ defenders;
Ensure: Local lag ($m_{i,i}^{\textbf{LL},\textbf{ATK}}$), local perception filters 1 and 2 ($m_{i,i}^{\textbf{LPF1},\textbf{ATK}}$,
    $m_{j,i}^{\textbf{LPF2},\textbf{ATK}}$), and local perception filter 2 ($m_{i,k}^{\textbf{LPF2},\textbf{DEF}}$, $m_{j,k}^{\textbf{LPF2},\textbf{DEF}}$);
Estimate the duration of actions using (7.7);
Estimate network latency;
for $i = 1$ to $N^{\textbf{ATK}}$ do
    for $j = 1$ to $N^{\textbf{ATK}}$ and $j \neq i$ do
        for $k = 1$ to $N^{\textbf{DEF}}$ do
            In attacker $i$'s view,
            Extend attacker $i$'s action using (7.20) and (7.23) to find the optimal $m_{i,i}^{\textbf{LPF1},\textbf{ATK}}$;
            Shorten attacker $j$'s action using (7.20) to find the optimal $m_{j,i}^{\textbf{LPF2},\textbf{ATK}}$;
            In defender $k$'s view,
            Shorten attacker $i$'s and $j$'s actions by the optimal $m_{i,k}^{\textbf{LPF2},\textbf{DEF}}$ and $m_{j,k}^{\textbf{LPF2},\textbf{DEF}}$ (using (7.22)-(7.24)) in order to compensate for the difference in network latency;
            In attacker $i$'s view,
            Delay the start of attackers $i$'s and $j$'s actions by the optimal $m_{i,i}^{\textbf{LL},\textbf{ATK}}$ (using (7.20)-(7.21), (7.23)-(7.24)) to let the actions complete exactly when the corresponding defender's response is received.
        end for
    end for
end for
```

Chapter 8

Conclusion

In this chapter we conclude our accomplished work in the thesis and list the contributions of our work on fast-paced interactive multimedia systems. We then present the limitations of our work and propose possible future work to overcome these limitations.

8.1 Summary of Accomplished Research

The following is a summary of the results of this thesis:

- Firstly, nowadays network condition is still insufficient for maintaining stable transmission of fast-paced interactive multimedia. Further protection in the network layer is a must for a non-interrupted session. The protection should be done only when the underlying requirements and assumptions can be satisfied; otherwise, the bandwidth used will be wasted.

- Secondly, the mapping between system controls and perceptual quality can be represented by a JND surface, which is a model that can provide human opinions on all possible pairs of alternative settings within the range of control.

- Thirdly, with a dominance property, our efficient methodology for measuring JND surface with offline subjective tests is correct in theory and precise in simulation.
- Fourthly, the offline measured JND surface can be generalized to online network conditions with our proposed transformation method.
- Fifthly, our methodology for combining both dependent and independent JND surfaces has been probed to be correct and have real-time performance. This provide a way for online optimizing the perceptual quality per the human opinions in real-time even for fast-paced interactions.
- Lastly, our methodology can be applied to both VoIP systems and online games and improve their perceptual quality. We further prove theoretically that our novel method for concealing delay artifacts in online game is correct.

The following are the contributions of this thesis:

- Our first contribution is the discovery of the dominance property that can significantly reduce the number of subjective tests to be conducted offline. By utilizing this property, we can represent the mappings from controls to perceptual quality by a JND surface, which is a compact representation of the space of all mappings from controls to perceptual quality. We further develop a systematic methodology for generating this JND surface using a small number of subjective tests.
- Our second contribution is on the online search of mappings from controls to perceptual quality with real-time performance.

With the dominance property and the JND surface, we have developed an online algorithm for finding a suitable combination of controls that attains good perceptual quality. We have further proposed a general network-control layer for fast-paced interactive multimedia systems which can provide an improved network condition for stable running.

- Our third contribution is on the design of the network-control layer and its implementation in online operation as a traffic interceptor. The large scale measurements of nowadays network condition provide a good support on the design of the network-control layer. This design and measurement results can be reused by other related works. Our traffic interceptor for improving proprietary videoconferencing systems is unique. It not only provides a way for improving network conditions, but also indicates a new method for comparing any algorithm with existing proprietary software.

