


2011

## Measuring And Improving Internet Video Quality Of Experience

Mukundan Venkataraman Iyengar  
*University of Central Florida*

 Part of the [Computer Engineering Commons](#)  
Find similar works at: <https://stars.library.ucf.edu/etd>  
University of Central Florida Libraries <http://library.ucf.edu>

This Doctoral Dissertation (Open Access) is brought to you for free and open access by STARS. It has been accepted for inclusion in Electronic Theses and Dissertations, 2004-2019 by an authorized administrator of STARS. For more information, please contact [STARS@ucf.edu](mailto:STARS@ucf.edu).

---

### STARS Citation

Iyengar, Mukundan Venkataraman, "Measuring And Improving Internet Video Quality Of Experience" (2011). *Electronic Theses and Dissertations, 2004-2019*. 1943.  
<https://stars.library.ucf.edu/etd/1943>

# MEASURING AND IMPROVING INTERNET VIDEO QUALITY OF EXPERIENCE

by

MUKUNDAN VENKATARAMAN IYENGAR

B.E., Visweswaraiah Technological University, 2003  
Computer Science and Engineering

A dissertation submitted in partial fulfillment of the requirements  
for the degree of Doctor of Philosophy  
in the Department of Computer Engineering  
in the College of Engineering and Computer Science  
at the University of Central Florida  
Orlando, Florida

Summer Term  
2011

Major Professor:  
Mainak Chatterjee

© 2011 by MUKUNDAN VENKATARAMAN IYENGAR

## ABSTRACT

Streaming multimedia content over the IP-network is poised to be the dominant Internet traffic for the coming decade, predicted to account for more than 91% of *all* consumer traffic in the coming years. Streaming multimedia content ranges from Internet television (IPTV), video on demand (VoD), peer-to-peer streaming, and 3D television over IP to name a few. Widespread acceptance, growth, and subscriber retention are contingent upon network providers assuring superior Quality of Experience (QoE) on top of today's Internet.

This work presents the *first* empirical understanding of Internet's video-QoE capabilities, and tools and protocols to efficiently *infer* and *improve* them. To infer video-QoE at arbitrary nodes in the Internet, we design and implement MintMOS: a lightweight, real-time, no-reference framework for capturing perceptual quality. We demonstrate that MintMOS's projections closely match with subjective surveys in assessing perceptual quality. We use MintMOS to characterize Internet video-QoE both at the link level and end-to-end path level. As an input to our study, we use extensive measurements from a large number of Internet paths obtained from various measurement overlays deployed using PlanetLab.

Link level degradations of *intra*- and *inter*-ISP Internet links are studied to create an empirical understanding of their shortcomings and ways to overcome them. Our studies show that *intra*-ISP links are often poorly engineered compared to peering links, and that



degradations are induced due to transient network load imbalance within an ISP. Initial results also indicate that overlay networks could be a promising way to avoid such ISPs in times of degradations.

A large number of end-to-end Internet paths are probed and we measure delay, jitter, and loss rates. The measurement data is analyzed offline to identify ways to enable a source to select alternate paths in an overlay network to improve video-QoE, without the need for background monitoring or apriori knowledge of path characteristics. We establish that for any unstructured overlay of  $N$  nodes, it is sufficient to reroute key frames using a random subset of  $k$  nodes in the overlay, where  $k$  is bounded by  $O(\ln N)$ . We analyze various properties of such random subsets to derive simple, scalable, and an efficient path selection strategy that results in a  $k$ -fold increase in path options for any source-destination pair; options that consistently *outperform* Internet path selection.

Finally, we design a prototype called source initiated frame restoration (SIFR) that employs random subsets to derive alternate paths and demonstrate its effectiveness in improving Internet video-QoE.

## ACKNOWLEDGMENTS

A Ph.D. thesis has been a long standing dream of mine ever since I was seventeen, and by the time I was in college an year later, I was already starting to align myself towards it. It has been a good 10 years to bring that dream to fruition, and I have been lucky to meet some wonderful people along the road who helped me get here. They have inspired me, lifted my spirit when I was down, and showed me to flow around problems rather than rue missed chances. To each and every one of you, I owe a big thanks.

I would like to begin by thanking my advisor, Dr. Mainak Chatterjee, for taking me under his aegis and supporting me for close to five years. My association with him started almost an year before I got admitted to UCF. Looking back, I realize just how lucky I have been to work under him. From his quick witted jokes each time he visited the lab to lift our spirits to the ominous “Mukund, you have 24 hours to get me that paper and I don’t care if there is an earthquake or a tornado coming”, it was all wonderful. I have a lot of things to thank him for, but perhaps the most important one would be the sense of freedom he bestowed on me. I really thrived on it. I had little or no experience in writing measurements/systems papers, but his sheer confidence in me inspired me to raise my bar. Despite some initial set-backs in my early attempts at it, his support is what saw me through and helped me cross many,

many hurdles. He has been a strong pillar of support, both on and off the workplace, and it is not possible to list the thousands of reason that I can thank him for.

Dr. Ratan Guha changed the course of my Ph.D., and my career, with a gift that I have been so fortunate to receive. He generously extended a slice of PlanetLab, which eventually led me to do some wonderful research. PlanetLab usage not only added depth to my research, it also deepened my understanding of the Internet profoundly. I carry this tremendous knowledge and confidence with me into my career and I only have Dr. Guha to thank for bringing about this transformation. I am fortunate to have him in my committee.

I would like to thank Dr. Mostafa Bassiouni for spending time discussing ideas, and for CNT 6707 which I thoroughly enjoyed. My taking that course coincided with my running extensive experiments on the Internet. It was because of his course that I read a wonderful series of papers directly related to the Internet, papers that I would probably have not read if I did not bother taking that course. Some of those papers offered great insights to my own work, and helped me prevent caveats and avoid bad methodology. Few of those papers also inspired me to figure out ways to deploy large-scale measurement overlays which I initially thought was nearly impossible on my own. But perhaps the most wonderful thing to take away from that course was the way he taught it and his love for the domain. It was amazing to see how inspired I felt after every lecture. I sincerely hope I grow up to be just as enthusiastic about networking research in my own academic career.

I thank Dr. Necati Catbas for serving in my Ph.D. committee, and taking time to read my thesis and grasp these concepts even though he is not directly related to computer science

or networking research. His enthusiasm and a desire to know inspire me, and I thank him for all the effort behind proof reading and asking thought provoking questions.

I have been extremely fortunate to receive the Presidential Doctoral Fellowship which covered multiple years of my stay at UCF. Being a fellowship student certainly took the pressure off me during various semesters of my Ph.D., and gave me that extra time to focus on my work without having to worry about financial support. I hope I lived up to its expectation, and I have every reason to thank the UCF admissions committee for extending me such a generous offer.

I undertook multimedia research thanks to Dr. Raja Neogi. I spent my first summer working under him at C-Cor in 2007, and he constantly pointed out just how important and unsolved some of these problems are, and he eventually convinced me into taking them up for my Ph.D. thesis. I remember my initial dislike of multimedia largely because I had very malformed notions of exactly what video streaming was and how complicated most of these processes were. But thanks to his patience, and those endless hours of free lectures that were so full of information, I learnt a great deal about multimedia over that summer. I really began getting a feel of these problems, and it was not long before I started designing experiments and proposing theories to overcome them.

I thank Dr. Siddhartha Chattopadhyay (of Google) who charted a way to measure perceptual quality and inspired me towards building a stand-alone tool, which eventually became MintMOS. He helped me through multiple phases of its evolution, and it was his spark that eventually led to the fire.

I have made many friends during my stay at UCF, and I would especially like to thank my room-mates over the years. Swastik Brahma and Zubair Amhad oversaw my transition to the US, and have helped me in numerous ways. Swastik taught me how to drive, and was brave enough to jump into the passenger seat during my first venture onto the highways. They have also been my lab-mates, and along with Shamik Sengupta, Wenjing Wang, Dipika Darshana, Saptarshi Debroy and Shameek Bhattacharjee, they gave me an environment which was a lot of fun. I cannot forget those endless coffee breaks and all night camping in the lab – nothing would have been the same without them.

Subhabrata and Bipasha were family to me – I cannot quite count the number of times I would come back to an apartment that actually felt like home with the smell of great food still hanging in the air. I will forever miss those backyard parties and wished they would never end.

Nida has arguably been the most wonderful thing that has happened to me, and has been a great companion over these years. Nothing would be the same without her, and to her I owe more than words can describe. Her constant support and a miraculously evergreen attitude has infused a lot of positivity in me over the years.

Lastly, I would like to thank my family – Appa, Amma and my sister (Kaushi) for their unconditional support for being the three pillars of my existence. To the three of you and to Nida, I dedicate this thesis.

## TABLE OF CONTENTS

|  |            |
|--|------------|
| <b>LIST OF FIGURES</b>                                       | <b>xvi</b> |
| <b>LIST OF TABLES</b>  | <b>xxi</b> |
| <b>CHAPTER 1 INTRODUCTION</b>                                | <b>1</b>   |
| 1.1 Quality of Experience: A Primer                          | 3          |
| 1.1.1 Why QoE?   | 4          |
| 1.2 Contributions of this Work                               | 7          |
| 1.2.1 Inferring QoE in real-time at arbitrary Internet nodes | 8          |
| 1.2.2 A link-level study of Internet’s QoE shortcomings      | 10         |
| 1.2.3 Video-QoE along an Internet path                       | 11         |
| 1.2.4 Improving Internet video-QoE with one-hop redirections | 12         |
| 1.3 Structure of this Dissertation                           | 14         |
| <b>CHAPTER 2 QUALITY OF EXPERIENCE</b>                       | <b>15</b>  |
| 2.1 Introduction to QoE                                      | 15         |

|  |  |           |
|--|--|-----------|
| 2.2  | Multimedia QoS . . . . .                                   | 16        |
| 2.3  | An MPEG-2 Perspective . . . . .                            | 19        |
| 2.3.1  | MPEG-2 GOP Structure . . . . .                             | 20        |
| 2.3.2  | QoS v/s QoE for MPEG-2 . . . . .                           | 21        |
| 2.4  | Factors affecting QoE . . . . .                            | 22        |
| 2.5  | Summarizing . . . . .                                      | 24        |
| <b>CHAPTER 3 MINTMOS: INFERRING VIDEO-QOE IN REAL TIME .</b> |  | <b>25</b> |
| 3.1  | Introduction . . . . .                                     | 25        |
| 3.2  | Framework Architecture . . . . .                           | 26        |
| 3.2.1  | QoE space . . . . .  | 27        |
| 3.2.2  | Choice of Number of Partitions ( $N$ ) . . . . .           | 29        |
| 3.2.3  | QoE Inference Engine (IE) . . . . .                        | 30        |
| 3.2.4  | Improving QoE: Suggestions Engine (SE) . . . . .           | 31        |
| 3.2.5  | Sniffer . . . . .  | 32        |
| 3.3  | Instrumenting MintMOS . . . . .                            | 33        |
| 3.3.1  | Creating a QoE space: Survey with Human Subjects . . . . . | 35        |
| 3.3.2  | Running SE . . . . .                                       | 37        |
| 3.4  | Validating the framework . . . . .                         | 39        |

|   |  |           |
|---|--|-----------|
| 3.5   | Complexity of the Framework . . . . .            | 43        |
| 3.5.1   | Experimental Setup . . . . .                     | 45        |
| 3.5.2   | Effect of number of samples ( $N$ ) . . . . .    | 45        |
| 3.5.3   | Effect of number of parameters ( $k$ ) . . . . . | 47        |
| 3.5.4   | Size of QoE space . . . . .                      | 48        |
| 3.6   | Discussions . . . . .                            | 48        |
| 3.7   | Conclusions . . . . .                            | 51        |
| <b>CHAPTER 4 VIDEO QOE DEGRADATIONS OF INTERNET LINKS .</b> |  | <b>53</b> |
| 4.1   | Introduction . . . . .                           | 54        |
| 4.2   | Data Collection Methodology . . . . .            | 56        |
| 4.2.1   | Choosing ISPs . . . . .                          | 57        |
| 4.2.2   | Phase-1: Extracting ISP Topologies . . . . .     | 57        |
| 4.2.3   | Phase-2: Studying Network Links . . . . .        | 60        |
| 4.2.4   | Caveats and Data Completeness . . . . .          | 61        |
| 4.3   | Estimating Video-QoE of Links . . . . .          | 63        |
| 4.4   | Internet Video Streaming . . . . .               | 65        |
| 4.4.1   | The typical Internet route . . . . .             | 65        |
| 4.5   | Intra ISP Routing Policies . . . . .             | 67        |



|       |   |    |
|-------|---|----|
| 4.5.1 | Delay and Jitter Distributions . . . . .                      | 68 |
| 4.5.2 | Case Study: Level3's link between Tampa and Houston . . . . . | 69 |
| 4.6   | Inter-ISP Routing Policies . . . . .                          | 73 |
| 4.6.1 | Delay and Jitter Distributions . . . . .                      | 73 |
| 4.6.2 | Case Study: Peering link between Sprint and Qwest . . . . .   | 75 |
| 4.7   | Playout Buffer Analysis . . . . .                             | 76 |
| 4.7.1 | Packet reception . . . . .                                    | 78 |
| 4.7.2 | Playout Buffer Contents . . . . .                             | 78 |
| 4.7.3 | An MPEG-2 playout perspective . . . . .                       | 80 |
| 4.8   | Discussions . . . . .   | 81 |
| 4.8.1 | BGP and Path Selection . . . . .                              | 82 |
| 4.8.2 | Increasing Capacity and expected QoE as a metric . . . . .    | 84 |
| 4.9   | A Case for Overlay Networks . . . . .                         | 85 |
| 4.9.1 | Methodology . . . . .   | 86 |
| 4.9.2 | Benefits of an Overlay Network . . . . .                      | 87 |
| 4.10  | Conclusions . . . . .   | 89 |

|                  |   |           |
|------------------|---|-----------|
| <b>CHAPTER 5</b> | <b>EFFECTS OF INTERNET PATH SELECTION ON VIDEO-</b> |           |
| <b>QOE</b>       | <b>. . . . .</b>                                    | <b>91</b> |

|       |   |     |
|-------|---|-----|
| 5.1   | Introduction . . . . .                          | 91  |
| 5.2   | Probing Internet Destinations . . . . .         | 93  |
| 5.2.1 | Vantage Points and Destination Sets . . . . .   | 93  |
| 5.2.2 | Probing Methodology . . . . .                   | 94  |
| 5.2.3 | Outage Locations . . . . .                      | 97  |
| 5.2.4 | Failure Rate . . . . .                          | 98  |
| 5.2.5 | Failure Duration . . . . .                      | 99  |
| 5.2.6 | Summarizing . . . . .                           | 100 |
| 5.3   | Impact on Perceptual Quality . . . . .          | 101 |
| 5.3.1 | Outage Impact on Perceived Quality . . . . .    | 101 |
| 5.3.2 | Reconstructing Video Clips for Survey . . . . . | 103 |
| 5.3.3 | Survey with Subjects . . . . .                  | 104 |
| 5.3.4 | Summarizing . . . . .                           | 105 |
| 5.4   | Using Routing Redirections . . . . .            | 106 |
| 5.4.1 | Methodology . . . . .                           | 107 |
| 5.4.2 | What kind of paths help QoE? . . . . .          | 109 |
| 5.4.3 | Suitability of Intermediaries . . . . .         | 109 |
| 5.4.4 | Useful Intermediaries . . . . .                 | 111 |
| 5.4.5 | Choosing Intermediaries . . . . .               | 111 |

|       |  |     |
|-------|--|-----|
| 5.4.6 | Path Switching with random-5 . . . . . | 113 |
| 5.4.7 | Robustness . . . . .                   | 116 |
| 5.4.8 | Preserving Interactivity . . . . .     | 116 |
| 5.4.9 | Summarizing . . . . .                  | 117 |
| 5.5   | Conclusions . . . . .                  | 119 |

## CHAPTER 6 RANDOMIZED PATH SELECTION IN LARGE UNSTRUCTURED OVERLAYS . . . . . 121

|       |                                       |     |
|-------|---------------------------------------|-----|
| 6.1   | Introduction . . . . .                | 121 |
| 6.2   | Data Collection Methodology . . . . . | 124 |
| 6.2.1 | Vantage Points . . . . .              | 124 |
| 6.2.2 | Experimental Setup . . . . .          | 125 |
| 6.2.3 | Caveats . . . . .                     | 126 |
| 6.3   | Overview of random- $k$ . . . . .     | 127 |
| 6.3.1 | $k$ nodes are enough . . . . .        | 128 |
| 6.3.2 | Composing $k$ . . . . .               | 129 |
| 6.3.3 | Overhead . . . . .                    | 131 |
| 6.3.4 | Summarizing . . . . .                 | 132 |
| 6.4   | Characterizing Random- $k$ . . . . .  | 132 |

|                           |                                    |            |
|---------------------------|------------------------------------|------------|
| 6.4.1                     | Can we improve random- $k$ ?       | 132        |
| 6.4.2                     | Fairness                           | 135        |
| 6.5                       | Discussions                        | 136        |
| 6.5.1                     | Reduction in RTT                   | 136        |
| 6.5.2                     | $k$ is bounded by $O(\ln N)$       | 138        |
| 6.5.3                     | Discussions                        | 139        |
| 6.5.4                     | Putting it all together            | 140        |
| 6.6                       | Source Initiated Frame Restoration | 142        |
| 6.6.1                     | Prototype description              | 142        |
| 6.6.2                     | Methodology                        | 143        |
| 6.6.3                     | Results                            | 144        |
| 6.7                       | Conclusions                        | 146        |
| <b>CHAPTER 7</b>          | <b>RELATED WORK</b>                | <b>148</b> |
| 7.1                       | Quality Evaluation                 | 149        |
| 7.2                       | Internet Measurements              | 151        |
| <b>CHAPTER 8</b>          | <b>CONCLUSIONS</b>                 | <b>154</b> |
| <b>LIST OF REFERENCES</b> |                                    | <b>158</b> |

## LIST OF FIGURES

|     |   |    |
|-----|---|----|
| 1.1 | QoE v/s QoS: While both clips experienced the same loss rate (QoS), the perceived quality can be very different. . . . .  | 5  |
| 1.2 | An example of a network transmission with induced degradations. (a) Playout at source of frame-9, (b) playout of frame-9 at destination, (c) Playout at source of frame-86, and (d) Playout of frame-86 at destination. . . . .   | 6  |
| 2.1 | An MPEG-2 GOP structure which shows the different types of frames (I, P and B) and their basic relationship with each other. . . . .  | 19 |
| 2.2 | Nature of artifact when loss affects different frames in a GOP: (a) freezing due to a corrupt B-frame, (b) slicing due to a corrupt P-frame and (c) ghosting/pixellation due to a corrupt I-frame. (Image originally appears in [20] and the Society of Motion Picture Television Engineers). . . . . | 20 |
| 3.1 | Architectural overview of MintMOS . . . . .   | 27 |

|     |   |    |
|-----|---|----|
| 3.2 | A sample 2-D QoE space constructed with two parameters $(\gamma_1, \gamma_2)$ . The solid dots denote precomputed QoE for known input values of $\gamma_1$ and $\gamma_2$ . The expected QoE for a new sample $(\gamma_1, \gamma_2)$ is obtained by calculating the Euclidean distance between the new observation point and known indices. . . . . | 28 |
| 3.3 | Overview of the Experiment Topology . . . . .   | 44 |
| 3.4 | Processing overhead for MintMOS: (a) Effect of increasing number of (log) partitions for $k = 11$ ; and, (b) Effect of increasing number of parameters for $N = 10^6$ . . . . .   | 46 |
| 3.5 | Duration of outages discovered by MintMOS over a one week period on a 22 node wide area measurement overlay built using PlanetLab. . . . .  | 50 |
| 4.1 | New information added by additional vantage points to trace data . . . . .  | 62 |
| 4.2 | Characterizing <i>intra</i> -ISP links: (a) Delay distributions, and (b) Jitter distributions for 800 intra-ISP links from six days of active probing . . . . .   | 67 |
| 4.3 | A 24-hour observation of an intra-ISP link by Level3 connecting Tampa and Houston: (a) Delay distribution, (b) Jitter distributions, (c) Mean loading of the peering link, (d) Estimated QoE at various times of the day . . . . .  | 70 |
| 4.4 | Characterizing <i>inter</i> -ISP Peering links: (a) Delay distributions, and (b) Jitter distributions for 1100 inter-ISP links from six days of active probing . . . . .  | 72 |

|     |  |     |
|-----|--|-----|
| 4.5 | A 24-hour observation of a peering link between Sprint and Qwest in California: (a) Delay distribution, (b) Jitter distributions, (c) Mean loading of the peering link, (d) Estimated QoE at various times of the day. . . . .   | 74  |
| 4.6 | (a) Received sequence number at destination for the first 3600 packets. The X axis denotes the time since the first arrival of a packet, while the Y axis indicates the anticipated Sequence Number. (b) Contents of the unoptimized playout buffer in various time slots. Horizontal bar represents the mean. . . | 77  |
| 4.7 | CDF of the amount of ‘intact’ information in the playout buffer. . . . .   | 79  |
| 4.8 | Ideal path versus actual Internet path for the probe train experiment . . . .  | 84  |
| 4.9 | Recovering from perceptual degradations: a case for overlays that avoid faulty AS’s. . . . .   | 88  |
| 5.1 | Fraction of location failures for classifiable failures; the last column shows fraction of unclassifiable failures observed for all path failures. . . . .   | 98  |
| 5.2 | Failure rate (log-log scale) of individual paths to servers and broadband hosts.   | 99  |
| 5.3 | Failure duration: number of consecutive frames impacted during outages. . .  | 100 |
| 5.4 | Outages and their impacts on paths to servers: (a) Best case video artifact durations for low and high motion clips, and (b) Corresponding worst case artifact durations. . . . .  | 102 |
| 5.5 | CDF of number of useful intermediaries as a function of the upper bound on tolerable delay. . . . .  | 108 |

|      |  |     |
|------|--|-----|
| 5.6  | Number of intermediaries that offered a better path in times of failures. . . .  | 110 |
| 5.7  | Fraction of outages recovered by transmitting the next GOP to $k$ -intermediate nodes; for dataset $D2$ , we additionally plot success for different delay bounds. | 113 |
| 5.8  | CDF of packet loss distributions in key frames. . . . .  | 115 |
| 5.9  | Benefits of switching early: probability of recovery after consecutive packet losses in key frames. . . . .  | 115 |
| 5.10 | Resilience of random-5 recovery time over IP-path self repairing itself. . . .   | 118 |
| 5.11 | Difference in round trip times resulting from random-5 path selection with bound = 500ms. . . . .  | 118 |
| 6.1  | Errors when embedding nodes to a co-ordinate space with TIV. Parentheses (right) show errors from embedding . . . . .  | 123 |
| 6.2  | Experimental set-up consists of 556 nodes deployed across the US/Canada, EU and AP . . . . .   | 125 |
| 6.3  | Percentage of potential detours taken using a random subset of $k$ nodes . . .   | 129 |
| 6.4  | Importance of composing a representative subset: effects of skew (PL-master)   | 130 |
| 6.5  | Variants of random selection and relative performance gains . . . . .  | 134 |
| 6.6  | Random selection leads to improved fairness in detour selection . . . . .  | 136 |
| 6.7  | Reduction in RTT from using detour paths with random- $k$ . . . . .  | 137 |
| 6.8  | The value of $k$ is bounded by $O(\ln(N))$ for any given $N$ nodes in the overlay  | 139 |



|     |  |     |
|-----|--|-----|
| 6.9 | (a) Default IP-path v/s (b) SIFR, following a degradation: SIFR recovers<br>from perceptual degradation by restoring key frames. . . . . | 146 |
|-----|--|-----|

## LIST OF TABLES

|     |  |     |
|-----|--|-----|
| 3.1 | The first 16 partitions of a QoE space with 54 partitions. QoE space of four dimensions (bitrate, motion complexity, best/worst case impact, and loss rate). The remaining entries in the table can be found in [62] . . . . . | 34  |
| 3.2 | A subset of “From-To” mappings and suggestions discovered by SE in the QoE space. . . . .  | 38  |
| 3.3 | Results from a survey conducted to measure the accuracy of MintMOS projections . . . . .   | 41  |
| 4.1 | The first 15 of the 51 ISPs used in our study. Refer to [60] for the remaining ISPs. . . . .   | 58  |
| 4.2 | Default parameter values used in the objective QoE function from ITU-T [25]  | 64  |
| 4.3 | Characterization of Internet routes studied . . . . .  | 66  |
| 5.1 | Overview of outage locations for paths to servers and P2PTV hosts observed from 62 vantage points in a one week period. . . . .  | 96  |
| 6.1 | Dataset Overview . . . . .   | 126 |

|     |   |     |
|-----|---|-----|
| 6.2 | Comparison of overhead incurred . . . . .                                 | 131 |
| 6.3 | Comparing perceptual quality of SIFR against default IP-routing . . . . . | 144 |

# CHAPTER 1

## INTRODUCTION

“Begin at the beginning,” the King said “and then go on till you come to the very  
end, then stop”

---

— Lewis Carroll (*Alice in Wonderland*)

Today, consumer generated video content alone accounts for one third of all consumer Internet traffic. The sum total of all Internet video traffic, including Internet Television (IPTV), Video on Demand (VoD) and peer-to-peer (P2P) sharing, will amount to 91% of all consumer traffic by 2014 [11]. Multimedia content is poised to *dominate* all Internet traffic in the coming decade. More people today opt for video-conferencing, live Internet television, and video-on-demand services than they did a decade ago. The Internet has evolved from being a platform for hosting web pages to be a playground for multimedia content.

As customers spend more and more time watching videos online, they are increasingly becoming unsatisfied by low bitrate videos and are embracing high-definition (HD) streaming services. Providing high quality video streaming services over a best-effort and shared infrastructure such as the Internet, however, is non-trivial. It is increasingly observed that

existing Internet Quality of Service (QoS) is insufficient at ensuring consistent consumer experience. As Internet based multimedia competes with traditional cable-based streaming, an ever increasing load is placed on network elements on the Internet to deliver streaming content with high *perceptual* quality; including content delivery networks (CDNs), overlay networks, VoD and IPTV infrastructures. To succeed, network service providers need to infer, predict and improve perceptual video quality on the Internet.

Service providers are hence trying to characterize a video stream in terms of *Quality of Experience* (QoE) [26] rather than QoS. Internet QoS has long attempted to assure *statistical* service guarantees for parameters like bandwidth, delay, loss, and jitter [52]. However, QoS lacks an important element in characterizing video streams: that of human perception. For a given loss rate, the *perceptual* degradation caused by a network outage can vary dramatically depending on the type of frame impacted, the motion complexity inherent in the clip, and the encoding bitrate of the clip, to name a few. For example, a 1% loss on an MPEG-2 transport stream can either result in a minor glitch that is barely noticeable, or can severely degrade playout for an entire second depending on the aforesaid factors [19, 20]. Inferring perceptual quality of a video stream continues to be an open problem.

Existing perceptual quality evaluation frameworks are often complex, computationally intensive, or require specialized information. Perceptual quality is often expressed in terms of a mean opinion score (MOS). A common way of inferring MOS is by comparing video frames before and after network transmission to check for degradations. MOS calculations are often hard (if not impossible) to perform inside the network core. This is because: (i) it

is cumbersome to deploy computationally intensive software at arbitrary nodes/routers, and (ii) QoE evaluation is often infeasible at arbitrary routers/nodes because the original frames are unavailable for reference.

Inferring QoE apart, understanding present day Internet QoE and improving it are even harder. Internet architectures and protocols are highly optimized for elastic content like `http`, `ftp` and e-mail. For these applications, time-to-deliver is less important than message integrity. Given the diversity and size of the Internet, degradations at the link level and end-to-end path level that effect video QoE are not well understood. Further, there are few architectures and protocols that can efficiently infer and improve video QoE on the Internet.

## **1.1 Quality of Experience: A Primer**

Quality of Experience (QoE) describes how well a service performs in meeting user expectations. It is a rating of performance from the users' perspective. For Internet based streaming to compete with existing cable based infrastructure, QoE delivered by streaming services has to match or outperform QoE from cable based streaming. This section looks at what QoE is, why it is important, and the areas of QoE that we address in this thesis.

### 1.1.1 Why QoE?

Traditionally, Internet quality of service (QoS) was aimed at enabling streaming services. The Internet Engineering Task Force (IETF) standardized IntServ and DiffServ router mechanisms to improve quality of streaming content, which required changes to every router in the Internet. Given the scale of the Internet, as well as the diversity of various autonomous systems (AS) that comprise it, these changes could not be completely co-ordinated. As a result, even after years of slow adoption there has been no significant performance enhancements in terms of video quality delivery, as the Internet continues to operate on a ‘best-effort’ delivery model. QoS mechanisms operate with a notion of providing service guarantees to enhance application performance. However, service guarantees alone are not sufficient to raise *perceptual* quality. QoS based quality assessments have often found to be grossly inaccurate at predicting user experience, and as such are not applicable in evaluating video quality [43, 61].

To understand why QoS guarantees do not promise perceptual quality and why objective quality evaluations often misrepresent quality, we consider two example scenarios. Consider a snapshot of a playout shown in Fig. 1.1(a) and Fig. 1.1(b). Both clips<sup>1</sup> were subject to the same loss rate (QoS), however, the perceptual quality of both these clips are very different. This is because the perceptual degradation depends not only on the loss rate, but the *type* of frame impacted as well. As a result, statistical service guarantees (like QoS) are

---

<sup>1</sup>We use the terms clips, samples, and video-sequence interchangeably



(a)



(b)

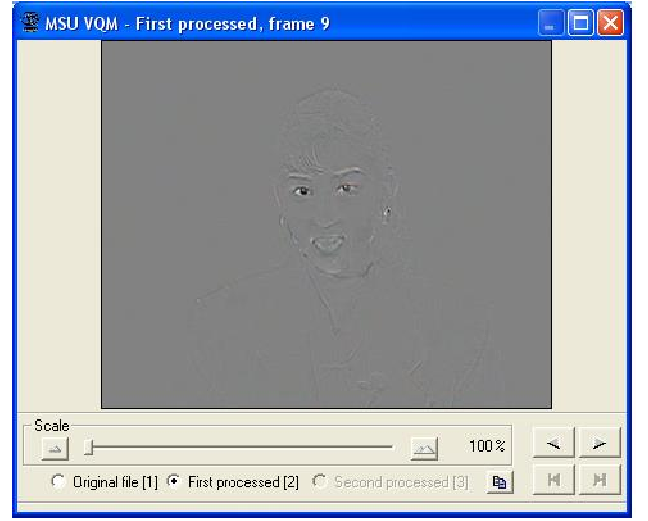
Figure 1.1: QoE v/s QoS: While both clips experienced the same loss rate (QoS), the perceived quality can be very different.

insufficient in assuring perceptual quality. To consider why objective functions misrepresent quality, consider Fig. 1.2(a) and Fig. 1.2(c). They show two instances of a video sequence at source before network transmission. In other words, these depict the expected playout at destination with perfect network transmissions. However, the actual playout at destination of these two instances was observed as in Fig. 1.2(b) and Fig. 1.2(d). If we ask any group of human observers to rate the quality of playout in Fig. 1.2(b) and Fig. 1.2(d), subjects would unanimously rate the latter playout to be far better than the former playout. Surprisingly, PSNR, the most common objective quality function, rates *both these sequences at par* at about 21db.