- Our forth contribution is on our methodology for designing a VoIP system with good perceptual quality based on offline collected human opinions. Our system can achieve similar performance as the best system developed previously but with fewer number of subjective tests.

- Our last contribution is on the theory for concealing delay artifacts in online games. Both the proofs and the subjective tests support the general framework for optimizing fast-paced online games can reduce the occurrences of delay artifacts and maintain the consistency between views of local and remote players.

8.2 Limitations and Future Work

In this section we discuss the limitations of our work, and provide possible way for overcoming them in future work.

The limitations of the methodology developed in this thesis are mainly related to the multiple-control problem. As have been stated in Table 1.2, our method cannot provide an efficient algorithm for combining JND surfaces with multiple independent controls, as there is no dominance property for reducing the search space.

To overcome this limitation, one possible way is to use more subjective tests to find the relation between these independent controls, and then utilize the relations to combine the JND surfaces of these controls. The core of such an approach is on the efficient method for discovering such relation.

We also have not completely solved the multiple-control problem with multiple dependent controls under complex constraints. By now, we have used application-dependent constraints to reduce the search space of the optimal set of settings, but have not provided a general method for such a reduction.

To overcome this limitation, one way is to investigate the different types of constraints, and study possible way on simplifying the optimization problems and on measuring the errors with the simplification.

Other limitations correspond to the assumptions of the network conditions. In this thesis we assume an one-way network latency to be less than 400 ms, the random loss rate to be less than 5%, and the available bandwidth to be higher than 100 Kbps. Under some extreme conditions, these assumptions may not be satisfied. Future works can focus on relaxing these conditions and studying specific methods for maintaining the minimum service level.

Other future works not related to the limitations are the extension of the proposed methods to other multimedia applications. We have presented the implementations and evaluations of VoIP and online games. Other possible applications include remote virtual-reality interactions, remote surgery, remote cooperation tasks, etc. As long as there is a need to optimize the perceptual quality in an online fast-paced interactive application, our methodology will be applicable.

□ **End of chapter.**

Bibliography

[1] B. Adelstein, T. Lee, and S. Ellis. Head tracking latency in virtual environments: psychophysics and a model. *Proc. of Human Factors and Ergonomics Society Annual Meeting*, 47(20):2083–2087, 2003.

[2] S. Aggarwal, H. Banavar, S. Mukherjee, and S. Rangarajan. Fairness in dead-reckoning based distributed multi-player games. In *Proc. of 4th ACM SIGCOMM workshop on Network and system support for games*, pages 1–10. ACM, 2005.

[3] J. Allen and S. Neely. Modeling the relation between the intensity just-noticeable difference and loudness for pure tones and wideband noise. *The Journal of the Acoustical Society of America*, 102:3628, 1997.

[4] S. Allin, Y. Matsuoka, and R. Klatzky. Measuring just noticeable differences for haptic force feedback: implications for rehabilitation. In *Proc. 10th Symp. on Haptic Interfaces for Virtual Environment and Teleoperator Systems (HAPTICS).*, pages 299–302. IEEE, 2002.

[5] A. Balachandran, V. Sekar, A. Akella, S. Seshan, I. Stoica, and H. Zhang. A quest for an Internet video quality-of-

experience metric. In *Proc. of the 11th ACM Workshop on Hot Topics in Networks*, pages 97–102. ACM, 2012.

[6] J. Beerends, A. Hekstra, A. Rix, and M. Hollier. Perceptual evaluation of speech quality (PESQ) the new ITU standard for end-to-end speech quality assessment. Part II: Psychoacoustic Model. *J. of Audio Engineering Society*, 50(10):765–778, 2002.

[7] P. Bourke. Rotate a point about an arbitrary axis. `http://paulbourke.net/geometry/rotate/`.

[8] B. Briscoe. Flow rate fairness: Dismantling a religion. *ACM SIGCOMM Computer Communication Review*, 37(2):63–74, 2007.

[9] N. Cardwell, Y. Cheng, C. S. Gunn, S. H. Yeganeh, and V. Jacobson. BBR: Congestion-based congestion control. *Queue*, 14(5):50, 2016.

[10] K.-T. Chen, C.-Y. Huang, P. Huang, and C.-L. Lei. Quantifying Skype user satisfaction. In *ACM SIGCOMM Computer Communication Review*, volume 36, pages 399–410. ACM, 2006.

[11] K.-T. Chen, P. Huang, and C.-L. Lei. How sensitive are online gamers to network quality? *Communications of the ACM*, 49(11):34–38, 2006.

[12] K.-T. Chen, C.-C. Tu, and W.-C. Xiao. Oneclick: A framework for measuring network quality of experience. In *Proc. IEEE INFOCOM*, pages 702–710. IEEE, 2009.

[13] P. Chen and M. El Zarki. Perceptual view inconsistency: an objective evaluation framework for online game quality of experience (QoE). In *Proc. 10th ACM SIGCOMM Workshop on Network and Systems Support for Games*, page Article 2. ACM, 2011.

[14] I. Cheng and P. Boulanger. A 3D perceptual metric using just-noticeable-difference. In *Eurographics (Short Presentations)*, pages 97–100, 2005.

[15] I. Cheng, A. Firouzmanesh, and A. Basu. Perceptual factors in graphics: from JND to PAM. In *Computer Vision, Third National Conf. on Pattern Recognition, Image Processing and Graphics (NCVPRIPG)*, pages 6–10. IEEE, 2011.

[16] Y.-W. Cheung and K. S. Lai. Lag order and critical values of the augmented dickey–fuller test. *Journal of Business & Economic Statistics*, 13(3):277–280, 1995.

[17] C.-H. Chou and Y.-C. Li. A perceptually tuned subband image coder based on the measure of just-noticeable-distortion profile. *IEEE Trans. on Circuits and Systems for Video Technology*, 5(6):467–476, 1995.

[18] B. Chun, D. Culler, T. Roscoe, A. Bavier, L. Peterson, M. Wawrzoniak, and M. Bowman. Planetlab: an overlay testbed for broad-coverage services. *ACM SIGCOMM Computer Communication Review*, 33(3):3–12, 2003.

[19] Cisco. Voice and Conferencing. http://www.cisco.com/cisco/web/solutions/small_business/products/voice_conferencing/index.html.

[20] M. Claypool and K. Claypool. Latency and player actions in online games. *Communications of the ACM*, 49(11):40–45, 2006.

[21] C. Diot and L. Gautier. A distributed architecture for multiplayer interactive applications on the internet. *IEEE Network*, 13(4):6–15, 1999.

[22] F. Dobrian, V. Sekar, A. Awan, I. Stoica, D. Joseph, A. Ganjam, J. Zhan, and H. Zhang. Understanding the impact of video quality on user engagement. In *ACM SIGCOMM Computer Communication Review*, volume 41, pages 362–373. ACM, 2011.

[23] E. N. Dzhafarov and H. Colonius. Fechnerian metrics in unidimensional and multidimensional stimulus spaces. *Psychonomic Bulletin & Review*, 6(2):239–268, 1999.

[24] F. B. Ecole Software. Dengeki Bunko: Fighting Climax Ignition. http://fightingclimax.sega.com/.

[25] H. E. Egilmez, S. Civanlar, and A. M. Tekalp. An optimization framework for QoS-enabled adaptive video streaming over openflow networks. *IEEE Trans. on Multimedia*, 15(3):710–715, 2013.

[26] B. Entertainment. Overwatch. https://playoverwatch.com/.

[27] G. Fechner. *Elements of psychophysics. Vol. I.* New York, 1966.

[28] J. Ferwerda. Psychophysics 101: how to run perception experiments in computer graphics. In *SIGGRAPH 2008 Classes*, page 87. ACM, 2008.

[29] B. A. Forouzan. *TCP/IP protocol suite*. McGraw-Hill, Inc., 2002.

[30] Google. Googla Talk Plug-in version 3.2.4.8431. `http://mail.google.com`.

[31] F. Hammer, P. Reichl, and A. Raake. The well-tempered conversation: interactivity, delay and perceptual VoIP quality. In *2005 IEEE Intl. Conference on Communications (ICC)*, volume 1, pages 244–249. IEEE, 2005.