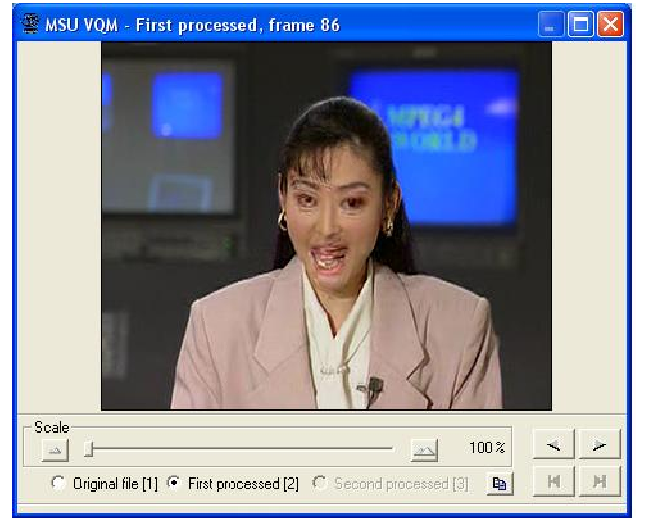
(a)



(b)



(c)



(d)

Figure 1.2: An example of a network transmission with induced degradations. (a) Playout at source of frame-9, (b) playout of frame-9 at destination, (c) Playout at source of frame-86, and (d) Playout of frame-86 at destination.

Clearly, there is a strong need to diverge from objective QoS based quality evaluation approaches towards QoE based quality evaluations. Monitoring and improving QoE seems to be the only way by which service providers can prevent churn and raise revenue. Service providers apart, QoE as a concept has been the driving factor for evaluating customer satisfaction in a wide variety of domains: from retails, airlines, food-services to customer support. Over the past few decades, QoE has been the most significant measure of human satisfaction in these domains. Understanding and improving QoE has had a great impact on the long term success of vendors in all of these domains.

## 1.2 Contributions of this Work

This thesis makes contributions to the following pertinent problems: (i) design a lightweight, no-reference tool called MintMOS which can infer the QoE of thousands of video streams in transit at arbitrary Internet nodes [57, 61, 62], (ii) provide the first empirical characterization of Internet *link-level* degradations and their impact on video-QoE [59, 60], (iii) provide the first large-scale characterization of *end-to-end* Internet paths in assuring video-QoE [58, 63], (iv) investigate one-hop Internet redirections in large, unstructured overlays that can scale to support millions of users [63], and (iv) propose a simple, scalable, and efficient path selection strategy called SIFR that applications can use to improve Internet video-QoE.

### 1.2.1 Inferring QoE in real-time at arbitrary Internet nodes

Inferring the perceptual quality of a video stream in transit at arbitrary nodes/routers in the Internet, where the original frame is not available for reference, continues to be an open problem. We present MintMOS: a loadable kernel module that is an accurate, lightweight, no-reference framework for capturing and offering suggestions to improve QoE inside the network core [57, 61, 62]. MintMOS can accommodate an arbitrary number of parameters to base quality inference decisions, and encompasses both network dependent and independent parameters. MintMOS internally consists of an inference engine (IE) to infer QoE, a suggestions engine (SE) to offer hints to improve QoE, a network sniffer to snoop traffic, and a QoE space. A QoE space is a known characterization of perceptual quality for various parameters that affect it. For any  $k$ -parameters that affect video quality, we begin by creating a  $k$ -dimensional QoE space, where each axis represents a parameter on which QoE is dependent. Hence, each point in the space is characterized by a  $k$ -tuple vector. For a given set of  $k$  parameters, we could get  $N$  “reference points” for MOS in the QoE space. To do this, we construct  $N$  versions of a given video by transporting the original video over a controlled environment i.e., for known values of the  $k$  parameters. The  $N$  video samples thus created are shown to a diverse population of human subjects<sup>2</sup> who assign a MOS to each of the samples. Given the  $N$  reference points in the  $k$ -dimensional QoE space, we can infer the MOS for a new set of parameters by calculating the least distortion between the

---

<sup>2</sup>UCF Institutional Review Board, IRB# 00001138

new values and the reference points in QoE space. MintMOS’s modular organization allows every component to evolve independently.

We instrument an actual QoE space with 54 partitions using four parameters: loss, encoding bitrate, motion complexity, and type of frame impacted. We generated 54 video samples using one low motion and one high motion clip, and requested 77 human subjects in a lab environment and 143 users online to assign a perceptual quality score to the samples. Their feedback was used to create our QoE space. We deployed MintMOS with this QoE space on a 22-node wide-area measurement overlay on PlanetLab. We streamed IP-traces of various clips from every node to every other node for one week and used MintMOS to predict quality and detect outages.

We validate the accuracy of MintMOS’s predictions using this QoE space by choosing 14 video samples reconstructed using the trace and asking 49 human subjects to assign a perceptual rating to them. MintMOS’s projections are compared to human perception, and we demonstrate a high degree of correlation between our predictions and human perception. MintMOS’s projections also outperform PSNR and VQM [43] predictions.

We demonstrate MintMOS’s ability to perform MOS calculations in real time by implementing and testing it both as a user level program and a kernel level module on standard, off-the-shelf, Linux terminals. Our experience shows that we can perform 20 MOS calculations per second for small sample spaces ( $< 100$  reference points and  $< 11$  parameters), and upto 4 MOS calculations per second for large spaces (1 million reference points and 11

parameters). In other words, for a reasonably accurate prediction, we can calculate MOS for 1000 flows per minute through a terminal.

In the end, we show that MintMOS can prove to be a valuable tool for measuring Internet path quality between any pair of nodes using it. MintMOS can suitably be used as a plugin for: (i) reporting QoE at destination in real time, (ii) various overlay/P2P networks for monitoring path quality, (iii) VoD and IPTV service providers to monitor QoE at critical nodes along the data path, and (iv) for CDNs to choose indirections based on path quality.

### **1.2.2 A link-level study of Internet's QoE shortcomings**

The capability of present day Internet links in delivering high perceptual quality streaming services is not completely understood. Link level degradations caused by intra-domain routing policies and inter-ISP peering policies are hard to obtain, as ISPs often consider such information proprietary. Understanding link level degradations will enable us in designing future protocols, policies, and architectures to meet the rising multimedia demands.

We present a trace driven study to understand QoE capabilities of present day Internet links using 51 diverse ISPs with a major presence in US, Europe and Asia-Pacific [59, 60]. We study their links from 38 vantage points in the Internet using both passive tracing and active probing for six days. We provide the first measurements of link level degradations and case studies of intra-ISP and inter-ISP peering links from a multimedia standpoint. Our

study offers surprising insights into intra domain traffic engineering, peering link loading, BGP and the inefficiencies of using AS-path lengths as a routing metric. Though our results indicate that Internet routing policies are not optimized for delivering high perceptual quality streaming services, we argue that alternative strategies such as overlay networks can help meet QoE demands over the Internet. Streaming services apart, our Internet measurement results can be used as an input to a variety of research problems.

### 1.2.3 Video-QoE along an Internet path

Having characterized link-level degradations, we next turn our attention to analyzing a large number of end-to-end Internet paths to better understand video-QoE capabilities of present day Internet. We seek answers to the following questions: (i) What *degrades* video QoE in the Internet and *where* in the path do these outages occur?, (ii) How does an Internet outage *affect* video-QoE?, and (iii) What fraction of these outages are *addressable* by using one-hop Internet redirections?

To answer the first question, we begin by probing 1000+ popular Internet video destinations from 62 geographically diverse PlanetLab vantage points for seven consecutive days. Our probing mimics “fetching” streaming content from each destination for a variety of low and high motion clips. Our destination set includes the 200 most popular IPTV/VoD servers, and a set of 1,200 IP addresses from crawls of popular P2PTV providers. We discovered a significant number of path outages that led to complete loss in path connectivity. We find

that such outages occur in various points in a path and vary significantly between paths to servers and P2P hosts. Of the outages on a round trip path to servers, we found that only 11% of these occur on the last hop, and therefore cannot be corrected by alternate routing. The remaining 89% are potentially recoverable by Internet routing. For P2P hosts, we found that over 40% of the outages are last hop, which indicates that alternate paths can potentially recover upto 60% of these outages.

To measure the perceptual degradation resulting from these outages, we reconstructed MPEG-2 video samples using the IP-traces collected from every destination set. We create a comprehensive list of 54 video clips that mirror the most commonly occurring loss patterns. We asked 77 subjects to review these clips to gain a deeper understanding of perceptual degradations. Network anomalies typically manifest as a video *artifact*, which is a visible distortion during playout that persists for a certain duration. These artifacts could range from slicing to freezing to extreme pixellation [19, 20]. These artifacts and their on-screen duration depend on the type of frame impacted, the motion complexity inherent in the clip (low v/s high), and encoding bitrate. Using the survey, we outline application specific policies that can improve perceptual quality.

#### **1.2.4 Improving Internet video-QoE with one-hop redirections**

Using insights from the previous round of study, we investigate ways of *improving* Internet video-QoE using one-hop redirections. Perceptual quality can be raised by path selection

strategies which preserve application specific policies. To make our results more generally applicable, we seek path selection strategies that *do not* require background monitoring of alternative routes or any apriori path quality information. We analyze a large number of Internet path measurements from five different overlays built using PlanetLab. Our datasets include weeklong measurements taken from overlays of: (i) 21 nodes in United States, (ii) 19 nodes in Europe, (iii) 22 nodes in Asia, and (iv) two different overlays (16 and 32 nodes each) spread across the globe. Using these datasets, we compare the performance of the “default” Internet path and other alternate paths derived by synthetically combining path metrics of disjoint nodes. Similar in spirit to randomized load allocation [12], we show that attempting to route key frames following a degradation using a random subset of 5 nodes is sufficient to recover from upto 90% of failures. We argue that our results are robust across datasets.

We further analyze the efficacy of a randomized path selection in large, unstructured overlays that can scale to service millions of users. Using weeklong measurements from 500+ vantage points in the Internet, we show that it is sufficient to reroute using ‘ $k$ ’ *randomly chosen* intermediate nodes for an overlay with  $N$  nodes. We show that the value of  $k$  is bounded by  $O(\ln N)$ ; which implies that  $k \approx 8$  for an overlay with 1000 nodes, and  $k$  is just 14 for an overlay with one-million nodes. We also observed that for random- $k$  to be effective, the subset  $k$  should be *uniformly representative* of the  $N$  nodes in the participating overlay.

Finally, we design and implement a prototype forwarding module in PlanetLab called source initiated frame restoration (SIFR). We evaluate the effectiveness of SIFR in improving



video-QoE against the default IP-path. We show that we can minimize and recover quickly from perceptual degradations, thereby raising perceptual quality on top of the best effort Internet. SIFR requires no modification to the Internet core, and can be seamlessly integrated into any source-destination pair that wishes to exchange multimedia content.

### **1.3 Structure of this Dissertation**

The rest of the dissertation is organized as follows. We provide a background on QoE and further motivate the contributions in Chapter 2. In Chapter 3, we develop a new tool called MintMOS that can capture video QoE in real time at arbitrary Internet nodes, without needing a reference frame to infer QoE. In Chapter 4, we study intra-ISP and inter-ISP links that make up today's Internet and look at their relative suitability for assuring superior QoE. Chapter 5 extends Internet measurements by analyzing end-to-end paths, and proposes a new redirection service to improve video QoE. We analyze the suitability of randomized path selection for large, unstructured overlays and build a prototype called SIFR that improves Internet path selection to raise video-QoE in Chapter 6. We analyze related research in Chapter 7, and conclude with pointers for future work in Chapter 8.

## CHAPTER 2

# QUALITY OF EXPERIENCE

“Quality in a product or service is not what the supplier puts in. It is what the customer gets out and is willing to pay for. A product is not quality because it is hard to make and costs a lot of money, as manufacturers typically believe. This is incompetence. Customers pay only for what is of use to them and gives them value.

Nothing else constitutes quality”

---

— Peter F. Drucker

### 2.1 Introduction to QoE

QoE is an attempt to characterize a vendors service from a customer standpoint. It is a combination of products, services, support, pricing and so on that make up a customers experience with a vendor. It tries to quantify, in the end, if the experience is consistent with what the customer wants or paid for. Finally, it postulates a set of principles that can ensure a consistent customer satisfaction for the price she pays. QoE as a concept has been successfully applied to a wide range of business domains to serve the customer better.

Understanding and improving QoE alone has had a profound impact on the long term success of many different organizations.

QoE is related to quality of service (QoS) but significantly differs from it as a concept. QoS is an *objective* measure of how a vendor performs, while QoE is a *subjective* measure of the same thing. QoS is closely tied to some form of a “contractual” agreement between the vendor and the client, and is likewise measured by the extent to which the vendor honors that contract.

Though a vendor may completely live up to such contractual agreements, the user of that service can be extremely unhappy with the experience. In such cases, though the vendor has a high QoS, the QoE perceived by the clients would rate very low. For example, a QoS in an airline industry might record the number of flights that depart and arrive at the scheduled times. An airline company may have a high QoS rating for achieving this objective, but its clients may be extremely dissatisfied with the airline for reasons such as bad food, lost luggage, and rude airline staff.

## 2.2 Multimedia QoS

The Internet was originally designed as a file sharing utility – it enabled a user to access information from another node in a remote location on the Internet with relative ease. For

much of the last decade, applications such as file transfer (**ftp**), web-browsing (**http**) and reliable message exchange (email) constituted a majority of all Internet traffic.

The Internet is further limited by a philosophy of end-system intelligence. This means that the core routers and packet forwarding modules that form the backbone of the Internet are necessarily “dumb”, while the end-nodes carry all the “intelligence”. This philosophy accounts for the extreme scalability of the Internet, allowing end-nodes to grow in terms of intelligence and processing capability while leaving the core untouched. This has made possible the interconnection of nodes that run on completely different hardware and software. This scalability has also come at a cost – the core routers need to be necessarily dumb and as such cannot guarantee any service quality. Core routers perform a basic functionality that can at best be summarized as “store-and-forward” while leaving the burden of packet recovery, out-of-order delivery and error resilience to the end nodes. This philosophy, however, proves detrimental for multimedia services when deployed on the Internet.

QoS was postulated to accommodate streaming content on the Internet. The earliest proposals towards this direction was the integrated services, or IntServ [6, 46, 47, 70]. IntServ operated with a notion of guaranteeing resources before transmission took place. In other words, every router along the path from source to destination would reserve resources before multimedia is exchanged. Reservations could include minimum statistical guarantees for latency, jitter, bandwidth and packet-drop rate. However, this violated the philosophy of end system intelligence, and it was practically impossible to co-ordinate such reservations across millions of routers on the Internet.

Another proposal to enhance Internet QoS was the differentiated services, or DiffServ [4, 5, 21, 39]. DiffServ operated with a notion that not all traffic in the Internet is equal, and likewise, routers should not behave consistently with every network traffic type. Internet traffic was classified into many different types, with each type emphasizing on a different metric. For example, applications like `ftp` often emphasize reliability over latency, while applications such as VoIP emphasize timely delivery over reliability. To differentiate and identify flows with different requirements, a type-of-service (ToS) field was introduced in the IP-header. This allowed routers to distinguish and apply differential treatment to various flows passing through it. DiffServ is hence a *flow* based mechanism, which is in contrast to IntServ which is a coarse-grained class based mechanism.

In the end, both DiffServ and IntServ could never be successfully implemented or deployed on a large scale in the Internet. This was largely because both these mechanisms were contrary to the end-system-intelligence philosophy of the Internet, and it was practically infeasible to co-ordinate its adoption across ISP's. As a result, present day Internet provides *no service guarantees for multimedia applications, and continues to work with a "best-effort" packet delivery model*. In a best-effort packet delivery model, the router tries its "best" to forward a packet. If the packet is lost or mis-route, the router takes no further responsibility since it tried its best. This simple packet forwarding philosophy leaves a wide gap between how multimedia performs on the Internet to what users expect multimedia to be like.

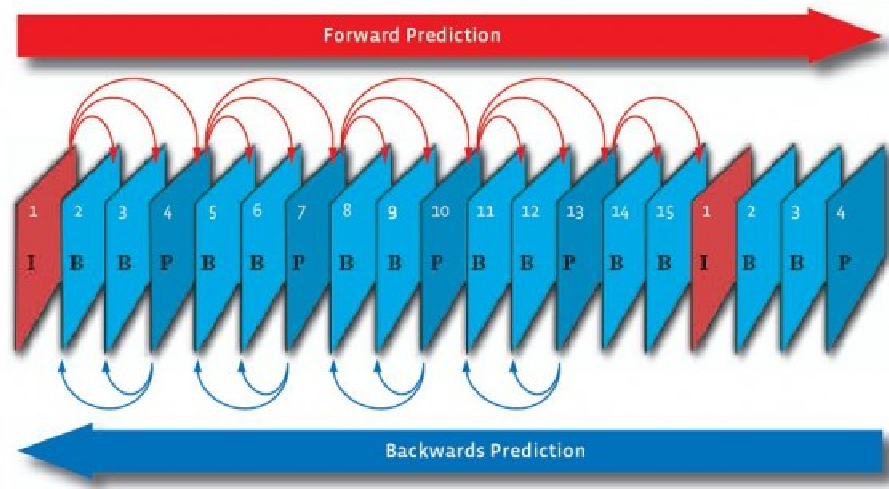


Figure 2.1: An MPEG-2 GOP structure which shows the different types of frames (I, P and B) and their basic relationship with each other.

## 2.3 An MPEG-2 Perspective

We next investigate how video streaming is achieved on the Internet, and provide an encoder/decoder perspective into QoE. We show how the same level of network QoS can manifest as completely different QoE based on the MPEG-2 frame structure, which motivates our need to investigate and improve Internet QoE.



Figure 2.2: Nature of artifact when loss affects different frames in a GOP: (a) freezing due to a corrupt B-frame, (b) slicing due to a corrupt P-frame and (c) ghosting/pixellation due to a corrupt I-frame. (Image originally appears in [20] and the Society of Motion Picture Television Engineers).

### 2.3.1 MPEG-2 GOP Structure

Streaming on the Internet is a process of encoding a video sequence, breaking it down into multiple IP-packets, transmitting them over the network, and re-assembling/decoding the packets into frames at the destination. A video streaming source (also called the “head end”) acquires video content from a third party or streams a stored content to its destination. In any case, the video content normally is an MPEG-2 stream, which is the most popular video container format in use today.

An MPEG-2 video stream is divided into multiple groups of pictures, or GOP. A GOP consists of three types of frames: (i) an intra-coded frame or I-frame carries a complete video picture which other frames use as a reference, (ii) a predictive or P-frame uses an I-frame for reference and accounts for motion, and (iii) a bi-directional or B-frame uses preceding I-

and P-frames to predict a frame. Figure 2.1 provides a high-level diagram of the internals of a GOP, showing the relationship between the various frames that constitute it.

### 2.3.2 QoS v/s QoE for MPEG-2

Each frame in a GOP is typically fragmented to multiple IP packets before transmission can take place. The Internet treats every packet alike, and network outages resulting in packet drops have an equal probability of affecting any frame. Since every frame in a GOP is not equally important, a network loss manifests artifacts that can range from a minor glitch to severe degradation. It is intuitive from the preceding discussion that an I-frame is the most important frame, and losses that affect an I-frame can be more damaging than other frames for the same given loss rate.

Packet losses that affect a B-frame result in frame “freezing”, which normally manifests as a motion picture that is stuck at one frame. A corrupt P-frame often results in “slicing”, which impairs a section of the playout while leaving others intact. A corrupt I-frame, on the other hand, often manifests extreme artifacts like “ghosting” or “pixellation” which render a playout barely perceptible. An example of slicing, freezing, and ghosting are presented in Figure 2.2. In other words, a similar level of network QoS (e.g., loss-rate) can result in completely different perceptual impairments depending on what frame was affected and how much of it is corrupt. This study establishes our need to divulge from QoS alone as a metric



for defining and assuring video quality on the Internet, and motivates us to investigate QoE as a factor and account for various other factors that might affect it.

## 2.4 Factors affecting QoE

QoE is the combined and cumulative result of a wide variety of factors. We discuss below some *measurable* parameters that are known to effect QoE, and motivate our contribution presented in this thesis.

**Media Quality:** The most important and obvious metric is an evaluation of how good a video sequence plays out to the user. Ideally, users would look for distortion free playouts. This implies that frames at a destination are available in-order (due to a playout buffer) with the same interspacing that they were sent. Users also look for high quality audio, and more importantly, synchronization of audio and video. Common network ailments include delay, packet loss, and delay jitter. Each of these degradations taken in isolation and combination leads to blockiness, blurriness, or even blackouts (like in Fig. 1.2(b)). Arguably, these areas are of primary importance to the networking research community which works towards minimizing and working around network related outages which result in these degradations.

**Availability:** Users would demand services round the clock, and especially when tuning into an event of importance (conference call, tele-surgery, live-sports etc.). Users want services that are stable and reliable. Most service providers target the “five nines” in reliability,

aiming to deliver services that are available 99.999% of the time. This implies that service providers need supporting middleware and services to *continuously* monitor degradations, and proactively fix problem before they arise.

**Pricing:** The pricing set by service providers will greatly dictate customer retention and service adoption. A competitive market will drive pricing, while customers look to maximize gains for the price they pay. Hence, there is a strive to optimize performance at reduced costs.

**Network Loss:** Loss contributes to missing information, and as such effects media playout. Network loss can occur due to a variety of reasons, and loss manifests as missing packets. The exact manifestation of loss, however, depends upon the type of frame impacted by loss. For example, a loss rate of 0.01% can manifest as a minor glitch that can go un-noticed, or can severely garble playout [20].

**Network Jitter:** Jitter is the variance in packet arrival times at the destination. Jitter can cause packets to arrive out of order, and most receivers implement a “playout buffer” to nullify the effects of jitter. Though the tolerance to network jitter is high, there is a cap to the amount of jitter a playout can tolerate. Jitter beyond a certain threshold is similar to network loss: if packets arrive out of sequence later than the buffering time, they are discarded at the receiver. The size of this playout buffer, however, can impact the interactivity presented by the system each time a user flips a channel. This is because channel change includes re-buffering delays: larger the buffer, more the tolerance to jitter, and more the time spent waiting for channel changes to take effect.

**Interactivity:** This reflects on the ability of system quickly and correctly change channels (zapping or flipping), forward, rewind, or service any user generated request. Acceptable channel change delay is established to be around 1 sec end-to-end delay. Channel zapping times of 100 – 200 ms are considered instantaneous. Much of the delay is incurred in transporting requests from a set-top-box (STB) to a server and back, which entails command processing, queuing delays, and video re-buffer delays. However, much of the functionality is implemented in hardware, making performance evaluation predictable and repeatable. Also, multicasting from edge-servers allows switching streams more amenable. Channel zapping is fairly understood and optimized by present day service providers.

## 2.5 Summarizing

This chapter provided a basic overview of QoE, and discussed its importance in the context of video streams encoded using MPEG-2. We looked at the structure of an MPEG-2 GOP, and how loss manifests as different artifacts based on the frame impacted. We additionally explored various factors that can affect QoE that go beyond encoding, such as media quality, interactivity, availability, pricing, and network QoS.

## CHAPTER 3

### MINTMOS: INFERRING VIDEO-QOE IN REAL TIME

Until you can measure something and express it in numbers, you have only the  
beginning of understanding

---

— Lord Kelvin

#### 3.1 Introduction

When a router or node in the Internet drops a video packet in transit, it has little knowledge of how this impacts a video streams' perceptual quality. Video-QoE is often computed by comparing video frames before and after transmission to check for differences. Inferring the perceptual quality of a video stream in transit at arbitrary nodes/routers in the Internet, where the original frame is *not* available for reference, continues to be an open problem.

In this chapter, we present MintMOS: a loadable kernel module that is an accurate, lightweight, no-reference framework for capturing and offering suggestions to improve QoE inside the network core [57, 61, 62]. MintMOS can accommodate an arbitrary number of parameters to base quality inference decisions, and encompasses both network dependent

and independent parameters. MintMOS internally consists of an inference engine (IE) to infer QoE, a suggestions engine (SE) to offer hints to improve QoE, a network sniffer to snoop traffic, and a QoE space. We take a closer look at each of these components and instrument an actual QoE space with 54 partitions using four parameters: loss, encoding bitrate, motion complexity, and type of frame impacted. We analyze the complexity of the framework in terms of memory and processing requirements. Finally, we deploy MintMOS with this QoE space on a 22-node wide-area measurement overlay on PlanetLab [82]. We stream IP-traces of various clips from every node to every other node for one week and used MintMOS to predict quality and detect outages. We validate the accuracy of QoE predictions using extensive subjective surveys, and comparing our result with the survey.

### 3.2 Framework Architecture

An architectural overview of MintMOS is shown in Figure 3.1. The framework consists of the following components: the sniffer, the QoE inference engine (IE), the suggestions engine (SE) and the QoE space. The sniffer sniffs packets from the networking interface, and generates statistics about the parameters in question (e.g., loss, delay, jitter and so on). The sniffer feeds these parameters as input to the QoE inference engine. The IE, alongwith the SE, operates on the QoE space to both predict the current QoE as well as offer suggestions to improve the perceptual quality of the video stream. Modularity allows these components to

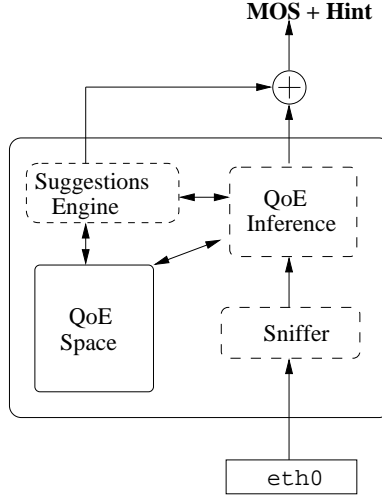


Figure 3.1: Architectural overview of MintMOS

evolve independently of each other. We begin with the QoE space and take a closer look at each of these components.

### 3.2.1 QoE space

The first step in putting together the MintMOS framework is the creation of a QoE space. A QoE space is a known characterization of expected QoE for various values of the  $k$  parameters that affect it. In general, if we assume  $k$  parameters that affect the quality of video then those  $k$  parameters can be used to express the QoE space in the form of a  $k$ -dimensional vector,  $\Gamma$ , as follows:

$$\Gamma = [\gamma_1, \gamma_2, \dots, \gamma_i, \dots, \gamma_k]$$

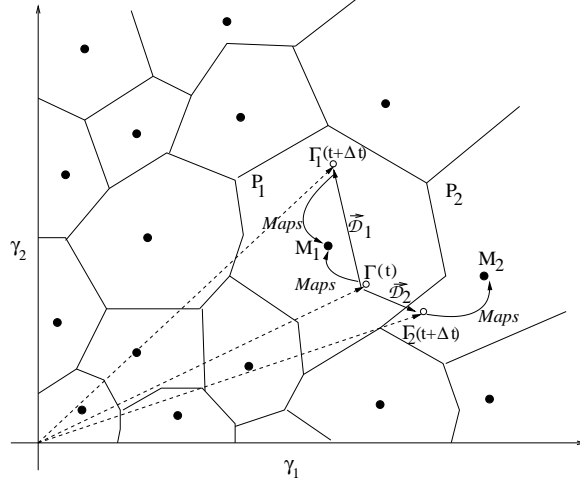


Figure 3.2: A sample 2-D QoE space constructed with two parameters  $(\gamma_1, \gamma_2)$ . The solid dots denote precomputed QoE for known input values of  $\gamma_1$  and  $\gamma_2$ . The expected QoE for a new sample  $(\gamma_1, \gamma_2)$  is obtained by calculating the Euclidean distance between the new observation point and known indices.

where  $\gamma_i$ ,  $1 < i < k$ , represents the instantaneous value of the  $i$ th parameter. Thus, the vector  $\Gamma$ , provides the instantaneous state of the video stream in transit. Due to the dynamism of the network, parameter  $\gamma_i$  constantly changes, and so does the vector  $\Gamma$ . The ever-changing vector  $\Gamma$  can be interpreted as the motion of a point in a  $k$ -dimensional QoE space. Borrowing concepts from Vector Quantization [17], we argue that associated with every point in this space is a *QoE index* which represents the expected quality of experience.

This is better depicted in Fig. 3.2, which represents an abstract 2-dimensional QoE space for two parameters with positive values. This space is partitioned into numerous non-overlapping regions  $(P_i)$ . Each of these regions will have a *reference* point, usually the

*centroid* (shown by solid dots), and any point in that region will map to that reference point. For example in Fig. 3.2, let  $\Gamma(t)$  be the position of the vector (2-dimensional) at time  $t$  belonging to partition  $P_1$  with reference point  $M_1$ . We consider two possible scenarios at time  $(t + \Delta t)$ : (i)  $\Gamma_1(t + \Delta t)$  undergoes a displacement of  $\vec{\mathcal{D}}_1$  and remains in the same partition  $P_1$ . Hence it continues to map to  $M_1$ , yielding the same QoE index; (ii)  $\Gamma_2(t + \Delta t)$  undergoes a displacement of  $\vec{\mathcal{D}}_2$  but moves to a different partition  $P_2$ , and maps to a different reference point  $M_2$ .

### 3.2.2 Choice of Number of Partitions ( $N$ )

For any  $k$ -parameters with  $x_i$  divisions ( $1 \leq i \leq k$ ), one would need  $N = \prod^k x_i$  number of partitions. The choice of  $x_i$  depends on the sensitivity of the parameter values to perceived quality. One can use both network dependent and network independent parameters that can affect QoE.

An alternative way of choosing partitions is to have enough pre-computed indices such that the projected MOS is within a tolerable *distortion*. Distortion is a measure of the level of inaccuracy in predicting MOS. Thus, the tolerable distortion can be used to back-calculate the number of QoE indices,  $N$ , to create the sample space. More explicitly, the average (mean square) distortion  $D$  is given by:

$$D = \sum_{i=1}^N \int_{\mathcal{R}_i} (x - M_i) f_X(x) dx, \quad (3.1)$$



where  $M_i$  is the QoE index in the region  $\mathcal{R}_i$  and  $f_X(x)$  is the probability mass function (pmf) of the random variable  $X$ . The entire QoE space is a summation of all such regions  $\mathcal{R}_i$ . The total distortion hence is a sum of the individual distortions in every region, taken over the number of QoE indices ( $N$ ). We integrate within each region  $\mathcal{R}_i$  because the QoE space is continuous, thus capturing the temporal aspect of the instantaneous values of the QoE parameters. Thus, if the tolerable distortion ( $D_{thresh}$ ) is given, the number of QoE indices ( $N$ ) can be obtained as:

$$N = \max \left\{ n : \sum_{i=1}^n \int_{\mathcal{R}_i} (x - M_i) f_X(x) dx < D_{thresh} \right\} \quad (3.2)$$

### 3.2.3 QoE Inference Engine (IE)

The IE is a component that takes instantaneous values of the  $k$  parameters (the vector  $\Gamma$ ) as input, and produces the expected QoE as the output. The IE searches the QoE space to find the closest match to the partition to which  $\Gamma$  belongs, and returns the QoE associated with that partition.