[32] M. Handley, S. Floyd, J. Padhye, and J. Widmer. TCP friendly rate control (TFRC): Protocol specification. Technical report, 2002.

[33] J. H. Hansen and B. L. Pellom. An effective quality evaluation protocol for speech enhancement algorithms. In *ICSLP*, volume 7, pages 2819–2822, 1998.

[34] N. Hariri, B. Hariri, and S. Shirmohammadi. A distributed measurement scheme for internet latency estimation. *IEEE Trans. on Instrumentation and Measurement*, 60(5):1594–1603, 2011.

[35] T. Hayashi, K. Yamagishi, T. Tominaga, and A. Takahashi. Multimedia quality integration function for videophone services. In *Global Telecommunications Conf., 2007. GLOBECOM'07. IEEE*, pages 2735–2739. IEEE, 2007.

[36] P. Hinterseer, S. Hirche, S. Chaudhuri, E. Steinbach, and M. Buss. Perception-based data reduction and transmission of haptic data in telepresence and teleaction systems. *IEEE Trans. on Signal Processing*, 56(2):588–597, 2008.

[37] T. Huang, P. Huang, K. Chen, and P. Wang. Could Skype be more satisfying? A QoE-centric study of the FEC mechanism in an Internet-scale VoIP system. *Network*, 24(2):42–48, 2010.

[38] Z. Huang and K. Nahrstedt. Perception-based playout scheduling for high-quality real-time interactive multimedia. In *Proc. INFOCOM*, pages 2786–2790. IEEE, 2012.

[39] Z. Huang, K. Nahrstedt, and R. Steinmetz. Evolution of temporal multimedia synchronization principles: a historical viewpoint. *ACM Trans. on Multimedia Computing, Communications, and Applications*, 9(1s):34, 2013.

[40] Z. Huang, B. Sat, and B. W. Wah. Automated learning of playout scheduling algorithms for improving perceptual conversational quality in multi-party VoIP. In *2008 IEEE Int'l Conf. on Multimedia and Expo (ICME)*, pages 493–496. IEEE, 2008.

[41] Z. Huang, W. Wu, K. Nahrstedt, R. Rivas, and A. Arefin. SyncCast: synchronized dissemination in multi-site interac-

tive 3D tele-immersion. In *Proc. of the Second Annual ACM Conference on Multimedia Systems (MMSys)*, pages 69–80. ACM, 2011.

[42] Huawei. IP Telephony. `http://e.huawei.com/en/products/enterprise-networking/unified-communications`.

[43] R. Huber and B. Kollmeier. PEMO-Q: A new method for objective audio quality assessment using a model of auditory perception. *IEEE Trans. on Audio, Speech, and Language Processing*, 14(6):1902–1911, 2006.

[44] HumanBenchmark. Reaction time test, 2016.

[45] Q. Huynh-Thu and M. Ghanbari. Scope of validity of psnr in image/video quality assessment. *Electronics Letters*, 44(13):800–801, 2008.

[46] ITU. Methods for subjective determination of transmission quality. *Recommendation P.800*, 1996.

[47] ITU. The E-model, a computational model for use in transmission planning. *Recommendation G.107*, 2005.

[48] ITU. Subjective video quality assessment methods for multimedia applications. *Recommendation P.910*, 2008.

[49] Y. Jia, W. Lin, and A. Kassim. Estimating just-noticeable distortion for video. *IEEE Trans. on Circuits and Systems for Video Technology*, 16(7):820–829, 2006.

[50] C. Keimel, M. Klimpke, J. Habigt, and K. Diepold. No-reference video quality metric for HDTV based on H. 264/AVC bitstream features. In *2011 18th IEEE Int'l Conf. on Image Processing*, pages 3325–3328. IEEE, 2011.

[51] A. Keller. Trace control for Netem. `http://tcn.hypert.net/`.

[52] J. Kenney and E. Keeping. Linear regression and correlation. *Mathematics of Statistics*, 1:252–285, 1962.

[53] J.-S. Lee. On designing paired comparison experiments for subjective multimedia quality assessment. *IEEE Trans. on Multimedia*, 16(2):564–571, 2014.

[54] J.-S. Lee, F. De Simone, and T. Ebrahimi. Subjective quality evaluation via paired comparison: Application to scalable video coding. *IEEE Trans. on Multimedia*, 13(5):882–893, 2011.

[55] L. Liang, Z. Sun, and H. Cruickshank. Relative QoS optimization for multiparty online gaming in DiffServ networks. *IEEE Communications Magazine*, 43(5):75–83, 2005.

[56] Y. J. Liang, N. Farber, and B. Girod. Adaptive playout scheduling and loss concealment for voice communication over ip networks. *IEEE Trans. on Multimedia*, 5(4):532–543, 2003.

[57] Z. Lu, W. Lin, E. Ong, X. Yang, and S. Yao. Psqm-based rr and nr video quality metrics. In *Visual Communications*

and Image Processing 2003, pages 633–640. Int'l Society for Optics and Photonics, 2003.

[58] M. Mauve, J. Vogel, V. Hilt, and W. Effelsberg. Local-lag and timewarp: providing consistency for replicated continuous applications. *IEEE Trans. on Multimedia*, 6(1):47–57, 2004.

[59] V. Menkovski, A. Oredope, A. Liotta, and A. C. Sánchez. Predicting quality of experience in multimedia streaming. In *Proc. of the 7th Intl. Conference on Advances in Mobile Computing and Multimedia*, pages 52–59. ACM, 2009.

[60] Microsoft. Windows Filtering Platform. `http://msdn.microsoft.com/en-us/library/windows/hardware/gg463267.aspx`.

[61] Microsoft. Windows Live Messenger 15.4.3555.308. `http://messenger.live.com`.

[62] D. Mills. Internet time synchronization: The network time protocol. *IEEE Trans. on Communications*, 39(10):1482–1493, 1991.

[63] E. Muzychenko. Virtual Audio Cable 4.12. `http://software.muzychenko.net/eng/vac.htm`.

[64] J. Myers, T. Riker, F. Thilo, D. Trowbridge, S. Morrison, A. Tupone, and D. Remenak. Bzflag 2.4.2, 2012.

[65] Nintendo. Mario Kart. `http://mariokart8.nintendo.com/`.

[66] J. Ott, S. Wenger, N. Sato, C. Burmeister, and J. Rey. Extended RTP profile for real-time transport control protocol (RTCP)-based feedback (RTP/AVPF). Technical report, 2006.

[67] W. Palant, C. Griwodz, and P. Halvorsen. Evaluating dead reckoning variations with a multi-player game simulator. In *Proc. of the 2006 Intl. Workshop on Network and Operating Systems Support for Digital Audio and Video*, page 4. ACM, 2006.

[68] F. Pedrielli, E. Carletti, and C. Casazza. Just noticeable differences of loudness and sharpness for earth moving machines. *Journal of the Acoustical Society of America*, 123(5):3164–3164, 2008.

[69] M. Pinson and S. Wolf. A new standardized method for objectively measuring video quality. *IEEE Trans. on Broadcasting*, 50(3):312–322, 2004.

[70] M. B. Priestley. Spectral analysis and time series. 1981.

[71] I. REC. Mean opinion score (MOS) terminology. *Intl. Telecommunication Union, Geneva*, 2006.

[72] I. S. Reed and G. Solomon. Polynomial codes over certain finite fields. *Journal of the Society for Industrial and Applied Mathematics*, 8(2):300–304, 1960.

[73] J. Robson and N. Graham. Probability summation and regional variation in contrast sensitivity across the visual field. *Vision Research*, 21(3):409–418, 1981.

[74] V. Romero, L. Swiler, and A. Giunta. Construction of response surfaces based on progressive-lattice-sampling experimental designs with application to uncertainty propagation. *Structural Safety*, 26(2):201–219, 2004.

[75] B. Sat, Z. Huang, and B. W. Wah. The design of a multi-party VoIP conferencing system over the Internet. In *Ninth IEEE Intl. Symp. on Multimedia (ISM)*, pages 3–10. IEEE, 2007.

[76] B. Sat and B. Wah. Analyzing voice quality in popular VoIP applications. *Multimedia, IEEE*, 16(1):46–59, 2009.