When a new set of values are provided to IE, the expected QoE is derived by selecting the least Euclidean distance between the given point and the centroids in the QoE space. One way to find the target QoE is to apply the 1-Nearest Neighbor algorithm, a special case of the  $k$ -Nearest Neighbor algorithm, which assumes that all instances are mapped to points in the  $k$ -dimensional space. Consider  $N$  pre-computed QoE points and call them  $M_j$ , where

$1 \leq j \leq N$ . The co-ordinates of  $M_j$  in the  $k$ -dimensional space is  $[M_{j1}, M_{j2}, \dots, M_{jk}]$ . The *distortion*  $d_j$  of an instance  $\Gamma = [\gamma_1, \gamma_2, \dots, \gamma_k]$  is simply the Euclidean distance from  $\Gamma$  to the target point [17]:

$$d_j = \sqrt{\sum_{i=1}^{i=k} (\gamma_i - M_{ji})^2}. \quad (3.3)$$

It is intuitive that more the number of pre-computed QoE indices (i.e.,  $M$ ) less would be the distortion and more accurate the QoE prediction.

### 3.2.4 Improving QoE: Suggestions Engine (SE)

Apart from inferring QoE, network service providers would also look for hints from the network to improve perceptual quality. For example, many existing QoS provisioning modules adapt encoding bitrate or frame rate based on feedback received from the network. These feedbacks are either round trip time (RTT) estimations or packet loss probabilities. However, these feedbacks in themselves offer little help in improving perceptual quality [14, 53].

The SE provides *application specific* hints or suggestions as a feedback to help improve perceptual quality. The SE views every partition in the QoE space as an *operating point*, and the parameter values associated with that partition as the default state of the system. By design, every operating point can be mapped to another operating point with a higher QoE by changing exactly *one* parameter value, unless the operating point is already at the highest possible QoE. For example, consider a QoE space of three parameters: loss, motion

complexity, and bitrate. For a partition operating at  $\Gamma_A = (L_i, C_i, B_i)$ , if there exists a  $\Gamma_B$  in the QoE space with a higher perceptual rating such that two parameters are in close agreement (e.g.,  $|L_i - L_j| < \delta_L$  and  $C_i = C_j$ ) while the third parameter is different,  $B_j > B_i$ , then there exists a possible transition to improve QoE by increasing bitrate. In this way,  $\Gamma_B$  can be further mapped to a  $\Gamma_C$  with a higher QoE by changing one parameter in  $\Gamma_B$ . In general, for any operating point, we note that there exists a set of transitions that can move a given operating point to the highest possible QoE. However, the transitions inherent in the QoE space may not always be practically applicable. To better leverage transitional relationships between operating points, *application specific* filters can be defined to infer feasible transitions within the QoE space. For example, one such filter could be to only suggest increasing bitrate when current path loss rates are within a certain threshold.

In general, SE offers suggestions to increase bitrate, decrease bitrate, choose alternate paths with lesser loss, or raise a flag asking a node to preserve certain packets carrying key frames. We leave the exact choice of adaptation to the application programmer. SE provides timely hints to make way for plethora of adaptation policies to be built on top of it.

### 3.2.5 Sniffer

The sniffer is a module that sniffs the network interface to capture packets from the live wire. It extracts parameters from the packet to compute networking statistics about a given flow (e.g., loss, delay, jitter). The Sniffer is written using the `libpcap` [76] library, which

is a low level packet capturing facility used to build popular tools like the `tcpdump`. The Sniffer cannot always capture network independent parameters (like encoding bitrate). To do this, the source must pass this information either in well defined headers or as a field in the payload. The Sniffer maintains a window of statistics in a FIFO manner, and computes the average value of every parameter after every successfully captured packet. For example, if one sets the window size to be 30, the Sniffer maintains statistics about the last 30 packets for a given flow. The Sniffer then passes the average value of 30 records for each parameter as an input to the IE.

### 3.3 Instrumenting MintMOS

In this section, we create a working MintMOS framework with an actual QoE space populated offline by subjective surveys. A QoE space is constructed by obtaining various video sequences with various values of the  $k$ -parameters. These video sequences are then shown to a large number of human subjects, who individually score the video samples. We create a four dimensional QoE space that involves the following parameters: loss, encoding-bitrate, type of frame impacted, and motion complexity inherent in the clip.

Loss refers to *missing information* in a 1000 msec window of observation. Missing information aggregates the result of both IP packet drops and jitter, which can cause packets to arrive out of order resulting in missing sequence numbers in the observation window. Typical loss fractions in the Internet are less than 0.1, and we choose five steps for loss:

Table 3.1: The first 16 partitions of a QoE space with 54 partitions. QoE space of four dimensions (bitrate, motion complexity, best/worst case impact, and loss rate). The remaining entries in the table can be found in [62]

| Seq.<br>No. | Bitrate<br>(kbps) | Motion<br>Complex. | Best(0)/<br>Worst(1)<br>Case | Loss<br>(frac.) | Avg.<br>MOS | Var   |
|-------------|-------------------|--------------------|------------------------------|-----------------|-------------|-------|
| 1           | 800               | Low                | 0                            | 0.01            | 2.55        | 0.267 |
| 2           | 800               | Low                | 0                            | 0.05            | 2.45        | 0.253 |
| 3           | 800               | Low                | 0                            | 0.1             | 2.375       | 0.212 |
| 4           | 800               | Low                | 0                            | 0.5             | 1.675       | 0.318 |
| 5           | 800               | High               | 0                            | 0.01            | 1.875       | 0.077 |
| 6           | 800               | High               | 0                            | 0.05            | 1.875       | 0.290 |
| 7           | 800               | High               | 0                            | 0.1             | 1.8         | 0.290 |
| 8           | 800               | High               | 0                            | 0.5             | 1.775       | 0.026 |
| 9           | 800               | Low                | 1                            | 0.01            | 1.65        | 0.245 |
| 10          | 800               | Low                | 1                            | 0.05            | 1.575       | 0.245 |
| 11          | 800               | Low                | 1                            | 0.1             | 1.475       | 0.139 |
| 12          | 800               | Low                | 1                            | 0.5             | 1.45        | 0.257 |
| 13          | 800               | High               | 1                            | 0.01            | 1.6         | 0.202 |
| 14          | 800               | High               | 1                            | 0.05            | 1.6         | 0.249 |
| 15          | 800               | High               | 1                            | 0.1             | 1.5         | 0.535 |
| 16          | 800               | High               | 1                            | 0.5             | 1.475       | 0.134 |

(0, 0.01, 0.05, 0.1, 0.5). For encoding bitrates, we choose three stops (kbps): 800, 3200, and 6400. We consider two cases for type of frame impacted: impact on an I-frame ( “*worst cast*”, denoted by 1) and impact on other (P,B) frames ( “*best case*”, denoted by 0). We used one high motion clip (*Football*) and one low motion clip (*Foreman*) to build our QoE space. High motion clips have frequently changing scenes, while low motion clips are less dynamic. These clips were encoded using MPEG-2 Main Profile. We used a 15:2 GOP structure at 30 frames per second (fps) to encode them. The transport stream was recorded at the IP layer using an Ineoquest Singulus video analyzer. We manually edited the IP trace to define the desired loss on various frame types.

### 3.3.1 Creating a QoE space: Survey with Human Subjects

We generated 48 clips for four parameter values for loss (excluding  $loss = 0$ ), two values for ‘type of frame’, three encoding bitrates and two types of motion complexity ( $4 \times 2 \times 3 \times 2 = 48$ ). For  $loss = 0$ , we additionally generated 6 clips for the three encoding bitrates and two motion complexities. In summary, we create  $48 + 6 = 54$  partitions of the QoE space with these clips<sup>1</sup>.

These clips were put to a survey with subjects, who were requested to rate these video sequences on a scale of 1 to 5. We requested 80 subjects in an indoor lab environment, and an additional 169 subjects online, to score these samples. Subjects were chosen with

---

<sup>1</sup>Visit [79] to examine all our video clips.

sufficient diversity in age, gender, and expertise in subject matter. Outliers were identified by interspersing a shown video sequence multiple times and recording their ratings each time. We identified a total of 3 outliers in the lab environment, and 26 outliers from our online surveys. In effect, we collected surveys from 77 subjects indoors, and 143 subjects online, to characterize our QoE space. Table 3.1 shows the values of our four choice parameters, along with the average MOS and variance obtained from the survey. A few observations are noteworthy about perceptual quality using these parameters:

**Low Motion Clips:** Loss of P- or B-frames seem to make little difference in perceptual quality. The difference in perceptual quality for the loss of an I-frame over a P-frame, however, is drastic (for e.g., compare sample-17 with sample-25 or sample-33 with sample-41). Low motion clips have larger I-frames, which increase their odds of getting impacted during an outage. Low motion clips have longer GOP structures with more P-frames, which enable more compression. Because of their long GOP structures, the *duration* of on-screen degradation tends to be longer. Interestingly, for both I- and P-frame losses, an increase in loss rate from 0.05 to 0.5 seems to make little difference.

**High Motion Clips:** An I-frame loss results in severe degradations, similar to low motion clips. However, because of the inherent dynamism, P-frames tend to be larger and more informative. Hence, loss of a P-frame draws a slightly more adverse reaction from the subjects than with low motion clips (sample-17 v/s sample-21). Increased loss of P-frames in a GOP gradually degrades quality. Because scenes change frequently, I-frames tend to

be shorter and less probable candidates for loss during outages. For I-frames, an increase in loss rates seems to hardly degrade quality any further.

### 3.3.2 Running SE

Once the QoE space is characterized, we run the SE upon it to discover all possible pairs of sample points such that the perceptual quality is improved by changing one parameter from a given operating point. The SE component mines through the QoE space and creates a one time mapping of all possible combinations that can improve QoE. This process is *one time only*: the relationships between entry points are static because the QoE space is static.

The SE returns a vector of the form  $[A|B]$ , where **A** suggests policies for preserving key frames, and **B** indicates the likely action that can improve QoE. The value **A** is set to 1 when key frames are corrupt and 0 otherwise. Values of **B** are reflective of actions that are both application specific and result in improving QoE.

We use the following rules to infer values of **B**. An ‘Increase Bitrate’ (**B=1**) is observed when current loss rates are no more than 0.1. Paths are suggested to be switched (**B=2**) when loss rates touch 0.5 or above. When frame preservation policies *alone* can improve QoE, we set **A=1** and **B=0**. When **A=1**, the value of **B** is reflective of the course of action *after* key frames are preserved.



Table 3.2: A subset of “From-To” mappings and suggestions discovered by SE in the QoE space.

| <b>Suggestion</b>                                  | <b>(From, To) Mapping Sets<br/>in the QoE space</b>  | <b>Hint</b> |
|--|--|-------------|
| Increase bitrate                                   | (1,17), (2,18), (3,19), (5,21), (6,22),<br>(7,23), (17,33), (18,34), (19,35)                             | [0 1]       |
| Decrease loss/<br>Switch path                      | (4,1), (8,5), (20,17), (24,21),<br>(36,33), (40,37)  | [0 2]       |
| Preserve Policy                                    | (41,33), (42,34), (43,35), (44,36),<br>(45,37), (46,38), (47, 30), (48, 40)                              | [1 0]       |
| Preserve Policy +<br>Increase bitrate              | (9,1), (10,2), (11,3), (13,5), (14,6),<br>(15,7), (25,17), (26,18), (27,19)<br>(29,21), (30,22), (31,23) | [1 1]       |
| Preserve Policy +<br>Decrease Loss/<br>Switch Path | (12,9), (16,13), (28,25), (32,29)  | [1 2]       |

Table 3.2 shows the output of SE when we ran it over our QoE space. For brevity, we present a subset of all possible suggestions. SE returns an integer which is interpreted as a possible course of action to improve QoE. A suggestion for an entry point is due to a “From-To” relationship between that and another partition with a higher QoE. For example, SE suggests an increase in bitrate for (1,17). This means that when the operating point is close to entry-1 in the QoE space with current loss rates around 0.01 (see Table 3.1), SE suggests an attempt to increase bitrate to move it closer to entry-17’s operating point. Likewise, SE discovered a ‘switch path’ to decrease loss rate for (4,1). Decreasing loss rates of 0.5 or above can move the operating point from sample-4 to that of sample-1 with a higher QoE. The other pairs of relationships shown can analogously be investigated in our QoE space in Table 3.1. SE returns a value ‘0’ when it discovers no relationship to with any other partition. Also, if multiple suggestions are possible for an operating point, SE by default returns that suggestion which moves the present operating point to the one with highest possible QoE.

When an operating point is input to SE in real-time, it simply returns the possible suggestions that it has discovered offline within the QoE space.

### 3.4 Validating the framework

With the QoE space thus characterized, we turn to validating the accuracy of MintMOS’s predictions in inferring perceptual quality. To do this, we deployed MintMOS on 22 Planet-Lab [82] nodes spread geographically across the globe. We specifically chose nodes in Europe,

Asia and the United States where streaming services are very popular. Between Jan. 14 to Jan. 21, 2010, we randomly selected one (out of a total of five clips) to be streamed to a randomly selected node every minute. We used two high motion clips (*Football(a)*, *Tennis*) and three low motion clips (*Coastguard*, *Mobile*, *Foreman*). These clips were recorded at the IP-level using an Ineoquest Singulus digital media analyzer with a fragmentation limit of 1024 bytes. These traces were used to generate UDP packets of 1024 bytes, and we passed information such as motion complexity, frame type, and encoding bitrate in the packet header. The ingress packets at the destination were input to MintMOS which analyzed loss rates in windows of 1000 ms.

This experiment resulted in a rich trace of Internet outages detected by MintMOS, and their corresponding quality rating. To begin with, we seek to understand the degree of accuracy of MintMOS’s projection to user quality perception. To do this, we reconstructed a set of 14 video clips using their packet traces at various loss rates observed on different paths. MintMOS’s predictions at these times were compared with ratings from a survey using 49 subjects in an indoor lab environment. We use the average scores assigned by subjects as the benchmark to evaluate the accuracy of various scoring schemes. We also compare our results with more established metrics like PSNR and VQM. The interested reader is invited to try out the test for herself by downloading all our clips from this anonymous site [79]. Table 3.3 tabulates our findings. Shown in the table are the input parameters used to create the sequence of 14 video samples, the type of clip used, the PSNR readings for the samples, the VQM ratings (as obtained from the MSU toolkit [81]), the mean MOS assigned

Table 3.3: Results from a survey conducted to measure the accuracy of MintMOS projections

| Seq.<br>No. | Clip<br>Name  | Bitrate<br>(kbps) | Loss<br>(rate) | Frame<br>Type (0/1) | PSNR<br>(dB) | VQM   | MOS from<br>Survey | MintMOS<br>Projection | Error<br>(%) | Hint  |
|-------------|---------------|-------------------|----------------|---------------------|--------------|-------|--------------------|-----------------------|--------------|-------|
| 1           | 'Coastguard'  | 800               | 0.035          | 0                   | 18.121       | 22.45 | 2.62               | 2.45                  | 3.4          | [0 1] |
| 2           | 'Coastguard'  | 3200              | 0.43           | 1                   | 20.227       | 4.83  | 1.125              | 1.10                  | 0.5          | [1 2] |
| 3           | 'Coastguard'  | 6400              | 0.17           | 0                   | 17.258       | 10.42 | 4.3                | 4.225                 | 1.5          | [0 0] |
| 4           | 'Mobile'      | 800               | 0.09           | 1                   | 22.193       | 32.47 | 1.46               | 1.475                 | 0.3          | [1 1] |
| 5           | 'Mobile'      | 3200              | 0.63           | 1                   | 6.731        | 31.23 | 1.25               | 1.10                  | 3.0          | [1 2] |
| 6           | 'Mobile'      | 6400              | 0.23           | 0                   | 19.075       | 11.23 | 4.35               | 4.225                 | 2.5          | [0 0] |
| 7           | 'Foreman'     | 800               | 0.015          | 0                   | 23.682       | 15.69 | 2.43               | 2.55                  | 2.4          | [0 1] |
| 8           | 'Foreman'     | 3200              | 0.07           | 1                   | 7.936        | 27.33 | 1.31               | 1.125                 | 3.7          | [1 1] |
| 9           | 'Football(a)' | 800               | 0.01           | 1                   | 21.239       | 5.31  | 1.55               | 1.6                   | 1.0          | [1 1] |
| 10          | 'Football(a)' | 800               | 0.1            | 1                   | 8.623        | 28.32 | 1.32               | 1.5                   | 3.6          | [1 1] |
| 11          | 'Football(a)' | 3200              | 0.42           | 0                   | 16.94        | 15.21 | 2.25               | 2.3                   | 1.0          | [0 2] |
| 12          | 'Tennis'      | 3200              | 0.33           | 1                   | 19.231       | 11.93 | 1.62               | 1.65                  | 0.6          | [1 2] |
| 13          | 'Tennis'      | 3200              | 0.2            | 0                   | 24.513       | 4.23  | 3.53               | 3.625                 | 1.9          | [0 1] |
| 14          | 'Tennis'      | 6400              | 0.05           | 0                   | 25.239       | 28.69 | 3.27               | 3.2                   | 1.4          | [0 0] |

by 49 subjects, the MOS assigned by our framework, as well as the error percentage of our predictions.

**PSNR**, which directly compares a processed sequence with the original, often fails to correctly correlate perceptual quality to degradation. For example, PSNR rates samples 1 and 3 with a similar reading ( $\approx 18$  db). Subjects, however, regard sample-3 of a relatively higher quality than sample-1. Likewise is the case with sample pairs (6,7) and (11,12).

**VQM**<sup>2</sup> projections provide a better indication of perceptual quality than PSNR. While VQM does identify clips with lower perceptual quality to some degree, the results are not too accurate. For example, VQM rates sample-14 relatively poorly while subjects rate this clip between ‘acceptable’ to ‘good’. Again, subjects rate samples 2 and 5 almost at par, while VQM predicts sample-2 to be far better than sample-5. In general, VQM’s ability to loosely classify a clip as very poor or very good is accurate, it often fails to successfully distinguish clips which share similar levels of degradation. Note that VQM is a full reference scheme, which requires the original frame at all times to provide quality evaluation.

**MintMOS**’s prediction was able to obtain the highest correlation with subjective perception. Noteworthy is the degree of correlation, with a maximum error rate of 3.7% for sample-8, and a minimum of 0.3% for sample-4. MintMOS was successfully able to differentiate between high, medium and poor perceptual quality. Our MOS projections are dependent on the accuracy of the QoE space. Since the QoE space is in turn populated *by* human subjects, the degree of correlation to a past event with a similar newer event is what makes our

---

<sup>2</sup>VQM projections work in an inverse way: higher the value indicated, lower shall be the perceived quality.

projections accurate. The mean distortion in using MintMOS with 54 reference points was determined to be 0.14. For smaller sample spaces ( $N = 30$ , by removing 24 QoE indices), we measured average distortion to 1.02.

**Improving QoE:** MintMOS also outputs a hint to improve QoE for every sample. For example, it suggests sample-1 to increase bitrate with a hint of [0|1], with the first value (0) indicating that key frames were not lost. Sample-1’s parameters suggest it is operating close to entry-2 in the QoE space. Increase bitrate is due to a (2,18) relationship (Table 2), where entry-18 has a higher MOS. Other suggestions may likewise be investigated by determining the entry point in QoE space to which a sample maps to, and their corresponding “From-To” relationship to other partitions.

### 3.5 Complexity of the Framework

For MintMOS to be an effective tool, it must scale well in servicing a large number of flows in real time at various points in the network. To measure the processing overhead of our framework on hardware, we implemented MintMOS in C++ with a precomputed QoE space. The module takes as input values of parameters in a specific range, and returns the anticipated MOS upon completion. We report our experience with running MintMOS on standard, off-the-shelf, Linux terminals (running a Pentium 4 with 2GB RAM). We measure the time it takes to perform MOS calculations with increasing number of partitions of the QoE space ( $N$ ) as well as the number of parameters to estimate QoE ( $k$ ). Our results create

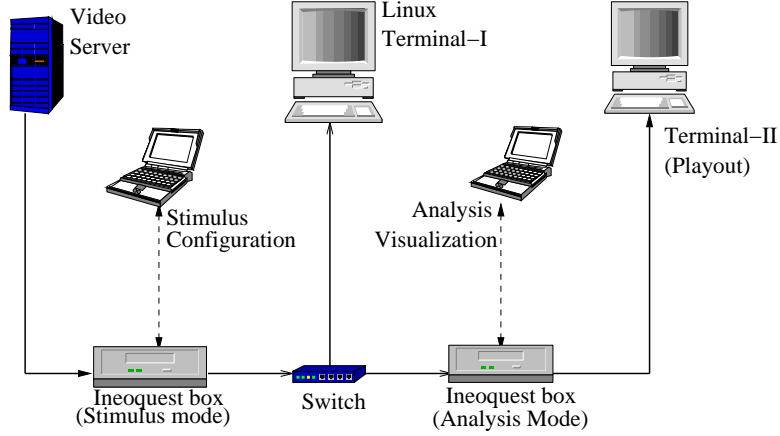


Figure 3.3: Overview of the Experiment Topology

a benchmark for using this framework on routers and end-user systems with comparable hardware.

Wall clock time (or the gross time between submission and getting the result) is a poor representative of complexity of the algorithm, since it takes into account all other resident processes in the system. Hence, we profile our module to gather the direct CPU cost to run it as a user level program as well as a kernel module. For the rest of this section, we report the *sum* of the user level time as well as the system (kernel) level time.

For these round of experiments, we use a QoE space with upto one million (spurious) entries. We report the *worst case* running time, where we assume that the desired QoE index is found in the last comparison. In other words, if there were 1000 reference points and 20 parameters, the desired MOS is found after having compared the 1000 distortions to all 20 QoE indices.

### 3.5.1 Experimental Setup

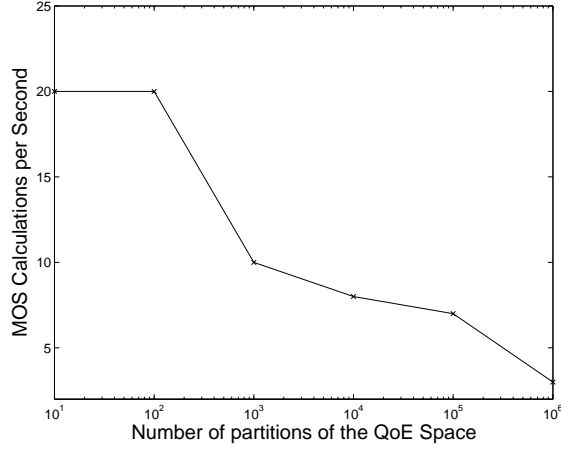
The topology used to measure complexity is shown in Figure 3.3. Our set-up consists of a video server which initiates a push based stream onto the network. The server can encode a video stream at various bit-rates and frame-rates. The server is connected an IneoQuest (IQ) Sigulus G1-T [77] box configured in the ‘stimulus’ mode. In this mode, the IQ can generate impairments to the flow by inducing loss, delay, or jitter. The flow is streamed to another IneoQuest G1-T configured in the ‘analysis and playback mode’. The stream is then passed on to a destination (Terminal-II) for visualization.

The boxes are connected through a switch, and we attach a Linux terminal (Terminal-I) to this switch. Terminal-I runs MintMOS by passively snooping network traffic and inferring QoE. Terminal-II, which is used for playback, also implements MintMOS where it measures perceived quality at destination. We report the measured complexity of MintMOS at Terminal-I.

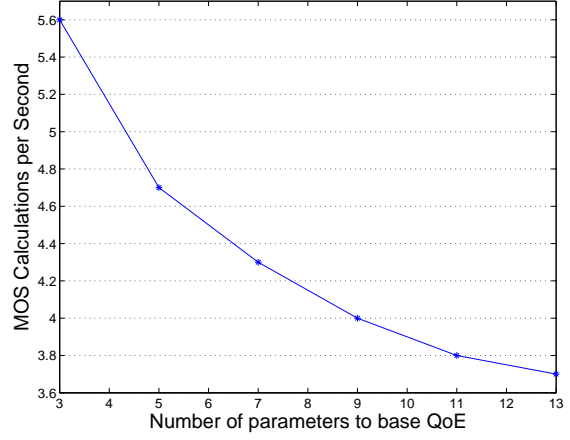
### 3.5.2 Effect of number of samples ( $N$ )

The accuracy of MOS projections gets richer with more reference points in the QoE space. An increase in the number of reference points means an increase in the number of comparisons required to infer MOS.





(a)



(b)

Figure 3.4: Processing overhead for MintMOS: (a) Effect of increasing number of (log) partitions for  $k = 11$ ; and, (b) Effect of increasing number of parameters for  $N = 10^6$

The effect of increasing the number of QoE space partitions (log scale) to the output rate is shown in Fig. 3.4(a). The output rate is the number of MOS calculations that can be performed *per second* for the given number of QoE indices to be compared with. For a small number of QoE indices, with less than 100 partitions, increasing the number of partitions has no real effect in the output rate of the program. The effect only begins to show when the number of partitions approaches 1000 or more. Shown in the plot is the effect of increasing the number of partitions to one million samples. Even at such a large number, we could perform 4 MOS calculations per second. Since our time measurement takes into account both the user level interaction with the kernel as well as the actual CPU time that the program consumes, this reflects on our ability to perform MOS calculations in real time on a router with comparable hardware. In other words, our module can compute 1200 MOS calculations

per minute for a small number of partitions ( $< 100$ ), and can compute 240 computations per minute even for one million reference points to compare to. For a reasonable number of reference points, MintMOS can service more than a thousand flows per minute on a router with a similar configuration.

### 3.5.3 Effect of number of parameters ( $k$ )

The cost of adding more parameters, or axes, to the QoE space should not come at a high cost. In fact, more the number of parameters used in inferring MOS, more accurate shall be the projections. We examine the effect of increasing the number of parameters in calculating MOS for a QoE space with one million partitions by varying the number parameters from 3 to 13.

Fig. 3.4(b) shows the effect of increasing the number of parameters. When increasing  $k$ , the only difference to the module is to compute  $k$  distances for each reference point. Even with a large number of reference points in the sample space, increasing the value of  $k$  from 3 to 13 has little effect on the output rate, which drops from around 5.5 MOS calculations per second to around 3.75 calculations per second. Again, this can be inferred as servicing 210 flows per minute for 3 parameters, and upto 23 flows per minute for 13 parameters for one million reference points to compare to.

### 3.5.4 Size of QoE space

We measure the memory consumption of MintMOS during execution. The executable code apart, the only significant source of memory consumption is the QoE space. We measure the amount of memory allocated to keep the *entire* QoE space in main memory during program execution.

For 13 parameters and upto 1 million entries in it, QoE space consumes about 21MB of resident memory. This is fairly lightweight even for such a large number of sample points to compare to. This means that even an extensive QoE space will not require database support, and can locally reside on a number of network nodes as a standalone module.

## 3.6 Discussions

During the week-long measurements using MintMOS on a 22-node wide area overlay network, we observed a total of 329 outages. Though a single packet can potentially corrupt playout, we distinguish an outage from a packet drop caused by a transient congestion by counting instances where *three* or more consecutive sequence numbers were lost, thereby corrupting an entire frame. Fig. 3.5 shows the CDF of the number of *consecutive* frames impacted during times of such outages. We note that more than 10 consecutive frames are lost in more than 60% of the cases. The more number of contiguous frames affected during an outage, more likely are the chances than an I-frame is subsequently corrupted. In effect, even best case

outages can easily be converted to worst case outages if the degradation goes unchecked and unreported. Real time path quality estimates from MintMOS can help detect *and* recover from outages in a timely fashion.

In this section, we discuss various extensions of MintMOS in scenarios which benefit from a knowledge of path quality measurement. Any pair of nodes running MintMOS can both measure and exchange path reachability information in terms of perceived quality. We look at the case of overlay networks, VoD/IPTV service providers, CDNs, and collecting end user QoE; all of which can use MintMOS to infer and improve perceptual quality.

**Overlay Networks:** An overlay network can be imagined as a network built on top of another network. Nodes in an overlay network are connected by virtual or logical links, which are the default IP unicast paths in case of the Internet. Though overlay networks do not have complete control over the virtual links built using the underlying network, they certainly have the ability to select the *sequence* of nodes traversed to reach a destination. As such, nodes running MintMOS can enable path switching based on perceived quality. Further, they can *switch* to paths with greater QoE in short times scales (seconds) compared to minutes of time lag before BGP advertises these outages.

**Collecting End-User's QoE:** Most network service providers are at a complete loss when it comes to keeping a log of QoE perceived by *all* of their clients. Our framework can be easily adapted to run on virtually any client laptop or desktop to monitor user perceived quality for a small set of flows that the user is viewing. These statistics can be bundled together and sent back to source providing continuous feedback and making way for a comprehensive

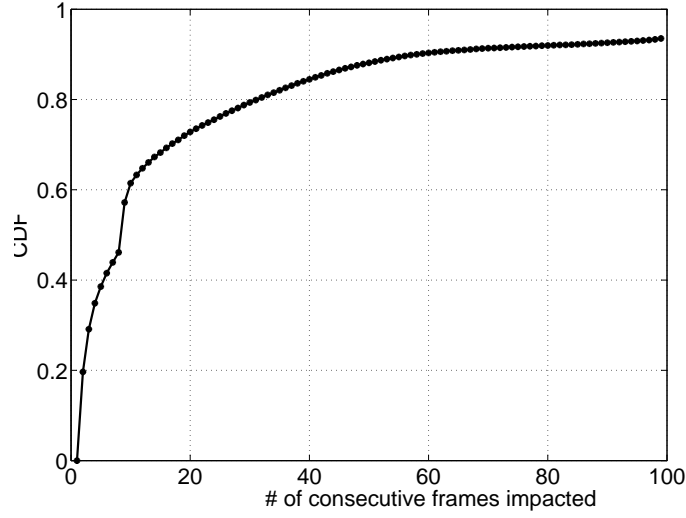


Figure 3.5: Duration of outages discovered by MintMOS over a one week period on a 22 node wide area measurement overlay built using PlanetLab.

log of QoE fluctuations. Such a log can be invaluable to service providers helping them plan capacity, monitor peak hour QoE, as well as for billing their clients.

**VoD and IPTV:** Service provers of VoD and IPTV need to continuously monitor QoE at five critical points in their network infrastructure: the source or server, the A/D servers, the MJoin, the EQAM and the end user set-top-boxes.

Our framework can seamlessly be integrated at these critical points and provide perceptual QoE at these points to be streamed to the control plane, even for a large number of flows. This would allow service providers to monitor QoE and its respective degradation at all five points. This has direct implications in fault isolation, billing clients, capacity planning, and purchasing new hardware to improve QoE.