[77] B. Sat and B. Wah. Statistical scheduling of offline comparative subjective evaluations for real-time multimedia. *IEEE Trans. on Multimedia*, 11(6):1114–1130, 2009.

[78] B. Sat and B. W. Wah. Analysis and evaluation of the Skype and Google-Talk VoIP systems. In *2006 IEEE Intl. Conference on Multimedia and Expo (ICME)*, pages 2153–2156. IEEE, 2006.

[79] B. Sat and B. W. Wah. Playout scheduling and loss-concealments in VoIP for optimizing conversational voice communication quality. In *Proc. of the 15th ACM Intl. Conference on Multimedia*, pages 137–146. ACM, 2007.

[80] C. Savery. *Consistency maintenance in networked games*. PhD thesis, Queen’s University, Kingston, Canada, 2014.

[81] C. Savery, T. Graham, and C. Gutwin. The human factors of consistency maintenance in multiplayer computer games. In

Proc. of the 16th ACM Intl. Conference on Supporting Group Work, pages 187–196. ACM, 2010.

[82] P. M. Sharkey, M. D. Ryan, and D. J. Roberts. A local perception filter for distributed virtual environments. In *Proc. 1998 IEEE Virtual Reality Annual Intl. Symp.*, pages 242–249. IEEE, 1998.

[83] W. Shi, J.-P. Corriveau, and J. Agar. Dead reckoning using play patterns in a simple 2D multiplayer online game. *Int'l J. of Computer Games Technology*, page Article 138596, 2014.

[84] Skype. Skype 5.10.0.116 Windows Edition. `http://www.skype.com`.

[85] J. Smed, H. Niinisalo, and H. Hakonen. Realizing the bullet time effect in multiplayer games with local perception filters. *Computer Networks*, 49(1):27–37, 2005.

[86] S. Stevens. On the theory of scales of measurement, 1946.

[87] R. C. Streijl, S. Winkler, and D. S. Hands. Mean opinion score (MOS) revisited: methods and applications, limitations and alternatives. *Multimedia Syst.*, 22(2):213–227, 2016.

[88] D. Stuckel and C. Gutwin. The effects of local lag on tightly-coupled interaction in distributed groupware. In *Proc. ACM Conf. on Computer Supported Cooperative Work*, pages 447–456, 2008.

[89] A. Takahashi, D. Hands, and V. Barriac. Standardization activities in the ITU for a QoE assessment of IPTV. *Communications Magazine, IEEE*, 46(2):78–84, 2008.

[90] M. Taylor and D. Creelman. PEST: Efficient estimates on probability functions. *The Journal of the Acoustical Society of America*, 41(4A):782–787, 1967.

[91] L. Thurstone. A law of comparative judgment. *Psychological Review*, 34(4):273, 1927.

[92] H. Tong and A. Venetsanopoulos. A perceptual model for JPEG applications based on block classification, texture masking, and luminance masking. In *Proc. 1998 Int'l Conf. on Image Processing (ICIP)*, pages 428–432. IEEE, 1998.

[93] H. von Helmholtz. *Kürzeste Linien im Farbensystem*. 1891.

[94] J. Wang and B. Bodenheimer. Computing the duration of motion transitions: an empirical approach. In *Proc. of the 2004 ACM SIGGRAPH/Eurographics Symp. on Computer Animation*, pages 335–344. Eurographics Association, 2004.

[95] Y. Wang, M. van der Schaar, S.-F. Chang, and A. C. Loui. Classification-based multidimensional adaptation prediction for scalable video coding using subjective quality evaluation. *IEEE Trans. on Circuits and Systems for Video Technology*, 15(10):1270–1279, 2005.

[96] Z. Wang. *Rate scalable foveated image and video communications*. PhD thesis, University of Texas at Austin, 2001.

[97] Z. Wang, L. Lu, and A. C. Bovik. Video quality assessment based on structural distortion measurement. *Signal Processing: Image Communication*, 19(2):121–132, 2004.

[98] A. Watson and L. Kreslake. Measurement of visual impairment scales for digital video. In *Proc. of SPIE*, volume 4299, pages 79–89, 2001.