**Content Delivery Networks:** CDNs move streaming content “closer” to clients by caching copies on multiple servers worldwide. CDNs perform extensive network and server measurements, and use these to redirect clients to different servers over short time scales (e.g., Akamai [74], the worlds largest and most popular CDN).

A study by Su *et. al.* [50] from 140 vantage points in the Internet reveals that Akamai redirections overwhelmingly correlate with network *latencies*. This implies that server redirection are based on delay more than any other parameter. Servers in a CDN can utilize their existing network measurements to estimate loss and delay rates, and utilize a QoE space to perform server redirections based on *estimated QoE* rather than network latencies. Feedback from MintMOS can further be used to decide encoding bitrates.

### 3.7 Conclusions

We presented MintMOS: a lightweight, scalable, no reference framework for inferring QoE of a video stream and offering suggestions to improve it. MintMOS is flexible enough to accommodate any number of parameters that can affect video quality, from network dependent to network independent. MintMOS revolves around a QoE space, which is a  $k$ -dimensional space for  $k$  parameters used to measure quality. The QoE space creates a mapping between parameter values and their associated perceptual quality. Inferring QoE or offering suggestions to improve it use the QoE space to base their decisions. We instrumented a

QoE space around 4 parameters and 54 partitions, and demonstrated its effectiveness in projecting MOS.

We deployed MintMOS using this QoE space on 22-node wide area overlay on top of the Internet using PlanetLab. We used this overlay to stream IP-traces of a variety of high and low motions clips, and used MintMOS to record perceptual quality. MintMOS’s ratings were compared offline to a survey with 49 subjects using 14 video clips reconstructed using their IP-traces. Our results indicate that MintMOS’s projects were very accurate, and outperform metrics like PSNR and VQM in assessing subjective perception.

We implemented MintMOS on standard, off-the-shelf Linux terminals. Testbed experiments demonstrated the lightweight nature of MintMOS which scales well with a large number of flows. We could measure the QoE for 1200 video flows per minute on standard Linux terminals for a reasonably sized QoE space. MintMOS’s modular and lightweight enable it to a viable tool for various networks elements.

Lastly, quality is an abstract concept. Even a numerical reduction of quality only helps in indicating it and never accurately scoring it. However, QoE is the most *significant* measure of human satisfaction, and has long been used to characterize customer experience with vendors in a wide variety of domains. Understanding and improving QoE has had a profound impact on the long term success of many organizations.

## CHAPTER 4

# VIDEO QOE DEGRADATIONS OF INTERNET LINKS

A chain is only as strong as its weakest link

---

— English Proverb

The Internet is comprised of hundreds of Internet service providers (ISPs) and millions of interconnecting links. Packets traverse through the Internet along these interconnecting links which make up the typical Internet path. An Internet link can be one of two types: an *intra*-ISP link that connects two routers within an ISP, and *inter*-ISP links or peering links which connect two border routers of different ISPs. Link level degradations caused by intra-domain routing policies and inter-ISP peering policies are hard to obtain, as ISPs often consider such information proprietary. Understanding link level degradations will enable us in designing future protocols, policies, and architectures to meet rising multimedia demands [59, 60].

This chapter presents a trace driven study to understand QoE capabilities of present day Internet links using 51 diverse ISPs with a major presence in US, Europe and Asia-Pacific. We study their links from 38 vantage points in the Internet using both passive tracing and



active probing for six days. We provide the first measurements of link level degradations and case studies of intra-ISP and inter-ISP peering links from a multimedia standpoint. Our study offers surprising insights into intra domain traffic engineering, peering link loading, (border gateway protocol) BGP and the inefficiencies of using AS-path lengths as a routing metric. Though our results indicate that Internet routing policies are not optimized for delivering high perceptual quality streaming services, we argue that alternative strategies such as overlay networks can help meet QoE demands over the Internet.

## 4.1 Introduction

The Internet is organized as an interconnection of independent autonomous systems (AS's). Each AS is under the purview of an Internet service provider (ISP), and AS's peer with each other to co-operatively forward packets. Routing in the Internet is a process of finding a series of paths traversing one or multiple AS's to reach a destination. Intra-domain routing policies, ISP-peering policies, as well as delay and jitter distributions of Internet links are hard to obtain because ISPs often consider such information proprietary. As a result, the quality of links both inside an AS and peering links used to exchange traffic between AS's are largely unknown. Since the quality of a video stream is as good as the quality offered by the worst link along its path, understanding link level degradations will help characterize the extent to which various factors in the Internet affect video QoE. This will help us gain an insight into designing future protocols, policies and supporting architectures to meet the

rising multimedia demands. Video QoE is known to be affected by three key network events: loss, delay and jitter. While path-inflation and loss characteristics have been studied in the past, no prior work has investigated the jitter levels of the Internet or the combined effect of various factors on perceptual video quality at the link level.

In this chapter, we systematically study both *intra*- and *inter*-ISP links that collaborate to perform present day Internet routing, and their respective video QoE capabilities. Our study involves 51 ISPs with a dominant presence in either the US, Europe, or Asia Pacific. We start by tracing all globally prefixed IP addresses for six days from 38 PlanetLab [82] vantage points in the Internet to extract ISP topologies. IP level paths obtained from the trace are converted to AS level paths, which reveal a rich collection on intra-ISP and inter-ISP peering links. We actively probe these links from vantage points close to these links to measure their response times and relative loading. We present 24-hour case studies of both an intra-ISP link run by Level3 and a peering link between Sprint and Qwest that are representative of a large fraction of the discovered links. Raw network statistics are mapped to QoE capabilities of these Internet links using an objective function. Finally, we study the combined interaction of traversing multiple links by analyzing an un-optimized playout buffer of an end-to-end transmission of 150,000 packets between UCLA and CMU, separated geographically by 2,400 miles.

Our major finding in this chapter can be summarized as follows: *(i)* Internet routing policies are not well suited for streaming services, and much of the pathology has to do with using AS path length as a routing metric and BGP itself, *(ii)* while ISPs are internally well

connected, intra domain paths show greater delay variations in short time scales than inter-ISP peering links. This suggests that ISPs either ignore load balancing or employ strategies that create transient load oscillations. These oscillations are enough to bring down video QoE, *(iii)* though inter-ISP peering links are inflated in terms of delay, we observed lesser load fluctuations in these links promising higher QoE. This implies significant co-operation amongst ISPs in peer link selection, *(iv)* the combined effect of various policies maps current Internet QoE to be just about “acceptable”, *(v)* while the Internet may not be completely multimedia ready, alternative approaches such as overlay networks are highly conducive to streaming services.

## 4.2 Data Collection Methodology

This section describes our data collection and correlation methodology in detail. Our data collection is broadly divided in two phases: (i) Phase-1: extract topology information of the ISPs we choose to study, and (ii) Phase-2: perform active tracing to estimate loss, delay and jitter inflation. In summary, phase-1 produced a dataset of more than 20 million traces from 38 PlanetLab [82] vantage points over 6 days to discover close to 50,293 router IP addresses that belong to the 51 ISPs we chose to study. We inferred an ISP’s topology by classifying a router as either a point-of-presence (POP) or a backbone router. In the second phase, active probing was used to measure network event inflation between end-points which could either be in the same ISP or use multiple ISPs to reach one another. Based on end-point proximity

in terms of ISPs traveled, we isolate the impact of network events for both intra-domain and inter-domain routing.

### 4.2.1 Choosing ISPs

We chose ISPs with great care to make our results as representative of Internet behavior as possible. We chose ISPs who are dominant players in their respective zone. This ensures that a chosen ISP carries a great portion of that zones traffic while ensuring geographical diversity. We look for the following criteria in choosing ISP: diversity, degree (peering links), and size. We also ensured an interesting mix of *tiers*, where tier-1 represents ISPs closest to the Internet core, and tier-3 represents ISPs farthest from it. We chose 19 of the 22 tier-1 ISPs, while keeping a diverse mixture of tier-2 and tier-3 ISPs. Table 4.1 shows the list of 51 ISPs used in this study sorted by their tier.

### 4.2.2 Phase-1: Extracting ISP Topologies

An ISP typically consists of a backbone network and various points of presence (POPs). The POPs peer with gateway backbone routers to connect traffic to and from spoke cities, and the backbone routers typically route data between cities. We discover an ISP's topology by identifying these POPs and backbone routers. As an input to our analysis, we use used

Table 4.1: The first 15 of the 51 ISPs used in our study. Refer to [60] for the remaining ISPs.

| Serial # | Tier | ISP<br>Name     | AS<br>number | Primary<br>Zone | Degree |
|----------|------|-----------------|--------------|-----------------|--------|
| 1        | 1    | ATT             | 7018         | US              | 1490   |
| 2        | 1    | Verizon         | 701          | US              | 2569   |
| 3        | 1    | Qwest           | 209          | US              | 887    |
| 4        | 1    | Level3          | 3356         | US              | 539    |
| 5        | 1    | Savvis          | 3561         | US              | 806    |
| 6        | 1    | Global Crossing | 3549         | US              | 585    |
| 7        | 1    | Genuity         | 7405         | US              | 622    |
| 8        | 1    | Globix          | 4513         | US              | 455    |
| 9        | 1    | Sprint          | 1239         | US              | 1735   |
| 10       | 1    | Verio           | 2914         | US              | 538    |
| 11       | 1    | Williams Comm   | 7911         | US              | 234    |
| 12       | 1    | XO              | 2828         | US              | 184    |
| 13       | 1    | Colt            | 8220         | Europe          | 161    |
| 14       | 1    | DTAG            | 3320         | Europe          | 111    |
| 15       | 1    | Equip           | 3300         | Europe          | 67     |

traceroute data collected from 38 PlanetLab vantage points. The vantage points were spread across the globe, especially in the three dominant zones (US, Europe and Asia-Pacific). From these vantage points, we traced to all of the 127,000 globally prefixed IP addresses extracted from the BGP tables of RouteViews [83], which peers with 60 large ISPs. Tracing typically took six consecutive days on any vantage point.

Traceroute data was processed to produce hop count, DNS names of routers (when available), and router IP addresses. The IP addresses found were then matched with the prefix advertised by our list of ISPs in Routeviews to find routers that belong to one of them, producing 50,293 unique matches. Of these routers, we discovered 19,832 POPs.

We inferred the AS numbers for every IP address based on prefix advertised by ISPs in RouteViews tables. This converted IP-level traceroute data to AS-level paths to reveal intra-ISP and inter-ISP peering links. A given link is intra-ISP if the source-destination pair belongs to the same ISP. Likewise, when a source-destination pair belongs to different ISPs, the link denotes an inter-ISP peering link.

We go one step further in processing these AS level paths to derive *city* level paths. We use the `undns` [85] tool to assign DNS names to cities. POPs were assigned to their respective cities by matching the DNS names with the ISPs naming convention. DNS names usually have an airport code, city, and/or state abbreviation to denote their location. For example, the DNS name `cr1.st6wa.ip.att.net` indicates a router in Seattle, WA (`st6wa`) run by ATT, while `te3-1.ccr01.lax01.atlas.cogentco.com` denotes a router in Los Angeles

(`1ax`) run by Cogent. City level paths give us a benchmark (speed-of-light direct fiber) of the expected network events on an idealized link connecting hosts to quantify degradation [49].

### 4.2.3 Phase-2: Studying Network Links

Once we discover all pairs of intra and inter-ISP links, we perform active probing from various vantage points to measure response times and loading level of the link at various times of the day. We analyze the effects of traversing multiple links using the probe train experiment.

**Active Probing:** Since it is practically impossible to run custom programs on arbitrary routers and hops all over the Internet to perform measurements, we use an alternative way of measuring these links from vantage points. A vantage point which discovers a link of the form  $A \rightarrow B$  from trace collection (usually less than 3 hops away to preserve accuracy [2]) sends out back-to-back TTL-limited probes, three at a time every 50 ms, towards the destination that produced  $A \rightarrow B$ . It sends out probes such that the first set of probes expire as soon as they arrive at  $A$  (TTL), and the second set of probes expire when they arrive at  $B$ . Both routers  $A$  and  $B$  send out an ICMP TTL expired message to the vantage point. The difference in RTTs provides a reasonable estimate to the level of loading experienced between  $A$  and  $B$  [3, 49]. We ensured that the replies came from the intended routers  $A$  and  $B$  to guard against routing changes. Also, the probes were sent in groups of three to break any synchronization with a routers maintenance tasks (usually 60 secs [41, 49]).

In addition to the above probes, we also use King [78] to estimate latencies between routers in the Internet. Since King works by recursive DNS queries, it does not always estimate the latency between every router pair. Hence, we include King estimates only when available. We also convert these raw statistics to an anticipated video-QoE from taking that link. To analyze the time-of-day effects of traversing such links, we perform two representative case studies of an intra-ISP between Tampa and Houston run by Level3 measured from a vantage point in Orlando, and a peering link between Qwest and Sprint measured from a vantage point in Berkeley.

**Probe Train:** We also conduct a probe train experiment, where a source in UCLA sent out a train of 512 byte sized high frequency UDP packets to a destination in CMU, mimicking a high-quality streaming application operating at 30 frames per second or higher. For more than 150,000 packets sent, we measure the amount of “intact” information in an unoptimized playout buffer at every time slot, as we analyze the effects of traversing multiple ISPs to reach a destination. We also extrapolate results based on subjective perceptions of low and high motion MPEG-2 clips.

#### 4.2.4 Caveats and Data Completeness

A trace driven study such as this is vulnerable to errors, failures, and information misinterpretation. We look at possible caveats in our data collection methodology and discuss steps taken to address them.



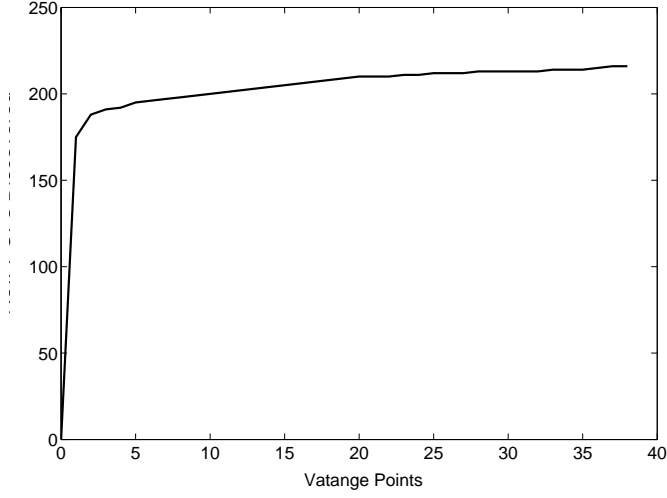


Figure 4.1: New information added by additional vantage points to trace data

**Dataset Consistency:** The dataset should be correct, albeit redundant or in excess. Though we use a small subset of ISPs, we ensured that these ISPs are major traffic carriers in their zone. In fact, 87% of our traceroutes traversed at least one of these ISPs. During phase-1 of our data collection, 3 of the vantage points failed during the six day event, and a power outage at our host node (web-server) led to an incomplete overnight-worth of data collection making us start afresh. However, we found that using multiple vantage points to collect data sufficiently thwarted these failures. To measure the completeness of our data set, we measured the number of *new* POPs discovered by the addition of each vantage point to infer ISP topologies [49]. Fig. 4.1 shows that a single node alone accounts for a majority of POPs learned, with additional vantage points adding marginally more information. With our usage of 38 vantage points, we are well into the knee of the curve. We also ensure that the routers we associate using prefix matching indeed belong to the ISPs that we infer

them to be by comparing AS numbers obtained from `aslookup` [75]. We take a conservative approach by using the common subset of ISP-router matches from these two methods.

**Transient Routing Changes:** Transient instabilities that happen on any one given day in the Internet do not necessarily paint a correct picture of the state-of-affairs. To account for transient routing changes or node failures, both phase-1 and phase-2 of data collection were spread across multiple days. It has been shown that such transient routing changes do not last more than a day [34, 35, 41, 49].

**Spurious Links:** Certain false links can appear due to transient routing changes, and TTL-based traceroute path discovery. We identify spurious links by applying the speed-of-light criterion: links that promise lower delay than possible are spurious.

### 4.3 Estimating Video-QoE of Links

A variety of investigations have been performed to infer and predict the QoE of a video stream. Unlike the R-Score in the VoIP industry, there is no universal consensus on a video QoE estimation model. However, researchers have consistently claimed two things: (i) QoE is a much stronger indication of human perception over QoS models, and (ii) QoE is affected by loss, delay and jitter as far as networking are concerned. To make our video QoE inference as general as possible, we present both uninterpreted statistics of loss, delay and jitter, as well as provide a mapping of QoE. We present three approaches to map trace data to QoE:

Table 4.2: Default parameter values used in the objective QoE function from ITU-T [25]

| Video Metric       | Default Value |
|--------------------|---------------|
| Codec Type         | MPEG-2        |
| Video Format       | VGA           |
| Key Frame Interval | 1 sec         |
| Video Display size | 9.2 inches    |

one using an objective function to return a numerical score, subjective surveys using video clips with artificially induced errors, and by estimating the amount of “intact” information at a receiver playback buffer.

**Objective Function model:** Objective mapping functions often return the inferred QoE as a number, or a mean opinion score (MOS). MOS ratings loosely classify a video stream as good, acceptable, poor, or anything in between. We use one such model recommended by the ITU-T [25]. This model takes into account the network delay of audio and video packets, the audio-video sync during playout, the encoding bitrate and framerate, as well as the type of receiver to predict video quality (in a scale of 1–5). Raw network measurements are used as an input to the model, which returns the anticipated quality degradation by using that link. The defaults assumed in estimating QoE using this model are shown in Table 4.2. In addition to the above, we also assume audio to be of the highest quality (not impaired).

**Playout Buffer Analysis:** We also look at the contents of a receivers “playback” buffer to estimate video QoE. Ideally, if every host had a direct fiber optic connection to every other

host in the network, streaming packets would arrive at a destination in the exact order and time interspacing that they were sent at. In such a case, the destination would consistently have a buffer's worth of data to ensure a smooth playout. Hence, the amount of information *absent* at a receiver buffer is a strong indication of what the network did to the video stream. Jitter causes packets to arrive badly out of sequence long enough to be considered missing, while loss directly contributes to missing information. We measure the amount of missing information in 100 ms of buffering at the receiver. We infer perceptual quality based on information available for playout.

## 4.4 Internet Video Streaming

We begin with a brief review of Internet routing requirements for streaming services, and present a high-level overview of the Internet paths studied in this chapter. We base this in the context of multimedia streaming over the Internet, and briefly discuss video buffering and its effects on preserving interactivity.

### 4.4.1 The typical Internet route

Internet routing is a process of finding a series of AS's to traverse from a source node to a destination node. The As's themselves are organized in a hierarchy of tier-1, tier-2 and tier-3

ISPs, and are typically characterized as carrier ISPs (tier-1) or stub ISP's (tier-2 and tier-3). Carrier ISPs peer with stub ISP's to route traffic between them, and this relationship is dictated by *provider-customer* contractual agreements between the carriers and stubs. Most source to destination pairs in the Internet traverse a combination of an ascending tier path (lower to higher tiers) and a corresponding descending tier path, in that stub ISP's are typically reached via an intermediate tier-1 ISP. A few exceptions to this rule include the presence of exchange points and specific peering relationships between stub ISP's.

We begin with a high level characterization of an Internet path in traversing from source to destination in terms of number of hops and the distribution of intra- and inter-ISP links we observed from our traceroute (Table. 4.3). The table shows that a majority of links used to traverse the Internet are *intra*-ISP links which account for 72% of all the links we studied. This implies that degradations caused within an ISP can have a more profound impact on video quality than due to peering links. An average Internet route today involves traversing about three AS's, and consists of a majority of intra-ISP paths and a few peering links, one each when an AS boundary is crossed.

Table 4.3: Characterization of Internet routes studied

| Path Metric               | Measured Value |
|---------------------------|----------------|
| Mean hop count            | 7.3            |
| Intra-ISP links           | 72%            |
| Inter-ISP (peering) links | 28%            |

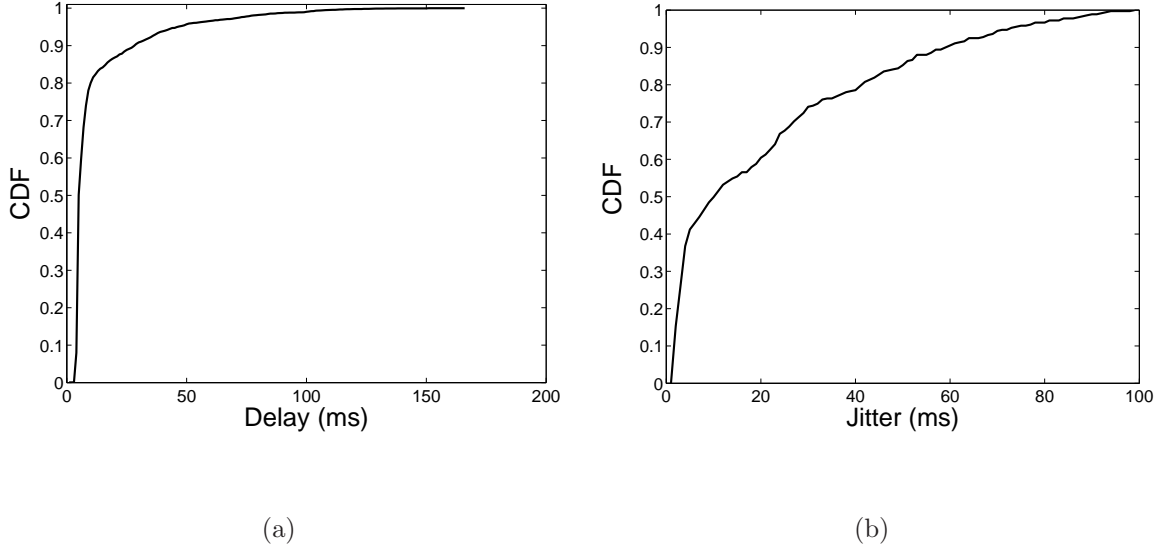


Figure 4.2: Characterizing *intra*-ISP links: (a) Delay distributions, and (b) Jitter distributions for 800 intra-ISP links from six days of active probing

## 4.5 Intra ISP Routing Policies

We begin by analyzing intra-ISP links, and their relative impact on video QoE. Intra domain links interconnect various routers that belong to an ISP, which are typically points of presence (POPs) in the same or different city.

We perform active probing on a set of 800 intra-ISP links from our 38 vantage points. Probing was continuously performed for an hour on each session, with at least 12 sessions per day for 6 days. Probing at various times of the day smoothens rate fluctuations that happen throughout the day, and probing for multiple days ensures we smoothen transient routing changes.

Our results offer surprising insights into intra-domain routing policies and their impact on video QoE. While we find that ISPs are internally well connected, there seems to be significant load *imbalance* in network links over short time scales. This suggests that ISPs either do not perform extensive load balancing amongst their internal links, or the applied strategies are topology insensitive resulting in load oscillations. Also, it is well known that RIP and OSPF, the most popular intra-ISP routing protocols, are load insensitive. Load imbalance creates excessive jitter, which is known to have the greatest impact on video QoE.

#### 4.5.1 Delay and Jitter Distributions

Active tracing reveals delay and jitter distributions for the set of intra-ISP links we study. Delay distributions are shown in Fig. 4.2(a), which shows that a majority of path lengths are fairly good ( $< 50$  ms), with a great percentage (70% of all links) less than 25ms. This means that ISPs are internally well connected. Though a majority of paths connect city pairs in close proximity, we found certain intra-ISP links connecting city pairs across countries and even continents. This was particularly true for tier-1 ISPs. We did not see any relative difference in the mean delay values based on geographic dominance of an ISP, which seems to indicate that ISP interconnection topologies are consistent (and possibly replicated) across continents.

The jitter distributions (Fig. 4.2(b)), however, show a surprisingly high value for some points. Jitter levels of more than 20ms are known to have a very adverse effect on perceptual

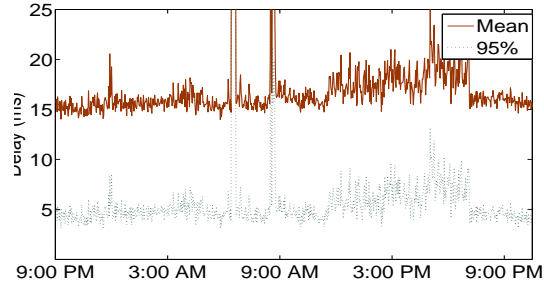
quality [8]. The plot shows that approximately half the links we study have jitter levels more than 20ms. Despite the fact that intra-domain routing is under the complete purview of an ISP, high jitter indicates that ISPs either do not perform any kind of load balancing, or perform load balancing which results in oscillations. In fact, the jitter distribution of intra-ISP links was observed to be higher than inter-ISP peering links. We also observed higher jitter values for many tier-1 ISPs links, which could be due to the fact tier-1 ISPs have a large number of nodes making traffic engineering rather complex.

#### **4.5.2 Case Study: Level3’s link between Tampa and Houston**

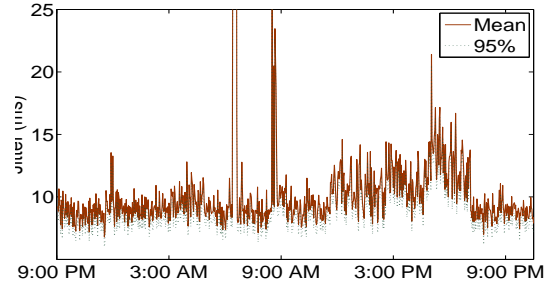
We wondered why intra-domain routing would produce such high levels of jitter, and sought out to investigate the true causes of it. We chose a representative link to perform sustained active probing for 24 hours determine if there was a change in the loading level of that link. A representative link would have mean delay and jitter levels below the knee of the curve of Fig. 4.2(a) and Fig. 4.2(b), which maps about 70% of all links discovered. We chose an intra-ISP link run by Level3 between Tampa (FL) and Houston (TX) from a vantage point in Orlando (FL). We probed this link continuously for 24 hours on July 23 & 24, 2009 and took measurements in bins of 2 minutes each. Each bin represents the result of 240 probes, with a 24 hour study accounting for 172,800 probes.

Delay variations in the 24 hour period are shown in Fig. 4.3(a). The plot shows that delay values vary depending upon the time of the day, where it peaks between 3:00 PM and

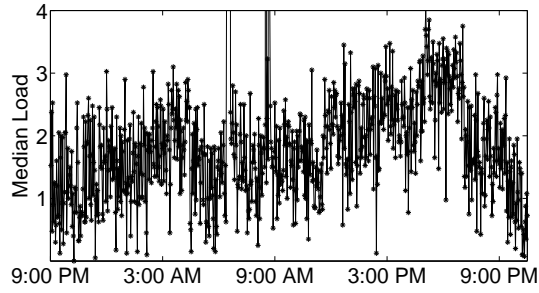




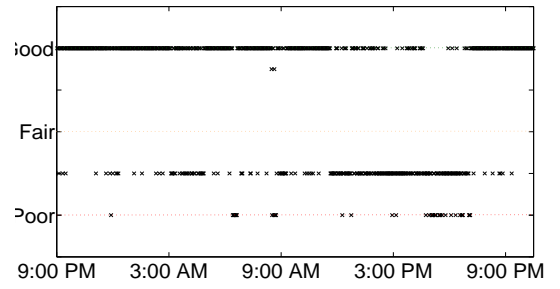
(a)



(b)



(c)



(d)

Figure 4.3: A 24-hour observation of an intra-ISP link by Level3 connecting Tampa and Houston: (a) Delay distribution, (b) Jitter distributions, (c) Mean loading of the peering link, (d) Estimated QoE at various times of the day

9:00PM for that day. We also discovered two significant outages, both before 9:00 AM on July 24, where a good fraction of the probes were lost, while others made it in more than 100 ms. Delay values show fluctuations indicating jitter.

Jitter deviations are shown in Fig. 4.3(b). Jitter patterns correlate with the delay patterns, in that they peak between 3:00 PM and 9:00 PM, and are reasonably high during the outage times. Jitter values at 95% confidence interval trail the mean values closely. The plot shows consistent jitter at around the 10ms range, which is enough to degrade video streaming.

To indeed verify that jitter at this link is because of a fluctuation in traffic directed to this link, we measured the median loading of the router at Tampa which connects to Houston. To measure loading, we recorded the router times-tamp on the ICMP packets. Variations in the time-difference at the destination router with no variations at the source are indications of loading at the source link [49]. Fig. 4.3(c) shows the median loading at the router in 2 minute bins. Indeed, there are significant fluctuations in the link load in very short time scales<sup>1</sup>. Also, loading continues to peak at around 6:00 PM, and was considerably high during the outage times.

MOS variations for this study are shown in Fig. 4.3(d), which are obtained from the ITU-T model [25]. The results suggest that MOS projections are overall in the “good” range, with jitter spikes ensuing in consistent MOS ratings of less than “acceptable”. MOS dropped to “poor” for an extended period of time at around 6:00 PM. Consistent jitter spikes, such as in

---

<sup>1</sup>with an OC3 link, median queuing delay of 1ms corresponds to 40 packets in the queue [41]

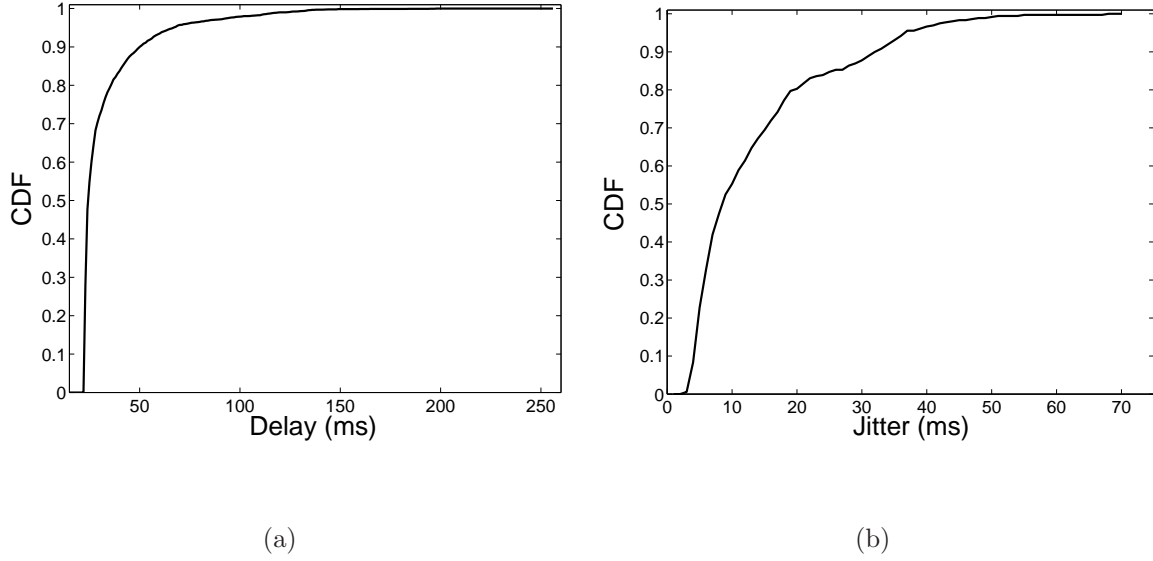


Figure 4.4: Characterizing *inter*-ISP Peering links: (a) Delay distributions, and (b) Jitter distributions for 1100 inter-ISP links from six days of active probing

this case, have subtle long term effects on QoE. Studies have also shown that subjects have a “forgiveness” effect, where users rate a video clip relatively high if the playout is smooth after a brief initial loss in quality. However, with a regular (almost periodic) degradation of quality, users would generally rate the video sequence lower on a longer time scale than the one we chose. The average MOS was around 2.46, which suggests quality between just above “acceptable”.

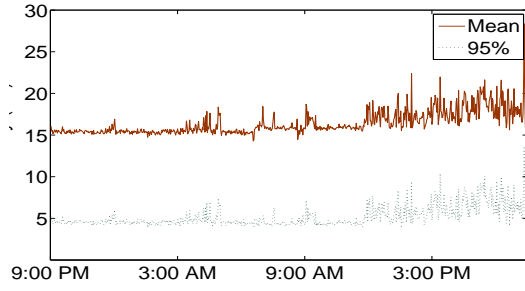
## 4.6 Inter-ISP Routing Policies

We next study the peering links used by ISPs to exchange traffic between one another. ISPs often use multiple peering links between one another, and the policies used by ISPs to choose peering links to exchange traffic is largely proprietary and driven by a multitude of factors. We measured 1100 such links that connect routers of different ISPs using active probing to measure the relative loading of each such link.

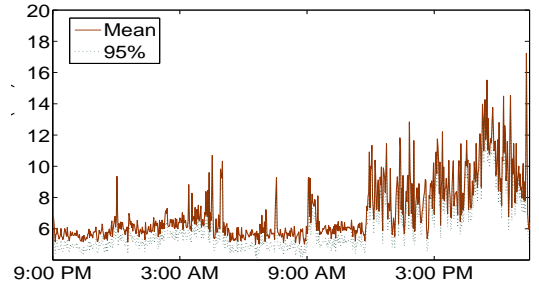
Our findings indicate that there is significant co-operation amongst ISPs in distributing load across links, and that peering links are well balanced and even. This was particularly the case for ISPs that used a large number of peering links between one another. Peering links were overall good for tier-1 ISPs compared to cases when a tier-3 ISP was involved. Peering links did not show fluctuations based on geographic presence. However, peering links connecting multiple continents were relatively sub-optimal.

### 4.6.1 Delay and Jitter Distributions

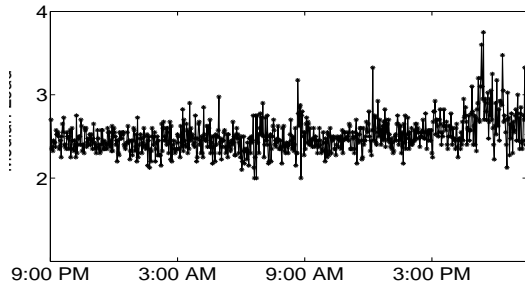
Delay distributions for peering links is shown in Fig. 4.4(a). The plot shows that at least half the links studied have delay distributions less than 25ms, and about 30% are in between 25 ms and 75 ms. Delay distributions are typically higher than intra ISP links, suggesting that peering links are relatively lesser than intra-ISP links, and that peering links often connect



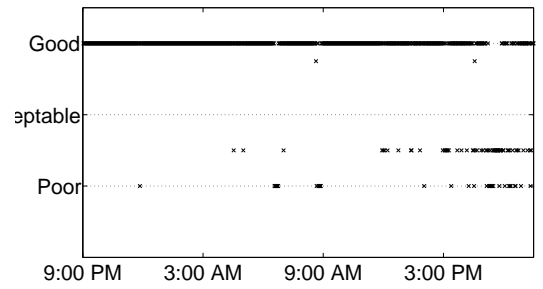
(a)



(b)



(c)



(d)

Figure 4.5: A 24-hour observation of a peering link between Sprint and Qwest in California:  
(a) Delay distribution, (b) Jitter distributions, (c) Mean loading of the peering link, (d)  
Estimated QoE at various times of the day.

distant routers. Certain peering links connect distant cities and even continents, which also account for higher delay distributions.

Jitter variations are shown in Fig. 4.4(b), which shows that more than roughly half of the links studied are in the “good” range ( $< 10\text{ms}$ ), and a greater majority of the remaining in the “acceptable” range. This means that delay variations of inter-ISP links are lesser than delay variations of intra-ISP links, suggesting that there is significant load balancing in peering link selection. We found jitter values to be lesser for peering links of tier-1 ISPs than between links of tier-3 ISPs.

#### **4.6.2 Case Study: Peering link between Sprint and Qwest**

To provide a basis to compare a typical intra-ISP link to a peering link, we chose a peering link that is representative of the delay and jitter distributions of Fig. 4.4(a) and Fig. 4.4(b). We chose a peering link between Sprint and Qwest in California, measured from a vantage point in Berkeley. We actively probed this link for 24 hours on July 23 & July 24, 2009 for a total of 1,72,800 probes.

Delay distributions over various times of the day are shown in Fig. 4.5(a). Delay averaged around 15 ms for most of the study, and peak around late afternoon to early evening. However, the relative difference in delay values at short time scales is not too high. Jitter

distributions for 24 hours is shown in Fig. 4.5(b). Jitter fluctuations are low at night and most times of the day, reaching a peak at around 6:00 PM.

We also measured the load on the links origin router at Sprint (Fig. 4.5(d)). We compared ICMP router timestamps to look for fluctuations at the Qwest router with little fluctuations at the Sprint router. However, we found that the link was evenly loaded for most of the day, with some fluctuations at around 6:00 PM. An even loading such as in this case strongly indicates that traffic is evenly distributed across peering links used between Sprint and Qwest, a trend that we predominantly saw for most of the peering links we studied.

The combined effect of these on video-QoE is shown in Fig. 4.5(d). A vast majority of the projections are in the “good” range. Of the points not in the good range, almost all of these are close to “acceptable” with no incidents of “poor” quality. The overall MOS for this study was 2.8, much closer to “good” than the 2.46 of the Level3 link.

## 4.7 Playout Buffer Analysis

We next put together the combined end-to-end effect of choosing a combination of links to traverse the Internet. We diverge from both active probing and an objective QoE model by looking at the receiver playout buffer of an end-to-end transmission that mimics a real-time, high quality multimedia stream. We round off this study by placing the results in the context of an MPEG-2 stream, and discuss the impact of degradation for low and high motion clips.

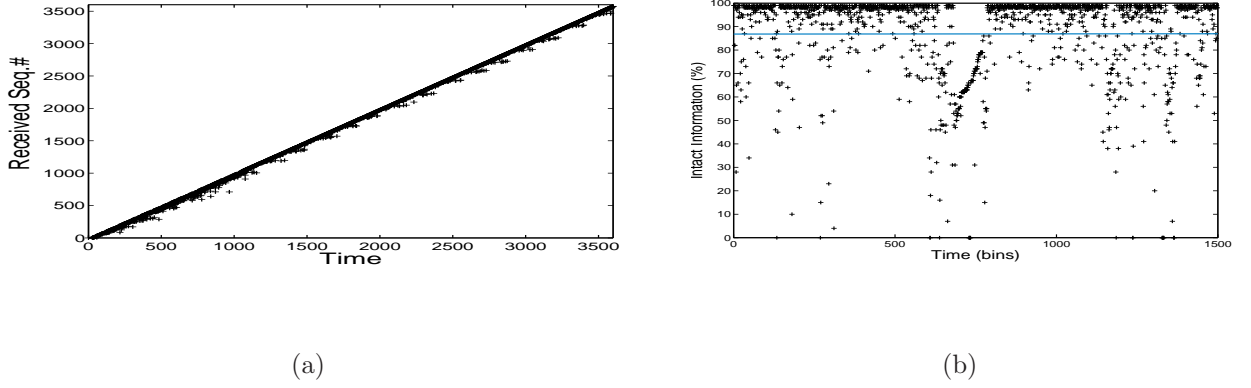


Figure 4.6: (a) Received sequence number at destination for the first 3600 packets. The X axis denotes the time since the first arrival of a packet, while the Y axis indicates the anticipated Sequence Number. (b) Contents of the unoptimized playout buffer in various time slots. Horizontal bar represents the mean.

We sent 150,000 packets from a source in California (UCLA) to a destination 2,400 miles north-east in Pennsylvania (CMU). The packets were 1024 bytes each, and were sent out at the rate of 1 packet every 10 milliseconds (or 100 packets per second). High data rate applications typically transmit more than 30 frames per second, which amounts to 120 packets per second assuming each frame takes 4 packets to encode.

The destination receives packets and places them in a “playout” buffer, which stalls for a certain time waiting for a batch of packets to arrive. A playout buffer of 100 ms is recommended for our sending data-rate [8]. The buffer is un-optimized and non-adaptive, and does not vary buffer limit based on the current reception count. Packets that do not arrive in their expected time slot are *discarded*, but packets that arrive out of order in a given time slot are received correctly. Certain enhancements to buffering strategies often



adaptively increase the buffering limit in the face of little or no packet reception, stalling playback at such times [16]. However, since our goal is to understand the raw capabilities of the Internet without optimizations, we under-estimate buffering by choosing this playout buffer.

### 4.7.1 Packet reception

We plot the reception time for each packet for the first 3600 packets received at the destination. Fig. 4.6(a) shows the time that each packet was received, with time starting from the first packet received at destination. If the source destination pair had a hypothetical direct optical fiber link between them, the data points would completely overlap the  $y = x$  line. Though we observed little loss ( $< 0.1\%$ ), we see that packets frequently arrive beyond their expected time.

### 4.7.2 Playout Buffer Contents

We now study the contents of the playout buffer at various time slots. We measure the amount of “intact” information available in the playout buffer. We measure the intactness by calculating the percentage of correctly received sequence numbers. The playout buffer

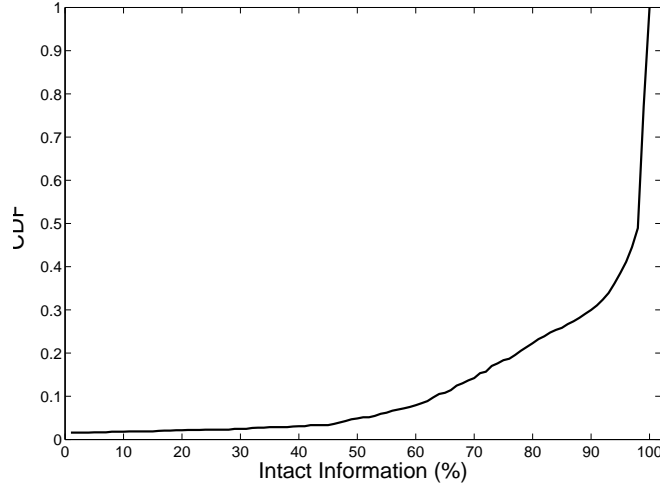


Figure 4.7: CDF of the amount of ‘intact’ information in the playout buffer.

infers the expected set of sequence numbers based on the slot number since the time of the first packets reception.

Fig.4.6(b) shows the intact information percentage at every time slot. Each time slot is of 1 sec duration, which amounts to 1500 time slots for all packets. The amount of content in the buffer shows great variation in time. Research has shown that even a marginal loss in information manifests as user dis-satisfaction, and more than 20% loss certainly degrades video [28]. The plot also shows that the buffer contains less than 40% of required information at regular intervals of time, even dipping to less than 10% every 4 minutes in the playout. This means that video quality will frequently dip to low ratings almost periodically. Consistent loss in video quality manifests as a strong dissatisfaction. The mean amount of information in the buffer was calculated to be 86.1%. Fig. 4.7 shows the CDF of the information available in the playout buffer. The plot shows that the buffer has less than

90% of intact information more than 30% of the time, and it has 100% intact information less than 20% of the time. The combined effects of multiple links that induce jitter are hence not merely additive.

### **4.7.3 An MPEG-2 playout perspective**

Streaming content in IP networks is commonly transported as a data stream encoded using the MPEG standard and transported via the real time protocol (RTP) over a UDP/IP stack. MPEG encodes video streams as a series of Intra (I), Predictive (P) and Bidirectional (B) frames. I-frames carry a complete video picture, and as such provide reference to the following B- and P-frames for decoding an MPEG stream. P-frames predict the frames to be coded using a preceding I or P-frame. Lastly, B-frames use the previous or next I-frame for motion compensation. Each frame is typically fragmented into multiple IP packets for transport over the Internet.

The frames are packed into a group of pictures (GOP), where each GOP consists of an I-frame at the start and a series of B and P-frames which use it as a reference. Depending upon the motion complexity inherent in a clip, the structure of a GOP can be very different: low motion clips (like a news program) have larger I-frames and a handful of P and B-frames to complete a GOP, while high motion (sports clip) clips have smaller I-frames and relatively larger P-frames for motion compensation.

**Low Motion Clips:** Loss of P- or B-frames seem to make little difference in perceptual quality. The difference in perceptual quality for the loss of an I-frame over a P-frame, however, is drastic (for e.g., compare sample-17 with sample-25 or sample-33 with sample-41). Low motion clips have larger I-frames, which increase their odds of getting impacted during an outage. Low motion clips have longer GOP structures with more P-frames, which enables more compression. Because of their long GOP structures, the *duration* of on-screen degradation tends to be longer.

**High Motion Clips:** I-frame loss is dominant, just like for low motion clips. However, because of the inherent dynamism, P-frames tend to be larger and more informative. Hence, loss of a P-frame draws a slightly more adverse reaction from the subjects than with low motion clips (sample-17 v/s sample-21). Increased loss of P-frames continues to degrade quality. Because scenes change frequently, I-frames tend to be shorter and less probable candidates for loss during outages. Increased loss rates within an I-frame do not seem to degrade quality any further [19, 20].

## 4.8 Discussions

We piece together the above studies to infer insights about the way the Internet operates, and suggest several ways to overcome these shortcomings. We look at a variety of causes that both contribute to these findings, and are affected by them. We take a closer look at intra and inter-domain policies of BGP which governs route selection, and suggest workarounds.

We analyze other solutions, such as increasing capacity, and multi homing as alternatives to providing better streaming quality. Lastly, we take a close look at the prospect of using overlay networks for deploying multimedia networks with one example deployment.

### 4.8.1 BGP and Path Selection

BGP is the de-facto routing protocol for inter-AS and intra-AS routing. It manages reachability information shared between two AS's, and allows for diverse networks to interconnect and become the Internet. BGP summarizes, and often hides, internal topological details about an AS to prevent routing oscillations in a process called 'route dampening'.

**Intra-domain (iBGP):** Intra domain routing is largely governed by iBGP sessions, which are an overlay of nodes over the OSPF substrate. While ISPs are known to perform certain optimizations intended towards load distribution, routing instabilities within an AS often lead to short term anomalies. Short term traffic fluctuations are typically not captured or accounted for, and are clearly not a metric for intra-domain routing. BGP convergence times are of the order of minutes, and it has been shown that topology sensitive load balancing is difficult with BGP. Prevalence of mechanisms such as early or late exit, and latency based route selection within an AS largely contribute to load imbalance. Our results call for even traffic distribution within an AS, which could help improve streaming quality. ISPs are known to be internally well connected, and discovering and utilizing redundant paths to distribute load could very well improve the quality of intra-ISP links [55].

**‘AS-path length’ based routing:** BGP determines the choice of AS’s used to traverse from source to destination. Route selection is typically performed by comparing AS-path lengths advertised by address prefixes. Research has shown that about 11% of all new paths learned by BGP are unreachable [34], and paths are often inflated in terms of distance travelled [49]. Fig. 4.8 captures the effect of using AS-path lengths as a routing metric for our probe train experiment between UCLA and CMU. Internet route selection took the packets from California (far west) to Louisiana (east), back to Houston (west), then Atlanta (north-east), and finally Washington (north) before making it to CMU. Clearly, the choice of ISPs traversed badly mismatches a direct series of links between these two geographic locations. Since our findings indicate that the quality or choice of peering links is not a factor for ISPs, we argue that inefficiencies in route selection arise from using AS-path length as a routing metric.

Given that a great majority of Internet traffic will carry multimedia content, we argue for BGP advertising “geographical” information at BGP speaking routers. That way, even an inexact reachability information at a router can be compensated by allowing video streams to “flow” towards the destinations geographical zone. Though far fetched into the future, we believe geographic co-ordinates embedded in BGP route advertisements can certainly enhance streaming applications.

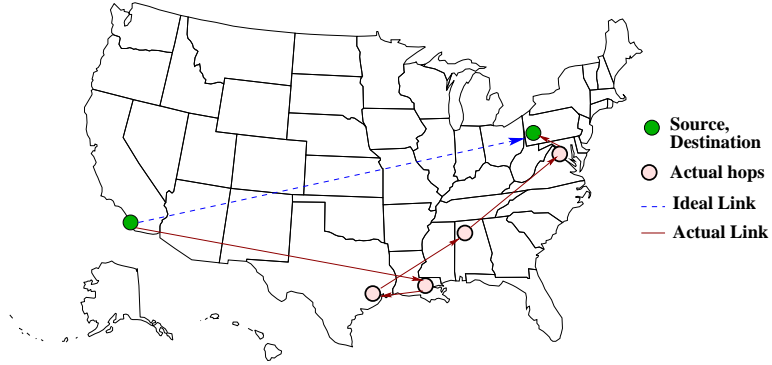


Figure 4.8: Ideal path versus actual Internet path for the probe train experiment

#### 4.8.2 Increasing Capacity and expected QoE as a metric

**Increasing Capacity:** A popular suggestion to improve performance is to increase capacity. However, increasing capacity with high jitter levels in the network only *exacerbates* the problem. This is because an increase in capacity provides higher peak levels for the traffic to reach. When links are unbalanced, jitter typically increases with network capacity. In fact, load balancing is known to have often increased the capacity of existing infrastructure.

**Expected QoE as a Routing Metric:** Intuitively, if inter-AS path selection was based on *expected QoE* rather than AS-path length alone, streaming content would deliver higher perceptual quality. However, this is not entirely practical for the following reasons: (i) it is not possible for every router to probe each of its links continuously to create a picture of expected QoE, (ii) fluctuations in expected QoE may cause route flapping, and (iii) changing the behavior of the entire Internet is known to be extremely slow. For example, QoS mechanisms

such as IntServ and DiffServ were envisioned to be implemented at all network elements from source to destination. The actual adoption, however, was sporadic and un-coordinated.

## 4.9 A Case for Overlay Networks

While change in the Internet may be slow, we look at alternative approaches that provide imminent solutions. We look at the case of network overlays tuned for multimedia. Information ‘detouring’ [45] and overlay networks [1, 52] have shown to be effective workarounds around Internet failures. Overlay networks are a collection of nodes in different autonomous systems that maintain a *virtual* link to one another. A virtual link is the normal IP-level path between these two nodes, which the nodes probe continuously to monitor signs of degradation. Overlay nodes exchange this reachability information with each other periodically to expose other *redundant* paths in the Internet across different AS’s that BGP cannot advertise. If the normal path between these two nodes experiences outages, nodes can switch to alternate paths that reach the destination. It has been shown that at most one such ‘reroute’ is enough to route around outages [1]. We bring out benefits of using an overlay based approach to route around IP failures using measurements from a measurement overlay we deployed.



### 4.9.1 Methodology

We analyze weeklong measurements of a large number of Internet paths all over the world to understand the benefits of quick redirections to preserve video-QoE and avoid transient fluctuations within an ISP. We created an overlay of 32 nodes deployed in geographically diverse locations, including the U.S., Europe and Asia-Pacific.

**Experimental Setup and Video Clips:** Between January 22 to 29, 2010, every node streamed 1024 byte UDP packets every 5 minutes to a randomly selected destination. The stream mimics the IP-packet trace of a randomly selected high or low motion clip from a set of five clips used in the previous round of study. We passed the name of the clip and the type of frame the packet carries in the packet payload, creating an IP-trace of the clip at the receiver. For any of our overlay with  $N$  nodes, the source indirectly probed the destination via the  $N - 2$  other intermediaries while streaming packets to the destination. This probing is performed only when transmitting key frames within the clips. For low motion clips, we mark the I-frame alone as the key frame. For high motion clips, we mark both the I- and P-frames as key frames. We record the receiver trace at the destination, and the probe responses from the intermediaries at source to analyze offline the suitability of alternate paths during outages.

### 4.9.2 Benefits of an Overlay Network

A potential workaround to transient degradations that occur within an ISP is to switch to a route that avoids the ISP in question, usually involving a one-hop detour through a node that avoids this ISP [45]. To determine the benefits of switching, and in particular switching early, we measure the probability of restoring key frames following a degradation in Fig. 4.9. This plot shows the probability of the next key frame being received successfully *after* observing a certain number of consecutive drops in a key frame. We plot this probability for upto 6 consecutive packet losses observed for the default IP-path, a random-1 path, and the “ideal” detour path. The random-1 path is derived by attempting to detour with *any random* intermediary from our set of 32 nodes. The “ideal” detour path is the most optimum path to take following a degradation, usually obtained by extensive background monitoring of alternative paths. In effect, we base the default-IP path against a measurement free (random-1) and measurement based informed detours.

After only two successive packet drops, the probability of the default IP-path restoring the next key frame seem to diminish to around 0.42. The “ideal” detour maintains a higher recovery probability of more than 80% for upto 5 consecutive drops. The plot also shows that even random-1 is able to provide higher returns than the default IP-path when 3 or more packets are lost in succession. This leads us to believe that an early detection and subsequent detour can help recover from degradations.

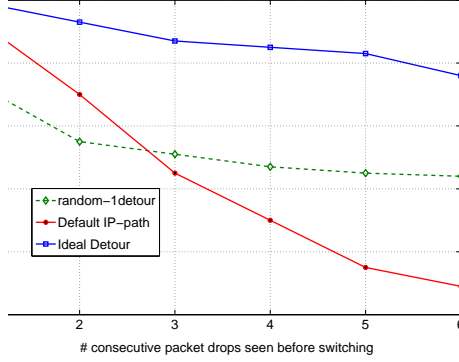


Figure 4.9: Recovering from perceptual degradations: a case for overlays that avoid faulty AS's.

In general, overlay networks can solve the following problems in terms of raising perceptual quality: (i) while Internet routers cannot individually monitor expected-QoE along links, a small set of overlay nodes can easily achieve this with low overhead probing, and re-route packets based on expected QoE; (ii) discover and expose redundant paths that cannot be exposed by BGP, providing greater routing options especially around ISPs which experience transient degradations, (iii) implement policy routing, and allow customized media dissemination, and (iv) converge faster than BGP, and recover from outages quickly. In other words, we believe overlay networks can provide a flexible way of implementing policy based networking changes that cannot be immediately brought about in the Internet.

## 4.10 Conclusions

Network service providers need to raise video QoE to attract customers and prevent churn. QoE has been found to be the most significant measure of human satisfaction, and has long been used to characterize customer experience with vendors in a wide variety of domains. Understanding and improving QoE has had a profound impact on the long term success of many organizations.

This chapter presented the first empirical observation of Internet links from a QoE perspective. We studied intra- and inter-ISP links of 51 major ISPs spread across the US, Europe and Asia-Pacific from 38 vantage points in the Internet. We closely studied these links for six days to infer their suitability for streaming services.

Our findings offer surprising insights at link level fluctuations in the Internet: (i) we find that intra-domain links are poorly engineered, showing significant loading fluctuations which can degrade video quality. This could largely be attributed to BGP, which makes topology sensitive load balancing hard; (ii) contrary to popular belief, ISP peering links are well engineered and quiet suitable for carrying streaming content. This implies that ISPs are not limited by their choice or quality of peering links, rather, AS-path lengths are insufficient as a routing metric; (iii) the overall effect of Internet route selection maps the current QoE to just about “acceptable”; and, (iv) overlay networks could overcome a majority of Internet’s shortcomings in delivering multimedia; both because they can be deployed with relative

ease compared to re-engineering the Internet, and provide alternative routes that avoid AS's which experience transient fluctuations in delivering packets.

## CHAPTER 5

### EFFECTS OF INTERNET PATH SELECTION ON VIDEO-QOE

“Alice if you don’t know which road to take then any road will get you there but just  
don’t choose the road that isn’t any road, you could end up in a terrible place and  
lose your muchness. That’s how you lost your muchness last time you chose the road,  
but I know this time you will choose the right road

---

— Lewis Carroll (Alice in Wonderland)

#### 5.1 Introduction

Having characterized Internet video-QoE at the link level, we now turn our attention towards studying the goodness of end-to-end Internet paths in assuring video-QoE. This chapter presents large scale Internet measurements to understand and improve the effects of Internet path selection on perceived video quality [58]. We seek answers to the following questions: (i) how do Internet outages effect video-QoE?, (ii) where in the path do these outages frequent?, (iii) how can redundant Internet paths be utilized to improve QoE?, and (iv) is there a simple, scalable approach that can utilize these redundant paths? An empirical understanding of

QoE degradations along an Internet path, and simple alternative path selection strategies that go beyond default IP-routing, would help us overcome some of these limitations.

To answer these questions, we begin by probing popular VoD/IPTV servers and PPLive hosts on the Internet for 7 days from 62 geographically diverse PlanetLab nodes in a manner that mimics “fetching” content. These probings reveal the nature of outages and where they occur along the path. High level results indicate that 89% of outages on paths to servers and 62% of outages to broad-band hosts can potentially be recovered from if the source used alternative routes. Further, we use the IP-level trace obtained from this experiment to reconstruct 54 video clips representative of a large fraction of the outages. These clips were used to conduct extensive surveys with 77 human subjects. Our surveys indicate that perceptual quality degradations depend of motion complexity of the clip, the type of frame impacted, and packet loss rates. We argue that perceptual quality can be increased by a combination of: (i) *application specific policies* (e.g., preserve key frames, provide bounds on response time etc.), and (ii) *path selection strategies* which can preserve application specific policies.

We also investigate ways to recover from QoE degradation by choosing alternate Internet paths that preserve application specific policies. We seek simple, scalable path selection strategies *without* the need for background path monitoring or apriori path knowledge of any kind. To do this, we deployed five measurement overlays: one each in the US, Europe, Asia-Pacific, and two spread across the globe. We used these to stream IP-traces of a variety of clips between source-destination pairs while probing alternate paths for an entire week. Our

results indicate that a source can recover from upto 90% of the degradations by attempting to restore QoE with any five *randomly* chosen nodes in an overlay. We argue that our results are robust across datasets.

## 5.2 Probing Internet Destinations

Streaming content on the Internet today is most commonly disseminated by VoD/IPTV service providers or by peer-to-peer (p2p) streaming (e.g., Joost, BBC iPlayer, PPLive etc.). Hence, we begin by measuring the round trip path to these destinations from geographically diverse client locations. We analyze outages on these paths, their recurring frequency, as well as their location along the path. We provide upper bounds on the fraction of outages that occur on the last hop, which cannot be recovered by using alternate paths. Overall, results presented in this section are crucial to understanding paths used to disseminate streaming content from popular sites/hosts all over the Internet.

### 5.2.1 Vantage Points and Destination Sets

**Vantage Points:** IP-based streaming services are currently popular in Germany, France, Belgium, United States, Korea, and China among other nations. Hence, we select vantage points that have a presence in these countries and are generally placed in United States,



Europe and Asia. We initially began with a list of 70 vantage points<sup>1</sup>. However, we removed data from 8 vantage points which had more than 24 hours worth of data loss due to downtimes, effectively reducing our vantage points to 62 nodes.

**Destination Sets:** To create our destination set, we gathered a list of the 200 most popular IPTV/VoD service providers from various Internet sources. To create a destination set for P2P video sharing hosts, we used 1,200 IP addresses of broadband hosts obtained from crawls of TVUNetworks and PPLive. In the end, our source-destination pairs are representative of typical round trip paths on the Internet used to disseminate streaming content.

### 5.2.2 Probing Methodology

Between January 08 and 14, 2010, we systematically studied paths between our vantage points and destination sets. We probed the destinations from our vantage points mimicking a “fetch” operation of streaming content using UDP probes of 1024 bytes. To do this, we timed our probes according to the IP-level trace of a variety of low and high motion clips. We use three representative low motion clips (*Foreman*, *Akiyo*, *Coastguard*) and two high motion clips (*Football*, *Tennis*) to obtain IP-level traces. The IP-level trace of these clips were recorded using an Ineoquest Singulus digital media analyzer [77] with a fragmentation limit of 1024 bytes. We use a 15:2 GOP at 30 frames per second to encode a given clip.

---

<sup>1</sup>All our vantage points and destination sets can be found at [80]

Every 5 minutes, each vantage point selects the IP-trace of a randomly chosen clip to probe a randomly chosen destination from its destination set. The destination’s response to these probes enables us to create an IP-level trace of the chosen clip at the receiver, which we use to infer path quality. We partitioned the destination sets across our vantage points ensuring an even mix of servers and P2P hosts.

**Failures v/s outages:** While even a single packet loss can potentially induce perceptual degradation, we strive to distinguish between short lived congestion drops and a true path outage in this round of study. We declare a path to experience a *failure event* if three or more consecutive probe packets fail to receive a response. As soon as this happens, we issue a traceroute from the vantage point to that destination. If the first traceroute after a failure event *also* fails, we declare a destination *outage*. Upon detecting an outage, we send a continuous stream of probes to the destination until the path return to normal. The path is deemed normal with the first incident of successfully receiving 10 probe responses. In the end, any definition of an outage based on probe loss patterns is arbitrary.

**Traceroutes:** When a path experiences a *failure event*, we used TCP traceroutes to determine the possible *location* of the failure. TCP traceroute return results faster than the standard ICMP based traceroute to determine failure location within milliseconds of its happening [22]. We broadly classify failure locations as source side, destination side, last hop, or middle core (backbone) [22, 51].

Table 5.1: Overview of outage locations for paths to servers and P2PTV hosts observed from 62 vantage points in a one week period.

| Event                         | Servers   | P2P hosts  |
|-------------------------------|-----------|------------|
| paths probed                  | 18,600    | 62,000     |
| Failure Events                | 4,181     | 16,724     |
| Path failures                 | 1829      | 6743       |
| Classifiable path<br>failures | 915       | 3439       |
| Last hop failures             | 101 (11%) | 1308 (38%) |
| Non last hop failures         | 814 (89%) | 2131 (62%) |
| Unclassifiable                | 914       | 3304       |

### 5.2.3 Outage Locations

We begin with characterizing failure locations summarized in Table 5.1. A ‘failure event’ is the loss of three consecutive probe packets. A ‘path failure’ is the additional failure of the first traceroute issued. Likewise, a ‘classifiable’ failure is when we can potentially isolate the location of failure from traceroute. We group the classifiable failures as either a ‘last hop’ failure or failures occurring elsewhere. Last hop failures are failures that happen on the last hop to the destination, and are very hard to recover from using alternate Internet paths. Lastly, we group outages as ‘unclassified’ if we cannot infer the location of failure from traceroute.