[99] A. Watson and D. Pelli. QUEST: A Bayesian adaptive psychometric method. *Perception & Psychophysics*, 33(2):113–120, 1983.

[100] A. B. Watson, J. Hu, and J. F. McGowan. Digital video quality metric based on human vision. *Journal of Electronic Imaging*, 10(1):20–29, 2001.

[101] Z. Wei, P. Qin, and Y. Fu. Perceptual digital watermark of images using wavelet transform. *IEEE Trans. on Consumer Electronics*, 44(4):1267–1272, 1998.

[102] S. Winkler. *Digital video quality: vision models and metrics*. John Wiley & Sons, 2005.

[103] A. S. Works. Guilty Gear Xrd. `http://guiltygear.us/ggxrdr/`.

[104] C.-C. Wu, K.-T. Chen, Y.-C. Chang, and C.-L. Lei. Crowdsourcing multimedia qoe evaluation: A trusted framework. *IEEE Trans. on Multimedia*, 15(5):1121–1137, 2013.

[105] C.-C. Wu, K.-T. Chen, C.-Y. Huang, and C.-L. Lei. An empirical evaluation of VoIP playout buffer dimensioning in Skype, Google talk, and MSN Messenger. In *Proc. of the 18th Intl. Workshop on Network and Operating Systems Support for Digital Audio and Video*, pages 97–102. ACM, 2009.

[106] W. Wu, A. Arefin, R. Rivas, K. Nahrstedt, R. Sheppard, and Z. Yang. Quality of experience in distributed interactive multimedia environments: toward a theoretical framework. In *Proc. of the 17th ACM Int'l Conf. on Multimedia*, pages 481–490. ACM, 2009.

[107] J. Xu and B. Wah. Concealing network delays in delay-sensitive online interactive games based on just-noticeable differences. In *2013 IEEE Int'l Conf. on Multimedia and Expo (ICME)*, pages 1–6. IEEE, July 2013.

[108] J. Xu and B. Wah. Optimizing the perceptual quality of real-time multi-metric multimedia applications using JND profiles. *IEEE Multimedia Magazine*, 22(4):14–28, 2015.

[109] J. Xu and B. Wah. Optimality of the greedy algorithm for generating just-noticeable-difference surfaces. *IEEE Trans. on Multimedia*, 18(7):1330–1337, 2016.

[110] J. Xu and B. W. Wah. Delay-aware loss-concealment strategies for real-time video conferencing. In *2011 IEEE Intl. Symp. on Multimedia (ISM)*, pages 27–34. IEEE, 2011.

[111] J. Xu and B. W. Wah. Exploiting just-noticeable difference of delays for improving quality of experience in video conferencing. In *Proc. of the 4th ACM Multimedia Systems Conference*, pages 238–248. ACM, 2013.

[112] J. Xu and B. W. Wah. Consistent synchronization of action order with least noticeable delays in fast-paced multiplayer

online games. *ACM Trans. on Multimedia Computing, Communications, and Applications (TOMM)*, 13(1):8, 2016.

[113] J. Xue and C. Chen. Mobile video perception: New insights and adaptation strategies. *IEEE Journal of Selected Topics in Signal Processing*, 8(3):390–401, 2014.

[114] A. Yahyavi and B. Kemme. Peer-to-peer architectures for massively multiplayer online games: a survey. *ACM Computing Surveys*, 46(1):9, 2013.

[115] M. Zampini, D. I. Shore, and C. Spence. Audiovisual temporal order judgments. *Experimental Brain Research*, 152(2):198–210, 2003.

[116] S. Zander and G. Armitage. Empirically measuring the qos sensitivity of interactive online game players. In *Proc. ATNAC*, pages 511–518, 2004.

[117] X. Zhang, W. Lin, and P. Xue. Improved estimation for just-noticeable visual distortion. *Signal Processing*, 85(4):795–808, 2005.

[118] Y. Zhang, L. Chen, and G. Chen. Globally synchronized dead-reckoning with local lag for continuous distributed multiplayer games. In *Proc. of 5th ACM SIGCOMM workshop on Network and System Support for Games*, page 7. ACM, 2006.