We observe that on paths to servers, only 11% of the failures happen at the last hop. This both indicates that servers are well provisioned and server side path outages are less frequent. This also implies that it is possible to potentially recover from 89% of outages on a path to a server. The last hop failure rate for broadband hosts is quite high (38%). This implies that routing around failures can potentially solve upto 62% of the outages. This has further implications for content providers: while a providers “walled garden network” may be well provisioned, performance is bound by the quality of their clients last hop links.

Of the classifiable failures, we present in Figure 5.1 the ratio of failures observed on each path segment to the total number of classifiable failures. To better group failure location, we divide a path into four segments [22]: *Last hop* failures are either the last access link failure or a ‘destination unreachable’ failure of traceroute, *Middle* failures occur in the backbone

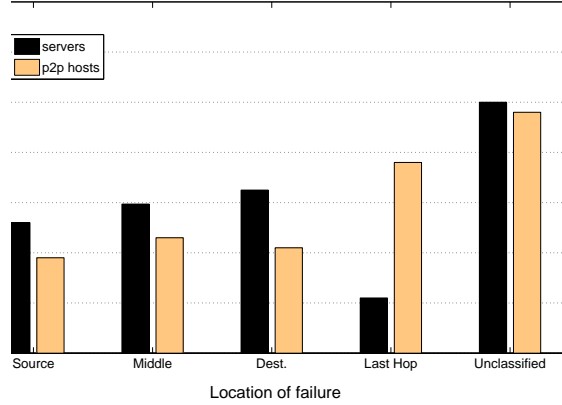


Figure 5.1: Fraction of location failures for classifiable failures; the last column shows fraction of unclassifiable failures observed for all path failures.

network (Tier-1 ISPs) that peer with a POP at the source’s ISP [51]. We infer a middle failure by checking the router addresses to infer a backbone link. Likewise, *Source* and *Destination* are the path segments before and after *Middle*. The plot shows that failures to servers and P2P hosts are equally likely at ‘source’ and ‘destination’. However, paths to servers experience lower last hop failures.

## 5.2.4 Failure Rate

We also measure the number of failures observed by *each* path over the seven day period (Figure 5.2), which represents failure rate for the seven day observation period. Paths are ranked by the number of failures they encountered sorted by highest to lowest. We observe that a fraction of the total paths measured in each destination set experience a majority of

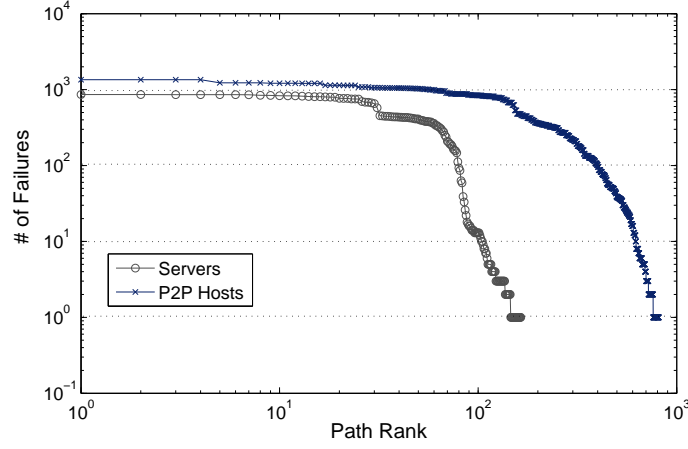


Figure 5.2: Failure rate (log-log scale) of individual paths to servers and broadband hosts.

the failures observed, with few paths registering loss free incidents. Paths to servers observe relatively lesser failures than paths to P2P hosts. Most paths that observe failures share similar number of failures.

### 5.2.5 Failure Duration

Also of interest is the duration of an outage, which gives us a measure of failure persistence. We report on failure duration in terms of the number of *consecutive* MPEG-2 frames impacted due an outage. We count a frame corrupt if at least one packet loss is observed in a given frame, and we continue counting corrupt frames until an intact frame reception is inferred.

Figure 5.3 shows the CDF of the number of consecutive frames impacted as a result of network induced degradation. In more than 50% of the case, there are more than 10 frames

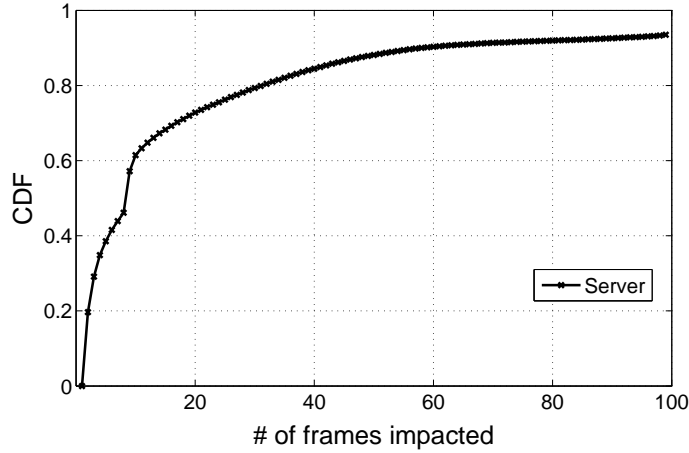


Figure 5.3: Failure duration: number of consecutive frames impacted during outages.

that are impacted during an outage. The probability that a key frame is lost increases with the number of frames impacted per outage. Also, 20% of the outages result in the corruption of more than 50 frames. This strongly brings out the need to detect and quickly recover from on-screen degradations as soon as they occur.

### 5.2.6 Summarizing

The above results make a strong case for Internet redirection, with upto 89% of paths to servers and 62% of paths to broadband hosts potentially recoverable by timely routing redirections; the few paths that observe very high failure rates would almost certainly benefit from redirections. A majority of the paths that evenly observe failures would benefit from timely redirections when outages start occurring. Since BGP convergence times are high

and Internet paths are not chosen based on QoE, route selection will not discover new paths to switch to until the outage continues to corrupt multiple frames. In general, outages that do not occur at last hops can potentially be alleviated by using alternate routes, provided the detection of QoE degradation and path switching happen in a timely fashion.

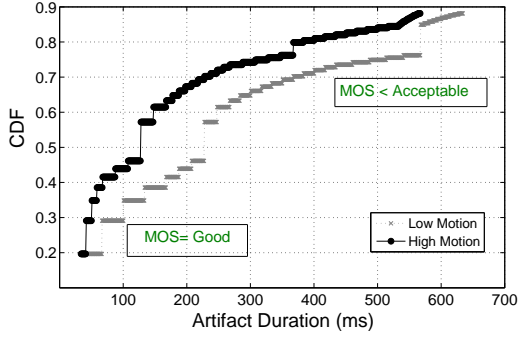
### 5.3 Impact on Perceptual Quality

This section analyzes the *perceptual* degradations caused by packet drops resulting from network anomalies in the IP-traces obtained from the previous round of study. We discuss our methodology of reconstructing MPEG-2 video clips using the IP-traces. These clips were used to conduct a survey with human subjects to better understand perceptual degradations, the factors that affect it, and user preferences. Finally, we summarize our key finding from this round of study to derive application specific parameters that can help preserve QoE.

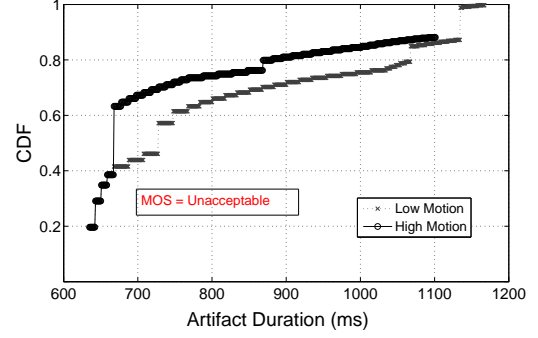
#### 5.3.1 Outage Impact on Perceived Quality

Since destinations were probed using the IP-trace of a given clip, the probe responses create an IP-trace at the receiver which contains round trip times and sequence numbers. We analyze this trace to look for missing information caused by network level degradations. Missing sequence numbers directly capture network drops. To account for jitter, we mark a





(a)



(b)

Figure 5.4: Outages and their impacts on paths to servers: (a) Best case video artifact durations for low and high motion clips, and (b) Corresponding worst case artifact durations.

received probe response as lost if the round trip time exceeds 1 sec (typical playout buffer sizes, e.g. VLC [86]). Hence, after this process, we have an IP-level trace of packet reception for a given video clip.

We quantify the *type* of visual impairment and its *expected* duration within a playout by observing the instantaneous contents in a 1 sec playout buffer and discuss user perception of these impairments from our survey. We make a distinction between the *actual* failure observed on a path and the *perceived* failure. The actual failure is measure of the number of packets lost. However, the perceived failure is the severity of perceptual degradation and its on-screen persistent caused by the actual failure. A *worst case* degradation is when the loss corrupts an I-frame. Likewise, *best case* degradation happen when a loss does not impact an I-frame.

While relative priorities of MPEG frames have been emphasized in the past, we seek to characterize the persistence of on-screen degradation that we infer from the IP-traces. The best case on-screen artifact duration as a result of the corrupted frames for different motion clips is shown in Figure 5.4(a). We note that the on-screen persistence can range from less than 100 ms to about 700 ms. Best case artifacts can range from minor glitches to frozen frames. The worst case artifact duration for the same number of corrupted frames is very different (Figure 5.4(b)). Worst case degradation occurs when the loss corrupts an I-frame, manifesting pixellization, ghosting, and extreme distortions. In this case, the remaining frames cannot reconstruct the scenes and depending upon motion complexity, the persistence of on screen degradation is longer. Even a *single* corrupt I-frame (10ms loss) results in impairments that persist for over 600 ms.

### 5.3.2 Reconstructing Video Clips for Survey

To better understand the *perceptual* experience of viewing clips with the aforesaid loss patterns and artifacts, we decided to recreate a set of clips that are representative of the most commonly occurring loss patterns. From our traces, we observed loss rates of less than 0.1 in a majority of cases, with typical loss rates crowding at around 0.01, 0.05 and 0.1. Loss rate occasionally reached 0.5 and above. We re-constructed video clips at various bitrates using these loss rates. To do this, we manually edited the IP-trace of the low and high motion clips originally obtained to induce these loss rates in a variety of frames. We consider two

possibilities of loss impacting an MPEG frame: (i) loss in key frames (or worst case losses), and (ii) loss in non-key frames (best case losses). We consider an I-frame as a key frame for all clips. We used three encoding bitrates of 800, 3200 and 6400 kbps to reconstruct the video clips. In summary, we recreated a set of 54 unique combinations of losses impacting key frames of high and low motion clips at three encoding bitrates<sup>2</sup>.

### 5.3.3 Survey with Subjects

The reconstructed video clips were used to conduct a survey with an initial set of 80 human subjects in an indoor lab environment. Subjects were shown the original video sequences, and were asked to rate the distorted sequences on a scale of 1 to 5. Subjects were chosen with sufficient diversity in age, gender, and expertise in subject matter. Outliers were identified by interspersing a shown video sequence multiple times and recording their ratings each time. We identified a total of 3 outliers in the lab environment, effectively reducing our survey strength to 77.

We observed many interesting patterns in subjective perception of these video clips with artifacts. For a given loss rate, the perceptual quality can *vary significantly* depending on the motion complexity of the clip and the type of frames impacted during loss. Subjects were less irritated with best case artifacts for low motion clips, and generally rate the clips as “good”. However, subjects were more irritated with increased P-frame losses in high motion

---

<sup>2</sup>All our clips can be examined at [80]

clips, and rate the clips as just about “acceptable”. Subjects consistently rated worst case degradations as “unacceptable”. Low motion clips have larger proportion of I-frames, hence it is more likely that an I-frame is impacted during a loss. Also, because of the longer sizes of GOPs for low motion clips, the persistence of playout distortion tends to be longer.

Interestingly, once the playout reaches “below acceptable” perception, subjects seemed to hardly react any further with continued losses within that GOP. Subjects also tend to “forgive” the degradation if the on-screen artifact suddenly heals with the new arrival of an intact I-frame.

### 5.3.4 Summarizing

Video-QoE is known to be multidimensional, and the overall perceived quality of a service provider depend on parameters that go beyond network efficiency. Next, we focus on discovering *network* induced degradations that are addressable by using alternate paths in the Internet.

We summarize our basic assumptions about QoE that we use for the rest of this chapter as follows. For each instance of a corrupted frame in a GOP, an artifact is produced. Not all artifacts induce the same user reaction. Subjective perception degrades to “below acceptable” when key frames are corrupted within a GOP. For low motion clips, we mark the I-frame as a key frame. For high motion clips, we mark both I- and P-frames as key frames.

While subjective perception degrades with the loss of key frames, immediate restoration of key frames following a degradation induce a “forgiveness” effect. Interactivity delay of more than 500 ms network round trip time degrades QoE.

## 5.4 Using Routing Redirections

Given that 89% of outages on paths to servers and 62% of paths to broadband hosts are recoverable by alternate routing, we investigate frame preserving policies and path selection strategies that can raise perceptual quality. We observe that a strategy of preserving key frames following a degradation can instantly convert worst case degradations to a best case degradation and raise perceptual experience. Given a degradation on an Internet path, however, we need a deeper understanding on how long an outage persists when a frame is corrupted, how soon should one switch paths, and what is the best strategy to utilize redundant Internet paths without the need for background monitoring. Choosing paths without background monitoring allows overlays to scale to large number of nodes. It also makes such strategies generally applicable in a wide variety of streaming services without burdening the existing infrastructure.

The Internet by default returns one path for a given destination to the source. Alternative paths can be derived by creating an *overlay* network. Overlays are not a new concept to computer networking: the Internet itself was built as an overlay on top of the telephone network [1, 22]. Current examples of overlays built on top of the Internet include: P2P

networks, content delivery networks or CDNs (e.g., Akamai [74]), multicast networks [27], OverQoS [52] etc. These networks often have a multitude of nodes in different ASs that can likely provide redirections around outages.

#### 5.4.1 Methodology

We analyze five different datasets that contain weeklong measurements of a large number of Internet paths all over the world. We streamed packets using IP-traces of a variety of low motion and high clips between source destination pairs. When transmitting key frames from the clip, we *simultaneously* probe every intermediary which indirectly probes the destination. We record the receiver trace at the destination, and the probe responses from the intermediaries at source to analyze offline the suitability of alternate paths during outages. We then derive ways to select alternate paths while preserving zapping delays *without* the need to perform background monitoring of any kind.

**Datasets:** We created five measurement overlays in different parts of the world, both within a continent and across continents. We created one overlay each within United States (US), Europe (EU) and Asia-Pacific (AP) consisting 19, 21, and 22 nodes respectively. In addition, we created two international overlays D1 and D2 consisting of 22 and 32 nodes *evenly* spread across the U.S., Europe and Asia-Pacific.

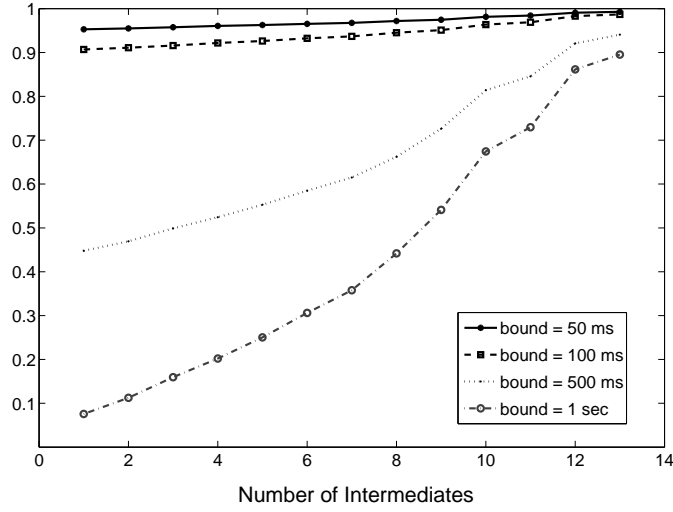


Figure 5.5: CDF of number of useful intermediaries as a function of the upper bound on tolerable delay.

**Experimental Setup and Video Clips:** Between January 22 to 29, 2010, every node streamed 1024 byte UDP packets every 5 minutes to a randomly selected destination. The stream mimics the IP-packet trace of a randomly selected high or low motion clip from a set of five clips used in the previous round of study. We passed the name of the clip and the type of frame the packet carries in the packet payload, creating an IP-trace of the clip at the receiver. For any of our overlay with  $N$  nodes, the source indirectly probed the destination via the  $N - 2$  other intermediaries while streaming packets to the destination. This probing is performed only when transmitting key frames. For low motion clips, we mark the I-frame alone as the key frame. For high motion clips, we mark both the I- and P-frames as key frames.

### 5.4.2 What kind of paths help QoE?

In general, preserving key frames and providing consistent zapping times are excellent network level support to raise perceptual quality. Hence, paths that can re-route subsequent key frames *and* ensure that zapping times don't exceed a given bound are excellent candidates for selection in times of an outage. By "bound", we mean that the difference in RTTs of the new path and the default IP path is within a given threshold, i.e.,  $RTT_{new} < RTT_{Default-IP} + bound$ . As a result, not all intermediaries can be of use if they exceed this bound even if they are determined to be loss free.

### 5.4.3 Suitability of Intermediaries

For a default IP-path from source (S) to destination (D), and intermediary (I) is considered "useful" when the alternative route stitched together by combining paths of (S,I) and (I,D) is loss free *and* whose round trip time is bounded by a given value. The choice of this bound limit can have a significant effect in the choice of intermediaries that can be considered useful.

Shown in Fig. 5.5 are the number of useful intermediaries each time a frame was corrupt at the receiver as a function of the delay bound. When bounds are tight (50 ms), the number of useful intermediaries in times of an outage are low. In fact, the same chosen few nodes tend to help recovery and the probability of finding newer nodes are low. When the bound



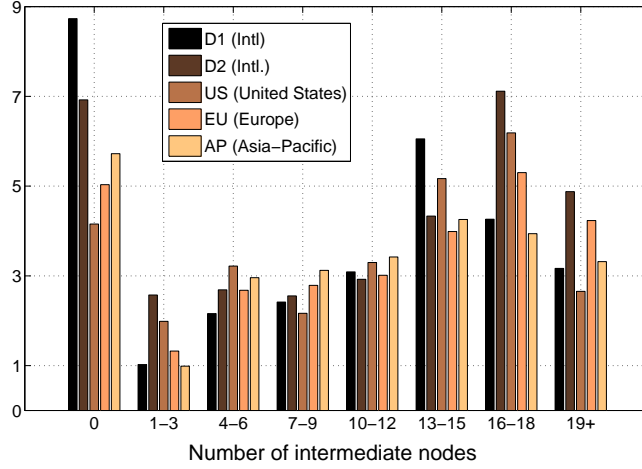


Figure 5.6: Number of intermediaries that offered a better path in times of failures.

is loosened to around 1 sec, we see an even likelihood of finding a varying number of useful intermediaries for every instance of an outage. We also observed (not shown in the plot) that the number of useful intermediaries tend to be more when the overlay is confined to a geographical area, largely because of the availability of paths with round trip times within the defined tolerable bounds. Note that this plot reports data from dataset D2, which is spread all over the world.

From our survey in the previous round, we note that worst case degradations often persist between 600 ms to 1 sec depending upon the severity of impact. Also, subjective perception does degrade any further with increases losses for the entire GOP given an impaired I-frame. Hence, given the duration of on-screen artifact persistence and subjective perception, we choose an upper bound for RTT as 500 ms for selecting intermediaries.

#### 5.4.4 Useful Intermediaries

For every instance of a corrupt frame at the receiver, we analyze the number of useful intermediaries that can help preserve subsequent frames with a bound of 500 ms. Fig. 5.6 shows the fraction of times the number of intermediaries (in bins of 3) were determined useful each time a frame was corrupt. We observe that few failures could be exclusively addressed by a small number of nodes (bin [1-3]). A large number of failures could find many intermediate nodes that prove helpful. We also observe that a fraction of these failures could not be recovered by any alternate route (the left most bar). This number seems to be relatively higher for international datasets (D1 and D2) than datasets derived from within a geographical area.

#### 5.4.5 Choosing Intermediaries

Given the number of useful intermediaries in times of an outage, we now look at path selection strategies that can improve perceived quality. Our key requirements in designing a path selection strategy is twofold: (i) path selection is done *without* the need for background monitoring or apriori path quality knowledge of any kind, and (ii) the approach is simple, lightweight and adds negligible computational overhead at the sender. We assume that the receiver can provide a feedback to source informing an outage whenever key frames are corrupt.

Similar in spirit to randomized load allocation [12, 15, 22], we employ a strategy of *randomly* selecting any  $k$  intermediate nodes in times of an outage, and simultaneously attempting to transmit the subsequent key frames through them. The first such intermediate node of the chosen  $k$  which is loss free and whose RTT is bounded is chosen as the best alternative and we continue streaming via that node. A subsequent failure on that path again triggers the random- $k$  strategy until a new path is found. In case of finding no paths, we re-invoke random- $k$  until we find a suitable intermediary or if the IP-path self repairs. By sending the next set of key frames of multiple paths, we maximize the chances of at least one of the paths to deliver the frames that helps restore quality.

A natural question then is the what value of  $k$  presents a reasonable tradeoff between reducing the number of nodes to be simultaneously attempted for recovery while maximizing gains in the resulting perceived quality. To answer this, we measure the fraction of outages recovered by various values of  $k$  across all our datasets. An outage is considered recovered if the subsequent key frames are corrupted in the default IP-path while the path through the intermediary was both loss free and within a desired round trip delay bounds.

Fig. 5.7 shows this for all five datasets, and additionally shows the results for different delay bounds with dataset D2. Each datapoint was obtained by calculating the recovery percentage using that value of  $k$  for the entire trace period on a given dataset. We observe that for all datasets, the value of  $k = 5$  presents a reasonable tradeoff in selecting intermediaries. Beyond  $k = 5$ , we observe the law of diminishing returns: attempting to recover from more number of intermediaries results in little gain. For datasets confined to a geographical

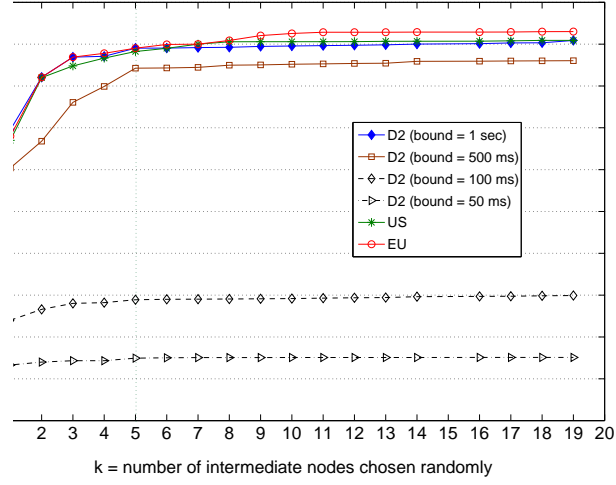


Figure 5.7: Fraction of outages recovered by transmitting the next GOP to  $k$ -intermediate nodes; for dataset  $D2$ , we additionally plot success for different delay bounds.

area (e.g.,  $US$  or  $EU$ ) we observe that the value of  $k = 4$  provides comparable gains owing to more intermediaries within the desired RTT bounds. For smaller RTT bounds, we observe that the gains hit a ceiling after a small number of intermediaries because only a few select intermediaries out of the available ones can help recover from outages.

#### 5.4.6 Path Switching with random-5

Path switching is performed when the destination reports a degradation which impairs perceptual quality. We now investigate the following question: how soon should a receiver inform of a degradation, and what are the benefits of switching paths early.

We begin by looking at the typical loss patterns in key frames that were corrupt at the receiver. For dataset D2, Fig. 5.8 shows the typical number of consecutive packet drops observed in key frames for low and high motion clips. The plot shows that reception is void of any losses 51% of the time. Consecutive losses of 2 or more packets are seen only seen in 29% of the cases. Since a single loss is sufficient to degrade quality, we argue that the receiver should inform source to switch paths after a single packet lost in a key frame.

The benefits of switching paths early are further shown in Fig. 5.9. This plot shows the probability of the next key frame being received successfully *after* observing a certain number of consecutive drops in a key frame. We plot this probability for upto 6 consecutive packet losses observed for  $k = (1, 5, 10)$  alongwith the performance of the default IP-path. After only two successive drops, the probability of the default IP-path restoring the next key frame seem to diminish to around 0.42. Both random-5 and random-10 maintain a higher recovery probability of more than 80% for upto 5 consecutive drops. Once again, the additional gains by selecting 10 nodes over 5 for transmitting key frames are marginal. This leads us to believe that paths should indeed be switched early. The plot also shows that even random-1 is able to provide higher returns than the default IP-path when 3 or more packets are lost in succession.

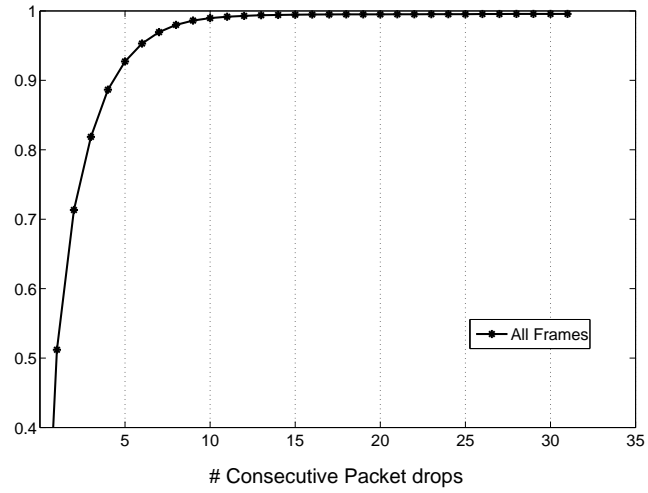


Figure 5.8: CDF of packet loss distributions in key frames.

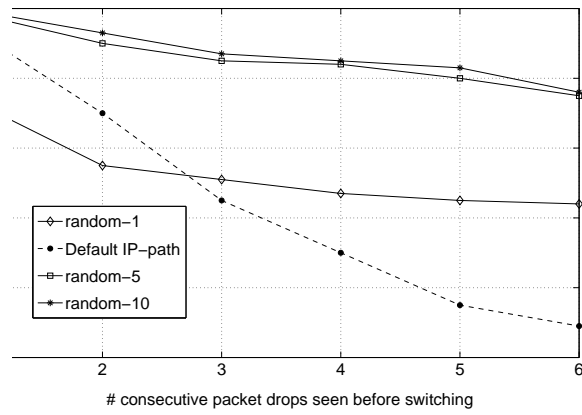


Figure 5.9: Benefits of switching early: probability of recovery after consecutive packet losses in key frames.

### 5.4.7 Robustness

We further elucidate the robustness of random-5 by next looking at the probability of recovering from a perceptual degradation either due to random-5 recovering from an outage or the IP-path self-repairing itself when random-5 cannot solve the problem after persistent efforts. Fig. 5.10 shows the CDF of all degradations recovered from either due to random-5 or the IP-path self repairing itself as a function of time elapsed since the destination reported a degradation. The success probability of random-5 continues to be 0.84 irrespective of the time elapsed since the degradation was reported. The IP-path, however, repairs itself with an increasing probability with elapsed time since an outage. The mean time to recover from outages for the default IP-path is typically 30 secs. The plot highlights the ability of random-5 to recover quickly from perceived degradations to restoring playout within the first few seconds.

### 5.4.8 Preserving Interactivity

When a source selects a path using the random-5 strategy, it automatically ensures two things: (i) the key frames make it to the destination in the *least* time possible using any of the five alternate paths, and (ii) the selection of the path with the minimum delay within specified maximum bounds ensures round trip times do not exceed the stated bound. Fig. 5.11 shows the difference between the mean round trip time of the default IP-path and paths due to

random-5 between all source destination pairs for all our datasets. While it is easy to observe that the additional round trip delay by choosing alternate paths is bounded (because the source will not consider a path successful until the round trip is within bounds), what is interesting to observe is the occasional *improvement* in round trip time. An improvement in RTT could be either due to: (i) the alternate path having a round trip time that is indeed less than the default IP-path, and (ii) during times of an outage, the round trip path to the source increases and the alternate paths which do not experience that outage have a smaller RTTs. We observe that the difference is smaller for overlays confined to a geographical area compared to overlays spanning multiple continents. This is largely because of the higher availability of alternate paths with desired RTT bounds within a continent.

#### 5.4.9 Summarizing

Though a number of intermediaries have loss free paths to the destination in times of a perceptual degradation, not all of them can be a viable alternative if the desired round trip delay is bounded to preserve interactivity. Using a bound of 500 ms, we investigated ways to restore perceptual quality by attempting to preserve key frames following an outage using intermediate nodes. Our results indicate that a source can restore perceptual quality by simultaneously transmitting key frames to five randomly chosen intermediaries following a degradation. We observe that random-5 is robust across all our datasets.



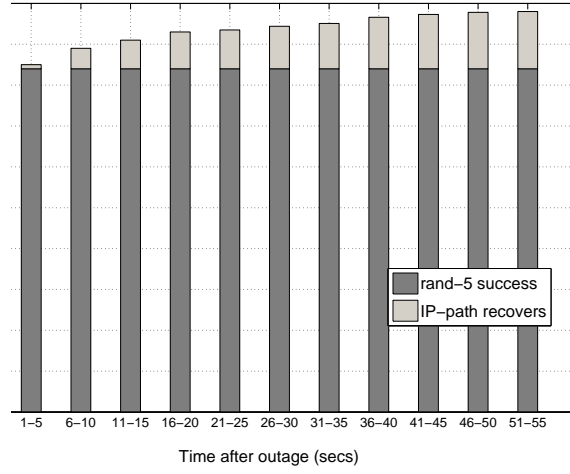


Figure 5.10: Resilience of random-5 recovery time over IP-path self repairing itself.

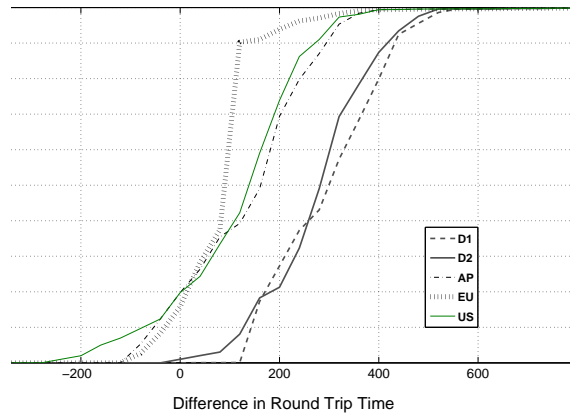


Figure 5.11: Difference in round trip times resulting from random-5 path selection with bound = 500ms.

## 5.5 Conclusions

This chapter presented large scale Internet measurements to understand the effects of Internet path selection on perceptual quality of MPEG-2 video and investigates ways to improve it. We began by performing repeated video “fetching” acts from top IPTV/VoD providers, PPLive hosts and random Internet destinations for one week from geographically diverse PlanetLab nodes. We mapped the probe responses to perceptual quality by reconstructing numerous representative low and high motion video sequences and conducted subjective surveys using them. Consistent with recent research, our findings indicate that degradation depend upon motion complexity and type of frame impacted among other things. High level results also indicate that upto 89% of paths to servers and 62% of paths to broadband are recoverable by using alternate paths.

To understand the benefits of using alternate paths, we collected weeklong measurements from five different datasets that both confine to and span multiple continents with a dominant presence of online streaming services. We observe that not all alternative paths can be useful even if they are loss free, and that a large fraction of degradations could be overcome by a large number of alternate paths. We investigated ways to restore quality by attempting to route successive key frames through  $k$  random intermediate nodes *without* relying on any kind of background path monitoring or apriori path knowledge. Our results indicate that  $k = 5$  provides a reasonable tradeoff between minimizing  $k$  and maximizing gains, and we argue that our results are consistent across datasets.

We believe our results have implications for any video source that streams content across the Internet. A technique of randomly choosing intermediaries requires little overhead. This promises large, scalable overlays to be easily build, deploy and maintain. We show that it is possible to achieve substantive gains in perceptual quality using our prototype on top of todays best effort Internet.

## CHAPTER 6

# RANDOMIZED PATH SELECTION IN LARGE UNSTRUCTURED OVERLAYS

“All I want to do is wake up and get out of this endless mess of twisted paths” Alice  
told the cat.

“If that was what you desire you would have already chosen that path. You’ll know  
which path to take if you just look closely” the cat replied.

---

— Lewis Carroll (Alice in Wonderland)

### 6.1 Introduction

In the previous chapter, we found that randomized path selection in routing overlays holds a promise for raising video-QoE. Routing overlays have been proposed to overcome many of Internet’s deficiencies by exposing numerous alternative paths. This allows end-hosts greater control over the path taken by its packet. Overlay networks have the potential to offer low delay paths and provide quicker path recovery. The biggest challenge in building

scalable multimedia overlays, however, is to consistently negotiate faster paths while scaling to thousands (even millions) of users.

Path selection in routing overlays is currently achieved either by background monitoring of alternate paths [1, 52] or by embedding overlay participants in a co-ordinate space [67]. With background monitoring, an overlay of  $N$  nodes requires constant monitoring of  $O(N^2)$  paths to choose the best detour at any given time. Though this provides the most accurate result, the adoptability of this approach quickly diminishes with increasing number of nodes. Furthermore, all-to-all probing makes available far more paths than necessary for reliability or latency reduction: most such paths are useless. Another approach to scalable overlay services is to embed node locations to a co-ordinate space, and predicting detour performance based on node location. Such a set-up leads to highly scalable overlay networks since it requires little or no background monitoring. However, end-to-end latencies in the Internet exhibit *triangle inequality violations* (TIV). Consider three nodes in Fig. 6.1 that form a triangle such that node-B can offer a detour to messages from node-A to node-C. This triangle ABC exhibits a TIV, since the (side)-AC > (side)-AB + (side)-BC. Research has shown that more than 5% of triples and more than 50% of pairs of nodes on the Internet exhibit TIV [32, 67]. Naturally, TIVs severely hamper latency estimation by introducing non-trivial errors in delay estimation. This again leads to sub-optimal performance. Hence, there is a pressing need to overcome fundamental limitations of building efficient, large scale overlays to improve Internet path selection for a variety of interactive multimedia applications.

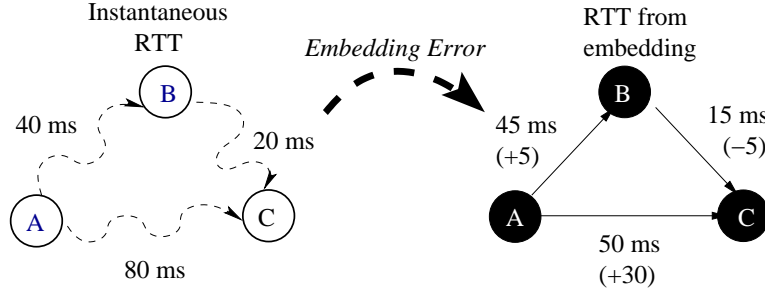


Figure 6.1: Errors when embedding nodes to a co-ordinate space with TIV. Parentheses (right) show errors from embedding

This chapter investigates ways of building scalable latency reducing overlays that *do not rely* on background monitoring or location information to select alternate paths. Using weeklong measurements from 500+ vantage points in the Internet, we show that it is sufficient to reroute using ‘ $k$ ’ *randomly chosen* intermediate nodes for an overlay with  $N$  nodes. We show that the value of  $k$  is bounded by  $O(\ln N)$ ; which implies that  $k = 8$  for an overlay with 1000 nodes, and  $k$  is just 14 for an overlay with one-million nodes. We also observed that for random- $k$  to be effective, the subset  $k$  should be *uniformly representative* of the  $N$  nodes in the participating overlay.

Key properties of an overlay built using the random- $k$  philosophy (and its variants) are further investigated. Our results indicate that: (i) random- $k$  takes more than 75% of all potential detours that an idealized path selection would take, (ii) random- $k$  is very effective at negotiating faster paths, and reduces RTT by more than 50 ms more than 80% the of time, (iii) random- $k$  increases availability of end-hosts, and (iv) random- $k$  leads to *improved* fairness amongst participating nodes.

## 6.2 Data Collection Methodology

This section describes our data collection and correlation methodology in detail, which we use as an input for the rest of this chapter. We discuss our choice of vantage points, some possible caveats and steps we took to address them.

### 6.2.1 Vantage Points

We observe that Internet based multimedia services are popular in the U.S., Canada, Germany, France, Belgium, U.K., China, Korea and Taiwan to name a few. To better reflect on these locations, our vantage points are generally located in US/Canada, Europe, and the Asia-Pacific. A total of 556 PlanetLab [82] vantage points were used on the wide area Internet. Fig. 6.2 shows a graphical representation of our experimental setup, highlighting the concentration of nodes in specific geographical locations. The vantage points were chosen with a ratio of 3:2:2 for US:EU:AP.

We initially began with 700 vantage points. However, we removed data from 144 vantage points which either failed during the weeklong probing or had more than 48 hours worth of trace data loss, effectively reducing our set of vantage points to 556.

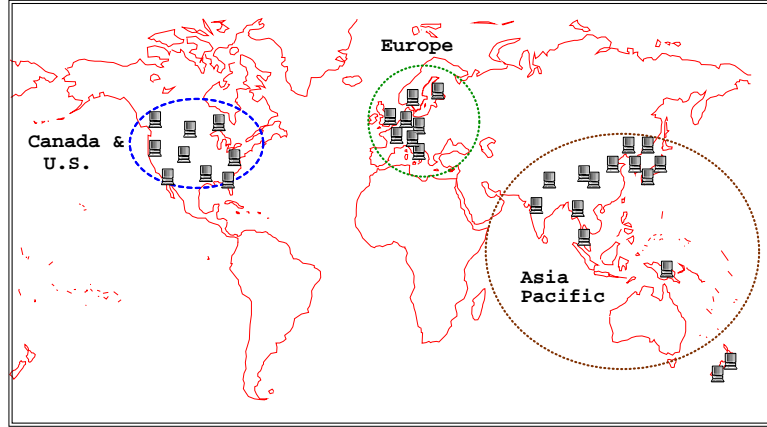


Figure 6.2: Experimental set-up consists of 556 nodes deployed across the US/Canada, EU and AP

### 6.2.2 Experimental Setup

**Probing:** Starting January 17, 2011, each vantage point continuously probed every other vantage point in our list for seven consecutive days. Probes were responded to by the destination node that allows each source node to record round trip times (RTT) from taking that path. Probing was performed in individual *sessions*, where each session consists of 120 probes spread over 2 minutes. As a result, every vantage point performed 720 sessions for every destination per day. When a vantage point (say ‘A’) probed a destination vantage point (B), it also simultaneously probed the destination B *indirectly* via all the other 554 vantage points. In the end, we obtained probing results for 154,290 direct paths which we could compare to 308,024 detour paths to analyze offline the path dynamics connecting every source-destination pair. This allows us to analyze offline ways to choose these paths without having to perform background monitoring.



Table 6.1: Dataset Overview

| Dataset Name | # of nodes | # of paths |
|--------------|------------|------------|
| PL-master    | 556        | 154,290    |
| US           | 239        | 28,441     |
| EU           | 159        | 12,561     |
| AP           | 158        | 12,403     |

**Data Correlation:** Trace data collected from probing was organized in a way that allowed us to analyze paths based on a list of participating nodes in the overlay. Though we collect data from one central global overlay, we could create multiple *views* of the same data to analyze local geographical behavior. Table 6.1 shows the four different views of the dataset we use. **PL-master** is the central repository of all our data, while datasets **US**, **EU** and **AP** are subsets of **PL-master** which only contain paths within the respective geographical area. The table also shows the number of paths within each overlay we report results from.

### 6.2.3 Caveats

We discuss some possible caveats in our data collection methodology, and discuss steps we took verify them.

**Data Completeness:** Path characteristics from our dataset should be representative of Internet behavior. Before we began probing paths connecting our vantage points, we issued

numerous traceroutes between every pair of vantage points. Traceroute data allows us to analyze the list of ISPs used by these paths. We find that 86% of these paths traverse at least one of the 51 major ISPs in US, EU and AP [49]. Also, we avoided sites that are known to use an Internet-2 connection, which is known to be superior.

**Transient routing changes:** Transient routing changes and load fluctuations in the Internet do not paint an accurate picture of the state of affairs. To smoothen transient changes data-collection was spread over multiple times of the day, including weekdays and weekend, to eliminate bias.

### 6.3 Overview of random- $k$

How can a source select alternate overlay paths to a destination to ensure the best RTT, without any apriori knowledge of detour path characteristics? This problem is akin to a distributed service that must track other participants to choose the best alternative. Ideally, this amounts to continuously tracking every participant to take the most informed decision; this is infeasible beyond a few hundred nodes. Randomized load allocation can reduce or even eliminate the need to continuously track participants, while holding the promise to scale to millions of users [12, 15, 22, 58].

**Using randomized load allocation,** a source can periodically (every  $\Delta_t$ ) attempt to concurrently re-route packets using a random subset of  $k$  nodes from the overlay. The

first such path that gets an acknowledgement from the receiver is deemed to be the best path and the source continues to use that path for the remainder of  $\Delta_t$ , until it retries again. Assuming a simple feedback mechanism from the destination, this strategy enables the source to discover and choose from upto  $k$  paths compared to the single default-IP path presented by the Internet. We next investigate the effectiveness of this strategy in its ability to improve IP-path selection.

### 6.3.1 $k$ nodes are enough

We begin with a high-level overview of the number of successful Internet detours provided by a random- $k$  strategy for all four datasets. A successful detour is one that either reduces the current latency, or provides a loss free path with an RTT increase of no more than 50 ms when the default path is deemed lossy. We compare the performance of random- $k$  against an *idealized* path selection which has complete global knowledge of all detour paths.

Fig. 6.3 provides a high level overview of these results. The plot indicates the ability of random- $k$  at successfully providing detours as a function of  $k$ . We report on the percentage of detours offered when compared to idealized path selection which returns 100% of all detours possible. By definition, random- $N$  is similar to the idealized path selection.

We observe that even random-1 takes 30% of the possible detours across the four datasets. For each dataset, we also notice a “knee of the curve”: an optimum value of  $k$  beyond which

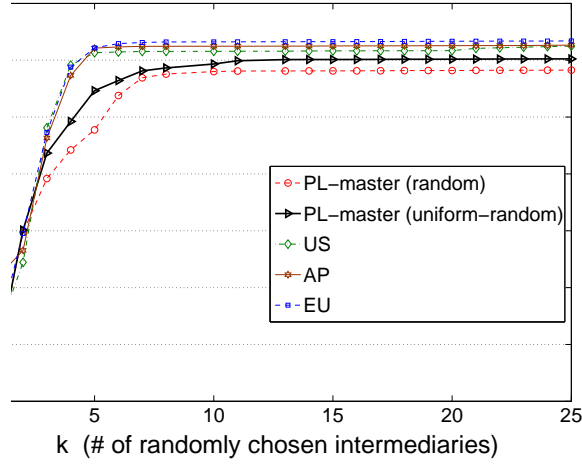


Figure 6.3: Percentage of potential detours taken using a random subset of  $k$  nodes

there are diminishing gains. For smaller overlays (US, EU, and AP), we observe that  $k = 5$  provides a reasonable tradeoff. For PL-master, the optimum value of  $k$  is approximately 7. Across all four datasets, we observe that random- $k$  offers more than 75% of detours available from idealized route selection.

### 6.3.2 Composing $k$

Given that  $k$  nodes are sufficient for providing successful detours, we next look at various ways of composing  $k$ . Specifically, we are interested in strategies that a node must employ to derive maximum gains from creating such random subsets. For e.g., consider a node in U.S. that participates in such an overlay that wishes to communicate with nodes in Europe.

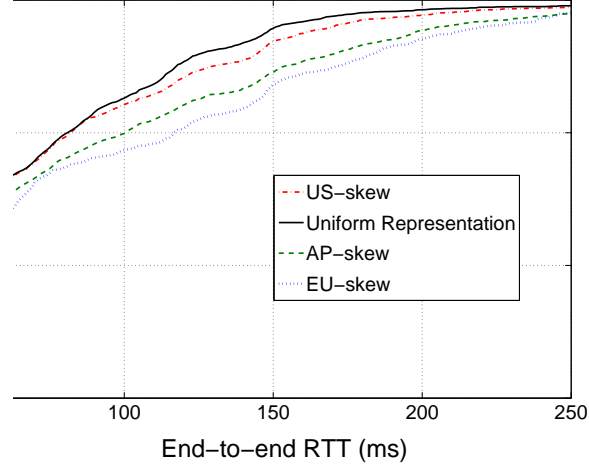


Figure 6.4: Importance of composing a representative subset: effects of skew (PL-master)

Intuitively, if the random subset is formed entirely of nodes from AP (owing to random selection), the promise of offering detours is likely to diminish for this scenario.

Our results indicate that the subset  $k$  should be *uniformly representative* of all participating nodes in the overlay. In other words, since the participating overlay consisted of nodes in the ratio 3:2:2 for US:EU:AP, the subset  $k$  should also follow a similar ratio for optimum performance. Fig. 6.4 shows the effects of *skew* in composing  $k$  on the end-to-end RTT between nodes: cases when this ratio was violated. We call a composition skewed when one of US, EU, or AP is more dominant in the makeup of the random subset. For example, when  $k$  is ‘US-skewed’, the ratio of  $k$ ’s makeup was 5:1:1. The plot shows that when  $k$  is skewed towards more nodes in US, EU or AP, the performance degrades.

Table 6.2: Comparison of overhead incurred

| Overlay Type                     | Overhead (packets) |
|----------------------------------|--------------------|
| random- $k$ ( $\Delta_t = 60$ )  | 550,000            |
| random- $k$ ( $\Delta_t = 120$ ) | 120,000            |
| random- $k$ ( $\Delta_t = 300$ ) | 40,000             |
| $O(N^2)$ monitoring              | $12 \times 10^8$   |

### 6.3.3 Overhead

We next look at the number of redundant transmissions required for the upkeep for an overlay using random- $k$  for various values of  $\Delta_t$ . Since the source attempts  $k$  simultaneous nodes every  $\Delta_t$  seconds to negotiate a better path, there amounts a certain number of redundant copies which form an overhead. We assume the worst case scenario: every node in the overlay continuously uses the overlay for the entire hour, seeking to negotiate a newer path every  $\Delta_t$  seconds. Note that overlay nodes are normally *not* required to invoke random- $k$  when not sending packets into the network.

Table 6.2 summarizes the number of overhead packets for one hour of usage for an overlay of 550 nodes. We observe that  $\Delta_t = 120$  provides a reasonable trade-off between number of redundant copies to frequent path selection. We also observe that for an overlay employing per-second  $O(N^2)$  monitoring, the amount of overhead required in this same period is greater by a factor of  $10^3$ .

### 6.3.4 Summarizing

High level results indicate that it is sufficient to reroute packets using a random subset of  $k$  nodes in an  $N$  node overlay for large scale latency reducing overlays with  $\Delta_t = 120$  seconds. Such a scheme can take upto 75% of all detours an idealized path selection would take. We also observed that random- $k$  performs at its optimum when the random subset  $k$  is uniformly representative of participating nodes.

## 6.4 Characterizing Random- $k$

We next look at *alternatives* of random- $k$  and their potential benefits and tradeoffs. We explore two variants of random- $k$  which use increased global state information to provide detours to compare with our stateless approach. We compare their ability to select better detour paths, as well as the ensuing fairness in detour selection.

### 6.4.1 Can we improve random- $k$ ?

We observed the law of diminishing returns in Fig. 6.3: the gains hit a ceiling with increasing values of  $k$ . We next look at ways to achieve additional gains by exploring two variants of random- $k$  which use increased global state knowledge. We compare random- $k$  to these variants to understand its true benefits and limitations.

**History- $k$ :** The first variant is a philosophy similar to random- $k$ , but the source node remembers and additionally retries the last successful detour. In other words, the source selects  $k - 1$  detour nodes at random (uniformly representative), and additionally chooses the last successful detour node to form a random subset. The rationale behind this is that the last successful detour node can likely provide further redirections if it indeed has superior connectivity. It has been observed that in many overlays, few nodes offer a majority of all detours within the overlay [32].

**Disjoint-AS- $k$ :** This variant assumes complete global knowledge of Internet autonomous system (AS) connectivity, and specifically selects a random subset that avoid AS's that the default path takes. The source node chooses the  $k$  most promising detours whose paths traverse AS's different than all the AS seen on the source-destination path. While this strategy may not be feasible in implementation (such global knowledge is impractical), this exercise allows us to evaluate the effectiveness of random- $k$  against an idealized version to understand its true potential.

Fig. 6.5 compares the effectiveness of random- $k$  against these policies using the **PL-master** dataset. The plot shows the percentage of valid detours each strategy takes for various values of  $k$  compared to idealized path selection. We notice that for small values of  $k$ , both history- $k$  and disjoint- $k$  outperform random- $k$  very quantitatively. However, with rising values of  $k$ , the gap between these strategies quickly diminishes. This leads us to believe that the gains are comparable. In fact, little separates random- $k$  with disjoint-AS- $k$  for high values of  $k$ . We also note that history- $k$  seems to perform slightly better amongst all three variants in



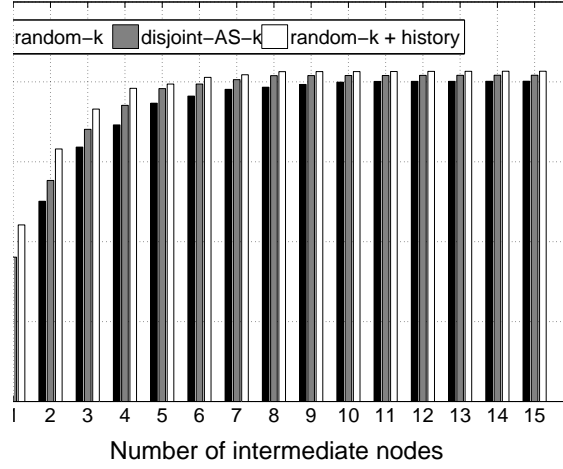


Figure 6.5: Variants of random selection and relative performance gains

taking a higher percentage of correct detours.

The benefits of using variants of random- $k$  with increased state information are marginal. In effect, these strategies bias their choice of subsets towards detours with higher promise, and this bias results in some additional gains. The gains, however, quickly diminish when we consider the increased state knowledge or the effects of bias on fairness. Disjoint-AS- $k$  assumes global knowledge of Internet connectivity, making it infeasible as a strategy of choice. History- $k$ , as we will soon see, leads to decreased fairness amongst participating nodes. The practicality and ease in implementation that comes with random- $k$ , with reduced deployment costs, makes it a more suitable choice of choosing subsets.

### 6.4.2 Fairness

While overlay networks make a diversity of paths available, it is often possible for nodes with superior connectivity to service a majority of all detours. This leads to a network that consists of a large fraction of freeloaders who use the remaining (smaller) fraction of nodes for detours. In the previous round of study, we noticed that history- $k$  often performs slightly better than random- $k$  in offering detours. We next look at fairness in selecting detours by comparing history- $k$  to random- $k$ .

Fig. 6.6 shows the frequency of a node offering a detour for all nodes across all four datasets for both history- $k$  and random- $k$ . Nodes are sorted by their ‘rank’, which is determined by the number of times they appear as a detour node for all source-destination pairs. By design, history- $k$  is biased towards a successful detour node, and ends up creating greater unfairness in load selection. Random- $k$ , on the other hand, uses no history data and generally results in spreading the detour load amongst a greater number of participating nodes. In general, randomness is a proven technique to improve fairness in many load-sharing systems.

**Summarizing:** We tested random- $k$  against two variants which assume global connectivity knowledge or rely on history to better select alternate paths. We observed that random- $k$  is comparable in terms of gains, while significantly reducing deployment costs and increasing fairness.

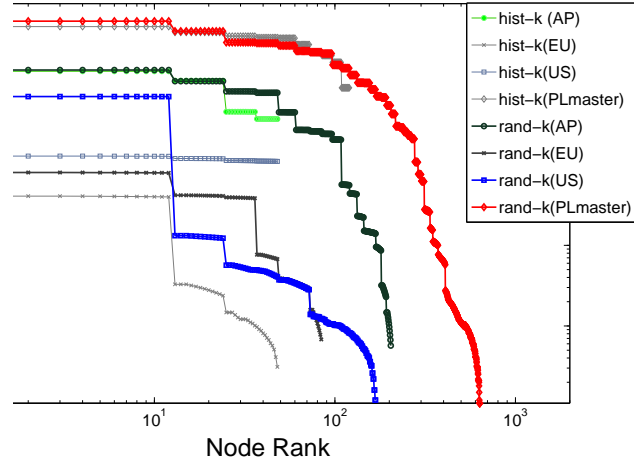


Figure 6.6: Random selection leads to improved fairness in detour selection

## 6.5 Discussions

We next look at real-world benefits of using random- $k$  to derive low latency paths and empirically characterize the value of  $k$  for any overlay of  $N$  nodes. We discuss the implications of our results, and summarize them as key guidelines for building an Internet path selection mechanism that a variety of multimedia applications can benefit from.

### 6.5.1 Reduction in RTT

We begin by looking at the reduction in RTT possible using the random- $k$  strategy. A source repeats the random- $k$  strategy every 120 secs to look for a better path that reduces latency and avoids loss.

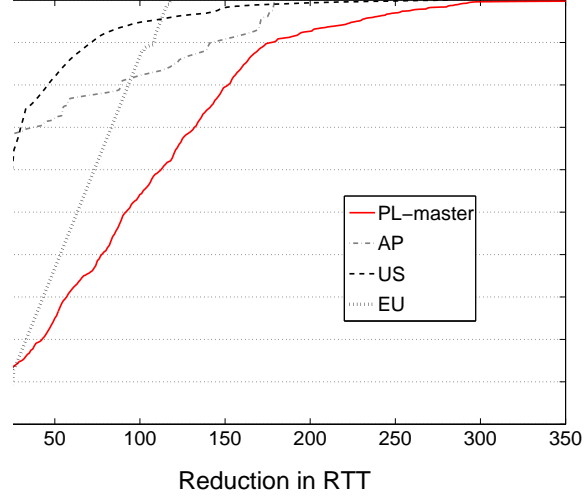


Figure 6.7: Reduction in RTT from using detour paths with random- $k$

Fig. 6.7 shows the reduction in RTT achieved by random- $k$  across all four datasets. For **PL-master**, the gains are consistently above 50 ms more than 70% of the time, and more than 150 ms at least 20% of the time. For datasets **US** and **AP**, the gains hit a ceiling at around 30 ms, as we observe that a majority of reductions are within that range. We observe a slightly larger reduction in RTT for the **EU** dataset.

The plot shows that RTT reduction is generally not a factor of geographical prescience since the gains are consistent across continents. This implies that ISP topologies and interconnects are possibly replicated. Also, we did not observe fluctuations in gains for weekdays/weekend.

### 6.5.2 $k$ is bounded by $O(\ln N)$

A key property in building a scalable overlay architecture is that the number of paths to be probed should grow *sub-linearly*. In other words, for an overlay of  $O(N)$  nodes, the value of  $k$  necessarily needs to be less than  $N$ . We next empirically characterize  $k$  as a function of  $N$ , and establish upper bounds on the value of  $k$  for random- $k$  to be universally applicable to any latency reducing overlay.

Fig. 6.8 shows the optimum value of  $k$  for overlays of different sizes. An optimum  $k$  is derived by results presented earlier in the chapter:  $k$  is uniformly representative of all nodes in the overlay, and  $\Delta_t = 120$  seconds. Each data point for the optimum value of  $k$  was observed by calculating the tipping point observed in Fig. 6.3 for different sizes of the overlay. The tipping point is the value of  $k$  beyond which there is little gains in random- $k$ 's ability to select better paths. We use the **PL-master** to calculate this tipping point.

We observe that the value of  $k$  completely contained within the curve  $y = \ln(x) + 3$ , where  $x$  is the number of nodes in the overlay. In other words, this provides an empirical characterization that  $k$  is bounded by  $O(\ln N)$  for an overlay with  $N$  nodes. We also note that  $O(\ln N)$  grows sub-linearly with  $N$ , which means that overlays built using this philosophy are scalable. We also note that our results have implications for *any* overlay service built on the Internet, including CDNs and peer-to-peer networks.

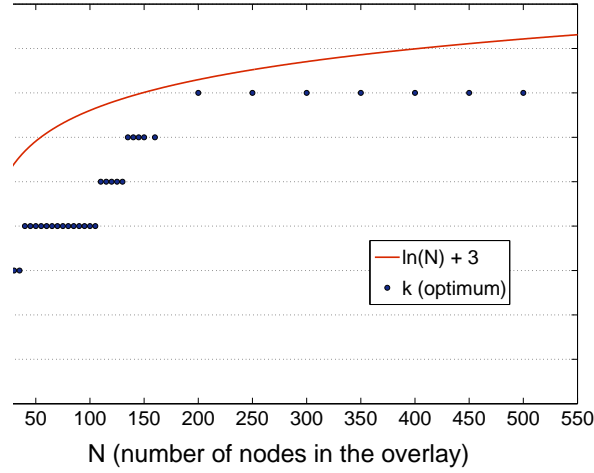


Figure 6.8: The value of  $k$  is bounded by  $O(\ln(N))$  for any given  $N$  nodes in the overlay

### 6.5.3 Discussions

**Group Membership:** Nodes wanting to join an overlay could simply send a ‘join’ message to any existing overlay node, which propagates (gossip) the new node’s presence in the network in return for a list of participating nodes.

**Availability:** The overlay should be available for nodes to freely query and negotiate faster paths. Notice that our design *does not* assume the support of a centralized server, and as such avoids single point of failures. Since nodes compute random subsets in a distributed manner, the scheme inherently promotes scalability. We believe such a system can also handle node churn seamlessly. Nodes may become unavailable due to a variety of reasons. If a node fails to respond to random- $k$  discovery for a threshold number of attempts, a source node can simply evict that option from future retries.

**Security:** The overlay is inherently secure against denial of service (DoS) attacks. Since detour nodes are chosen randomly, the overlay can never become compromised even if a subset of its nodes come under a DoS attack. Also, the random subsets calculated by a node are not shared with any other node in the overlay, making it impossible for any attacker to selectively target overlay nodes to disrupt the infrastructure. Though not investigated as a part of this chapter, we plan to conduct further studies to ground our beliefs as future work.

#### 6.5.4 Putting it all together

The end result of this study is to build an architecture on the Internet that a variety of multimedia applications can query to consistently obtain low-delay, loss-free paths that go beyond Internet path selection. Akin to CDNs transparently redirecting clients to a nearby server, we envision a routing underlay which transparently discovers suitable alternate Internet paths for multimedia applications querying it. Such a system shall operate requiring little state information to do this while scaling to millions of nodes.

Nodes participating in the overlay are envisioned to use the overlay using a simple API such as `SendMessage(dest)`. The underlying substrate at every source transparently revokes random- $k$  every  $\Delta_t$  or based on feedback from destination. Such an overlay could comprise of end-user clients and multimedia dissemination servers (video, audio, games). As we have seen, this mechanism leads to a  $k$ -fold increase in path options for a source; options that often *outperform* Internet path selection.

### Summary of key findings

1. Nodes sending/receiving multimedia content participate in an overlay; nodes include clients/servers. Such an overlay provides  $k$ -fold increase in path options.
2. Source randomly selects  $k$  intermediaries every  $\Delta_t$
3.  $k$  is uniformly representative of participating nodes
4. Source attempts to simultaneously route packets via  $k$  nodes to reach the destination
5. Destination ACKs back the path-of-choice that the sources uses for the remainder of  $\Delta_t$
6. At  $\Delta_t$ , go back to Step. 2
7. Alternatively, if dest. reports an 'outage' goto Step. 2



## 6.6 Source Initiated Frame Restoration

Using the insights from the previous section, we now design and implement a prototype called SIFR, or source initiated frame restoration. SIFR employs frame preserving policies coupled with a random-5 path selection strategy for improving perceptual quality of streaming content. We deploy SIFR on 32 PlanetLab nodes which were used to obtain dataset D2. We evaluate the effectiveness SIFR over the default IP-path in restoring and improving perceptual quality.

### 6.6.1 Prototype description

SIFR requires deployment at source, destination, and intermediate nodes. SIFR at source applies application specific policies to packet generation following a degradation, and chooses intermediate paths based on the random-5 strategy to restore key frames. The receiver takes ingress packets and counts the number of correctly received frames in a 1 sec playout buffer. When losses that manifest in perceptual degradation are observed, the destination issues an “outage” feedback to the source. Upon reception of this, the source tries to recover from the degradation by sending the next set of key frames simultaneously through five randomly chosen intermediaries. When the destination receives an intact GOP from a path after an outage, it reports of this successful reception using this path back to the source. We use a

custom header to capture feedback from the destination. Finally, the intermediaries simply forward ingress packets to the announced destination.

### 6.6.2 Methodology

We select three pairs of PlanetLab nodes to act as a source, with one pair each in the US, Germany and Korea. Each pair of nodes in a country belong to the same site (university) that hosts them, and are as such geographically co-located (i.e., within the same campus); one runs SIFR while the other uses the default IP path to reach destinations. We verify that the source pairs take similar ASs to reach a variety of destinations to eliminate bias in our results. Our destination and intermediary set consists of 32 PlanetLab nodes used in the previous round to obtain dataset D2. Our goal is to compare the perceptual quality of video streams that use SIFR over using the default IP-path.

Every minute, each source pair cycles through a list of five low and high motion clips used in our previous rounds to stream to a destination. The destination is likewise cycled through each of the 32 intermediaries every instance. Using the IP packet trace of the clip, each source pair generate packets with a fragmentation limit of 1024 bytes to the destination. The destination records the packet trace received from each pair of source, which enables us to compare the performance of our prototype over the default IP-path in recovering from outages. We ran this experiment for a little over 48 hours starting Feb 08, 2010.

Table 6.3: Comparing perceptual quality of SIFR against default IP-routing

| Performance Metric                             | SIFR    | Default<br><b>IP-path</b> |
|--|---------|---------------------------|
| total # of GOP degradations                    | 303     | 779                       |
| # of degradation “episodes”                    | 251     | 293                       |
| Mean # of corrupt GOP per episode              | 1.167   | 2.65                      |
| % of times episodes were<br>limited to one GOP | 96%     | 82%                       |
| Mean time to restore quality                   | < 1 sec | 5.23 secs                 |

### 6.6.3 Results

Table 6.3 summarizes a comparison of receiver traces of source nodes that implement SIFR against source nodes that only used the default IP path. We report the number of events when playout degraded to ‘below acceptable’, the number of ‘episodes’ where the degradation persisted on screen, the percentage of times playout could be restored in the very next GOP, and the mean time to restore on screen perceptual quality.

A GOP is considered ‘degraded’ when a key frame is corrupt within a GOP, which manifest artifacts resulting in strong user dissatisfaction. We observed a total of 303 such instances on source nodes that implement SIFR and 779 degraded GOPs using the default IP-path. Overall, this indicates that SIFR could preserve about 61% of GOPs that the

default IP-path could not. Since SIFR reroutes key frames *after* the destination reports of a degradation, a fraction of degradations cannot be prevented. To further elucidate this, we analyze the number of degradation ‘episode’. A degradation episode begins with a degraded GOP and lasts until the first arrival of an intact GOP. We observe that on paths using SIFR, there were 251 episodes of degradation. For every such episode, the destination would have sent a feedback to source requesting a route change. The default IP-path registers about 293 degradations, which seems to indicate that the combination of alternative paths used by SIFR were marginally better in terms of episodes observed.

Of interest then is the mean number of degraded GOPs per episode, which dictates the mean on screen degradation time. For SIFR, we observe that this amounts to 1.2, which indicates that SIFR is able to restore a GOP on most occasions following a degradation. For the default IP-path the mean is about 2.65, which indicates that SIFR could improve episode duration by about 55%.

To better estimate recovery using SIFR, we measure the percentage of times the degradation episode was limited to one GOP. Our results indicate that 96% of the time, SIFR could restore playout following a degradation using alternate paths. For default IP-paths this is around 82%, which in a way reflects on the IP-path in self healing itself. In effect, the availability of higher on screen perceptual quality benefits by 14% with SIFR. We also measure the mean time to restore quality when it degrades. The mean time for IP-path to recover is around 5.23 secs, while SIFR takes less than one second on average.

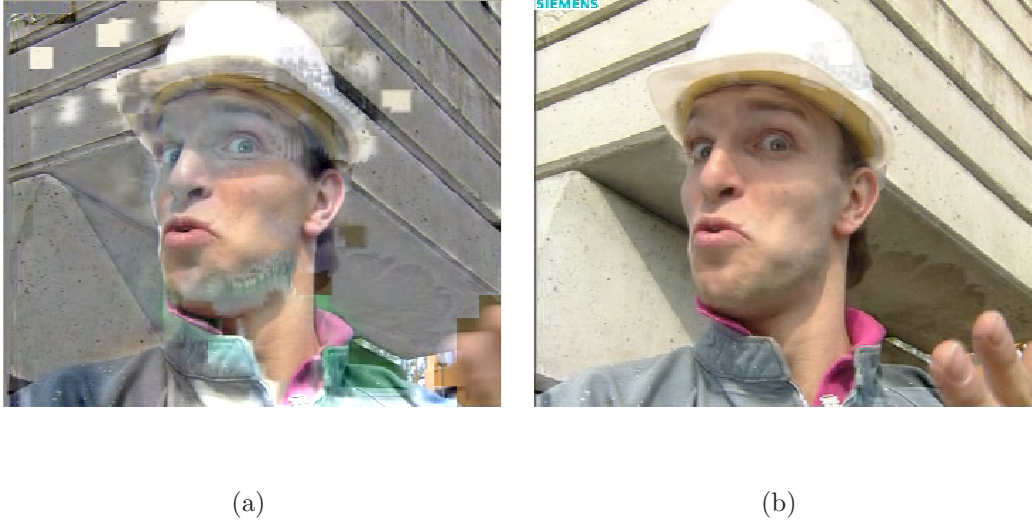


Figure 6.9: (a) Default IP-path v/s (b) SIFR, following a degradation: SIFR recovers from perceptual degradation by restoring key frames.

The perceptual benefits of preserving key frames is substantive. To better illustrate the perceptual benefits of restoring frames, consider the screenshots in Fig. 6.9(a) and Fig. 6.9(b). After an degradation on the default IP-path, the quality of playout degraded to below acceptable. SIFR successfully restored playout by quickly rerouting frames (Fig. 6.9(b)), while the IP-path continued to experience a longer episode of degradation (Fig. 6.9(a)).

## 6.7 Conclusions

This chapter explored a basic mechanism of randomized overlay route selection that a source can use to increase its Internet routing options  $k$ -fold. We argue that it is sufficient to randomly pick  $k$  nodes in any overlay of  $N$  nodes, where  $k$  is bounded by  $O(\ln(N))$ , to

derive low-delay and loss-free paths from the Internet. We explored various properties of such random subsets, like composing an optimum  $k$ , and variants of random- $k$ .

Our results indicate that an overlay built using such a philosophy can reduce RTT by more than 50 ms in more than 70% of the time. We show that such path selection mechanisms require little or no state information, do not burden existing infrastructure, can scale to millions of nodes, improve fairness, and are generally robust because they are completely distributed. We believe this design can be used to build an underlay in the Internet that multimedia applications can query to derive suitable paths, and that such paths can easily outperform Internet path selection.

## CHAPTER 7

### RELATED WORK

If I have seen further it is by standing on the shoulders of giants

---

— Sir Issac Newton

We round off this discussion by taking a closer look at related research. Work related to the material presented in this thesis can be broadly classified as Internet measurement based studies and work on inferring perceptual quality. Though work has been done on both the fronts, little has been done to quantify, infer or improve Internet QoE. We believe our work compliments much of prior Internet based measurement studies and directions towards improving Internet video-QoE. Our measurements of popular Internet destinations and the benefits of using alternative routes can provide valuable insights to service providers and ISPs with major commercial and technical implications.

## 7.1 Quality Evaluation

Depending upon the amount of information needed to perform video quality assessment, architectures can be broadly classified as no-reference (NR), full reference (FR) or reduced reference (RR). FR schemes requires the entire copy of the original frame to infer quality, while RR schemes require partial (key parameters extracted from video) information. The only feasible scheme which can reside as a standalone module in the network core are the NR types.

Video quality assessment can be either objective or subjective. Objective video quality assessment [40] usually involves a characterizing function of measurable parameters (networking or transcoding) that evaluates the quality of a reconstructed frame. Objective quality assessment often fails to correlate with subjective perception. For example, the media delivery index (MDI) [69] is a no-reference framework that can reside in network terminals to report loss and delay rates. While it can provide a comprehensive log of these statistics, it does not map them to subjective perception. Other objective quality assessment techniques like PSNR that are FR in nature are both infeasible inside the network core and grossly inaccurate at inferring perceptual quality.

There has been a recent surge of interest in subjectively assessing quality of a video stream. Even at a time when there is little or no consensus on the exact nature of such a model, the basic idea has been to provide a subjective interpretation of various events on a video stream, such as loss, delay, jitter and so on. For example, authors in [37] train



a random neural network to adjust to viewer responses, and infer the quality of a video stream based on what is observed along the data path. The nature of their deployment, the size of the neural network base, or its suitability of operation at arbitrary network nodes is unknown. Similarly, the video quality metric (VQM) [43] provides a reasonably good indication of subjective perception. However, VQM requires the original frame for reference and is likewise infeasible inside the network core. Apart from these, there are other proprietary video quality metrics exclusive to organizations that own them (e.g., Agilent [73], Symmetricom [84] etc.). These metrics are not open to the public for free inspection or usage.

Subjective quality evaluation usually culminate in a mean opinion score (MOS), which is recommended by the ITU-T [24]. MOS on a chosen scale rates the quality of a video sequence. MOS as a metric has been known to have its share of drawbacks [10, 28, 68]: (i) subjects tend to avoid the extreme scores, (ii) “forgiveness effect”, where users tend to give higher rating when a playout is long and smooth, is not uncommon; and (iii) quality in itself is not a very well defined notion, and has many dimensions to it.

Alternates to using MOS have also been suggested. Work by Tasaka *et. al.* [54] expresses QoE in terms of interval scale or psychological scale, where the proposed SCS strategy switches from error concealment to frame skipping based on a threshold of error. Paired comparisons as an alternate to gathering subjective surveys was proposed in [10]. A multi-disciplinary approach to map QoS metrics to QoE for distributed interactive multimedia was discussed in [72]. Some other studies ask the user to provide certain specific feedbacks instead of MOS. For example, users could be asked to report visibility of certain artifacts

on screen [38], or to suggest changes to artifacts to make the sequence more appealing to them [44]. However, MOS is a convenient way of quantifying scores because it provides a uniform interpretation of quality for various purposes (billing, peak hour usage, capacity planning, customer satisfaction, quality assurance etc.).

## 7.2 Internet Measurements

Internet pathologies have been well investigated in the past [30, 42]. The first such large scale study revealed that Internet path selection was often sub-optimal with persistent routing loops, failures, and network unreachability [42]. Even after a decade since that paper, researchers have consistently found that Internet outages are both unpredictable, and worse, can go undetected for a while [1, 34, 35, 45, 49, 52]. Recent work continues to highlight path inflation in Internet [49], and likewise, BGP anomalies and mis-configurations [34]. However, much of prior research has concentrated on improving reliability of Internet paths with respect to elastic applications. It is a well known fact that Internet path selection today is primarily driven by ISP customer-provider relationships, and is solely based on hop-count and AS-reachability. Authors in [22] investigate ways to improve path selection to increase availability of end-hosts using randomized load allocation. The ensuing gains for elastic applications, however, was marginal. As far as streaming services are concerned, VoIP performance study on the Internet backbone links reveal that a significant number of paths yield poor quality (e.g., [36]).

Internet QoS was aimed at enabling streaming services. The IETF standardized IntServ and DiffServ router mechanisms to improve quality of streaming content, which required changes to every router in the Internet. Given the scale of the Internet, as well as the diversity of various AS's that comprise it, these changes could not be completely co-ordinated. Even after years of slow adoption, there has been no significant performance enhancements in terms of video quality, as the Internet continues to operate on a 'best-effort' delivery model.

OverQoS proposed to overcome QoS deployment issues by instead enforcing QoS mechanisms on a small overlay of nodes, thereby providing statistical QoS guarantees [52]. However, OverQoS operates by probing  $O(N^2)$  links and cannot scale to a large number of nodes. Further, QoS mechanisms operate with a notion of providing service guarantees to enhance application performance. However, service guarantees alone are not sufficient to raise *perceptual* quality. Perceptual quality is best characterized by QoE, which attempts to infer quality from a user's perspective. Random subsets to build scalable application-layer overlay trees by background monitoring  $k$  paths was presented in [29]. Our results reinforce the potential of randomized load allocation, as we establish several key features of composing random subsets while experimentally providing upper bounds to the value of  $k$ .

Scalable overlays can be built by embedding node locations to a co-ordinate space [13]. However, RTT variations in the Internet exhibit triangle inequality violations, often reducing the accuracy of such approaches. It has been shown that the error from embedding is non-trivial [32, 67], limiting their applicability. To the best of our knowledge, no prior work has

been performed on developing a generic large scale architecture that a variety of multimedia applications can query to improve Internet path selection.

## CHAPTER 8

### CONCLUSIONS

Nothing is as easy as it looks in the beginning

Or as difficult as it looks in the middle

Or as complete as it looks in the end

---

— Chinese Proverb

This work presented the first empirical observation of Internet links and paths from a QoE perspective, and tools and protocols to efficiently infer and improve video QoE. To enable QoE inference at arbitrary nodes in the Internet, we presented MintMOS: a lightweight, scalable, no reference framework for inferring QoE of a video stream and offering suggestions to improve it. MintMOS is flexible enough to accommodate any number of parameters that can affect video quality, from network dependent to network independent. MintMOS revolves around a QoE space, which is a  $k$ -dimensional space for  $k$  parameters used to measure quality. The QoE space creates a mapping between parameter values and their associated perceptual quality. Inferring QoE or offering suggestions to improve it use the QoE space to base their decisions. We instrumented a QoE space around 4 parameters and 54 partitions, and

demonstrated its effectiveness in projecting MOS. Subjective surveys verify the accuracy of MintMOS projections.

We studied intra- and inter-ISP links of 51 major ISPs spread across the US, Europe and Asia-Pacific from 38 vantage points in the Internet. We closely studied these links for six days to infer their suitability for streaming services. Our findings offered surprising insights at link level fluctuations in the Internet: (i) we find that intra-domain links are poorly engineered, showing significant loading fluctuations which can degrade video quality. This could largely be attributed to BGP, which makes topology sensitive load balancing hard; (ii) contrary to popular belief, ISP peering links are well engineered and quiet suitable for carrying streaming content. This implies that ISPs are not limited by their choice or quality of peering links, rather, AS-path lengths alone are insufficient as a routing metric; (iii) the overall effect of Internet route selection maps the current QoE to just about "acceptable"; and, (iv) overlay networks could overcome a majority of Internet's shortcomings in delivering multimedia; both because they can be deployed with relative ease compared to re-engineering the Internet, and provide alternative routes that avoid AS's which experience transient fluctuations in delivering packets.

We then studied a large number of end-to-end Internet paths to empirically characterize location, persistence, duration and recurring frequency of Internet outages and their effect on perceptual video quality. To do this, we systematically studied end-to-end paths of source-destination pairs on the Internet to the top 100+ VoD and IPTV service providers. Our results indicate that upto 89% of outages on paths to servers and 62% of outages on

paths to P2P clients are potentially recoverable by using one-hop detours. We collected weeklong measurements from five different datasets that both confine to and span multiple continents with a dominant geographical presence of online streaming services. We observe that not all alternative paths can be useful even if they are loss free, and that a large fraction of degradations could be overcome by a large number of alternate paths. For small scale overlays, we additionally observed that a simple strategy of choosing five random intermediaries in an overlay is sufficient to overcome 90% of outages. We call this strategy “random-5”.

To make randomized route selection a practical solution for large unstructured overlays, we deployed a measurement overlay consisting of 550+ nodes on the wide area Internet for one week. Using this overlay, we explored a basic mechanism of randomized overlay route selection (called “random- $k$ ”) that a source can use to increase its Internet routing options  $k$ -fold. We argue that it is sufficient to randomly pick  $k$  nodes in any overlay of  $N$  nodes, where  $k$  is bounded by  $O(\ln(N))$ , to derive low-delay and loss-free paths from the Internet. We explored various properties of such random subsets, like composing an optimum  $k$ , and variants of random- $k$ . Our results indicate that an overlay built using such a philosophy can reduce RTT by more than 50 ms in more than 70% of the time. We show that such path selection mechanisms require little or no state information, does not burden existing infrastructure, can scale to millions of nodes, improve fairness and are generally robust because they are completely distributed. We believe this design can be used to build an

underlay in the Internet that multimedia applications can query to derive suitable paths, and that such paths can easily outperform Internet path selection.

Finally, we designed a prototype called SIFR for choosing intermediate nodes in a simple, lightweight, yet efficient manner to improve perceptual quality. SIFR outperforms default IP-routing over a 2 day period across wide-area links on the Internet. We show that it is possible to achieve substantive gains in perceptual quality using SIFR on top of today's best effort Internet.



## LIST OF REFERENCES

- [1] D. G. Andersen, H. Balakrishnan, M. F. Kaashoek, R. Morris, “Resilient Overlay Networks”, *Proc. 18th ACM SOSP*, Oct 2001.
- [2] A. Akella, S. Seshan, and A. Shaikh, “An Empirical Evaluation of Wide-Area Internet Bottlenecks”, *ACM Sigmetrics’03*, June 2003.
- [3] K. Anagnostakis, M. Greenwald and R. S. Ryger, “cing: Measuring network internal delays using only existing infrastructure”, *IEEE Infocom*, March 2003.
- [4] J. Babiarez, K. Chan, and F. Baker, “Configuration Guidelines for DiffServ Service Classes”, IETF RFC# 4594. Aug. 2006.
- [5] S. Blake *et. al.*, “An Architecture for Differentiated Services”, IETF RFC 2475. Dec. 1998.
- [6] R. Braden, D. Clark and S. Shenker, “Integrated Services in the Internet Architecture: an Overview”, IETF RFC# 1633. June 1994.
- [7] R. Braden, L. Zhang, S. Berson, S. Herzog and S. Jamin, “Resource ReSerVation Protocol (RSVP)”, IETF RFC# 2205. Sept. 1997.
- [8] P. Calyam, M. Sridharan, W. Mandrawa, and P. Schopis, “Performance Measurement and Analysis of H.323 Traffic”, *Passive and Active Measurements (PAM)*, April 2004
- [9] M. Cha, P. Rodriguez, J. Crowcroft, S. Moon, and X. Amatriain, “Watching television over an IP network”, *Proc. ACM IMC*, Vouliagmeni, Greece. Oct. 2008.
- [10] K. T. Chen, C. C. Wu, Y. C. Chang, and C. L. Lei, “A crowdsorceable QoE evaluation framework for multimedia content”, *Proc. Intl. Conf. on Multimedia*, Beijing, China, Oct. 2009.
- [11] Cisco White Paper, “Cisco Visual Networking Index: Forecast and Methodology, 2009–2014”, Cisco Inc. Availabale: [www.cisco.com](http://www.cisco.com). June 2010.
- [12] A. Czumaj and V. Stemann, “Randomized Allocation Processes”, *Symp. on Foundations of Computer Science*, Miami, FL. Oct. 1997.

- [13] F. Dabek, R. Cox, M. F. Kaahoe, and R. Morris, “Vivaldi: A Decentralized Network Coordinate System”, *ACM Sigcomm*, Portland, OR. Aug. 2004.
- [14] P. de Cuetos and K. W. Ross, “Optimal Streaming of Layered Video: Joint Scheduling and Error Concealment”, *Proc. ACM Intl. Conf. on Multimedia*, Berkeley, CA. Nov 2003.
- [15] D. Eager, E. Lazowska, and J. Zahorjan, “Adaptive load sharing in homogeneous distributed systems”, *IEEE Trans. on Software Engg.*, vol. 12(5), May 1986.
- [16] K. Fujimoto, S. Ata, and M. Murata, “Adaptive playout buffer algorithm for enhancing perceived quality of streaming applications”, *IEEE Globecom*, Nov. 2002.
- [17] A. Gersho and R. M. Gary, *Vector Quantization and Signal Compression*, Kluwer Academic Publishers, 1991.
- [18] R. Govindan and V. Paxson, “Estimating router ICMP generation delays”, *Passive and Active Measurements (PAM)*, March 2002.
- [19] J. Greengrass, J. Evans, and A. C. Begen, “Not All Packets Are Equal, Part I: Streaming Video Coding and SLA Requirements”, *IEEE Internet Computing*, vol. 13(1), March 2009.
- [20] J. Greengrass, J. Evans, and A. C. Begen, “Not All Packets Are Equal, Part II: The Impact of Network Packet Loss on Video Quality ”, *IEEE Internet Computing*, vol. 13(2), March 2009.
- [21] D. Grossman, “New Terminology and Clarifications for Diffserv”, IETF RFC# 3260. April 2002.
- [22] K. Gummadi, H. Madhyastha, S. Gribble, H. Levy, and D. Wetherall, “Improving the reliability of internet paths with one-hop source routing”, *Proc. OSDI*, San Fransico, CA. Dec. 2004.
- [23] D.S. Hands, “A basic multimedia quality model”, *IEEE Trans. on Multimedia*, Vol. 6(6), pp. 806 – 816, Dec. 2004.
- [24] International Telecommunication Union, “Subjective video quality assessment methods for multimedia applications”, Rec. ITU-T P.910, Sept. 1999.
- [25] Internation Telecommunication Union, “Opinion model for video-telephony applications”, Rec. ITU-T G. 1070, Nov. 2009.
- [26] R. Jain, “Quality of Experience”, *IEEE Multimedia*, Vol. 11(1), pp 95–96, March 2004.
- [27] J. Jannotti, D. Gifford, K. Johnson, M. F. Kaashoe, and J. O’Toole, “Overcast: Reliable Multicasting with an Overlay Network”, *Proc. OSDI*, San Diego, CA. Oct. 2000.

- [28] S. Kanumuri, P. C. Cosman, A. R. Reibman, and V. A. Vaishampayan, “Modeling packet-loss visibility in MPEG-2 video”, *IEEE Trans. on Multimedia*, 8(2), pp. 341–355, April 2006.
- [29] D. Kostic, A. Rodriguez, J. Albrecht, A. Bhirud, and A. Vahdat, “Using Random Subsets to Build Scalable Network Services”, *Proc. USITS*, Seattle, WA. March 2003.
- [30] C. Labovitz, R. Malan, and F. Jahanian, “Internet Routing Instability”, *IEEE/ACM Trans. on Networking*, 6(5). 1998.
- [31] M. Lu, J. Wu, K. Peng, P. Huang, J. Yao, and H. Chen, “Design and Evaluation of a P2P IPTV System for Heterogeneous Networks”, *IEEE Trans. on Multimedia*, Vol. 9(8), Dec. 2007.
- [32] C. Lumezanu, R. Baden, D. Levin, N. Spring, and B. Bhattacharjee, “Symbiotic relationships in Internet routing overlays”, *Usenix NSDI*, Boston, MA. April 2009.
- [33] C. Ly, C. Hsu, and M. Hafeeda, “Improving Online Gaming Quality using Detour paths”, *ACM Multimedia*, Firenze, Italy. Oct 2010.
- [34] R. Mahajan, N. Spring, D. Wetherall, and T. Anderson, “Inferring link weights using end-to-end measurements”, *ACM Sigcomm Internet Measurement Workshop*, 2002.
- [35] R. Mahajan, N. Spring, and T. Anderson, “Understanding BGP Misconfiguration”, *ACM Sigcomm*, Aug 2002.
- [36] A. Markopoulou, F. Tobagi, and M. Karam, “Assessment of VoIP quality over Internet backbones”, *Proc. IEEE Infocom*, June 2002
- [37] S. Mohamed and G. Rubino, “A Study of Realtime Packet Video Quality Using Random Neural Networks”, *IEEE Transactions On Circuits and Systems for Video Technology*, vol. 12(12), pp. 1071 – 1083, Dec. 2002.
- [38] M. S. Moore, S. K. Mitra, and J. M. Foley, “Defect visibility and content importance implications for the design of an objective video fidelity metric”, *Proc. IEEE ICIP*, Rochester, NY. June 2002.
- [39] K. Nichols, S. Blake, F. Baker and D. Black, “Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers”, IETF RFC# 2474. Dec. 1998.
- [40] O. Olsson, M. Stoppiana, and J. Baina, “Objective methods for assessment of video quality: State of the art”, *IEEE Trans. on Broadcasting*, vol. 43, pp. 487–495, Dec. 1997
- [41] K. Papagiannaki *et. al.*, “Analysis of measured single hop delay from an operational backbone network”, *IEEE Infocom*, June 2002.

- [42] V. Paxson, “End-to-end routing behavior in the Internet”, *IEEE/ACM Trans. on Networking*, 5(5), pp. 601–615, 1997.
- [43] M. H. Pinson and S. Wolf, “A New Standardized Method for Objectively Measuring Video Quality”, *IEEE Trans. on Broadcasting*, Vol. 50(3). Sept 2003.
- [44] M. G. Ramos and S. S. Hemami, “Suprathreshold wavelet co-efficient quantization in complex stimuli: Pshychophysical evaluation and analysis”, *J. Opt. Soc. Amer.*, 20 (7), pp. 1164–1180, July 2003.
- [45] S. Savage *et. al.*, “Detour: A Case for informed internet routing and transport”, *IEEE Micro*, 19(1), Jan. 1999.
- [46] S. Shenker, C. Partridge, and R. Guerin, “Specification of Guaranteed Quality of Service”, IETF RFC# 2212. Sept. 1997.
- [47] S. Shenker and J. Wroclawski, “General Characterization Parameters for Integrated Service Network Elements”, IETF RFC# 2215. Sept. 1997.
- [48] M. Siller and J. Woods, “QoS arbitration for improving the QoE in multimedia transmission”, *Proc. Intl. Conf. on Visual Information Engineering*, 2003.
- [49] N. Spring, R. Mahajan and T. Anderson, “Quantifying the Causes of Path Inflation”, *ACM SIGCOMM*, Sept. 2006.
- [50] A. Su, D. Choffnes, A. Kuzmanovic, and F. Bustamante, “Drafting Behind Akamai”, *ACM SIGCOMM*, Pisa, Italy, Sept 2006.
- [51] L. Subramanian, S. Agarwal, J. Rexford, and R. H. Katz, “Characterizing the Internet hierarchy from multiple vantage points”, *IEEE Infocom’02*, New York. June 2002.
- [52] L. Subramanian, I. Stoica, H. Balakrishnan, and R. Katz, “OverQoS: An Overlay Based Architecture for Enhancing Internet QoS”, *Usenix NSDI*, March 2004.
- [53] S. Tao and R. Guerin, “Application-Specific Path Switching: A Case Study for Streaming Video”, *Proc. ACM Multimedia*, New York, NY. October 2004.
- [54] S. Tasaka, H. Yoshimi, A. Hirashima, and T. Nunome, “The Effectiveness of a QoE-Based Video Output Scheme for Audio-Video IP Transmission”, *ACM Multimedia*, Vancouver, Canada, Oct. 2008.
- [55] R. Teixeira , K. Marzullo, S. Savage, and G. Voelker, “In Search of Path Diversity in ISP Networks”, *ACM Internet Measurement Conference (IMC)*, 2003.
- [56] C. J. van den Branden Lambrecht, and O. Verscheure, “Perceptual quality measure using a spatio temporal model of the human visual system”, *Proc. IST/SPIE Conference Digital Video and Compression: Algorithms and Technologies 1996*, vol 2668, Feb 1996

- [57] M. Venkataraman and M. Chatterjee, “Inferring video-QoE in Real Time”, *IEEE Network*, 25(1), Jan 2011.
- [58] M. Venkataraman and M. Chatterjee, “Effects of Internet Path Selection on Video-QoE”, *ACM Multimedia Systems (MMSys)*, San Jose, CA. Feb. 2011 .
- [59] M. Venkataraman and M. Chatterjee, “Case Study of Internet Links: What degrades Video QoE?”, *IEEE Globecom*, Miami, FL. Dec 2010.
- [60] M. Venkataraman and M. Chatterjee, “Quantifying Video QoE-degradation of Internet links”, *IEEE/ACM Trans. on Networking* (revision submitted).
- [61] M. Venkataraman, M. Chatterjee, and S. Chattopadhyay, “Lighweight, real-time, no-reference framework for inferring subjective-QoE”, *IEEE Globecom*, Dec. 2009.
- [62] M. Venkataraman and M. Chatterjee, “MintMOS: A Lightweight, Real Time, No-Reference Tool for Inferring Video-QoE along Internet paths”, *IEEE Trans. on Multimedia* (submitted), March 2011.
- [63] M. Venkataraman and M. Chatterjee, “Effects of Internet Path Selection on Video-QoE: Analysis and Improvements”, *IEEE/ACM Trans. on Networking* (submitted) April 2011.
- [64] M. Venkataraman, S. Sengupta, M. Chatterjee and R. Neogi, “Towards a Video-QoE Definition in Converged Networks”, *International Conference on Digital Communications (ICDT)*, San Jose, CA. July 2007.
- [65] M. Venkataraman, S. Sengupta, M. Chatterjee and R. Neogi, “Designing a Collector Overlay Architecture for Fault Diagnosis in Video Networks”, *Elsevier Computer Communications*, May 2011.
- [66] M. Venkataraman, S. Sengupta, M. Chatterjee and R. Neogi, “A Collector Overlay Architecture for Fault Diagnosis in Access Networks”, *IEEE CCNC*, Las Vegas, NV. Jan. 2009.
- [67] G. Wang, B. Zhang, and T. S. Eugene Ng, “Towards network triangle inequality violation aware distributed systems”, *ACM IMC*, San Diego, CA. Oct. 2007.
- [68] A. Watson and M. A. Sasse, “Measuring perceived quality of speech and video in multimedia conferencing applications”, *Proc. ACM Multimedia*, pp. 55–60, Apr 1998.
- [69] J. Welch and J. Clark, “A Proposed Media Delivery Index (MDI)”, IETF RFC 4445. Apr 2006
- [70] J. Wroclawski, “Specification of the Controlled-Load Network Element Service”, IETF RFC# 2211. Sept. 1997.

- [71] H. R. Wu, T. Ferguson, and B. Qiu, “Digital video quality evaluation using quantitative quality models”, *Proc. 4th Intl. Conf. on Signal Processing*, Beijing, China, Oct. 1998.
- [72] W. Wu, A. Arefin, R. Rivas, K. Nahrstedt, R. Sheppard, Z. Yang, “Quality of experience in distributed interactive multimedia environments: toward a theoretical framework”, *Proc. ACM Multimedia*, Beijing, China. Oct. 2009.
- [73] Agilent Technologies Inc., [www.agilent.com](http://www.agilent.com)
- [74] Akamai Inc., [www.akamai.com](http://www.akamai.com)
- [75] AS Number lookup Utility tool. <http://www.bugest.net/software/aslookup/index-e.html>, July 2009.
- [76] The libpcap project at SourceForge. <http://sourceforge.net/projects/libpcap/>
- [77] Ineoquest Singulus G1-T Equipment. [www.ineoquest.com/singulus-family](http://www.ineoquest.com/singulus-family)
- [78] *King* : A tool to estimate latency between any two Internet hosts, from any other Internet host. Available: <http://www.mpi-sws.org/~gummadi/king/>
- [79] MintMOS Validation Clips: <http://sites.google.com/site/anonmintmos> Houses all sample clips used in this thesis.
- [80] PlanetLab vantage points, discussions, and video clips used to conduct subjective surveys. <http://sites.google.com/site/anonqoe/>
- [81] Video Evaluation Toolkit (Moscow State University, Moscow, Russia). <http://graphics.cs.msu.ru>. Available for free download. Aug. 2008.
- [82] PlanetLab Consortium. <http://www.planet-lab.org/>
- [83] The “RouteViews” Project at University of Oregon. <http://www.routeviews.org/>, July 2009.
- [84] Symmetricom Inc., <http://www.symmetricom.com/>
- [85] Undns tool, router hostname to location decoder. <http://www.scriptroute.org/source/>
- [86] VLC Media Player, <http://www.videolan.org/vlc>
- [87] YouTube(R). [www.youtube.com](http://www.youtube.com). April 2009.